TESIS – KS142501

MODEL KLASIFIKASI UNTUK DETEKSI SITUS PHISING DI INDONESIA

FEBRY EKA PURWIANTONO
NRP. 5215201006

DOSEN PEMBIMBING
Dr. Ir. Aris Tjahyanto, M.Kom
NIP. 196503101991021001

PROGRAM MAGISTER
JURUSAN SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

(Halaman ini sengaja dikosongkan)

THESIS – KS142501

CLASSIFICATION MODEL FOR DETECTION PHISHING WEBSITE IN INDONESIA

FEBRY EKA PURWIANTONO
NRP. 5215201006

SUPERVISOR
Dr. Ir. Aris Tjahyanto, M.Kom
NIP. 196503101991021001

MAGISTER PROGRAM
DEPARTMENT OF INFORMATION SYSTEMS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom)
Di
Institut Teknologi Sepuluh Nopember

Oleh :
Febry Eka Purwiantono
NRP. 5215201006

Tanggal Ujian : 11 Juli 2017
Periode Wisuda : September 2017

Disetujui Oleh :

1. Dr. Ir. Aris Tjahyanto, M.Kom
   NIP. 195810051986031003 (Pembimbing)

2. Dr. Eng. Febriliyan S., S.Kom., M.Kom
   NIP. 197302191998021001 (Penguji)

3. Nur Aini R., S.Kom., M.Sc.Eng., Ph.D
   NIP. 198201202005012001 (Penguji)

Dekan
Fakultas Teknologi Informasi

Dr. Agus Zainal Arifin, S.Kom., M.Kom
NIP. 197208091995121001

(Halaman ini sengaja dikosongkan)

MODEL KLASIFIKASI UNTUK DETEKSI SITUS PHISING DI INDONESIA

Nama Mahasiswa	: Febry Eka Purwiantono
NRP	: 5215201006
Pembimbing	: Dr. Ir. Aris Tjahyanto, M.Kom

ABSTRAK

Penelitian ini mengusulkan sebuah model klasifikasi yang dapat digunakan untuk mendeteksi situs phising di Indonesia (berbahasa Indonesia, berserver di Indonesia atau sering digunakan oleh pengguna internet dari Indonesia) secara akurat. Teknik deteksi yang diusulkan berdasarkan analisis situs menggunakan pendekatan berbasis fitur konten dan URL. Model klasifikasi ini mengkombinasikan beberapa fitur unik dari penelitian sebelumnya dan fitur baru berbasis konten dan URL untuk meningkatkan kinerja deteksi agar mampu mengungguli model klasifikasi pada penelitian sebelumnya. Dataset yang digunakan dalam penelitian ini kurang lebih terdiri dari 340 situs phising dan 340 situs non-phising. Selain itu, pada model klasifikasi yang diusulkan dibuat sebuah *web crawler* berbasis PHP dan API *(Application Programming Interface)* untuk mengekstraksi fitur pada penelitian ini, sehingga memudahkan peneliti dalam pengolahan data menggunakan software Weka.

Penelitian ini menggunakan 4 algoritma berbeda antara lain SMO *(Sequential Minimal Optimization)*, Naive Bayes, Bagging dan Multilayer Perceptron. Hasilnya, SMO, Naive Bayes, Bagging dan Multilayer Perceptron memiliki akurasi kurang lebih 95,88%, 96,91%, 97,35% dan 96,91%. Dimana algoritma dengan akurasi terbaik yaitu Bagging akan digunakan dalam model klasifikasi ini untuk dibandingkan dengan model klasifikasi pada penelitian sebelumnya menggunakan dataset yang sama. Hasilnya, akurasi dari model klasifikasi pada penelitian ini mengungguli akurasi dari model klasifikasi pada penelitian sebelumnya. Model klasifikasi pada penelitian ini unggul 16,76% terhadap model klasifikasi pada penelitian sebelumnya yang mana hanya menghasilkan akurasi 80,59%.

Kata kunci : model klasifikasi, deteksi, situs phising, Indonesia, fitur

(Halaman ini sengaja dikosongkan)

CLASSIFICATION MODEL FOR DETECTION PHISHING WEBSITE IN INDONESIA

By	: Febry Eka Purwiantono
Student Identity Number	: 5215201006
Supervisor	: Dr. Ir. Aris Tjahyanto, M.Kom

ABSTRACT

This research proposed a classification model that can be used to detect phishing website in Indonesia (using Bahasa Indonesia, hosted in Indonesia or frequently accessed by Internet users from Indonesia) accurately. The proposed detection technique based on website analysis using the URL and content feature based approach. This classification model combines some unique feature vectors of previous research and new feature vector based on URL and content approach to improve detection performance to be able to outperform classification model in previous research. Dataset used in this research consisted of approximately 340 authentic websites and 340 phishing websites. Moreover, in the proposed classification model created a web crawler based on PHP and API (Application Programming Interface) to extract feature vectors in this research, so it can support researcher in data processing using software Weka.

This research uses four different algorithms such as SMO (Sequential Minimal Optimization), Naive Bayes, Bagging and Multilayer Perceptron. The result, SMO, Naive Bayes, Bagging and Multilayer Perceptron have accuracy of approximately 95.88%, 96.91%, 97.35% and 96.91%. Algorithm has the best accuracy is Bagging, it will be used in this classification model to compare with classification model in previous research using same dataset. The result, accuracy of classification model in this research outperformed accuracy of classification model in previous research. The classification model in this research outperform 6.01% against classification model in previous research which only yielded 80.59% accuracy.

Keywords: classification model, detection, phishing website, Indonesia, feature

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas berkat, rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan tesis dengan judul "MODEL KLASIFIKASI UNTUK DETEKSI SITUS PHISING DI INDONESIA". Penyusunan tesis ini dibuat sebagai salah satu syarat kelulusan program magister jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Penulis menyadari selama menempuh pendidikan dan proses penyelesaian tesis ini penulis memperoleh bantuan dan dukungan dari berbagai pihak. Pada kesempatan kali ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang membantu pengerjaan tesis ini, antara lain:

1. Kedua orang tua, adik dan keluarga yang telah memberikan doa, motivasi serta dukungan kepada penulis tanpa henti.
2. Bapak Dr. Ir. Aris Tjahyanto, M.Kom yang telah sabar dan telaten membimbing serta membagikan ilmu dan waktunya kepada penulis dalam pengerjaan tesis ini.
3. Bapak Dr. Eng. Febriliyan S., S.Kom, M.Kom dan Ibu Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D yang telah memberikan banyak kritik dan saran untuk perbaikan penelitian ini.
4. Seluruh Bapak dan Ibu dosen serta karyawan di program magister jurusan Sistem Informasi ITS yang telah membagikan ilmu dan inspirasi kepada penulis.
5. Kekasih tercinta yang selalu memberikan semangat dan motivasi saat suka maupun duka agar penulis mampu menyelesaikan penelitian ini hingga titik darah penghabisan.
6. Pemain DotA 2 dan OSM (Online Soccer Manager) yang selalu menemani dan menghibur penulis saat suntuk, sehingga membuat penulis mampu berfikir jernih dan menemukan inspirasi-inspirasi baru untuk menyelesaikan penelitian ini.
7. Rekan bisnis yang selalu mendukung penulis untuk menyelesaikan studi.
8. Teman-teman dan keluarga besar program magister Sistem Informasi ITS angkatan 2015 yang telah memberikan bantuan dan dukungan kepada penulis selama mengikuti perkuliahan dan proses penelitian ini berlangsung.

9. Teman-teman dan pihak lain yang tidak dapat penulis cantumkan namanya satu per satu yang telah mendoakan, memberikan bantuan, dukungan serta sumbangan pemikiran dalam proses penyelesaian tesis ini.

Semoga Allah SWT senantiasa memberikan berkat, rahmat dan anugerah-Nya, serta membalas semua kebaikan dan dukungan yang telah diberikan kepada penulis. Penulis menyadari masih banyak kekurangan yang terdapat pada penelitian ini, oleh karena itu kritik dan saran yang bersifat membangun akan selalu diterima oleh penulis. Semoga penelitian ini dapat memberikan manfaat dan wawasan yang berguna bagi pengembangan ilmu pengetahuan dan bagi para pembaca.

Surabaya, 10 Juni 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN i
ABSTRAK iii
ABSTRACT v
KATA PENGANTAR vii
DAFTAR ISI ix
DAFTAR GAMBAR xiii
DAFTAR TABEL xv
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 6
1.3 Tujuan 6
1.4 Ruang Lingkup Penelitian 7
1.5 Kontribusi Penelitian 7
1.5.1 Kontribusi Teoritis 7
1.5.2 Kontribusi Praktis 8
1.6 Sistematika Penulisan 8
BAB 2 LANDASAN TEORI DAN KAJIAN PUSTAKA 11
2.1 Situs Phising 11
2.2 Klasifikasi 12
2.3 Penelitian Terkait 16
2.4 Fitur Deteksi 28
2.5 Kinerja Deteksi 28
2.6 Algoritma Pada Klasifikasi 29
2.6.1 SMO *(Sequential Minimal Optimization)* 29
2.6.2 Naive Bayes 32
2.6.3 Bagging 34
2.6.4 Multilayer Perceptron 35
BAB 3 METODOLOGI PENELITIAN 39
3.1 Gambaran Umum Penelitian 39

3.2 Pengumpulan Data 40
3.3 Fitur 41
3.4 *Pre-Processing Data* 45
3.4.2 Prefiksasi 45
3.4.2 Ekstraksi Fitur 46
3.5 Model Klasifikasi Deteksi Situs Phising 48
3.5.1 Model Klasifikasi 49
3.5.2 Kinerja Klasifikasi 50
3.5.3 Algoritma Klasifikasi 51
3.6 Skenario Uji Coba 52
3.7 Jadwal Penelitian 54
BAB 4 UJI COBA DAN ANALISIS HASIL 57
4.1 Data Uji Coba 57
4.2 Lingkungan Uji Coba 58
4.3 Persiapan Uji Coba 59
4.3.1 Pembuatan *Web Crawler* 59
4.3.1.1 *Web Crawler I* 60
4.3.1.2 *Web Crawler II* 66
4.3.2 Hasil Prefiksasi 72
4.3.3 Hasil Ekstraksi Fitur 73
4.4 Uji Coba 74
4.4.1 Uji Coba Algoritma Klasifikasi 74
4.4.1.1 Uji Coba Algoritma SMO *(Sequential Minimal Optimization)* 75
4.4.1.2 Uji Coba Algoritma Naive Bayes 76
4.4.1.3 Uji Coba Algoritma Bagging 77
4.4.1.4 Uji Coba Algoritma Multilayer Perceptron 78
4.4.2 Uji Coba Model Klasifikasi Pada Penelitian Sebelumnya 84
4.4.3 Uji Coba Data Baru 85
4.4.4 Uji Coba Data Mining Clustering 93
4.5 Analisis Hasil 96
4.6 Kontribusi 98

4.6.1 Kontribusi Secara Teoritis.....98
4.6.2 Kontribusi Secara Praktis.....101
BAB 5 KESIMPULAN DAN SARAN.....107
5.1 Kesimpulan.....107
5.2 Saran.....108
DAFTAR PUSTAKA.....111
LAMPIRAN A.....115
LAMPIRAN B.....129
BIODATA PENULIS.....135

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 1.1 Perbandingan Situs Phising (Kiri) Dengan Non-Phising (Kanan).....2
Gambar 1.2 Grafik *Phishing Activity Trend* Kuartal Keempat (APWG, 2017).....3
Gambar 2.1 Proses Klasifikasi (Jiawei Han, 2006).....13
Gambar 2.2 Model Klasifikasi Sentimen (Cagatay Catal, 2017).....14
Gambar 2.3 Model Klasifikasi Untuk Memprediksi Aktivator Pada CAR (Kyungro Lee, 2016) 15
Gambar 2.4 Model Klasifikasi Berbasis SVM (Karthik Thirumala, 2017) 16
Gambar 2.5 Hasil Perbandingan Algoritma Klasifikasi Deteksi Website *E-Business* Phising (Dongsong Zhang, 2014) 20
Gambar 2.6 Hasil Perbandingan Algoritma Klasifikasi Deteksi Website Phising (Yuangcheng Li, 2016) 23
Gambar 2.7 Hasil Perbandingan Algoritma Klasifikasi Deteksi Situs Phising (Nelda Abdelhamid, 2014).....27
Gambar 2.8 Ilustrasi Algoritma Bagging 34
Gambar 2.9 Arsitektur Multilayer Perceptron 36
Gambar 3.1 Gambaran Umum Penelitian 39
Gambar 3.2 Tampilan *Web Crawler* 47
Gambar 3.3 Contoh Hasil Ekstraksi Fitur 47
Gambar 3.4 Model Klasifikasi Deteksi Situs Phising 49
Gambar 4.1 Tampilan *Web Crawler I*.....66
Gambar 4.2 Tampilan *Web Crawler II* 72
Gambar 4.3 Hasil Perbandingan Kinerja Algoritma Klasifikasi.....80
Gambat 4.4 Hasil Perbandingan Kinerja Model Klasifikasi 97
Gambar 4.5 Rancangan Implementasi *Service* Menjadi API.....103
Gambar 4.5 Rancangan Implementasi *Service* Menjadi *Standalone Website* 104

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2.1 Rasio Fitur.....26
Tabel 2.2 Analogi JSB (Jaringan Syaraf Biologis) Dan JST (Jaringan Syaraf Tiruan) 35
Tabel 2.3 Multilayer Perceptron 37
Tabel 3.1 Contoh Hasil Proses Prefiksasi 46
Tabel 3.2 *Confusion Matrix* 50
Tabel 3.3 Detail Data 53
Tabel 3.4 Jadwal Rencana Kegiatan Penelitian 55
Tabel 4.1 Spesifikasi Perangkat Keras.....58
Tabel 4.2 Spesifikasi Perangkat Lunak.....58
Tabel 4.3 *Confusion Matrix* Algoritma SMO 75
Tabel 4.4 Kinerja Algoritma SMO.....76
Tabel 4.5 *Confusion Matrix* Algoritma Naive Bayes.....76
Tabel 4.6 Kinerja Algoritma Naive Bayes.....77
Tabel 4.7 *Confusion Matrix* Algoritma Bagging 77
Tabel 4.8 Kinerja Algoritma Bagging.....78
Tabel 4.9 *Confusion Matrix* Algoritma Multilayer Peceptron.....78
Tabel 4.10 Kinerja Algoritma Multilayer Peceptron 77
Tabel 4.11 Perbandingan *Confusion Matrix* Tiap Algoritma 80
Tabel 4.12 Rank Fitur 81
Tabel 4.13 *Confusion Matrix* Model Klasifikasi Pada Penelitian Dongsong Zhang (2014) Menggunakan Algoritma SMO 84
Tabel 4.14 Kinerja Model Klasifikasi Pada Penelitian Dongsong Zhang (2014)..85
Tabel 4.15 Data Sampel.....86
Tabel 4.16 Hasil Prefiksasi Data Sampel.....87
Tabel 4.17 Hasil Uji Coba Data Baru Pada Model Klasifikasi Di Dalam Penelitian Ini 90
Tabel 4.18 Hasil Uji Coba Data Baru Pada Model Klasifikasi Di Dalam Penelitian Dongsong Zhang (2014) 92

Tabel 4.19 Hasil Uji Coba Data Baru Menggunakan Data Mining Clustering 94
Tabel 4.20 Hasil Perbandingan Kinerja Model Klasifikasi 98

BAB 1
PENDAHULUAN

Pada bab awal ini akan dijelaskan mengenai gambaran penelitian mulai dari latar belakang, rumusan masalah, tujuan, ruang lingkup penelitian, kontribusi penelitian hingga sistematika penulisan dokumen.

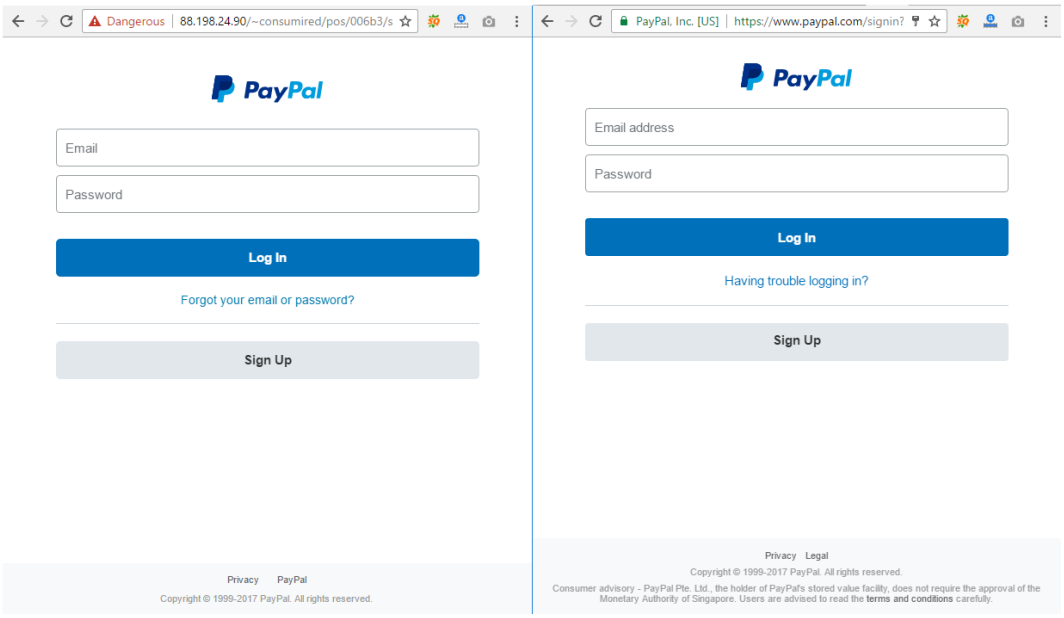
1.1 Latar Belakang

Dari tahun ke tahun pengguna internet semakin meningkat. Berbagai transaksi maupun aktifitas yang semula dilakukan secara offline, sekarang sudah dapat diakses dan dilakukan dari mana saja secara online dan *real time*. Di sisi lain, kemudahan ini dapat menjadi celah keamanan bagi para pengguna internet yang masih awam mengenai keamanan bertransaksi di dunia maya yang dapat dimanfaatkan oleh penjahat internet untuk mendapatkan informasi rahasia seperti data pribadi, kata sandi e-mail, bahkan informasi finansial seperti data kartu kredit dan *online banking* tanpa disadari oleh pengguna internet. Untuk melakukan hal ini, penjahat internet biasanya menggunakan situs phising sebagai alat bantunya.

Situs phising merupakan sebuah website yang didesain oleh penjahat internet sedemikian rupanya agar menyerupai situs otentik (tampilan, konten, URL domain atau lainnya) untuk mengelabui korbannya (pengguna internet) dengan membuat korban seolah-olah sedang mengakses halaman situs dari sumber yang sah [1]. Tampilan situs akan dibuat semirip mungkin dengan situs aslinya agar korban yakin sedang berada pada situs yang benar. Selain itu, ada pula situs phising yang didesain khusus untuk memberikan informasi atau petunjuk palsu yang menyesatkan. Jika korban berhasil dikelabui dan memasukkan informasi yang diminta, penjahat internet dapat dengan mudah menggunakan informasi tersebut pada situs yang sah untuk melakukan aktifitas-aktifitas yang tidak diinginkan dan tentunya hal ini akan menimbulkan kerugian yang cukup signifikan bagi para korbannya mulai dari kerugian finansial hingga *data loss*.

Situs jual beli dan *online banking* adalah situs yang paling banyak dijadikan sasaran phishing oleh penjahat internet, karena potensi keuntungan yang bisa diraup

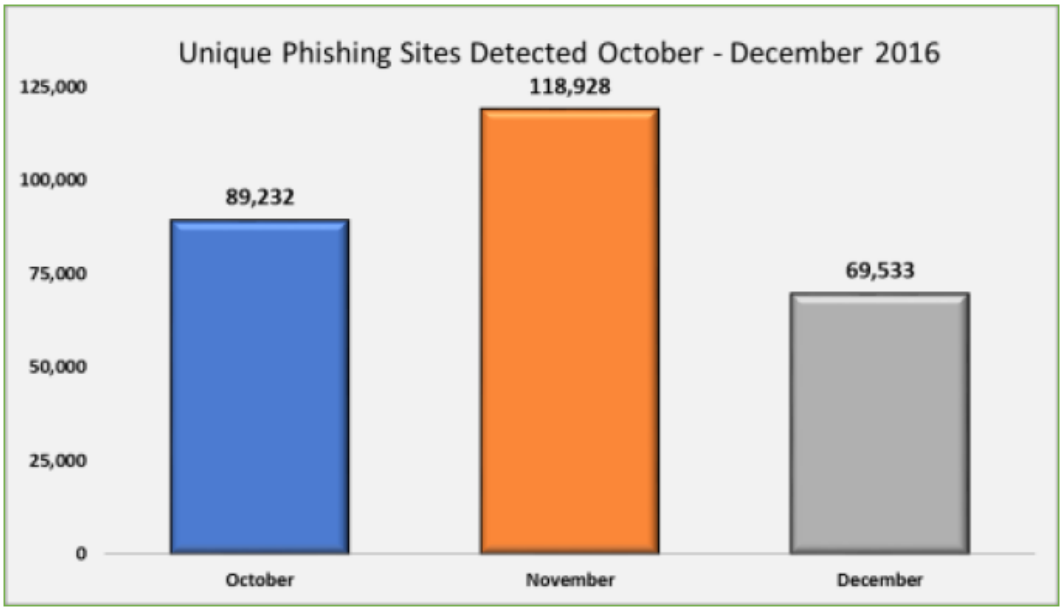
oleh penjahat internet lumayan besar bila dibandingkan situs-situs lain. Situs jual beli dan *online banking* yang paling populer menjadi sasaran empuk para penjahat internet di antaranya adalah eBay dan PayPal [2]. Akan tetapi tidak sedikit pula situs berbasis *Social Media* menjadi sasaran para penjahat internet seperti Facebook, Twitter, Instagram dan sejenisnya. Selain digunakan untuk melakukan pencurian data, situs phising juga digunakan untuk melakukan tindakan penipuan yang mengatasnamakan situs yang sah dan sebagai media penyebaran malware/virus komputer oleh penjahat internet. Di bawah ini adalah perbandingan halaman situs phising dan situs non-phising yang menyerupai situs *online banking* PayPal yang mana situs phising tersebut beralamatkan di *http://88.198.24.90/~consumired/pos/006b3/* :



Gambar 1.1 Perbandingan Situs Phising (Kiri) Dengan Non-Phising (Kanan)

Bila diamati situs phising tersebut sangat mirip sekali dengan halaman situs aslinya, bahkan sangat sulit sekali membedakannya menggunakan mata telanjang. Menurut APWG *(Anti-Phishing Working Group)* [3], kesadaran masyarakat terhadap situs phising meningkat dari tahun ke tahun, akan tetapi jumlah situs phising dan kerugian yang ditimbulkan tumbuh lebih cepat. Pada laporan APWG kuartal keempat 2016 (Gambar 1.2), *phishing activity trend* pada bulan Oktober

2016 terdapat 89,232 situs yang terdeteksi sebagai situs phising, sedangkan di bulan November dan Desember 2016 masing-masing terdapat 118.928 dan 69.533 situs yang terindikasi sebagai situs phising. Pada laporan tersebut juga ditemukan kurang lebih 17 juta malware baru.



Gambar 1.2 Grafik *Phishing Activity Trend* Kuartal Keempat 2016 (APWG, 2017)

Hal ini dapat menimbulkan ketakutan dan menurunnya kepercayaan pengguna internet bertransaksi secara online, padahal transaksi secara online saat ini sedang *booming* di Indonesia. Oleh sebab itu dibutuhkan sebuah sistem yang mampu mendeteksi situs phising secara akurat untuk mencegah dan menghindari kerugian yang ditimbulkan oleh situs phising kepada pengguna internet khususnya di Indonesia. Terlebih lagi minimnya penelitian mengenai situs phising di Indonesia menjadi awal mula pembuatan penelitian ini. Beberapa penelitian sebelumnya menggunakan pengklasifikasian (model klasifikasi) untuk membedakan situs phising dan situs non-phising. Model klasifikasi dipilih karena mampu memberikan keputusan dan mengklasifikasikan data yang ada untuk menentukan dan memastikan apakah situs yang dimaksud termasuk situs phising atau non-phising berdasarkan persyaratan-persyaratan (fitur-fitur) yang telah ditentukan sebelumnya.

Sistem deteksi situs phishing muncul sebagai mekanisme penting untuk memberantas situs phising yang ada di internet. Karena sebagian besar serangan phising biasanya mencuri informasi penting dari pengguna dengan menyamar sebagai situs yang dapat dipercaya. Berdasarkan penelitian terdahulu, teknik deteksi yang paling banyak digunakan adalah analisis situs. Di dalam analisis situs disebutkan bahwa terdapat beberapa pendekatan untuk mendeteksi situs phising antara lain pendekatan berbasis *blacklist, visual similarity*, fitur konten dan URL, dan *third-party search engine* [2]. Sedangkan menurut [4] ada beberapa pendekatan yang dapat digunakan untuk menghandle situs phising antara lain pendekatan berbasis *blacklist, fuzzy rule, machine learning*, CANTINA *(Carnegie Mellon Anti-phishing and Network Analysis Tool)* dan gambar *(visual similarity).*

Beberapa penelitian sebelumnya lebih condong menggunakan pendekatan berbasis fitur konten dan URL untuk mendeteksi situs phising. Contohnya Dongsong Zhang dan kawan-kawan [2] membuat sebuah model klasifikasi untuk mendeteksi website *e-business* phising menggunakan 15 fitur seperti jumlah dot (.), usia domain, sertifikat *e-commerce* dan sebagainya, lalu mengolah datanya menggunakan beberapa algoritma klasifikasi antara lain SMO *(Sequential Minimal Optimization)*, Naive Bayes, Random Forest dan Logistic Regression, sedangkan Yuancheng Li dan kawan-kawan [5] menggunakan 12 fitur untuk mendeteksi situs phising secara global (bukan hanya untuk website *e-business* saja) di antaranya seperti rata-rata *inbound link (link masuk)*, rata-rata *outbound* link (link keluar), rata-rata *internal link* (link dalam) dan lain-lain, kemudian data yang ada diolah menggunakan algoritma BVM *(Ball-based Support Vector Machine)*, SVM *(Support Vector Machine)*, Naive Bayes, Simple Logistic dan beberapa algoritma lain.

Nilai akurasi deteksi yang paling baik didapatkan oleh Dongsong Zhang dan kawan-kawan [2] ketika menggunakan algoritma SMO yaitu 95,38%, sedangkan Yuancheng Li dan kawan-kawan [5] mendapatkan nilai TP *(True Posistive)* terbaik yaitu 0,965 ketika menggunakan algoritma SVM dan Simple Logistic. TP pada penelitian tersebut diartikan sebagai proporsi situs yang benar-benar positif diantara keseluruhan situs yang menunjukkan hasil tes positif. Akan tetapi peneliti pada penelitian tersebut lebih memilih algoritma BVM yang notabene memiliki nilai TP

lebih rendah dari algoritma SVM dan Simple Logistic yaitu kurang lebih sekitar 0.964 atau -0,001. Hal itu dikarenakan perhitungan yang dilakukan oleh algoritma BVM (0,15 detik) lebih cepat dibandingkan algoritma SVM (0,35 detik) maupun Simple Logistic (30 detik).

Sedangkan Neda Abdelhamid dan kawan-kawan [4] menggunakan pendekatan berbasis *fuzzy rule*, *machine learning* dan CANTINA untuk mendeteksi situs phising secara global. Pendekatan berbasis *fuzzy rule* dan *machine learning* sendiri masih termasuk ke dalam pendekatan berbasis fitur konten dan URL, sedangkan CANTINA lebih condong ke pendekatan berbasis *third-party search engine*. Sama seperti penelitian [2] dan [4], peneliti pada penelitian ini juga membandingkan beberapa algoritma klasifikasi antara lain CBA *(Classification Based on Association)*, PART *(hybrid classification)*, C4.5, JRip, MCAC *(Multi-label Classifier based Associative Classification)* dan MCAR *(Multi-class Classification based on Association Rule).* Dimana nilai akurasi dari algoritma MCAC mengungguli nilai akurasi dari algoritma Jrip, C4.5, PART, CBA dan MCAR masing-masing 1,86%, 1,24%, 4,46%, 2,56% dan 0,8%.

Pada penelitian ini, peneliti juga akan melakukan pendekatan berbasis fitur konten dan URL yang mengadopsi *fuzzy rule* dan *machine laearning* untuk membuat model klasifikasi deteksi situs phising di Indonesia. Beberapa fitur-fitur yang akan digunakan dalam penelitian ini diambil dari [2], [4]-[5], akan tetapi sebagian kecil fitur pada penelitian [2] akan dihilangkan karena hanya cocok untuk website *e-business* saja. Padahal model klasifikasi ini diperuntukkan untuk keseluruhan situs yang ada di Indonesia secara global (bukan hanya website *e-business* saja). Selain itu peneliti juga akan menambahkan fitur baru, membuat ekstraksi fitur berbasis *web crawler* dan menguji beberapa algoritma seperti SMO, Naive Bayes, Bagging dan Multilayer Perceptron untuk memastikan bahwa model klasifikasi yang dibuat mampu meningkatkan kinerja deteksi situs phising, sehingga akurasinya jauh lebih akurat bila dibandingkan hanya menggunakan fitur dasar pada penelitian sebelumnya [2].

Software *data mining* yang digunakan dalam pengolahan data pada penelitian ini adalah Weka. Weka adalah software *open source* yang bisa digunakan secara gratis untuk mendukung berbagai tugas standar pada *data mining* antara lain

clustering (pengelompokan), *association* (asosiasi) maupun *classification* (klasifikasi) [6]. Weka berisikan koleksi proses yang meliputi berbagai teknik *pre-processing* dan teknik permodelan data, sehingga dapat membantu peneliti menguji model klasifikasi yang dibuat. Dengan demikian kinerja deteksi yang dihasilkan dapat terukur secara matematis dan validasinya dapat dipercaya, sehingga hasilnya dapat digunakan untuk menunjang penelitian lain yang sejenis dikemudian hari. Tentunya apabila model klasifikasi ini diimplementasikan, maka dapat menghindarkan dan mengurangi resiko pengguna internet terkena serangan *malware* atau *hijacking* dari situs phising.

1.2 Rumusan Masalah

Di bawah ini adalah rumusan-rumusan masalah atau sesuatu yang menjadi pertanyaan-pertanyaan yang harus diselesaikan oleh peneliti pada penelitian yang akan di buat :

a. Bagaimana cara membedakan situs phising dengan situs aslinya?
b. Fitur-fitur apa saja yang digunakan untuk mendeteksi situs phising dan bagaimana cara mendapatkan fitur-fitur tersebut?
c. Bagaimana memastikan bahwa sistem/model yang dibuat mampu meningkatkan kinerja deteksi situs phising?

1.3 Tujuan

Di bawah ini adalah tujuan sekaligus memecahkan rumusan-rumusan masalah yang ada dan menjadi jawaban untuk pertanyaan-pertanyaan pada penelitian ini :

a. Membuat sebuah model klasifikasi yang mampu mendeteksi situs phising.
b. Menentukan fitur-fitur yang relevan berdasarkan pendekatan berbasis konten dan URL untuk meningkatkan kinerja deteksi.
c. Menguji model klasifikasi yang dibuat.

1.4 Ruang Lingkup Penelitian

Untuk membatasi lingkup penelitian yang terlalu luas, pembatasan penelitian dilakukan agar penelitian lebih terarah dan sesuai dengan tujuan penelitian. Di bawah ini adalah batasan dan ruang lingkup dari penelitian ini :

a. Metode pengumpulan data yang dilakukan oleh peneliti dalam penelitian ini yaitu melakukan observasi, membaca dan mencari data di internet, email maupun jurnal Internasional.
b. Data situs phising didapatkan dari penelitian Dongsong Zhang [2], PhishTank, e-mail, dan sumber internet.
c. Data situs non-phising (situs otentik) didapatkan dari penelitian Dongsong Zhang [2], Moz, Alexa dan sumber internet.
d. Situs-situs yang akan digunakan sebagai data training adalah situs berbahasa Indonesia, berserver di Indonesia atau sering diakses oleh pengguna internet yang berasal dari Indonesia.

1.5 Kontribusi Penelitian

Kontribusi dari penelitian ini dibagi menjadi dua yaitu kontribusi secara teoritis dan kontribusi secara praktis. Di bawah ini adalah penjelasan dari masing-masing kontribusi:

1.5.1 Kontribusi Teoritis

Kontribusi dari penelitian ini secara teoritis adalah mengusulkan sebuah model klasifikasi untuk deteksi situs phising di Indonesia berdasarkan pendekatan berbasis fitur konten dan URL yang mana dapat membedakan situs phising dan situs non-phising dengan kinerja yang baik. Sehingga model klasifikasi tersebut dapat digunakan pada penelitian lain untuk membuat sistem deteksi situs phising yang lebih spesifik lagi (contohnya : sistem deteksi bank online phising, sistem deteksi sosial media phising, sistem deteksi situs jual beli phising di Indonesia atau sejenisnya).

1.5.2 Kontribusi Praktis

Sedangkan kontribusi penelitian ini secara praktis yaitu mengusulkan dan mempermudah peneliti pada penelitian selanjutnya dalam pengembangan sistem deteksi situs phising, karena model klasifikasi ini dapat diimplementasikan menjadi sebuah s*ervice*, sehingga informasi yang diberikan dapat menghindarkan pengguna internet terkena serangan *malware* atau *hijacking* dari situs phising maupun mengurangi resiko kerugian finansial dan *data loss* yang ditimbulkan oleh dari situs phising.

1.6 Sistematika Penulisan

Sistematika penulisan dokumen pada laporan penelitian ini dibagi menjadi 5 bab yakni sebagai berikut :

- **BAB 1 PENDAHULUAN**

  Pada bab ini dijelaskan mengenai latar belakang, rumusan masalah, tujuan, ruang lingkup penelitian, kontribusi penelitian, dan sistematika penulisan dalam pemaparan penelitian.

- **BAB 2 LANDASAN TEORI DAN KAJIAN PUSTAKA**

  Pada bab ini dijelaskan mengenai landasan teori dan kajian pustaka dari berbagai penelitian yang memiliki keterkaitan dengan penelitian ini. Kajian pustaka berguna untuk memperkuat dasar dan alasan dilakukannya penelitian ini. Selain kajian pustaka, pada bab ini juga dijelaskan mengenai teori-teori terkait yang bersumber dari buku, jurnal, ataupun artikel di blog yang berfungsi sebagai dasar dalam melakukan penelitian agar dapat memahami konsep atau teori penyelesaian permasalahan yang ada.

- **BAB 3 METODOLOGI PENELITIAN**

  Pada bab ini akan dijelaskan mengenai langkah-langkah penelitian beserta metode yang digunakan. Langkah-langkah penelitian dijelaskan dalam sebuah diagram alur yang sistematis dan akan dijelaskan tahap demi tahap.

- **BAB 4 UJI COBA DAN ANALISIS HASIL**

  Pada bab ini akan dilakukan uji coba terhadap metode yang digunakan. Uji coba ini dilakukan berdasarkan skenario uji coba yang telah dirancang sebelumnya. Selain itu pada bab ini juga dijelaskan mengenai analisis hasil uji coba tersebut.

- **BAB 5 KESIMPULAN DAN SARAN**

  Bab ini berisi kesimpulan dari penelitian ini dan juga saran bagi penelitian mendatang yang berasal dari kekurangan ataupun temuan dari penelitian ini.

(Halaman ini sengaja dikosongkan)

BAB 2
LANDASAN TEORI DAN KAJIAN PUSTAKA

Pada bab ini akan dijelaskan mengenai teori-teori yang mendasari penelitian dan kajian pustaka mengenai penelitian-penelitian yang terkait. Teori yang dijelaskan antara lain mengenai situs phising, klasifikasi, fitur deteksi, kinerja deteksi dan algoritma klasifikasi.

2.1 Situs Phising

Istilah *phishing* dalam bahasa Inggris berasal dari kata "*fishing*" (memancing), dalam hal ini "*fishing"* berarti memancing informasi dan kata sandi pengguna internet. *Phishing* adalah usaha untuk mendapatkan suatu informasi penting dan rahasia secara tidak sah, seperti user id, password, PIN, informasi rekening bank, informasi kartu kredit, atau informasi rahasia yang lain, sedangkan situs phising merupakan sebuah situs yang didesain oleh penjahat internet sedemikian rupanya agar menyerupai situs aslinya (tampilan, konten, URL domain dan sejenisnya) untuk mengelabui korbannya (pengguna internet) dengan membuat korban seolah-olah sedang mengakses halaman situs dari sumber yang sah [1]. Selain itu situs phising juga seringkali digunakan sebagai media penyebaran malware dan penipuan mengatasnamakan situs otentik (situs asli) oleh penjahat internet.

Aktivitas *phishing* tersebut biasanya dilakukan secara sengaja oleh orang dalam, hacker atau penjahat internet yang berhasil menyusupi sebuah website melalui celah keamanan yang ada pada website tersebut, lalu meletakkan halaman *phishing* maupun membuat halaman *phishing* baru yang serupa. Sarana yang sering digunakan oleh orang-orang yang tidak bertanggungjawab tersebut adalah sebagai berikut :

a. Penggunaan alamat e-mail palsu dan grafik untuk menyesatkan pengguna internet, sehingga pengguna internet terpancing menerima keabsahan e-mail atau alamat situs. Agar tampak meyakinkan, penjahat internet juga seringkali memanfaatkan logo atau merk dagang milik lembaga resmi, seperti bank atau

penerbit kartu kredit. Pemalsuan ini dilakukan untuk memancing korban menyerahkan data pribadi, seperti password, PIN dan nomor kartu kredit.

b. Membuat situs jaringan palsu yang sama persis dengan situs resmi, sehingga jika ada pengunjung yang mengisikan data pribadi maka informasi akan direkam oleh pembuat situs palsu tersebut.
c. Membuat hyperlink ke situs jaringan palsu melalui e-mail atau *instant message*.
d. Membuat kloningan situs yang sama persis dengan situs aslinya untuk menyebarkan malware atau melakukan tindakan penipuan.

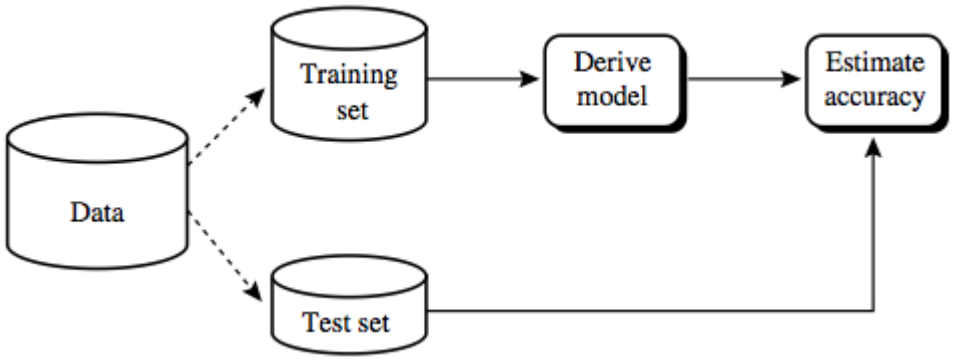
2.2 Klasifikasi

Menurut [7], klasifikasi adalah sebuah metode dari *data mining* yang digunakan untuk memprediksi kategori atau *class* dari suatu *data instances* (data sampel) berdasarkan sekumpulan atribut-atribut dari data tersebut. Atribut yang digunakan mungkin bersifat *categorical* (misalnya golongan darah : "A", "B", "O", dan seterusnya), *ordinal* (misalnya urutan : kecil, sedang, dan besar), integer-valued (misalnya banyaknya suatu kata pada suatu paragraf), atau *real-valued* (misalnya suhu). Kebanyakan algoritma yang menggunakan metode klasifikasi ini hanya menggunakan data yang bersifat diskrit dan untuk data yang bersifat kontinu (*real-valued* dan *integer-valued*) maka data tersebut harus dijadikan diskrit dengan cara memberikan *threshold* (misal lebih kecil dari 5 atau lebih besar dari 10) supaya data dapat terbagi menjadi grup-grup. Sebagai contoh dari metode klasifikasi adalah menentukan e-mail yang masuk termasuk kategori spam atau bukan spam atau menentukan diagnosis dari pasien berdasarkan umur, jenis kelamin, tekanan darah, dan sebagainya.

Algoritma yang mengimplementasikan metode klasifikasi disebut dengan classifier. Istilah "classifier" ini juga terkadang direferensikan sebagai fungsi matematika yang digunakan untuk memetakan input data dengan kategori-kategori tertentu. Cara kerja dari model klasifikasi adalah sebuah proses 2 langkah. Langkah pertama adalah *learning*, pada langkah ini classifier dibangun berdasarkan sekumpulan kelas atau kategori yang sudah ditentukan dari data. Langkah ini disebut *learning step* atau *training step*, dimana sebuah algoritma klasifikasi membangun classifier dengan menganalisis atau belajar dari sebuah *training set*.

Sebuah *tuple*, yang direpresentasikan dengan n-dimensi *attribute vector*, yang menggambarkan n sebagai buah pengukuran yang dibuat pada *tuple* pada *n attribute*. Setiap *tuple* diasumsikan termasuk dalam kelas atau kategori yang sudah ditentukan oleh *attribute* yang disebut dengan *class label attribute*. *Class label attribute* mempunyai nilai diskrit, tidak berurutan dan tiap nilai berfungsi sebagai kelas atau kategori. Langkah pertama dari klasifikasi ini juga sering disebut sebagai *learning of mapping* atau *function*, suatu fungsi pemetaan yang bisa memprediksi class label *y* pada suatu *tuple x*. Pemetaan ini direpresentasikan dalam bentuk *classification rules*, *decision tree* atau formula matematika. Dari *rules* atau *tree* tersebut dapat digunakan untuk mengklasifikasi tuple baru.

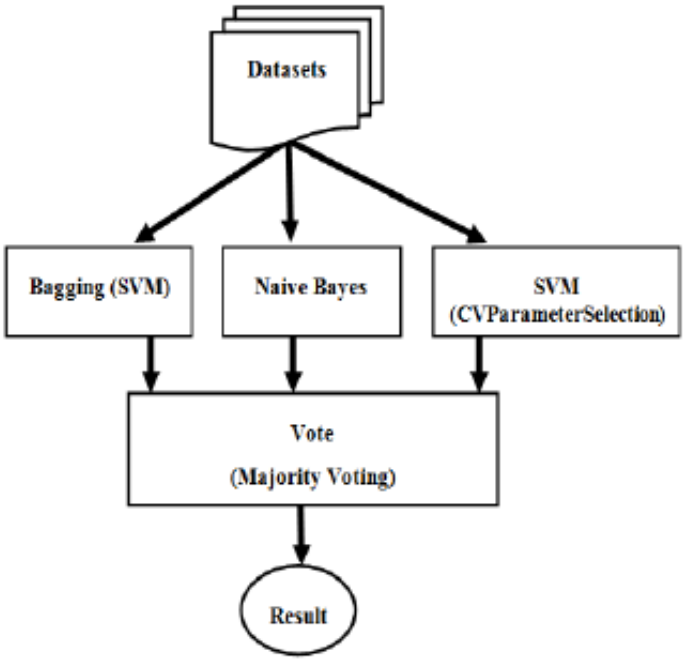
Langkah kedua adalah *classification*. Pada langkah ini, classifier yang sudah dibangun akan digunakan untuk mengklasifikasi data. Dimana akurasi dari prediksi classifier tersebut akan diperkirakan. Jika menggunakan *training set* untuk mengukur akurasi dari classifier, maka estimasi akan optimal karena data yang digunakan untuk membentuk classifier adalah *training set* juga. Oleh karena itu, digunakan *test set*, yaitu sekumpulan *tuple* beserta *class* labelnya yang dipilih secara acak dari dataset. *Test set* bersifat independen dari *training set* dikarenakan *test set* tidak digunakan untuk membangun classifier. Akurasi dari classifier yang diestimasikan dengan *test set* adalah persentase dari *tuple test set* yang diklasifikasi secara benar oleh classifier. *Class label* dari setiap *tuple* dari *test set* dibandingkan dengan prediksi *class label* dari classifier. Jika akurasi dari classifier dapat diterima, maka *classifer* dapat digunakan untuk mengklasifikasi data baru.



Gambar 2.1 Proses Klasifikasi (Jiawei Han, 2006)

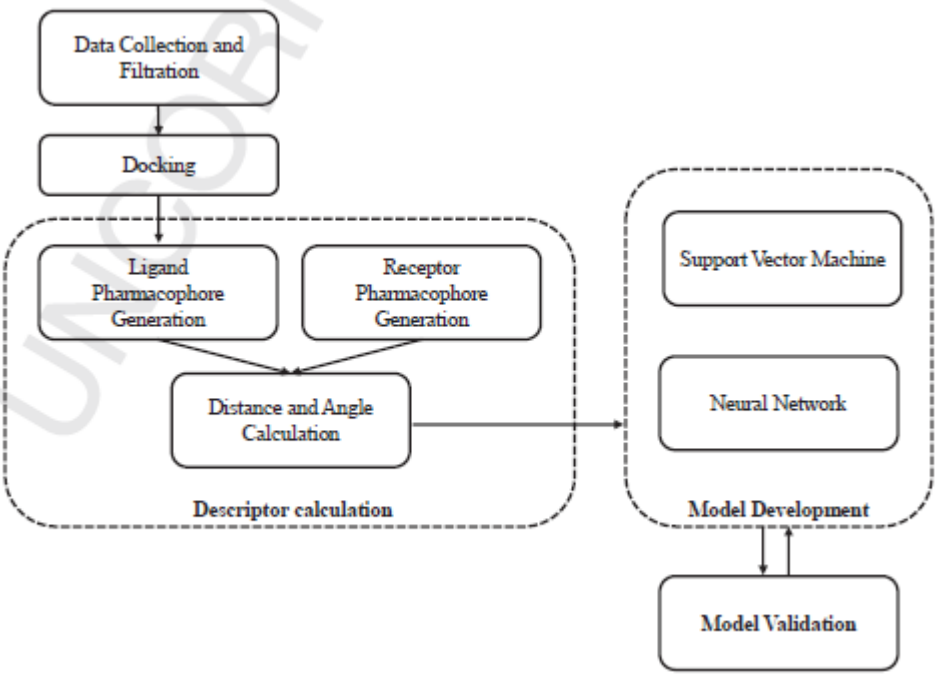
Klasifikasi termasuk *supervised learning* karena *class label* dari setiap *tuple* sudah disediakan. Berbeda dengan *unsupervised learning* di mana *class label* dari *setiap tuple* tidak diketahui. Metode yang menggunakan *unsupervised learning* adalah clustering. Terdapat beberapa algoritma *data mining* yang menggunakan metode klasifikasi antara lain seperti SMO *(Sequential Minimal Optimization)*, BVM *(Ball-based Support Vector Machine)*, Logistic Regression, Naive Bayes, Multilayer Perceptron dan beberapa algoritma lainnya. Untuk contoh dari proses klasifikasi bisa dilihat pada Gambar 2.1. Dimana gambaran atau rangkaian proses dari klasifikasi tersebut sering disebut sebagai model klasifikasi.

Pada penelitian yang dilakukan oleh Cagatay Catal dan kawan-kawan [8] dibuat sebuah model klasifikasi yang dapat digunakan untuk mengklasifikasikan review pelanggan pada blog, forum dan jejaring sosial yang ada di Turki yang mana dikelompokkan berdasarkan sentimen (review negatif, positif dan netral). Pada penelitian tersebut digunakan 3 algoritma antara lain Bagging, Naive Bayes dan SVM *(Support Vector Machine)* untuk melakukan klasifikasi sentimen. Dimana hasil klasifikasi pada ketiga algoritma tersebut akan dikombinasikan menggunakan algoritma Voting untuk mendapatkan kinerja yang lebih baik. Di bawah ini adalah penampakan dari model klasifikasi yang dibuat :



Gambar 2.2 Model Klasifikasi Sentimen (Cagatay Catal, 2017)

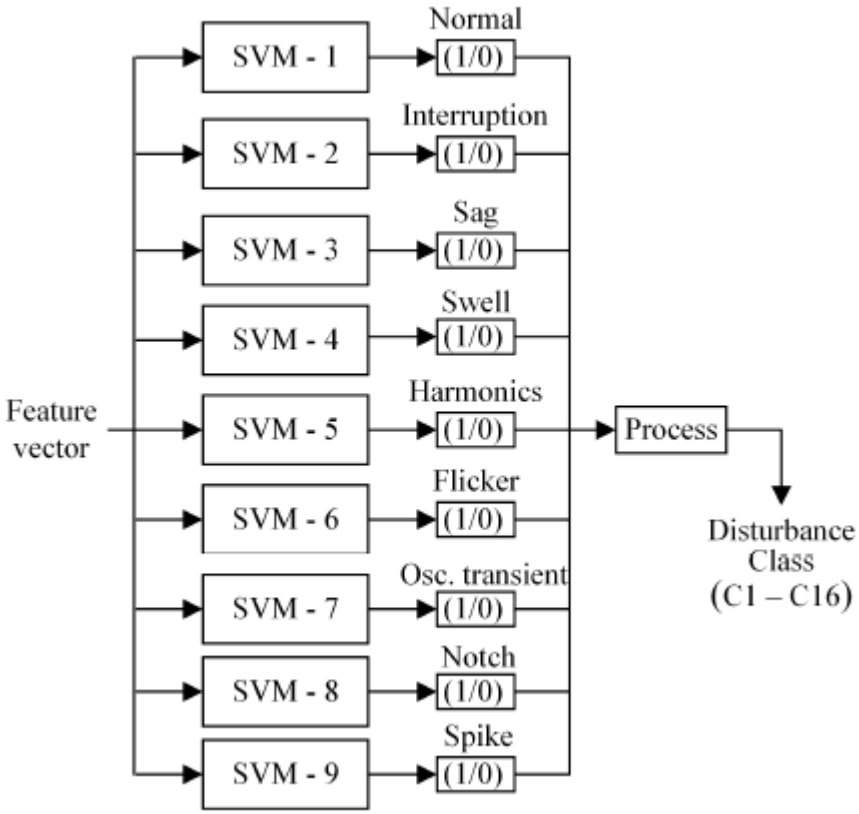
Sejatinya data mining klasifikasi memang bisa digunakan dalam berbagai macam studi kasus termasuk studi kasus di dalam bidang kesehatan. Contohnya seperti penelitian yang dilakukan oleh Kyungro Lee dan kawan-kawan [9]. dimana pada penelitian tersebut dibuat sebuah model klasifikasi untuk memprediksi *activator* pada CAR *(Constitutive Androstane Receptor)* dan menawarkan informasi struktural mengenai interaksi ligan/protein di dalam hati. CAR adalah organ di dalam hati yang berguna untuk mengatur metabolisme dan transportasi obat [9]. Di dalam penelitian ini, peneliti tersebut membangun model klasifikasi menggunakan 2 algoritma yaitu SVM dan NN *(Neural Network)* dan di bawah ini adalah desain model klasifikasi yang dibuat :



Gambar 2.3 Model Klasifikasi Untuk Memprediksi Aktivator Pada CAR (Kyungro Lee, 2016)

Sedangkan pada penelitian lain dibuat sebuah model klasifikasi untuk mendeteksi gangguan pada PQ *(Power Quality)* yang dilakukan oleh Karthik Thirumala dan kawan-kawan [10]. Dimana model klasifikasi yang diusulkan pada penelitian ini hanya berfokus pada SVM saja sebagai classifier tunggal untuk

mengklasifikasikan gangguan PQ skala kecil maupun besar seperti yang terlihat pada gambar di bawah ini :



Gambar 2.4 Model Klasifikasi Berbasis SVM (Karthik Thirumala, 2017)

2.3 Penelitian Terkait

Menurut [2], sistem deteksi phishing muncul sebagai mekanisme penting untuk memberantas situs phising yang ada di internet. Karena sebagian besar serangan *phishing* mencuri informasi penting dari pengguna dengan menyamar sebagai situs yang dapat dipercaya. Berdasarkan penelitian terdahulu, teknik deteksi yang paling banyak digunakan adalah analisis situs. Dalam penelitian tersebut disebutkan bahwa terdapat beberapa pendekatan untuk mendeteksi situs phising, peneliti mengelompokkan pendekatan tersebut menjadi 4 kelompok yaitu :

a. Pendekatan berbasis *blacklist*.

   Pendekatan berbasis *blacklist* bergantung pada list URL situs phising yang diketahui. Jika URL situs target sesuai dengan URL salah satu situs phising yang dikenal pada daftar *blacklist*, maka situs tersebut akan diberi label sebagai situs

phising. PhishNet misalnya, menggunakan sebuah algoritma pencocokan untuk membedah URL ke dalam beberapa komponen yang cocok secara individu terhadap URL situs phishing yang diketahui di dalam daftar *blacklist*. Pendekatan berbasis *blacklist* adalah metode deteksi situs phising yang paling sederhana, sehingga pendekatan ini tidak memiliki kemampuan untuk mendeteksi situs phishing baru. Selain itu, pendeketan berbasis *blacklist* membutuhkan update secara berkala agar mampu mendeteksi situs phising terbaru.

b. Pendekatan berbasis *visual similarity*.

   Pendekatan berbasis *visual similarity* menentukan situs phishing menggunakan pencocokan gambar/tampilan (konten, warna, block boundary, font). Biasanya pendekatan berbasis *visual similarity* membagi isi sebuah website menjadi sejumlah bagian, lalu menganalisis dan membandingkan kesamaan antara ciri-ciri visual dari bagian-bagian tersebut dengan situs aslinya.

c. Pendekatan berbasis fitur konten dan URL.

   Pendekatan berbasis fitur konten dan URL berfokus pada analisis karakteristik konten dan URL dari situs target. Tantangan utama yang dihadapi bila menggunakan pendekatan berbasis fitur konten dan URL adalah kesulitan dalam menentukan fitur konten dan URL yang berpengaruh terhadap deteksi itu sendiri.

d. Pendekatan berbasis *third-party search engine.*

   Tipe pendekatan lainnya yaitu pendekatan berbasis *third-party search engine*. Cara kerja dari pendekatan ini adalah mencari informasi yang relevan tentang URL pada mesin pencari (contoh : *google.com*, *bing.com* dan sejenisnya), kemudian menggunakan hasil pencarian untuk membuat keputusan deteksi. Dalam pendekatan ini, URL dari situ yang ditargetkan akan digunakan sebagai kueri pencarian. Jumlah dan ranking situs yang dikembalikan oleh mesin pencari akan digunakan untuk klasifikasi.

Pada penelitian yang dilakukan oleh Dongsong Zhang dan kawan-kawan [2] dibuat sebuah model klasifikasi yang dapat mendeteksi website *e-business* phising dengan melibatkan fitur domain yang unik. Model yang diusulkan tidak bergantung

pada pengetahuan sebelumnya atau asumsi-asumsi mengenai situs otentik (non-phising). Model dalam penelitian tersebut lebih mengutamakan adopsi pedekatan berbasis fitur konten dan URL, karena pendekatan ini adalah pendekatan yang paling umum digunakan, sebab pendekatan ini mampu menggabungkan dan mengevaluasi fitur deteksi pada sebuah domain. Dengan mengintegrasikan fitur baru pada website *e-business* dengan beberapa prediksi fitur deteksi yang digunakan pada penelitian sebelumnya, dibuat sebuah vektor fitur untuk model yang diusulkan. Di bawah ini adalah fitur-fitur yang dimaksud :

- F1 : Apakah URL berisi sebuah alamat IP?

  Biasanya sebuah website phising menggunakan alamat IP. Jika URL menggunakan alamat IP, maka F1 akan diberi nilai 1, jika tidak, maka F1 diberi nilai 0.
- F2 : Apakah URL berisi simbol '@'?

  Biasanya website phising akan menggunakan simbol '@' di dalam URL. Jika URL menggunakan simbol '@', maka F2 akan diberi nilai 1, jika tidak, maka F2 diberi nilai 0.
- F3 : Apakah karakter dalam URL dikodekan ke dalam UNICODE?

  Biasanya website phising menggunakan URL yang dikodekan ke dalam UNICODE untuk menyembunyikan URL aslinya. Jika URL dikodekan ke dalam UNICODE, maka F3 akan diberi nilai 1, jika tidak, maka F3 diberi nilai 0.
- F4 : Jumlah dot (.) dalam URL

  Pada penelitian terdahulu [11] menyatakan bahwa semakin banyak dot dalam sebuah URL, maka semakin besar kemungkinan website tersebut terindikasi sebagai website phising.
- F5 : Jumlah sufiks (imbuhan di belakang) dalam nama domain

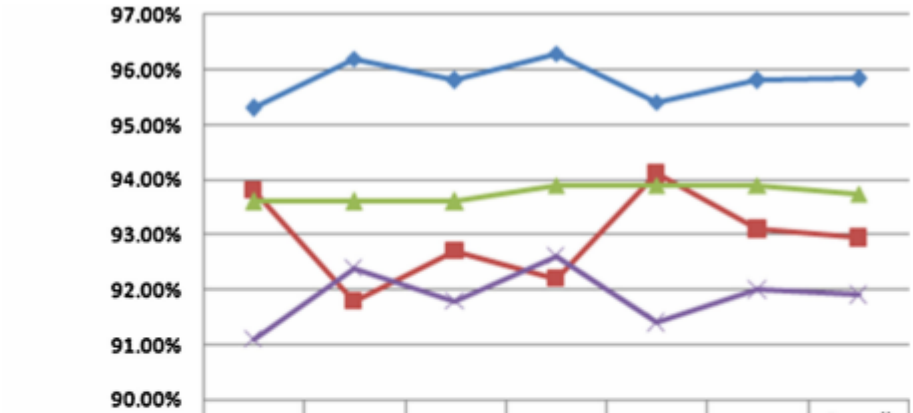
  Biasanya website phising menggunakan 2 sufiks domain, pada umumnya pengguna internet hanya akan melihat bagian pertama dan mengabaikan bagian lainnya, sehingga akan menuju website phising.
- F6 : Usia domain

  Usia domain dihitung sejak domain diregistrasi oleh *registrar*.
- F7 : Masa aktif domain *(expired)*

Jumlah hari yang dihitung adalah jumlah hari sebelum masa aktif domain tersebut berakhir.

- F8 : Apakah domain memiliki DNS *(Domain Name Server)*?
  DNS adalah alamat di mana domain tersebut dihostingkan. Jika domain memiliki DNS, maka F8 akan diberi nilai 1, jika tidak, maka F8 diberi nilai 0.
- F9 : Apakah website memiliki informasi pendaftaran domain (WHOIS)?
  Jika website memiliki informasi tersebut, maka F9 akan diberi nilai 1, jika tidak, maka F9 diberi nilai 0.
- F10 : Apakah domain didaftarkan oleh perusahaan?
  Jika domain didaftarkan oleh perusahaan, maka F10 akan diberi nilai 1, jika tidak, maka F10 diberi nilai 0.
- F11 : Apakah domain diprivasi?
  Jika domain diprivasi, maka F11 akan diberi nilai 1, jika tidak, maka F11 diberi nilai 0.
- F12 : Apakah di dalam website tercantum nomor lisensi ICP *(Internet Content Provider)*?
  Jika website mencantumkan nomor lisensi ICP, maka F12 akan diberi nilai 1, jika tidak, maka F12 diberi nilai 0.
- F13 : Jumlah *dead link* (link mati) dalam website
  Dalam penelitian terdahulu menyatakan bahwa website phising memiliki jumlah link mati lebih banyak dibandingkan website aslinya.
- F14 : Jumlah *outbound link* (link keluar) dalam website
  Normalnya setiap website pasti memiliki link keluar, akan tetapi jika jumlah link keluar terlalu banyak, website tersebut patut dicurigai sebagai website phising.
- F15 : Apakah di dalam website *e-business* terdapat informasi sertifikat *e-commerce*?

Jika website menampilkan informasi sertifikat *e-commerce,* maka F15 akan diberi nilai 1, jika tidak, maka F15 diberi nilai 0.



	Phishing Websites(n=425)			Real Websites(n=439)			Overall Accuracy (n=864)
	Precision	Recall	F1	Precision	Recall	F1	
SMO	0.953	0.962	0.958	0.963	0.954	0.958	95.83%
Naïve Bayes	0.938	0.918	0.927	0.922	0.941	0.931	92.94%
RandomForest	0.936	0.936	0.936	0.939	0.939	0.939	93.75%
Logistic	0.911	0.924	0.918	0.926	0.914	0.92	91.90%

Gambar 2.5 Hasil Perbandingan Algoritma Klasifikasi Deteksi Website *E-Business* Phising (Dongsong Zhang, 2014)

Peneliti pada penelitian tersebut membandingkan 4 algoritma antara lain SMO *(Sequential Minimal Optimization),* Naive Bayes, Random Forest dan Logistic Regression (Gambar 2.5). Hasil dari penelitian tersebut menyatakan bahwa algoritma SMO memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan tiga algoritma lainnya. SMO memiliki tingkat akurasi 95,83% diikuti oleh Random Forest, Naive Bayes dan Logistic Regression masing-masing 93,75%, 92,94% dan 91,90%. Nilai *Precision* untuk algoritma SMO adalah 0,953 dengan nilai *Recall* sebesar 0,962 dan *F-Measure* 0,958. Perlu diketahui bahwa *Precision* adalah pengukuran tingkat ketepatan dalam kegiatan penelusuran [11], sedangkan *Recall* merupakan rasio jumlah dokumen relevan yang ditemukan kembali dengan total jumlah dokumen dalam kumpulan dokumen yang dianggap relevan [12] dan *F-Measure* merupakan salah satu perhitungan evaluasi dalam pemulihan informasi yang mengkombinasikan *Recall* dan *Precision* [13].

Dimana algoritma dengan hasil kinerja terbaik (SMO) akan digunakan pada model klasifikasi yang dibuat untuk dibandingkan dengan model klasifikasi yang dibuat oleh [15] dan [16] berdasarkan hipotesa yang telah ditentukan sebelumnya :

- H1 : Fitur berbasis konten dan URL pada model klasifikasi yang diusulkan untuk deteksi website *e-business* akan mengungguli model klasifikasi yang menggunakan fitur tradisional bila dilihat dari segi *Precision*.
- H2 : Fitur berbasis konten dan URL pada model klasifikasi yang diusulkan untuk deteksi website *e-business* akan mengungguli model klasifikasi yang menggunakan fitur tradisional bila dilihat dari segi *Recall*.
- H3 : Fitur berbasis konten dan URL pada model klasifikasi yang diusulkan untuk deteksi website *e-business* akan mengungguli model klasifikasi yang menggunakan fitur tradisional bila dilihat dari segi *F-Measure*.

Sedangkan di dalam penelitian yang berjudul "*A Minimum Enclosing Ball-Based Support Vector Machine Approach For Detection Of Phishing Websites*" yang dikerjakan oleh Yuancheng Li dan kawan-kawan [5] digunakan 12 jenis indikator sebagai fitur topologi website. Fitur topologi website sendiri masih termasuk di dalam pendekatan berbasis fitur konten dan URL. Peneliti tersebut menggunakan *web crawler* untuk mengekstrak 12 fitur topologi website tersebut menjadi *DOM (Document Object Model) tree*. Detail fitur topologi website pada penelitian tersebut antara lain sebagai berikut :

c. Jumlah total halaman web yang terdapat dalam website

   Jumlah total halaman web yang terkandung di dalam situs yang dianalisis.

d. Rata-rata jumlah *inbound links* (link masuk)

   Tautan (link) yang mengarah ke halaman situs yang dianalisa yang berasal dari situs lain sering disebut sebagai *inbound links* (link masuk). Jumlah rata-rata link masuk dapat dihitung dengan rata-rata jumlah link masuk di semua situs yang terdeteksi.

e. Rata-rata jumlah *outbound links* (link keluar)

   Biasanya sebuah situs memiliki tautan (link) yang mengarah ke halaman situs lain yang mana tautan-tautan tersebut sering disebut sebagai *outbound links* (link

keluar). Jumlah rata-rata link keluar dapat dihitung dengan rata-rata jumlah link keluar di dalam situs yang dianalisis.

d. Rata-rata jumlah *internal links* (link dalam)

   Biasanya sebuah situs memiliki tautan (link) yang mengarah ke halaman lain di dalam situs yang mana tautan-tautan tersebut sering disebut sebagai *internal links* (link dalam). Jumlah rata-rata link dalam dapat dihitung dengan rata-rata jumlah link dalam pada situs yang dianalisis.

e. Rata-rata jumlah gambar

   Jumlah rata-rata gambar yang terkandung di dalam semua halaman web yang dianalisis.

f. Rata-rata jumlah file CSS

   Jumlah rata-rata file CSS yang terkandung di dalam semua halaman web yang dianalisis.

g. Rata-rata jumlah file JS

   Jumlah rata-rata file JS (JavaScript) yang terkandung di dalam semua halaman web yang dianalisis.

h. Rata-rata jumlah form

   Jumlah rata-rata tag <form> pada situs yang dianalisis.

i. Rata-rata jumlah input box selain password

   Jumlah rata-rata tag <input> selain password pada situs yang dianalisis.

j. Rata-rata jumlah input box bertipe password

   Jumlah rata-rata tag <input> bertipe password pada situs yang dianalisis.

k. Proporsi dari link form

   Jumlah link form dari website yang menunju ke halaman lain dibagi dengan jumlah total link form halaman web.

l. Proporsi halaman web dinamis

   Halaman web dinamis berakhiran .php, .aspx, dan .jsp, sedangkan halaman web statis berakhiran .htm dan .html. Rasio jumlah halaman web dinamis dengan jumlah total halaman web.

Pada penelitian tersebut, peneliti juga membandingkan beberapa algoritma antara lain Bayes Nets, Naive Bayes, Logistic Regression, RBFN *(Radial Basis Function Network)*, Simple Logistic, Decision Table, Decision Stump, SVM *(Support Vector Machine)* dan BVM *(Ball-based Support Vector Machine)* seperti yang terlihat pada Gambar 2.6. Akan tetapi yang unik dari penelitian tersebut adalah peneliti memilih algoritma dengan *training time* yang paling cepat yaitu BVM (unggul 0,197 detik dari SVM) tanpa memperdulikan nilai TP *(True Positive)*, FP *(False Positive)*, *Presicion*, *Recall* atau bahkan *F-Measure*. TP pada penelitian tersebut diartikan sebagai proporsi situs yang benar-benar positif diantara keseluruhan situs yang menunjukkan hasil tes positif, sedangkan FP adalah persentase dari semua situs yang benar-benar negatif diantara semua situs yang menunjukkan hasil tes negatif.

Classifier	TP Rate	FP Rate	Precision	Recall	*F*-value	Training time(s)
Bayes net	0.962	0.039	0.963	0.962	0.962	5
Native bayes	0.960	0.041	0.961	0.960	0.960	2
Logistic	0.962	0.039	0.962	0.962	0.962	6
RBF network	0.925	0.076	0.931	0.925	0.925	9
Simple logistic	0.965	0.035	0.967	0.965	0.965	30
Decision table	0.954	0.046	0.955	0.954	0.954	1.3
Decision stump	0.951	0.050	0.951	0.951	0.951	0.2
SVM	0.965	0.037	0.995	0.964	0.963	0.347
BVM	0.964	0.037	0.996	0.964	0.963	0.15

Gambar 2.6 Hasil Perbandingan Algoritma Klasifikasi Deteksi Website Phising (Yuancheng Li, 2016)

Pada penelitian yang sejenis yang dilakukan oleh Neda Abdelhamid dan kawan-kawan [4] yang berjudul "*Phishing Detection Based Associative Classification Data Mining*" dikatakan ada empat pendekatan yang dapat digunakan untuk mendeteksi situs phising antara lain pendekatan berbasis *blacklist*, *fuzzy rule*, *machine learning*, CANTINA *(Carnegie Mellon Anti-phishing and Network Analysis Tool)* dan gambar *(visual similarity)*. Peneliti pada penelitian ini mengusulkan 16 fitur berbasis pendekatan *fuzzy rule*, *machine learning* dan CANTINA untuk mendeteksi situs phising dan menggunakan 3 indikator untuk mengelompokkan situs phising yaitu *Legit* (situs non-phising), *Suspicious* (terindikasi sebagai situs phising) dan *Phishy* (situs phising). Dimana pendekatan berbasis *fuzzy rule* dan *machine learning* tersebut masih berbau pendekatan berbasis fitur konten dan URL, sedangkan pendekatan berbasis CANTINA lebih

condong ke pendekatan berbasis *third-party search engine*. Di bawah ini adalah fitur-fitur yang dimaksud :

- F1 : Apakah URL berisi sebuah alamat IP?

  Biasanya sebuah website phising menggunakan alamat IP. Jika URL menggunakan alamat IP, maka F1 akan diberi keterangan *"Phishy"*, jika tidak, maka F1 diberi keterangan *"Legit"*.
- F2 : Panjang URL

  Situs phising biasanya memiliki panjang URL yang tidak lazim. Jika panjang URL < 54 karakter, maka F2 akan diberi keterangan *"Legit"*, jika panjang URL >= 54, maka F2 diberi keterangan *"Suspicious"*, selain itu, maka F2 diberi keterangan *"Phishy"*.
- F3 : Apakah URL berisi simbol '@'?

  Biasanya website phising akan menggunakan simbol '@' di dalam URL. Jika URL menggunakan simbol '@', maka F3 akan diberi keterangan *"Phishy"*, jika tidak, maka F3 diberi keterangan *"Legit"*.
- F4 : Apakah URL memiliki prefiks (imbuhan di depan) atau sufiks (imbuhan di akhir)?

  Jika URL mengandung karakter "-", maka F4 akan diberi keterangan *"Phishy"*, jika tidak, maka F4 diberi keterangan *"Legit"*.
- F5 : Jumlah sub domain

  Jika jumlah dot (.) < 3, maka F5 akan diberi keterangan *"Legit"*, jika jumlah dot (.) = 3, maka F5 diberi keterangan *"Suspicious"*, selain itu, maka F5 diberi keterangan *"Legit"*.
- F6 : Apakah website menggunakan https?

  Pada dasarnya setiap situs besar pasti memiliki protokol keamanan yang disebut https. Jika website memiliki https, https-nya disertai sertifikat SSL *(Secure Socket Layer)* dan umur SSL-nya >= 2 tahun, maka F6 akan diberi keterangan *"Legit"*, jika website memiliki https dan https-nya tidak disetai sertifikat SSL, maka F6 diberi keterangan *"Suspicious"*, selain itu, maka F6 diberi keterangan *"Phishy"*.
- F7 : Presentase permintaan URL

Semakin banyak permintaan URL (gambar, video, objek atau file) yang berasal dari luar situs dapat mengindikasikan bahwa situs tersebut adalah situs phising. Jika presentase permintaan URL < 22%, maka F6 akan diberi keterangan *"Legit"*, jika presentase permintaan URL >= 22% dan < 61%, maka F6 diberi keterangan *"Suspicious"*, selain itu, maka F6 diberi keterangan *"Phishy"*.

- F8 : Presentase *URL of anchor*

  URL of anchor yang dimaksud pada penelitian ini yaitu *outbound links* (link keluar). Jika presentase URL of anchor < 31%, maka F8 akan diberi keterangan *"Legit"*, jika presentase URL of anchor >= 32% dan <= 67%, maka F8 diberi keterangan *"Suspicious"*, selain itu, maka F8 diberi keterangan *"Phishy"*.
- F9 : Apakah website memiliki SFH *(Server Form Handler)*?

  Pada dasarnya situs phising tidak meneruskan informasi ke server situs yang sah karena tidak memiliki SFH. Jika website tidak memiliki SFH, maka F9 akan diberi keterangan *"Phishy"*, jika website memiliki SFH dan diarahkan ke domain lain, maka F9 diberi keterangan *"Suspicious"*, selain itu, maka F9 diberi keterangan *"Legit"*.
- F10 : Apakah URL tidak normal?

  Jika identitas website *(hostname)* tidak sesuai dengan WHOIS, maka website tersebut dapat dikategorikan sebagai website phsiing. Jika tidak ada hostname dalam URL, maka F10 akan diberi keterangan *"Phishy"*, jika ada, maka F10 diberi keterangan *"Legit"*.
- F11 : Apakah website menggunakan jendela *pop-up*?

  Pada dasarnya situs otentik tidak menyuruh pengguna internet untuk memasukkan informasi pribadinya melalui jendela *pop-up*. Jika tidak bisa diklik kanan, maka F11 akan diberi keterangan *"Phishy"*, jika bisa diklik kanan tetapi muncul pesan peringatan, maka F11 diberi keterangan *"Suspicious"*, selain itu, maka F11 diberi keterangan *"Legit"*.
- F12 : Jumlah halaman *redirect*

  Jika jumlah halaman *redirect* >= 1, maka F12 akan diberi keterangan *"Legit"*, jika jumlah halaman *redirect* > 1 dan < 4, maka F12 diberi keterangan *"Suspicious"*, selain itu, maka F12 diberi keterangan *"Phishy"*.
- F13 : Apakah domain memiliki DNS *(Domain Name Server)*?

DNS adalah alamat di mana domain tersebut dihostingkan. Jika domain tidak memiliki DNS, maka F10 akan diberi keterangan *"Phishy"*, jika tidak, maka F10 diberi keterangan *"Legit"*.

- F14 : Apakah website menyembunyikan link?

  Jika mouse diarahkan ke link pada situs dan terjadi perubahan, maka F14 akan diberi keterangan *"Phishy"*, jika mouse diarahkan ke link pada dan tidak terjadi apa-apa, maka F14 diberi keterangan *"Suspicious"*, selain itu, maka F14 diberi keterangan *"Legit"*.
- F15 : Pengunjung website

  Indikator yang digunakan untuk mengukur pengunjung pada penelitian ini adalah Alexa Rank. Jika Alexa Rank < 150.000, maka F15 akan diberi keterangan *"Legit"*, jika Alexa Rank > 150.000, maka F15 diberi keterangan *"Suspicious"*, selain itu, maka F15 diberi keterangan *"Legit"*.
- F16 : Usia domain

  Usia domain dihitung sejak domain diregistrasi oleh *registrar*. Jika usia domain <= 6 bulan, maka F16 akan diberi keterangan *"Phishy"*, jika lebih, maka F16 diberi keterangan *"Legit"*.

Sedangkan di bawah ini adalah tabel rasio fitur yang mengindikasikan bahwa situs tersebut adalah situs phising menurut [4] :

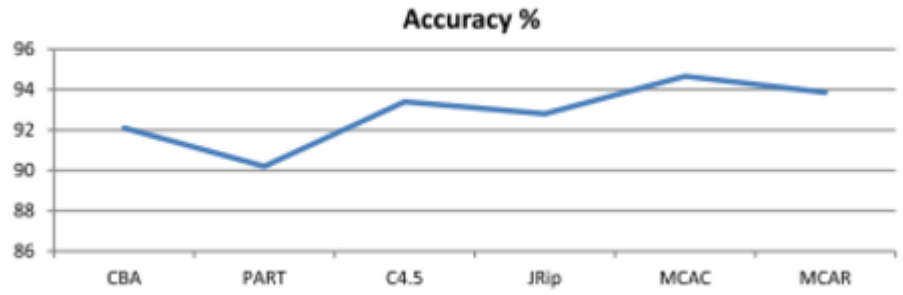
Tabel 2.1 Rasio Fitur

Fitur	Rasio
Alamat IP	20,5%
Panjang URL	51%
Simbol '@'	6,8%
Prefiks atau sufiks	25,4%
Sub domain	42,8%
HTTPS	89,2%
Permintaan URL	100%
URL of anchor	5,7%

SFH	22,3%
URL tidak normal	20,5%
Jendela pop-up	14,3%
Halaman redirect	11%
DNS	7,6%
Link tersembunyi	21%
Pengunjung website	93,2%
Usia domain	97,4%

Sumber : Neda Abdelhamid, 2014

Sama seperti penelitian [2] dan [5], peneliti dalam penelitian tersebut juga membandingkan beberapa algoritma klasifikasi untuk mendapatkan algoritma dengan akurasi terbaik (Gambar 2.7). Hasilnya MCAC *(Multi-label Classifier based Associative Classification)* menghasilkan nilai akurasi tertinggi bila dibandingkan dengan algoritma lain seperti CBA *(Classification Based on Association)*, PART *(hybrid classification)*, C4.5, JRip dan MCAR *(Multi-class Classification based on Association Rule)*. MCAC sendiri merupakan sebuah algoritma yang mana mempelajari *rules* yang terkait dengan beberapa *class* dari data label tunggal. Algoritma MCAC mengekstrak classifier dari keseluruhan data training yang mana mengandung semua *class* yang mungkin terhubung dengan *rules* selama classifier memiliki representasi *data training* [16].



Gambar 2.7 Hasil Perbandingan Algoritma Klasifikasi Deteksi Situs Phising (Nelda Abdelhamid, 2014)

2.4 Fitur Deteksi

Fitur merupakan karakteristik yang mungkin dimiliki atau tidak dimiliki oleh objek [17], sedangkan deteksi merupakan suatu cara untuk mengetahui jenis masalah atau cara untuk menyelesaikan suatu permasalahan untuk membuat keputusan maupun pengambilan kesimpulan [18]. Sehingga fitur deteksi dapat diartikan sebagai karakteristik-karakteristik atau kriteria-kriteria yang dapat digunakan untuk mengetahui jenis masalah atau cara untuk menyelesaikan permasalahan tersebut.

Di era modernisasi, deteksi seringkali digunakan dalam beberapa teknologi dalam bidang pencitraan seperti deteksi fitur wajah maupun deteksi objek. Selain itu jika mengacu pada era globalisasi dan teknologi yang berkembang sangat pesat, secara teoritis dan praktis sistem deteksi juga dapat digunakan dalam berbagai studi kasus berbasis online, contohnya deteksi situs phising. Deteksi situs phising berfungsi sebagai pengambil keputusan untuk menentukan situs mana yang termasuk situs phising dan situs non-phising. Sehingga hasil dari deteksi dapat dimanfaatkan sebagai alternatif pencegahan terhadap serangan malware atau pencurian data melalui situs phising yang dilakukan oleh penjahat internet.

Sebelum melakukan deteksi, proses awal yang biasanya harus dilakukan adalah melakukan pemilihan fitur deteksi untuk menentukan kriteria-kriteria yang mengindikasikan bahwa situs tersebut merupakan situs phising. Pemilihan fitur sendiri merupakan salah satu tahapan yang dilakukan untuk mengurangi dimensi data dan fitur-fitur yang tidak relevan. Pemilihan fitur yang relevan akan meningkatkan efektifitas dan efisiensi kinerja dari algoritma klasifikasi [19]. Contoh fitur pada studi kasus deteksi situs phising antara lain yaitu jumlah sufiks (imbuhan), jumlah . (dot), usia domain dan lain-lain.

2.5 Kinerja Deteksi

Menurut [21] *performance* atau kinerja merupakan hasil atau keluaran dari suatu proses. Dengan kata lain kinerja deteksi situs phising merupakan hasil atau keluaran dari proses klasifikasi situs phising dan situs non-phising. Dalam model klasifikasi menggunakan software Weka, kinerja dari model klasifikasi dapat dilihat dari *Correctly Classified Instances*, *Precision*, *Recall*, *F-Measure* dan

training time. *Correctly Classified Instances* atau akurasi adalah ukuran seberapa dekat suatu hasil pengukuran dengan nilai yang benar atau diterima dari kuantitas yang diukur [22]. Untuk *Precision* lebih condong sebagai pengukuran tingkat ketepatan dalam kegiatan penelusuran [11], sedangkan *Recall* merupakan rasio jumlah dokumen relevan yang ditemukan kembali dengan total jumlah dokumen dalam kumpulan dokumen yang dianggap relevan [12] dan *F-Measure* merupakan salah satu perhitungan evaluasi dalam pemulihan informasi yang mengkombinasikan *Recall* dan *Precision* [13].

Untuk memastikan bahwa kinerja dari suatu proses klasifikasi meningkat atau berkembang lebih baik dari sebelumnya adalah nilai dari *Correctly Classified Instances*, *Precision*, *Recall* dan *F-Measure* harus berdekatan satu sama lain dan nilainya lebih besar dari nilai sebelumnya. Selain itu *time* dalam *pre-processing* juga mempengaruhi kinerja dalam model klasifikasi. Sebagai contoh, apabila model klasifikasi deteksi situs phising mampu mengidentifikasi sebuah situs dan memberikan keputusan dengan waktu yang sangat singkat, hal itu juga bisa disebut sebagai peningkatan kinerja. Tidak harus semua elemen yang meningkat tetapi salah satu atau sebagian besar yang meningkat juga bisa disebut sebagai peningkatan kinerja.

2.6 Algoritma Pada Klasifikasi

Di bawah ini adalah penjelasan mengenai beberapa algoritma yang ada pada data mining klasifikasi :

2.6.1 SMO *(Sequential Minimal Optimization)*

Algoritma pada klasifikasi biasanya selalu identik dengan SVM *(Support Vector Machine)*. SVM sendiri merupakan salah satu metode *supervised learning* yang biasanya digunakan untuk klasifikasi data dan pada umumnya diimplementasikan untuk menangani dataset yang hanya memiliki dua kelas [23]-[24]. Dalam rangka memisahkan data terhadap kelasnya, SVM akan membangun sebuah *hyperplane* (bidang pemisah). Sebuah *hyperplane* (bidang pemisah) yang baik, bukan hanya *hyperplane* yang bisa digunakan untuk memisahkan data, akan tetapi *hyperplane* yang baik adalah *hyperplane* yang memiliki batasan *(margin)*

yang paling besar. Pencarian bidang pemisah terbaik inilah yang menjadi inti dari SVM. Akan tetapi, dalam proses pencarian *hyperplane* tersebut, akan muncul permasalahan baru yaitu sebuah formula yang sangat sulit untuk dipecahkan, yang disebut dengan permasalahan QP *(Quadratic Programming)*.

Permasalahan QP ini sangat sulit sekali untuk diselesaikan, apalagi jika masih menggunakan *primal form* dalam rangka pencarian *hyperplane* terbaik. Salah satu teknik penyelesaian QP yang paling sering digunakan adalah *lagrange multiplier*. DImana bentuk *primal form* yang tadinya sangat susah untuk dipecahkan, akan dirubah kedalam bentuk *dual form* yang hanya akan mengandung nilai α. Berbagai algoritma telah dikembangkan untuk mencari nilai α tersebut. Akan tetapi, algoritma-algoritma tersebut memerlukan waktu yang lama, apalagi jika dipakai untuk data yang berukuran besar, karena algoritma tersebut menggunakan *numerical quadratic programming* sebagai *inner loop* [25].

Oleh karena itu, dibuatlah sebuah algoritma yang dapat menangani masalah pemecahan nilai α tersebut. Algoritma ini disebut algoritma SMO *(Sequential Minimal Optimization)*. Berbeda dengan algoritma lainnya, SMO menggunakan *analytic quadratic programming* sebagai *inner* loop-nya. Algoritma ini dapat memecahkan masalah tersebut dengan cara menggunakan dua buah data pada setiap langkahnya. SMO menggunakan dua buah data pada setiap iterasinya sehingga pencarian solusi optimal dapat dilakukan. Hal ini tentunya akan mengakibatkan jumah iterasi semakin bertambah, akan tetapi karena waktu yang dibutuhkan dalam setiap iterasi sangat kecil maka waktu total pelatihan menjadi lebih singkat [26].

SMO secara luas digunakan untuk mendukung pelatihan SVM dan diimplementasikan menggunakan LIBSVM. LIBSVM adalah sebuah *open source library* untuk SVM yang bisa ditemukan pada software *data mining* seperti Weka. Publikasi algoritma SMO pada tahun 1998 telah menghasilkan banyak kegembiraan dalam komunitas SVM, bila dibandingkan dengan algoritma lainnya pada SVM, SMO jauh lebih simpel dan tidak memerlukan sumber daya yang cukup besar untuk memecahkan QP. Mengacu pada masalah klasifikasi biner dengan dataset (x_1, y_1), ..., (xn, yn), dimana xi adalah vektor input dan $yi \in \{-1, 1\}$ adalah label biner yang sesuai untuk bentuk itu, maka sebuah *hyperplane* dalam *SVM* akan dilatih dengan

memecahkan masalah QP yang dinyatakan dalam rumus sebagai berikut :

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j K(x_i, x_j) \alpha_i \alpha_j,$$

menjadi

$$0 \leq \alpha_i \leq C, \quad \text{for } i = 1, 2, \ldots, n,$$
$$\sum_{i=1}^{n} y_i \alpha_i = 0$$

Dimana C adalah *hyperparameter* SVM dan *k (xi, xj)* adalah fungsi kernel, keduanya disediakan oleh pengguna dan variabel α_1 adalah *lagrange multipliers*. SMO membuat serangkaian terkecil (sub-masalah) untuk menyelesaikan masalah yang kemudian diselesaikan secara analitis. Karena *equality constraint* linear melibatkan *lagrange multipliers* α_1, maka kemungkinan terkecil masalah yang terjadi bisa melibatkan kedua *multipliers.* Kemudian untuk menghindari hal tersebut, rumus α_1 dan α_2 disederhanakan menjadi seperti di bawah ini :

$$0 \leq \alpha_1, \alpha_2 \leq C,$$
$$y_1 \alpha_1 + y_2 \alpha_2 = k,$$

Sehingga dengan adanya penyederhanaan ini, masalah dapat diselesaikan secara analitis. Salah satu yang dibutuhkan untuk menemukan minimum dari fungsi kuadrat satu dimensi *k* adalah negatif dari sisa jumlah teratas dalam *equality constraint* yang tetap di setiap iterasi. Di bawah ini adalah proses dari algoritma yang dimaksud :

a. Pertma-tama cari *lagrange multiplier* α_1 yang melanggar Karush-Kuhn-Tucker (KKT) kondisi untuk masalah optimasi.
b. Kemudian pilih *multiplier* kedua α_2 dan mengoptimalkan pasangan α_1, α_2).
c. Ulangi langkah 1 dan 2 hingga konvergensi.

Ketika semua *lagrange multiplier* memenuhi kondisi KKT (dalam toleransi yang ditetapkan pengguna), masalah telah diselesaikan. Meskipun algoritma ini dijamin untuk menghasilkan konvergensi, heuristik tetap digunakan untuk memilih sepasang *multiplier* sehingga mempercepat laju konvergensi. Hal ini penting untuk set data yang besar karena ada *n (n - 1)* mungkin pilihan untuk α_1 dan α_2. Di bawah ini adalah alasan orang-orang menggunakan SMO untuk mengklasifikasikan data :

a. Metode SMO berbasis optimasi numerik cocok dengan data yang digunakan.
b. Sering memanggil *library* untuk memecahkan masalah optimasi.
c. Manipulasi matriks dengan skala besar yang mana memungkinkan terjadinya kesalahan.
d. Membutuhkan memori eksponensial.

2.6.2 Naive Bayes

Menurut [26], Naive Bayes adalah salah satu algoritma pembelajaran induktif yang paling efektif dan efisien untuk *machine learning* dan data mining. Performa Naive Bayes sangat kompetitif dalam proses klasifikasi walaupun menggunakan asumsi keindependenan atribut (tidak ada kaitan antar atribut). Asumsi keindependenan atribut ini pada data sebenarnya jarang terjadi, namun walaupun asumsi keindependenan atribut tersebut dilanggar, performa dari pengklasifikasian Naive Bayes masih cukup tinggi, hal ini dibuktikan pada berbagai penelitian empiris lainnya.

Naive Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai teorema Bayes. Teorema tersebut dikombinasikan dengan ”*naive*” dimana diasumsikan kondisi antar atribut saling bebas. Pada sebuah dataset, setiap baris/dokumen *I* diasumsikan sebagai vector dari nilai-nilai atribut $<x_1,x_2,\ldots,x_3>$ dimana tiap nilai-nilai menjadi peninjauan atribut X_i $(i \in [1,n]))$. Setiap baris mempunyai label kelas $c_i \in \{c_1,c_2,\ldots,c_k\}$ sebagai nilai variabel kelas *C*, sehingga untuk melakukan klasifikasi dapat dihitung nilai probabilitas $p(C=c_i|X=x_j)$, dikarenakan pada Naive Bayes diasumsikan setiap atribut saling

bebas, maka persamaan yang didapat adalah sebagai berikut :

- Peluang $p(C=c_i|X=x_j)$ menunjukkan peluang bersyarat atribut X_i dengan nilai x_i diberikan kelas c, dimana dalam Naive Bayes, kelas *C* bertipe kualitatif sedangkan atribut X_i dapat bertipe kualitatif ataupun kuantitatif.
- Ketika atribut X_i bertipe kuantitatif maka peluang $p(X=x_i|C=c_j)$ akan sangat kecil sehingga membuat persamaan peluang tersebut tidak dapat diandalkan untuk permasalahan atribut bertipe kuantitatif. Maka untuk menangani atribut kuantitatif, ada beberapa pendekatan yang dapat digunakan seperti distribusi normal (Gaussian) ataupun *Kernel Density Estimation* (KDE).

Selain dua pendekatan distribusi tersebut, ada mekanisme lain untuk menangani atribut kuantitatif (numerik) yaitu diskritisasi. Proses diskritisasi sendiri terjadi saat proses persiapan data atau saat data preprocessing, dimana atribut numerik *X* diubah menjadi atribut nominal *X**. Performansi klasifikasi Naive Bayes akan lebih baik ketika atribut numerik didiskritisasi daripada diasumsikan dengan pendekatan distribusi seperti di atas. Nilai-nilai numerik akan dipetakan ke nilai nominal dalam bentuk interval yang tetap memperhatikan kelas dari tiap-tiap nilai numerik yang dipetakan, sehingga penggambaran perhitungan Naive Bayes-nya menjadi seperti berikut ini :

$$p(I=i_i/C=c_j)=\frac{p(I=i_j)p(C=c_i|I=i_j)}{p(C=c_i|)}$$

Keterangan :

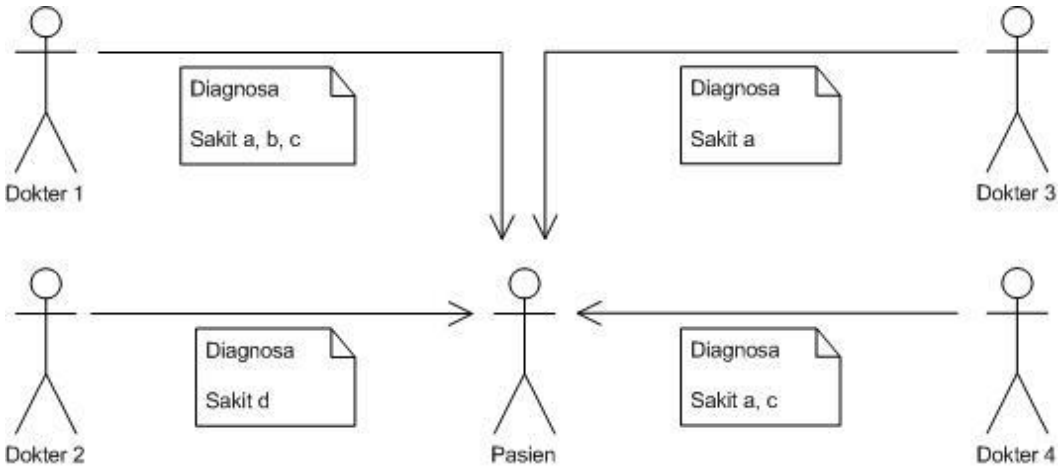
- $p(I=i_i|C=c_j)$: peluang interval *i* ke *j* untuk kelas c_i
- $p(I=i_j)$: peluang sebuah interval ke *j* pada semua interval yang terbentuk
- $p(C=c_i|I=i_j)$: peluang kelas c_i pada interval *i* ke *j*
- $p(C=c_i)$: peluang sebuah kelas ke *i* untuk semua kelas yang ada di dataset

Sedangkan di bawah ini adalah langkah-langkah yang ada pada algoritma Naive Bayes :

- Menghitung jumlah *class*.
- Menghitung jumlah kasus yang sama dengan *class* yang sama.
- Kalikan semua hasil variabel *class* satu dengan yang lainnya.
- Bandingkan hasil masing-masing *class*.

Selain itu algoritma Naive Bayes memiliki kelebihan yaitu mudah diimplementasikan dan mampu memberikan hasil yang baik untuk banyak kasus, sedangkan kelemahannya adalah harus mengasumsi bahwa antar fitur tidak terkait (independen). Dalam kenyataannya, keterkaitan itu ada dan keterkaitan tersebut tidak dapat dimodelkan oleh algoritma Naive Bayes.

2.6.3 Bagging



Gambar 2.8 Ilustrasi Algoritma Bagging

Bagging adalah singkatan dari *bootstrap aggregating* merupakan algoritma klasifikasi untuk pengambilan keputusan yang mana menggunakan beberapa suara yang digabung menjadi prediksi tunggal [27]. Prediksi tunggal yang dihasilkan oleh algoritma Bagging didapatkan dari bobot yang sama dari suara terbanyak. Ilustrasi algoritma Bagging seperti halnya seorang pasien yang ingin mendapatkan suatu diagnosa dari dokter terhadap gejala yang dirasakan [28]. Semestinya seorang

pasien hanya mengunjungi seorang dokter, namun pasien tersebut mengunjungi beberapa dokter untuk mendapatkan diagnosanya. Jika diagnosa yang diberikan sering muncul sama lebih dari satu, maka bisa dianggap diagnosa itu merupakan diagnosa terbaik, yang artinya diagnosa didapat dari suara mayoritas yang sama (Gambar 2.8). Sedangkan di bawah ini adalah prosedur dari algoritma Bagging :

Model Generation

```
Let n be the number of instances in the training data.
For each of t interations:
        Sample n instances with replacement from training data.
        Apply the learning algoritm to the sample.
        Store the resulting model.
```

Classification

```
For each of the t models.
        Predict class of instance using model.
Return class that has been predicted most often.
```

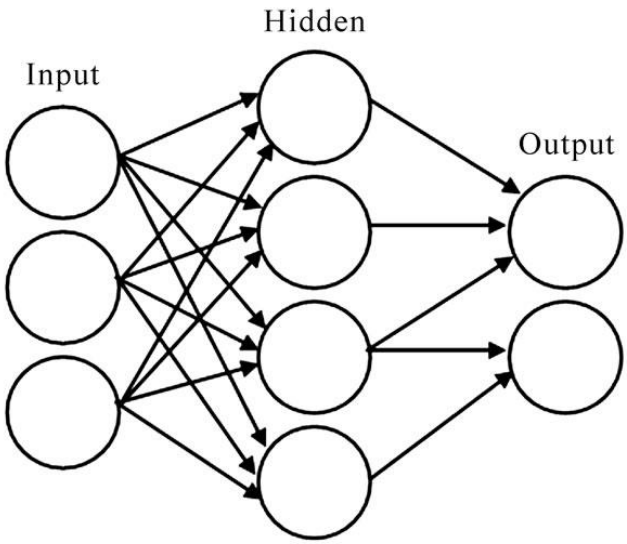
2.6.4 Multilayer Perceptron

Tabel 2.2 Analogi JSB (Jaringan Syaraf Biologis) Dan JST (Jaringan Syaraf Tiruan)

JSB	JST
Soma	Neuron
Dendrite	Masukan (Input)
Axon	Keluaran (Output)
Synapse	Bobot (Weight)

Perlu diketahui bahwa Multilayer Perceptron termasuk jenis JST. JST sendiri merupakan sebuah algoritma yang disusun dengan struktur dan fungsi otak manusia sebagai model untuk ditiru. Pada sebuah JST terdapat sejumlah neuron. Satu neuron bisa terhubung ke banyak neuron lain, dan setiap koneksi *(link)* tersebut

mempunyai bobot *(weight)*. Tabel 2.2 merupakan analogi bagian-bagian dari JST terhadap JSB. Pembelajaran merupakan karakteristik dasar dari JSB. JST melakukan proses pembelajaran melalui penyesuaian bobot pada koneksi antar neuronnya [29]. Multilayer Perceptron sendiri merupakan topologi paling umum dari JST di mana perceptron-perceptronnya terhubung membentuk beberapa lapisan (layer). Multilayer Perceptron mempunyai lapisan masukan *(input layer)*, minimal satu lapisan tersembunyi *(hidden layer)*, dan lapisan luaran *(output layer)*. Arsitektur dari Multilayer Perceptron ditunjukkan pada gambar di bawah ini :



Gambar 2.9 Arsitektur Multilayer Perceptron

Metode yang paling banyak digunakan dalam pembelajaran atau pelatihan Multilayer Perceptron adalah propagasi balik *(back-propagation)*. Terdapat empat langkah yang harus dilakukan dalam metode ini yaitu inisialisasi *(initialization)*, aktivasi *(activation)*, pelatihan bobot *(weight training)*, dan iterasi *(iteration)*. Pada langkah inisialisasi, nilai awal bobot dan ambang batas *(threshold)* ditentukan secara acak namun dalam batasan tertentu. Pada tahapan aktivasi, diberikan masukan dan nilai keluaran yang diharapkan *(desired output)*. Proses penyesuaian bobot terjadi pada tahap pelatihan bobot, nilai luaran sebenarnya *(actual output)* dibandingkan dengan *desired output* dan dilakukan penyesuaian bobot. Langkah kedua dan ketiga diulangi sampai dengan tercapai kondisi yang ditentukan.

Untuk contoh studi kasus dari Multilayer Perceptron yaitu

merepresentasikan masukan/keluaran dari biopolar (-1 untuk *false* dan 1 untuk *true*), dimana bobot dan *threshold* masing-masing adalah 1 dan 0 dengan fungsi aktivasi 1 jika net > *threshold*, 0 jika net = *threshold* dan -1 jika net < *threshold*, sehingga hasilnya akan tampak seperti tabel di bawah ini :

Tabel 2.3 Multilayer Perceptron

Input 1	Input 2	Bobot	Output
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	1	-1

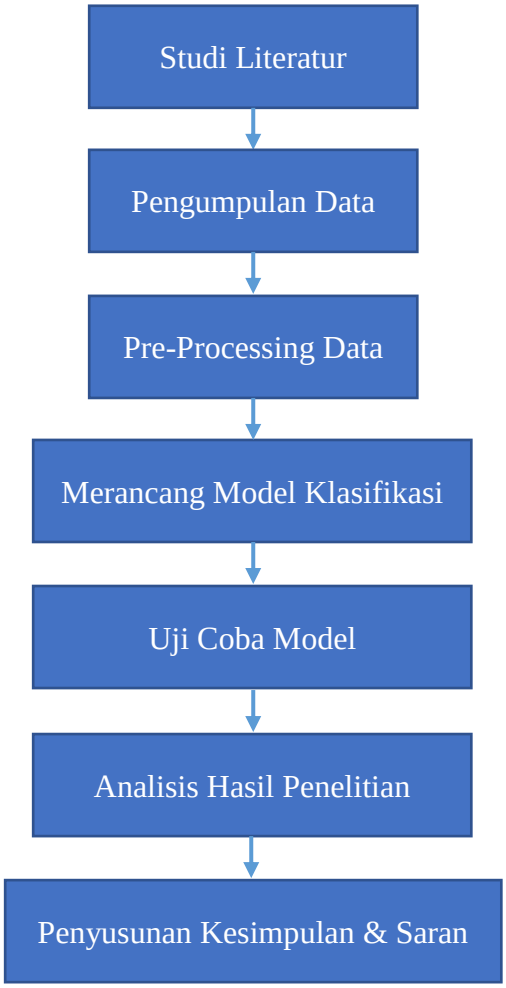
(Halaman ini sengaja dikosongkan)

BAB 3
METODOLOGI PENELITIAN

Dalam bab ini diuraikan tahap-tahap yang akan dilakukan pada penelitian ini. Tahapan-tahapan yang dimaksud antara lain yaitu tahap pengumpulan data, *pre-processing data*, pembuatan model klasifikasi hingga skenario uji coba yang akan dilaksanakan dalam penelitian ini.

3.1 Gambaran Umum Penelitian

Langkah-langkah atau metodologi dalam penelitian ini tergambar dalam gambaran umum penelitian seperti diagram di bawah ini :



Gambar 3.1 Gambaran Umum Penelitian

3.2 Pengumpulan Data

Untuk metode pengumpulan data, peneliti melakukan observasi dan pencarian data berupa list URL situs phising dan situs non-phising (situs otentik) melalui internet, email, dan referensi dari penelitian Dongsong Zhang dan kawan-kawan [2]. Sumber dari internet yang dimaksud berasal dari *https://moz.com/top500* (Moz) dan *http://www.alexa.com* (Alexa) untuk situs non-phising, sedangkan untuk list situs phising didapatkan dari PhishTank yang beralamatkan di *http:// phishtank.com* dan beberapa website penyedia informasi mengenai situs phising (khususnya di Indonesia). Hasilnya, peneliti memperoleh sekitar 680 situs, masing-masing terdiri dari 340 situs otentik dan 340 situs phishing. Di bawah ini adalah contoh URL situs otentik yang berhasil didapatkan :

- *https://ebay.com*
- *https://facebook.com*
- *https://gmail.com*
- *https://ibank.bankmandiri.co.id*
- *https://kaskus.co.id*
- *https://paypal.com*
- *https://pb.garena.co.id*
- *http://serbasepeda.com*
- *https://taobao.com*

Sedangkan di bawah ini adalah contoh dari URL situs phising yang telah dikumpulkan :

- *http://www.ebay.com.tw*
- *http://www.facebok.com*
- *http://hack-gmail-password.com*
- *http://ablytube.com/clip/Personal*
- *https://kaskusbluemoviess.allalla.com*
- *http://88.198.24.90/~consumired/pos/006b3/*
- *http://radeemnowevents.ye.vc*
- *http://serba-sepeda.blogspot.co.id*
- *https://taobaohacks.wordpress.com*

3.3 Fitur

Menentukan/memilih fitur adalah langkah awal yang harus dilakukan pada penelitian ini sebelum melakukan klasifikasi. Pemilihan fitur dilakukan untuk mendapatkan fitur-fitur yang sesuai dan revelan berdasarkan pendekatan yang ada agar dapat digunakan untuk mendeteksi situs phising. Peneliti melakukan studi literatur dan mengkaji beberapa teori yang ada untuk mendapatkan fitur-fitur dimaksud. Studi literatur dan kajian yang ada diharapkan dapat memecahkan rumusan masalah terkait sulitnya memilih fitur yang relevan sekaligus yang mampu meningkatkan kinerja deteksi. Setelah melakukan studi literatur dan mengkaji beberapa teori yang ada, akhirnya peneliti mengusulkan untuk menggunakan beberapa fitur pada penelitian [2], [4]-[5] dan menambahkan fitur baru berbasis konten dan URL.

Pada penelitian yang dilakukan oleh Dongsong Zhang dan kawan-kawan [2] terdapat beberapa fitur deteksi yang tidak dibutuhkan dalam penelitian ini sebagai contoh yaitu F12 dan F15. F12 adalah fitur yang berfungsi untuk mengetahui apakah di dalam website tercantum nomor lisensi ICP *(Internet Content Provider)*, sedangkan F15 adalah fitur yang berguna untuk mengetahui apakah di dalam website *e-business* tersebut terdapat informasi sertifikat *e-commerce*. Peneliti membuang fitur-fitur tersebut, karena secara garis besar fitur-fitur tersebut tidak dibutuhkan pada model klasifikasi yang akan dibuat pada penelitian ini. Pasalnya model klasifikasi pada penelitian ini mencakup keseluruhan situs secara global khususnya di Indonesia (bukan hanya untuk website *e-commerce* saja).

Peneliti juga mengambil dan memodifikasi beberapa fitur deteksi berbasis pendekatan *fuzzy rule* dan *machine learning* berbau pendekatan berbasis konten dan URL dengan rasio frekuensi tinggi pada penelitian yang dilakukan oleh Neda Abdelhamid dan kawan-kawan [4] antara lain yaitu panjang URL, https dan pengunjung website (Alexa Rank). Sedangkan pada penelitian yang dilakukan oleh Yuancheng Li dan kawan-kawan [5], peneliti mengambil dan memodifikasi sebuah fitur yang mendukung kontribusi praktis dari penelitian ini terkait implementasi di penelitian selanjutnya yang mana apabila model/sistem yang dibuat diimplementasikan, maka dapat menghindarkan pengguna internet dari serangan malware atau virus, fitur yang dimaksud adalah fitur untuk mendeteksi rata-rata

jumlah file JS (JavaScript). Karena pada dasarnya semakin banyak file JS pada suatu situs, semakin banyak juga peluang file tersebut disisipi malware atau virus.

Selain itu peneliti juga menambahkan fitur baru berbasis konten dan URL yaitu skor halaman web yang diambil dari *PageSpeed Insights Google*. Sehingga kurang lebih peneliti menggunakan 11 fitur untuk mendeteksi situs phising dan di bawah ini adalah fitur-fitur yang dimaksud :

a. F1 : Apakah URL berisi sebuah alamat IP?

   Biasanya sebuah website phising menggunakan alamat IP. Pada penelitian [2] dan [4] juga menggunakan fitur ini untuk mendeteksi situs phising. Jika URL menggunakan alamat IP, maka F1 akan diberi nilai 1, jika tidak, maka F1 diberi nilai -1.

b. F2 : Apakah URL berisi simbol '@'?

   Biasanya website phising akan menggunakan simbol '@' di dalam URL. Fitur ini juga digunakan pada penelitian [2] dan [4]. Jika URL menggunakan simbol '@', maka F2 akan diberi nilai 1, jika tidak, maka F2 diberi nilai -1.

c. F3 : Jumlah afiks (imbuhan)

   Penjahat internet biasanya memodifikasi URL situs phising dengan menambahkan beberapa afiks untuk menipu pengguna internet yang mana seolah-olah website tersebut adalah situs ontentik. Afiks sendiri dibagi menjadi 4 jenis yaitu prefiks (imbuhan di depan), infiks (imbuhan di tengah), sufiks (imbuhan di belakang) dan konfiks (imbuhan di depan dan di belakang). Pada penelitian [2] hanya digunakan sufiks saja untuk mendeteksi situs phising, karena peneliti pada penelitian tersebut percaya bahwa website phising menggunakan 2 sufiks domain dan biasanya pengguna internet hanya akan melihat bagian pertama dan mengabaikan bagian lainnya, sehingga akan menuju website phising. Sedangkan pada penelitian [4] menggunakan prefiks dan sufiks untuk mendeteksi situs phising. Akan tetapi pada penelitian ini akan digunakan afiks yang mana mencakup semua jenis imbuhan termasuk imbuhan yang digunakan pada penelitian yang telah disebutkan di atas. Contoh afiks pada penelitian ini adalah "-" dan beberapa ekstensi domain yang dianggap aneh (tidak wajar). Perlu diketahui bahwa atribut dari F3 bertipe data numerik (non-boolean).

d. F4 : Usia domain

Usia domain dihitung sejak domain diregistrasi oleh registrar. Pada penelitian [2] dan [4] juga menggunakan fitur ini untuk mendeteksi situs phising, karena pada dasarnya semakin muda umur domain tersebut, kredibilitasnya sebagai situs non-phising semakin dipertanyakan. Perlu diketahui bahwa atribut dari F4 bertipe data numerik (non-boolean).

e. F5 : Apakah domain didaftarkan oleh perusahaan?

Pada penelitian [2], fitur ini digunakan untuk memeriksa apakah domain didaftarkan oleh sebuah perusahaan atau non-perusahaan (pribadi). Biasanya situs phising menggunakan nama pribadi, nama samara, nama palsu atau bahkan menyembunyikan nama pendaftarnya agar tidak mudah dilacak oleh orang lain. Jika domain didaftarkan oleh perusahaan, maka F5 akan diberi nilai 1, jika tidak, maka F5 diberi nilai -1.

f. F6 : Apakah domain diprivasi?

Dalam penelitian [2], fitur ini digunakan untuk memastikan bahwa informasi domain tidak diprivasi oleh sang pemilik, karena kebanyakan domain yang dijadikan sebagai situs phising oleh penjahat internet diprivasi informasinya sehingga orang lain tidak bisa melacak atau mencari informasi mengenai pemilik situs phising tersebut. Jika domain diprivasi, maka F6 akan diberi nilai 1, jika tidak, maka F6 diberi nilai -1.

g. F7 : Panjang URL

Situs phising biasanya memiliki panjang URL yang tidak lazim. Pada penelitian [4] panjang URL memiliki dampak rasio 51% terhadap sistem deteksi situs phising yang dibuat. Karena nilai rasionya cukup tinggi, maka peneliti menggunakan fitur tersebut pada penelitian ini. Perlu diketahui bahwa atribut dari F7 bertipe data numerik (non-boolean).

h. F8 : Apakah website menggunakan https?

Pada dasarnya setiap situs besar pasti memiliki protokol keamanan yang disebut https, dimana protokol tersebut tidak dimiliki oleh sebagian besar situs phising karena selain digunakan untuk melakukan pencurian data dan tindakan kriminalitas seperti penipuan, situs phising juga difungsikan sebagai media penyebaran malware, virus dan *hacking* oleh penjahat internet yang mana hal

tersebut kontradiksi dengan https. Pada penelitian [4] https memiliki dampak rasio 89,2% terhadap sistem deteksi situs phising yang telah dibuat. Karena nilai rasionya sangat tinggi, maka peneliti mencoba memodifikasi sedikit lalu menggunakan fitur tersebut pada penelitian ini. Jika website memiliki https dan https-nya disertai sertifikat SSL *(Secure Socket Layer)*, maka F8 akan diberi nilai -1, jika website memiliki https dan https-nya tidak disetai sertifikat SSL, maka F8 diberi nilai 0, selain itu, maka F8 diberi nilai 1.

i. F9 : Apakah website memiliki Alexa Rank?

   Alexa Rank adalah perankingan web yang dapat digunakan untuk mengukur seberapa banyak pengunjung ataupun popularitas sebuah website. Nilai Alexa Rank sendiri kurang lebih berkisar antara 1 hingga 25.000.000, semakin kecil rankingnya semakin bagus. Normalnya setiap web dari perusahaan besar pasti memiliki Alexa Rank. Berbeda dengan situs phising yang kebanyakan tidak memiliki Alexa Rank karena kunjungan per harinya terlalu sedikit. Pada penelitian [4] menggunakan fitur pengunjung website yang mana indikator yang digunakan adalah Alexa Rank. Dimana dampak rasio dari pengunjung website terhadap sistem deteksi situs phising yang dibuat sangat besar yaitu 93,2%. Peneliti pada penelitian ini memodifikasi justifikasi pada fitur ini menjadi : jika website memiliki Alexa Rank = 0, maka F9 akan diberi nilai 1, jika website memiliki Alexa Rank > 10.000.000, maka F9 diberi nilai 0, selain itu, maka F9 diberi nilai -1.

j. F10 : Jumlah JS

   Kebanyakan situs phising memiliki jumlah file JS yang tidak wajar. Hal itu terjadi karena situs phising menggunakan file JS tersebut untuk menyebarkan malware atau virus. Pada penelitian [4] juga digunakan fitur serupa tetapi dengan perhitungan yang berbeda yaitu rata-rata jumlah JS, sedangkan pada penelitian ini akan dihitung jumlah file JS. Perlu diketahui bahwa atribut dari F10 bertipe data numerik (non-boolean).

k. F11 : Skor halaman web

   Situs phising biasanya memakan waktu lebih ketika diakses, karena pada dasarnya situs phising mengandung banyak script, JS, pop-up atau malware. Pada penelitian [30] dikatakan bahwa kecepatan loading web (*response time* dan

latency) dapat mempengaruhi ranking dari sebuah halaman website. Sehingga dapat ditarik kesimpulan bahwa sebagian besar halaman website yang memiliki skor bagus pasti memiliki *loading web* yang cepat. *PageSpeed Insights Google* adalah salah satu fitur dari Google Inc. yang dapat digunakan untuk menghitung skor pada sebuah halaman web dengan memperhitungkan kriteria-kriteria yang telah disebutkan di atas. Perlu diketahui bahwa atribut dari F10 bertipe data numerik (non-boolean).

Untuk semua fitur yang memiliki tipe data numerik (non-boolean) akan dinormalisasi dalam ekstraksi fitur untuk menjaga hubungan nilai antar fitur dan menyederhanakan nilai fitur tetapi tidak menghilangkan bobot dari nilai fitur itu sendiri, sehingga mampu meningkatkan kinerja klasifikasi.

3.4 *Pre-Processing Data*

Menurut [31] *pre-processing data* adalah suatu proses/langkah yang dilakukan untuk membuat data mentah menjadi data yang berkualitas (input yang baik untuk *data mining tools*). Langkah ini dilakukan untuk menyiapkan data agar mudah diolah pada tahap berikutnya. *Pre-processing data* pada penelitian ini dibagi menjadi 2 tahap yaitu :

3.4.1 Prefiksasi

Dalam tahap awal *pre-processing data* pada penelitian ini akan dilakukan prefiksasi. Prefiksasi adalah proses pembentukan kata dengan cara menambahkan afiks pada bentuk dasar dan melekatkannya di depan bentuk dasar [32]. Contoh prefiks (imbuhan di depan) pada kata berbahasa Indonesia adalah meng-, di-, per- dan lain-lain. Penelitian ini tidak menggunakan prefiks huruf, melainkan prefiks angka (1 dan -1) yang mana melambangkan situs phising dan situs non-phising. Pada data yang telah dikumpulkan oleh peneliti, setiap kata atau baris mewakili sebuah URL situs. Setiap URL situs akan diberi prefiks 1 dan -1 berdasarkan statusnya seperti yang telah dijelaskan sebelumnya (1 untuk situs phising dan -1 untuk situs non-phising). Pemberian prefiks ini dilakukan secara manual oleh

peneliti. Pada Tabel 3.1 adalah contoh hasil penerapan prefiksasi pada data situs yang telah dikumpulkan.

Tabel 3.1 Contoh Hasil Proses Prefiksasi

Masukan	Hasil
http://88.198.24.90/~consumired/pos/006b3/	*1 http://88.198.24.90/~consumired/pos/006b3/*
http://ablytube.com/clip/Personal	*1 http://ablytube.com/clip/Personal*
http://hack-gmail-password.com	*1 http://hack-gmail-password.com*
http://radeemnowevents.ye.vc	*1 http://radeemnowevents.ye.vc*
http://serba-sepeda.blogspot.co.id	*1 http://serba-sepeda.blogspot.co.id*
http://serbasepeda.com	*-1 http://serbasepeda.com*
http://www.ebay.com.tw	*1 http://www.ebay.com.tw*
http://www.facebok.com	*1 http://www.facebok.com*
https://ebay.com	*-1 https://ebay.com*
https://facebook.com	*-1 https://facebook.com*
https://gmail.com	*-1 https://gmail.com*
https://ibank.bankmandiri.co.id	*-1 https://ibank.bankmandiri.co.id*
https://kaskus.co.id	*-1 https://kaskus.co.id*
https://kaskusbluemoviess.allalla.com	*1 https://kaskusbluemoviess.allalla.com*
https://paypal.com	*-1 https://paypal.com*
https://pb.garena.co.id	*-1 https://pb.garena.co.id*
https://taobao.com	*-1 https://taobao.com*

Hasil dari proses prefiksasi tersebut akan dikonversi menjadi file berformat .txt atau .csv agar mudah diolah pada tahap selanjutnya.

3.4.2 Ekstraksi Fitur

Ekstraksi fitur merupakan pengambilan ciri/fitur dari suatu bentuk dimana

nilai yang didapatkan akan dianalisis untuk proses selanjutnya [33]. Pada penelitian ini ekstraksi fitur dilakukan untuk mengekstrak fitur-fitur yang ada menggunakan data yang telah dikumpulkan menjadi sebuah file berformat ARFF *(Attribute-Relation File Format)* berisi header (*relation* dan *attribute*) dan data (nilai fitur) agar bisa langsung diolah menggunakan software Weka, selain itu ekstraksi fitur pada penelitian ini juga memiliki fungsi lain yaitu meminimalisir kesalahan validasi nilai fitur yang dilakukan oleh manusia secara manual dan melakukan normalisasi semua nilai fitur dari atribut yang bertipe data numerik.

Normalisasi ini dilakukan untuk menjaga hubungan nilai antar fitur dan menghasilkan nilai fitur yang lebih sederhana tetapi dengan bobot yang sama, sehingga mampu meningkatkan kinerja klasifikasi. Sebagai contoh F7 (panjang URL) memiliki nilai minimum 14 karakter dan nilai maksimum 785 karakter. Bila diamati jarak nilai terendah dan tertinggi sangat jauh yaitu selisih 771 karakter. Selisih yang sangat jauh tersebut bisa mempengaruhi hubungan nilai antar fitur, sehingga akan mempengaruhi kinerja klasifikasi. Oleh sebab itu perlu dilakukan normalisasi untuk menyederhanakan nilai fitur tanpa merubah bobot dan mengoptimasi kinerja klasifikasi. Di bawah ini adalah rumus yang digunakan untuk melakukan normalisasi nilai fitur :

$$N = \frac{\boldsymbol{n - min}}{\boldsymbol{max - min}}$$

Dimana :

- N : Nilai normalisasi
- n : Nilai fitur
- min : Nilai minimum pada fitur
- max : Nilai maksimum pada fitur

Untuk ekstraksi fitur deteksi situs phising ini, peneliti membuat sebuah *web crawler* berbasis PHP menggunakan teknologi yang berkembang saat ini. Teknologi yang dimaksud adalah API *(Application Programming Interface)*. API merupakan satu set instruksi pemograman untuk mengakses aplikasi berbasis web.

Sebuah perusahaan perangkat lunak merilis API kepada publik sehingga pengembang perangkat lunak lain dapat merancang produk yang didukung oleh layanannya [34]. Pada penelitian ini API digunakan untuk mendeteksi beberapa fitur antara lain F4 (usia domain), F5 (Apakah domain didaftarkan oleh perusahaan?), F6 (Apakah domain diprivasi?) dan fitur-fitur lainnya. Di bawah ini adalah sampel justifikasi dan kode PHP berdasarkan F6 (Apakah domain diprivasi?) :

Sampel Justifikasi

```
if ((name = 'Registration Private') || (name == 0)) {
    F6 = 1
} else {
    F6 = -1
}
```

Sampel Kode

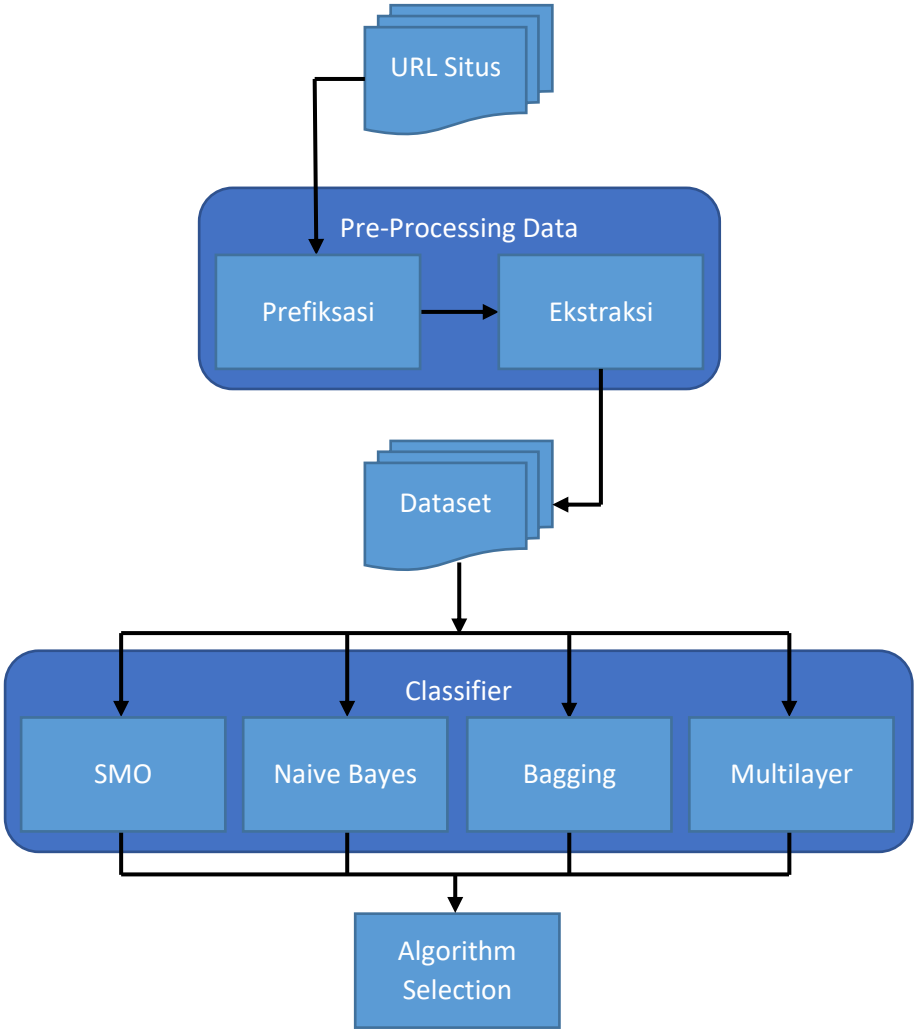
```
function privasi() {
        if ($this->whois == null) {
                $this->whois();
        }
        $status = (($this->whois['registrant_contact']['name']=='Registration
        Private') || ($this->whois['registrant_contact']['name']=='')?1:-1);
        $this->attr[] = '@attribute privasi { 1,-1 }';
        $this->real[] = false;
        $this->hasil[] = $status;
}
```

3.5 Model Klasifikasi Deteksi Situs Phising

Pada penelitian yang dilakukan oleh Dongsong Zhang dan kawan-kawan [2] pembuatan model klasifikasi deteksi website *e-business* phising dimulai dari pembuatan vektor fitur dengan cara menentukan fitur yang relevan, memodifikasi fitur yang ada dan menambahkan fitur baru berbasis pendekatan fitur konten dan URL, lalu membandingkan beberapa classifier dan memilih classifier dengan

kinerja terbaik. Dimana classifier dengan kinerja deteksi terbaik diterapkan pada model klasifikasi tersebut untuk dibandingkan dengan hasil kinerja pada penelitian [15] dan [16] yang hanya menggunakan fitur tradisional saja untuk deteksi situs phising.

3.5.1 Model Klasifikasi



Gambar 3.4 Model Klasifikasi Deteksi Situs Phising

Model klasifikasi deteksi situs phising pada penelitian ini seperti yang terlihat pada Gambar 3.4 mengacu pada penelitian yang dilakukan oleh Dongsong Zhang [2]. Akan tetapi pada tahap *pre-processing* pada penelitian ini dilakukan prefiksasi dan ekstraksi fitur berbasis *web crawler* yang mana hal tersebut tidak

dilakukan pada Dongsong Zhang [2]. Tentunya hal ini menjadi pembeda penelitian ini dengan penelitian tersebut. Pembuatan dari esktraksi fitur itu sendiri merujuk pada penelitian yang dilakukan oleh dan Yuancheng Li [5] untuk memudahkan peneliti dalam mengolah data skala besar. Sedangkan untuk desain dari model klasifikasi yang dibuat pada penelitian ini diadopsi dari pada penelitian yang dilakukan oleh Cagatay Catal [8], Kyungro Lee [9] dan Karthik Thirumala [10] yang mana menggunakan beberapa classifier, lalu memilih classifier dengan kinerja terbaik *(algoritm selection).*

3.5.2 Kinerja Klasifikasi

Sama seperti penelitian [2], penelitian ini juga akan menggunakan P *(Precision),* R *(Recall)* dan F *(F-Measure)* untuk mengevaluasi kinerja dari model klasifikasi yang dibuat. Tetapi yang menjadi pembeda penelitian ini dengan penelitian tersebut adalah penambahan aspek lain untuk mengevaluasi kinerja dari model klasifikasi yang dibuat yaitu A (akurasi) dan T *(training time)*. Jika mengacu pada *confusion matrix,* maka pada penelitian ini terdapat 4 kemungkinan hasil yang didapatkan dari klasifikasi yaitu TP *(True Positive),* FP *(False Positive),* TN *(True Negative)* dan FN *(False Negative)*. *Confusion matrix* adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep *data mining*. Di bawah ini adalah tabel *confusion matrix* yang dimaksud :

Tabel 3.2 *Confusion Matrix*

Prediksi	**Hasil**	
	1	0
1	*TP (True Positive)*	FN *(False Negative)*
0	FP *(False Positive)*	TN *(True Negative)*

Menurut [2] *Precision* juga disebut sebagai nilai prediksi positif yang mana adalah presentase dari prediksi benar yang digambarkan dengan *TP / (TP + FP),* nilai *Recall* adalah proporsi aktual positif di dalam data populasi yang telah diuji yang mana ditulis sebagai *TP / (TP + FN),* sedangkan *F-Measure* adalah nilai mean

dari kombinasi *Precision* dan *Recall* yang dapat dihitung menggunakan rumus *2 (P x R) / (P + R)*. Nilai dari *Precision, Recall* dan *F-Measure* berkisar antara 0 hingga 1. Pada penelitian tersebut, Dongsong Zhang dan kawan-kawan [2] mengemukakan hipotesa bahwa fitur berbasis konten dan URL pada model klasifikasi yang diusulkan untuk deteksi website *e-business* akan mengungguli model klasifikasi yang menggunakan fitur tradisional bila dilihat dari segi *Precision, Recall* dan *F-Measure*. Sedangkan hipotesa pada penelitian ini yaitu fitur baru berbasis pendekatan konten dan URL yang diusulkan pada penelitian ini dapat membuat kinerja deteksi situs phising jauh lebih baik daripada fitur dasar pada penelitian sebelumnya [2].

Dimana kinerja deteksi situs phising yang dimaksud mencakup TP, FN, FP, TN, P, R, F, A dan T. Akan tetapi pada penelitian ini, peneliti lebih memprioritaskan nilai FP dan TN daripada nilai akurasi, karena peneliti memiliki asumsi bahwa lebih baik situs phising diprediksi sebagai situs non-phising daripada situs non-phising diprediksi sebagai situs phising dan kebetulan asumsi tersebut maknanya terkandung di dalam nilai FP dan TN. Sehingga tak jadi masalah bila akurasi yang dihasilkan bukanlah akurasi terbaik di antara algoritma lain, asalkan asumsi tersebut dan akurasi dari model klasifikasi yang dihasilkan tidak lebih buruk daripada penelitian sebelumnya [2]. Selain itu jumlah fitur yang digunakan pada penelitian ini jauh lebih sedikit bila dibandingkan dengan penelitian [2], [4]-[5]. Apabila kinerja deteksi situs phising meningkat dari sebelumnya menggunakan fitur yang jauh lebih sedikit, hal ini tentunya bisa menjadi kontribusi teoritis.

3.5.3 Algoritma Klasifikasi

Pada penelitian ini akan digunakan 4 algoritma klasifikasi berbeda di dalam tahap uji coba antara lain :

a. SMO *(Sequential Minimal Optimization)*

   Algoritma SMO dipakai karena dapat memecahkan masalah QP *(Quadratic Programming)* yang timbul selama pelatihan SVM (*Support Vector Machine*) dimana pada penelitian ini akan digunakan data dalam skala besar yang mana memungkinkan terjadinya kesalahan ketika memanipulasi matriks. Selain itu pada penelitian yang dilakukan oleh Dongsong Zhang dan kawan-kawan [2],

algoritma tersebut menghasilkan nilai akurasi terbaik.

b. Naive Bayes

Pada penelitian yang sejenis (deteksi situs phising) [2] dan [4], classifier Naive Bayes adalah algoritma yang paling sering dipakai. Itu tak lepas dari fungsi Naive Bayes sebagai classifier yang dapat digunakan untuk memprediksi sesuatu berdasarkan data yang ada menggunakan metode probabilitas dan statistik termasuk untuk memprediksi apakah situs tersebut termasuk situs phising atau non-phising.

c. Bagging

Dalam pembuatan model klasifikasi sentimen review pelanggan pada blog, forum dan sosial media di Turki, Cagatay Natal [8] menggunakan Bagging. Bagging digunakan dalam model tersebut karena mampu memberikan sebuah keputusan menggunakan beberapa suara yang digabung menjadi prediksi tunggal.

d. Multilayer Perceptron

Peneliti mengusulkan menggunakan algoritma baru Multilayer Perceptron berbasis Jaringan Syarat Tiruan (JST) untuk digunakan dalam penelitian ini, karena pada penelitian yang dilakukan oleh Kyungro Lee dan kawan-kawan [9] digunakan algoritma sejenis berbasis JST yaitu NN *(Neural Network)* untuk membangun sebuah model klasifikasi yang mampu memprediksi *activator* pada CAR *(Constitutive Androstane Receptor)* dan menawarkan informasi struktural mengenai interaksi ligan/protein di dalam hati. Oleh sebab itu peneliti ingin mencoba menggunakan algoritma sejenis tetapi dengan studi kasus yang berbeda.

3.6 Skenario Uji Coba

Untuk membantu menjawab rumusan masalah dan tujuan dari penelitian ini, maka dirancang skenario uji coba. Uji coba ini dilakukan untuk menentukan algoritma mana yang akan dipakai oleh model klasifikasi dan memastikan bahwa model klasifikasi yang dibuat mampu mendeteksi situs phising dengan kinerja baik. Algoritma dengan hasil uji coba terbaik nantinya akan digunakan dalam penelitian ini, karena dengan hasil uji coba yang baik, maka dipastikan model klasifikasi yang

dibuat dapat meningkatkan kinerja deteksi situs phising. Aspek-aspek yang perlu diperhatikan untuk menganalisa hasil uji coba pada penelitian ini antara lain adalah nilai TP *(True Positive)*, FN *(False Negative)*, FP *(False Positive)*, TN *(True Negative)*, P *(Precision)*, R *(Recall)*, F *(F-Measure)*, A (akurasi) dan T (*training time)*.

Untuk mendapatkan hasil yang secara langsung dapat dibandingkan, maka pada penelitian ini digunakan dataset dan *fold* (interasi) yang sama. Dataset yang digunakan merupakan kumpulan URL situs yang telah melalui tahap *pre-processing data* sehingga dataset tersebut dapat digunakan untuk mengukur tingkat keberhasilan dari penelitian yang dilakukan. Data tersebut didapatkan dari penelitian yang dilakukan oleh Dongsong Zhang [2]. PhishTank, Moz, Alexa, email, dan berbagai sumber kredibel lainnya yang ada di internet. Dimana setiap situs otentik (non-phising) memiliki minimal satu situs phising (*one to one*/*one to many*). Di bawah ini adalah detail dari data yang akan digunakan dalam tahap uji coba nantinya :

Tabel 3.3 Detail Data

Objek	Jumlah	Presentase
Situs non-phising	340 situs	50%
Situs phising	340 situs	50%
Total	**680 situs**	**100%**

Uji coba dalam penelitian ini sendiri dibagi menjadi 4 bagian antara lain sebagai berikut :

a. Uji Coba Algoritma Klasifikasi

   Tahap pertama dalam skenario uji coba ini adalah uji coba algoritma klasifikasi. Uji coba ini bertujuan untuk mendapatkan algoritma klasifikasi dengan kinerja deteksi terbaik. Tahap pertama pada penelitian ini, dibagi menjadi empat uji coba antara lain uji coba algoritma SMO *(Sequential Minimal Optimization)*, uji coba algoritma Naive Bayes, uji coba algoritma Bagging dan uji coba algoritma

Multilayer Perceptron. Dimana algoritma dengan hasil kinerja deteksi terbaik akan digunakan dalam penelitian ini.

b. Uji Coba Model Klasifikasi Pada Penelitian Sebelumnya

Uji coba model klasifikasi pada penelitian sebelumnya dilakukan untuk mengukur kinerja model klasifikasi pada penelitian sebelumnya [2] bila menggunakan fitur dasar pada penelitian tersebut dan data yang sama dengan penelitian ini. Sehingga hasil dari uji coba tersebut dapat dijadikan perbandingan untuk menilai kinerja deteksi dari model klasifikasi yang telah dibuat. Untuk mendukung uji coba tahap kedua ini, maka peneliti menggunakan *web crawler* khusus untuk mengekstraksi fitur yang ada dalam penelitian tersebut yang disebut sebagai *Web Crawler II*.

c. Uji Coba Data Baru

Tahap ketiga dalam skenario uji coba pada penelitian ini adalah uji coba data baru. Uji coba ini dilakukan untuk menguji model klasifikasi yang telah dibuat dan model klasifikasi pada penelitian sebelumnya [2] apakah mampu membedakan situs phising atau situs non-phising secara akurat. Peneliti akan menggunakan data baru maupun data yang sudah ada berupa list situs phising dan situs non-phising sebagai sampel untuk melakukan uji coba ini. Data tersebut tentunya akan melewati tahap *pre-processing data* (prefiksasi dan ektsraksi fitur) terlebih dahulu sebelum di uji.

d. Uji Coba Data Mining Clustering

Uji coba data mining clustering ini dilakukan untuk mengetahui kecenderungan data dan memastikan bahwa model klasifikasi ini memang sangat cocok digunakan untuk membedakan situs phising dan situs non-phising bila dibandingkan dengan menggunakan data mining clustering. Pada uji coba ini, peneliti menggunakan algoritma K-Means. Algoritma ini digunakan karena mampu mengkategorikan data berdasarkan centroid (titik tengah) dari nilai fitur yang bersangkutan.

3.7 Jadwal Penelitian

Jadwal kegiatan pada penelitian ini akan dilakukan dalam kurun waktu kurang lebih enam bulan. Rincian rencana kegiatan pada penelitian ini dapat dilihat

pada tabel di bawah ini :

Tabel 3.4 Jadwal Rencana Kegiatan Penelitian

Kegiatan	Jan 17				Feb 17				Mar 17				Apr 17				Mei 17				Jun 17			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah, Rumusan Masalah dan Objek Penelitian																								
Studi Literatur dan Teori Dasar																								
Rancangan Penelitian																								
Pengumpulan Data																								
Pengolahan Data																								
Analisis Hasil																								
Pembuatan Laporan Penelitian																								

(Halaman ini sengaja dikosongkan)

BAB 4
UJI COBA DAN ANALISIS HASIL

Bab ini menjelaskan mengenai proses uji coba dan analisis dari hasil yang didapatkan dari penelitian ini. Proses uji coba pada penelitian ini meliputi lingkungan uji coba, pelaksanaan uji coba, hasil uji coba dan analisis hasil uji coba.

4.1 Data Uji Coba

Data yang digunakan dalam penelitian ini adalah kumpulan dari URL situs otentik berbahasa Indonesia, berserver di Indonesia atau sering digunakan oleh pengguna internet dari Indonesia dimana setiap situs non-phising (otentik) memiliki 1 atau lebih situs phising *(one to many)*. Kurang lebih terdapat 680 situs yang berhasil dikumpulkan, situs-situs tersebut terdiri dari 340 situs non-phising dan 340 situs phising (50:50). Data situs phising dan situs non-phising ini sendiri diperoleh dari berbagai sumber antara lain yaitu dari penelitian yang dilakukan oleh Dongsong Zhang [2], email dan internet seperti PhishTank, Moz, Alexa, dan beberapa website penyedia informasi mengenai situs phising khususnya di Indonesia.

Data tersebut akan melewati tahap *pre-processing data* (prefiksasi dan ekstraksi fitur) sebelum diolah menggunakan *data mining tool* (Weka versi 3.8) beserta classifier yang telah ditentukan sebelumnya untuk mengukur kinerja dari sistem/model klasifikasi deteksi situs phising yang dibuat. Sama seperti penelitian yang dilakukan oleh Bonda [35], penelitian ini juga akan menggunakan metode *cross validation* dengan jumlah interasi sebanyak *10 folds* dimana pembagian data latih dan data uji dilakukan secara acak/otomatis oleh software Weka 3.8 pada tahap uji coba, karena menurut penelitian tersebut metode *cross validation* dapat menghasilkan kinerja klasifikasi yang jauh lebih baik daripada metode *percentage split*.

4.2 Lingkungan Uji Coba

Lingkungan uji coba merupakan kriteria perangkat pengujian yang digunakan dalam menguji sistem/model yang dibuat pada penelitian ini. Lingkungan uji coba terdiri dari perangkat keras dan perangkat lunak. Adapun perangkat keras yang digunakan ditunjukkan pada tabel di bawah ini :

Tabel 4.1 Spesifikasi Perangkat Keras

Nama	Keterangan
Laptop	ASUS X455L Intel Core i3, Memory 4 GB dan Harddisk 500 GB
Mouse	Logitech B100
Internet Kabel	MyRepublic
Modem	Smartfren Andromax M2P 4G LTE

Selain perangkat keras juga digunakan beberapa perangkat lunak untuk uji coba dalam penelitian ini yang ditunjukkan pada tabel di bawah ini :

Tabel 4.2 Spesifikasi Perangkat Lunak

Nama	Keterangan
Sistem Operasi	Windows 10
Dokumen Editor	Microsoft Word 2016, Microsoft Excel 2016, Microsoft PowerPoint 2016, Microsoft Visio 2007, Microsoft Translator dan Wordpad
Code Editor	Notepad++ dan Notepad
Web Browser	Google Chrome dan Mozilla Firefox
Data Mining Tool	Weka 3.8
Hosting	MasterWeb kapasitas hosting 500 MB dan bandwidth unlimited
Domain	Ekstensi .com
Bahasa Pemrograman	PHP

4.3 Persiapan Uji Coba

Di bawah ini adalah persiapan yang dilakukan oleh peneliti sebelum melakukan proses uji coba :

4.3.1 Pembuatan *Web Crawler*

Web crawler pada penelitian ini digunakan untuk mengekstraksi fitur deteksi situs phising yang ada pada penelitian ini sehingga dapat memudahkan peneliti dalam pengolahan data di tahap berikutnya sekaligus mempercepat proses uji coba bila menggunakan *big data*. Selain itu *web crawler* berfungsi untuk meminimalisir kesalahan validasi nilai fitur yang dilakukan oleh manusia secara manual. Untuk membuat *web crawler* ini, peneliti membutuhkan sumber daya antara lain hosting dengan kapasitas kurang lebih 100 MB dengan bandwidth unlimited serta sebuah domain berekstensi .com yang bersever di MasterWeb, karena peneliti membuat *web crawler* ini secara online (bukan di localhost) agar bisa diakses secara global oleh pengguna internet dan dapat digunakan untuk menunjang penelitian selanjutnya ataupun penelitian sejenis.

Bahasa pemrograman yang digunakan untuk membangun *web crawler* ini adalah PHP dan didukung oleh API *(Application Programming Interface)* sebagai *grabber* beberapa nilai fitur pada penelitian ini. Peneliti melakukan kerjasama dengan Alexa, Google Inc. dan Enclout untuk *grabbing* beberapa nilai fitur menggunakan API. Kerjasama yang dilakukan bersama Alexa bertujuan untuk *grabbing* nilai Alexa Rank, sedangkan kerjasama yang dilakukan bersama Google Inc. ditujukan untuk mendapatkan nilai halaman web dan jumlah JS berdasarkan *PageSpeed Insights Google* serta kerjasama yang dilakukan bersama Enclout difungsikan untuk berbagai macam tujuan yaitu *grabbing* umur domain, masa aktif domain, DNS dan lain-lain.

Selain itu peneliti juga menanamkan *normalization function* pada *web crawler* ini, sehingga mampu melakukan normalisasi terhadap fitur deteksi dengan tipe data *numeric* menggunakan rumus $N = (n - min) / (max - min)$, dimana N adalah *n*ilai normalisasi, n adalah nilai fitur, *min* adalah nilai minimum pada fitur dan *max* adalah nilai maksimum pada fitur. Sebagai contoh, website *http://serbasepeda.com* memiliki umur kurang lebih 1.076 hari, dimana nilai

maksimal dan minimal pada fitur umur domain adalah 11.766 hari dan 0 hari. Sehingga nilai *N* untuk domain *http://serbasepeda.com* pada fitur umur domain adalah *(1.032-0) / (11.766-0) = 0,0914499405*. Sedangkan untuk menunjang penelitian ini, peneliti membuat 2 *web crawler* dengan fungsi berbeda antar lain :

4.3.1.1 *Web Crawler I*

Pada penelitian ini, *Web Crawler I* segaja dibuat untuk membantu peneliti mengekstraksi 11 fitur yang diusulkan pada model klasifikasi deteksi situs phising di Indonesia. Ada 2 langkah dalam pembuatan *Web Crawler I* ini. Langkah pertama adalah menentukan justifikasi dan *function* dari masing-masing fitur yang ada. Justifikasi pada penelitian ini dilakukan untuk memberi keputusan terhadap nilai fitur berdasarkan hipotesa yang ada, sedangkan *function* digunakan untuk mencari nilai fitur berdasarkan hipotesa yang ada. Di bawah ini adalah justifikasi dan *function* dari ke 11 fitur yang dimaksud :

a. F1 : Apakah URL berisi sebuah alamat IP?

   Biasanya sebuah website phising menggunakan alamat IP. Pada penelitian [2] dan [4] juga menggunakan fitur ini untuk mendeteksi situs phising. Jika URL menggunakan alamat IP, maka F1 akan diberi nilai 1, jika tidak, maka F1 diberi nilai -1.

```
if (URL contains IP Address) {
    F1 = 1
} else {
    F1 = -1
}
```

b. F2 : Apakah URL berisi simbol '@'?

   Biasanya website phising akan menggunakan simbol '@' di dalam URL. Fitur ini juga digunakan pada penelitian [2] dan [4]. Jika URL menggunakan simbol '@', maka F2 akan diberi nilai 1, jika tidak, maka F2 diberi nilai -1.

```
if (URL contains @) {
    F2 = 1
} else {
    F2 = -1
}
```

c. F3 : Jumlah afiks (imbuhan)

Penjahat internet biasanya memodifikasi URL situs phising dengan menambahkan beberapa afiks untuk menipu pengguna internet yang mana seolah-olah website tersebut adalah situs ontentik. Afiks sendiri dibagi menjadi 4 jenis yaitu prefiks (imbuhan di depan), infiks (imbuhan di tengah), sufiks (imbuhan di belakang) dan konfiks (imbuhan di depan dan di belakang). Pada penelitian [2] hanya digunakan sufiks saja untuk mendeteksi situs phising, karena peneliti pada penelitian tersebut percaya bahwa website phising menggunakan 2 sufiks domain dan biasanya pengguna internet hanya akan melihat bagian pertama dan mengabaikan bagian lainnya, sehingga akan menuju website phising. Sedangkan pada penelitian [4] menggunakan prefiks dan sufiks untuk mendeteksi situs phising. Akan tetapi pada penelitian ini akan digunakan afiks yang mana mencakup semua jenis imbuhan termasuk imbuhan yang digunakan pada penelitian yang telah disebutkan di atas. Contoh afiks pada penelitian ini adalah "-" dan beberapa ekstensi domain yang dianggap aneh (tidak wajar). Atribut F3 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti pada halaman di balik ini :

F4 = (jumlah afiks – minimal afiks) / (maksimal afiks – minimal afiks)

d. F4 : Usia domain

Usia domain dihitung sejak domain diregistrasi oleh registrar. Pada penelitian [2] dan [4] juga menggunakan fitur ini untuk mendeteksi situs phising, karena pada dasarnya semakin muda umur domain tersebut, kredibilitasnya sebagai situs non-phising semakin dipertanyakan. Atribut F4 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

usia domain = tgl sekarang – tgl daftar *F5 = (usia domain – minimal usia domain) / (maksimal usia domain – minimal usia domain)*

e. F5 : Apakah domain didaftarkan oleh perusahaan?

Pada penelitian [2], fitur ini digunakan untuk memeriksa apakah domain didaftarkan oleh sebuah perusahaan atau non-perusahaan (pribadi). Biasanya situs phising menggunakan nama pribadi, nama samara, nama palsu atau bahkan

menyembunyikan nama pendaftarnya agar tidak mudah dilacak oleh orang lain. Jika domain didaftarkan oleh perusahaan, maka F5 akan diberi nilai 1, jika tidak, maka F5 diberi nilai -1.

```
if (organization != 0) {
    F5 = 1
} else {
    F5 = -1
}
```

f. F6 : Apakah domain diprivasi?

Dalam penelitian [2], fitur ini digunakan untuk memastikan bahwa informasi domain tidak diprivasi oleh sang pemilik, karena kebanyakan domain yang dijadikan sebagai situs phising oleh penjahat internet diprivasi informasinya sehingga orang lain tidak bisa melacak atau mencari informasi mengenai pemilik situs phising tersebut. Jika domain diprivasi, maka F6 akan diberi nilai 1, jika tidak, maka F6 diberi nilai -1.

```
if ((name = 'Registration Private') || (name == 0)) {
    F6 = 1
} else {
    F6 = -1
}
```

g. F7 : Panjang URL

Situs phising biasanya memiliki panjang URL yang tidak lazim. Pada penelitian [4] panjang URL memiliki dampak rasio 51% terhadap sistem deteksi situs phising yang dibuat. Karena nilai rasionya cukup tinggi, maka peneliti menggunakan fitur tersebut pada penelitian ini. Atribut F7 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

```
F7 = (panjang URL – minimal panjang URL) / (maksimal panjang URL –
     minimal panjang URL)
```

h. F8 : Apakah website menggunakan https?

Pada dasarnya setiap situs besar pasti memiliki protokol keamanan yang disebut https, dimana protokol tersebut tidak dimiliki oleh sebagian besar situs phising karena selain digunakan untuk melakukan pencurian data dan tindakan kriminalitas seperti penipuan, situs phising juga difungsikan sebagai media penyebaran malware, virus dan *hacking* oleh penjahat internet yang mana hal tersebut kontradiksi dengan https. Pada penelitian [4] https memiliki dampak rasio 89,2% terhadap sistem deteksi situs phising yang telah dibuat. Karena nilai rasionya sangat tinggi, maka peneliti mencoba memodifikasi sedikit lalu menggunakan fitur tersebut pada penelitian ini. Jika website memiliki https dan https-nya disertai sertifikat SSL *(Secure Socket Layer)*, maka F8 akan diberi nilai -1, jika website memiliki https dan https-nya tidak disetai sertifikat SSL, maka F8 diberi nilai 0, selain itu, maka F8 diberi nilai 1.

```
if ((website has https) && (https has SSL)) {
    F8 = -1
} elseif ((website has https) && (https has't SSL)) {
    F8 = 0
} else {
    F11 = 1
}
```

i. F9 : Apakah website memiliki Alexa Rank?

Alexa Rank adalah perankingan web yang dapat digunakan untuk mengukur seberapa banyak pengunjung ataupun popularitas sebuah website. Nilai Alexa Rank sendiri kurang lebih berkisar antara 1 hingga 25.000.000, semakin kecil rankingnya semakin bagus. Normalnya setiap web dari perusahaan besar pasti memiliki Alexa Rank. Berbeda dengan situs phising yang kebanyakan tidak memiliki Alexa Rank karena kunjungan per harinya terlalu sedikit. Pada penelitian [4] menggunakan fitur pengunjung website yang mana indikator yang digunakan adalah Alexa Rank. Dimana dampak rasio dari pengunjung website terhadap sistem deteksi situs phising yang dibuat sangat besar yaitu 93,2%. Peneliti pada penelitian ini memodifikasi justifikasi pada fitur ini menjadi : jika website memiliki Alexa Rank = 0, maka F9 akan diberi nilai 1, jika website memiliki Alexa Rank > 10.000.000, maka F9 diberi nilai 0, selain itu, maka F9

diberi nilai -1.

```
if (Alexarank = 0) {
    F9 = 1
} elseif (Alexarank > 10000000) {
    F9 = 0
} else {
    F9 = -1
}
```

j. F10 : Jumlah JS

Kebanyakan situs phising memiliki jumlah file JS yang tidak wajar. Hal itu terjadi karena situs phising menggunakan file JS tersebut untuk menyebarkan malware atau virus. Pada penelitian [4] juga digunakan fitur serupa tetapi dengan perhitungan yang berbeda yaitu rata-rata jumlah JS, sedangkan pada penelitian ini akan dihitung jumlah file JS. Atribut F10 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

```
F10 = (jumlah js – minimal js) / (maksimal js – minimal js)
```

k. F11 : Skor halaman web

Situs phising biasanya memakan waktu lebih ketika diakses, karena pada dasarnya situs phising mengandung banyak script, JS, pop-up atau malware. Pada penelitian [30] dikatakan bahwa kecepatan loading web (*response time* dan *latency*) dapat mempengaruhi ranking dari sebuah halaman website. Sehingga dapat ditarik kesimpulan bahwa sebagian besar halaman website yang memiliki skor bagus pasti memiliki *loading web* yang cepat. *PageSpeed Insights Google* adalah salah satu fitur dari Google Inc. yang dapat digunakan untuk menghitung skor pada sebuah halaman web dengan memperhitungkan kriteria-kriteria yang telah disebutkan di atas. Atribut F11 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

```
F11 = (page score – minimal page score) / (maksimal page score – minimal
      page score)
```

Langkah kedua adalah mengimplementasikan justifikasi dan *function* dari masing-masing fitur ke dalam website berbasis PHP dan API *(Application*

Programming Interface), sekaligus mendesain tampilan dari halaman *Web Crawler I* menggunakan HTML dan CSS. Di bawah ini adalah contoh kodenya setelah diimplementasikan :

F1 : Apakah URL berisi sebuah alamat IP?

```
function ip() {
        $this->attr[] = '@attribute ip  { 1,-1 }';
        $this->real[] = false;
        $this->hasil[] = (filter_var($this->base_url(), FILTER_VALIDATE_IP) ?
"1" : "-1");
}
```

F8 : Apakah website menggunakan https?

```
function https() {
        $this->attr[] = '@attribute url_https  { 1,0,-1 }';
        $this->real[] = false;
        $status=((strpos($this->url, 'https://') === FALSE) ? "-1" : "0");
        if($status=='0'){
                $ch = curl_init();
                curl_setopt($ch, CURLOPT_URL, $this->url);
                curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
                curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, true);
                curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, true);
                $content = curl_exec($ch);
                curl_close($ch);
                if($content){
                        $status='1';
                }
        }
        $this->hasil[] = $status;
}
```

Sedangkan pada Gambar 4.1 adalah tampilan dari halaman *Web Crawler I* tersebut.

Gambar 4.1 Tampilan *Web Crawler I*

4.3.1.2 *Web Crawler II*

Agar hasil penelitian ini dapat dibandingkan dengan hasil penelitian dari Dongsong Zhang dan kawan-kawan [2], maka peneliti pada penelitian ini juga membuat sebuah *web crawler* lain bernama *Web Crawler II*. *Web Crawler II* ini dibuat untuk mengekstrak ke 15 fitur yang ada pada penelitian [2]. Sama seperti *Web Crawler I*, pembuatan *Web Crawler II* ini dibagi menjadi 2 langkah. Langkah pertama adalah menentukan justifikasi dan *function* dari masing-masing fitur yang ada. Justifikasi pada penelitian ini dilakukan untuk memberi keputusan terhadap nilai fitur berdasarkan hipotesa yang ada, sedangkan *function* digunakan untuk mencari nilai fitur berdasarkan hipotesa yang ada. Justifikasi yang digunakan dalam *Web Crawler II* mengikuti hipotesa dari penelitian [2]. Akan tetapi nilai justifikasinya yang semula 1 (phising) dan 0 (non-phising) dirubah menjadi 1 (phising) dan -1 (non-phising), selain itu ada pula fitur yang nilai justifikasinya ditukar untuk memudahkan peneliti dalam pembuatan *Web Crawler II* ini antara

lain F8, F9, F10, F12 dan F15, sedangkan untuk *function* yang digunakan mengikuti penelitian ini. Sehingga hasilnya menjadi seperti di bawah ini :

a. F1 : Apakah URL berisi sebuah alamat IP?

   Biasanya sebuah website phising menggunakan alamat IP. Jika URL menggunakan alamat IP, maka F1 akan diberi nilai 1, jika tidak, maka F1 diberi nilai -1.

```
if (URL contains IP Address) {
    F1 = 1
} else {
    F1 = -1
}
```

b. F2 : Apakah URL berisi simbol '@'?

   Biasanya website phising akan menggunakan simbol '@' di dalam URL. Jika URL menggunakan simbol '@', maka F2 akan diberi nilai 1, jika tidak, maka F2 diberi nilai -1.

```
if (URL contains @) {
    F2 = 1
} else {
    F2 = -1
}
```

c. F3 : Apakah karakter dalam URL dikodekan ke dalam UNICODE?

   Biasanya website phising menggunakan URL yang dikodekan ke dalam UNICODE untuk menyembunyikan URL aslinya. Jika URL dikodekan ke dalam UNICODE, maka F3 akan diberi nilai 1, jika tidak, maka F3 diberi nilai -1.

```
if (URL contains UNICODE) {
    F3 = 1
} else {
    F3 = -1
}
```

d. F4 : Jumlah dot (.) dalam URL

   Pada penelitian terdahulu [11] menyatakan bahwa semakin banyak dot dalam sebuah URL, maka semakin besar kemungkinan website tersebut terindikasi

sebagai website phising. Atribut F4 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

F4 = (jumlah dot – minimal dot) / (maksimal dot – minimal dot)

e. F5 : Jumlah sufiks (imbuhan di belakang) dalam nama domain

Biasanya website phising menggunakan 2 sufiks domain, pada umumnya pengguna internet hanya akan melihat bagian pertama dan mengabaikan bagian lainnya, sehingga akan menuju website phising. Atribut F5 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

F5 = (jumlah sufiks – minimal sufiks) / (maksimal sufiks – minimal sufiks)

f. F6 : Usia domain

Usia domain dihitung sejak domain diregistrasi oleh *registrar*. Atribut F6 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

usia domain = tgl sekarang – tgl daftar

F6 = (usia domain – minimal usia domain) / (maksimal usia domain – minimal usia domain)

g. F7 : Masa aktif domain *(expired)*

Jumlah hari yang dihitung adalah jumlah hari sebelum masa aktif domain tersebut berakhir. Atribut F7 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

sisa masa aktif domain = tgl expired – tgl sekarang

F7 = (sisa masa aktif domain – minimal sisa masa aktif domain) / (maksimal sisa masa aktif domain – minimal sisa masa aktif domain)

h. F8 : Apakah domain memiliki DNS *(Domain Name Server)*?

DNS adalah alamat di mana domain tersebut dihostingkan. Jika domain memiliki DNS, maka F8 akan diberi nilai -1, jika tidak, maka F8 diberi nilai 1.

```
if (website has DNS) {
    F8 = -1
} else {
    F8 = 1
}
```

i. F9 : Apakah website memiliki informasi pendaftaran domain (WHOIS)?

Jika website memiliki informasi tersebut, maka F9 akan diberi nilai -1, jika tidak, maka F9 diberi nilai 1.

```
if (website has WHOIS) {
    F9 = -1
} else {
    F9 = 1
}
```

j. F10 : Apakah domain didaftarkan oleh perusahaan?

Jika domain didaftarkan oleh perusahaan, maka F10 akan diberi nilai 1, jika tidak, maka F10 diberi nilai -1.

```
if (organization != 0) {
    F10 = 1
} else {
    F10 = -1
}
```

k. F11 : Apakah domain diprivasi?

Jika domain diprivasi, maka F11 akan diberi nilai 1, jika tidak, maka F11 diberi nilai -1.

```
if ((name = 'Registration Private') || (name == 0)) {
    F11 = 1
} else {
    F11 = -1
}
```

l. F12 : Apakah di dalam website tercantum nomor lisensi ICP *(Internet Content Provider)*?

Jika website mencantumkan nomor lisensi ICP, maka F12 akan diberi nilai 1, jika tidak, maka F12 diberi nilai -1.

```
if (website has ICP number) {
    F12 = -1
} else {
    F11 = 1
}
```

m. F13 : Jumlah *dead link* (link mati) dalam website

Dalam penelitian terdahulu menyatakan bahwa website phising memiliki jumlah link mati lebih banyak dibandingkan website aslinya. Atribut F13 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

```
jumlah dead link = jumlah link – jumlah link hidup

F13 = (jumlah dead link – minimal jumlah dead link) / (maksimal dead link
      –  minimal jumlah dead link)
```

n. F14 : Jumlah *outbound link* (link keluar) dalam website

Normalnya setiap website pasti memiliki link keluar, akan tetapi jika jumlah link keluar terlalu banyak, website tersebut patut dicurigai sebagai website phising. Atribut F14 bertipe data *numeric* (bukan *boolean*), sehingga *function* yang digunakan menjadi seperti di bawah ini :

```
jumlah outbound link = jumlah link – jumlah internal link

F14 = (jumlah outbound link – minimal jumlah outbound link) / (maksimal
      outbound link – minimal jumlah outbound link)
```

o. F15 : Apakah di dalam website *e-business* terdapat informasi sertifikat *e-commerce*?

Jika website menampilkan informasi sertifikat e-commerce, maka F15 akan diberi nilai 1, jika tidak, maka F15 diberi nilai -1.

```
if ((website has certificate number) || (website has certificate link) {
    F15 = -1
} else {
    F15 = 1
}
```

Langkah kedua adalah mengimplementasikan justifikasi dan *function* dari masing-masing fitur ke dalam website berbasis PHP dan API *(Application Programming Interface)*, sekaligus mendesain tampilan dari halaman *Web Crawler II* menggunakan CSS. Di bawah ini adalah contoh kodenya setelah diimplementasikan :

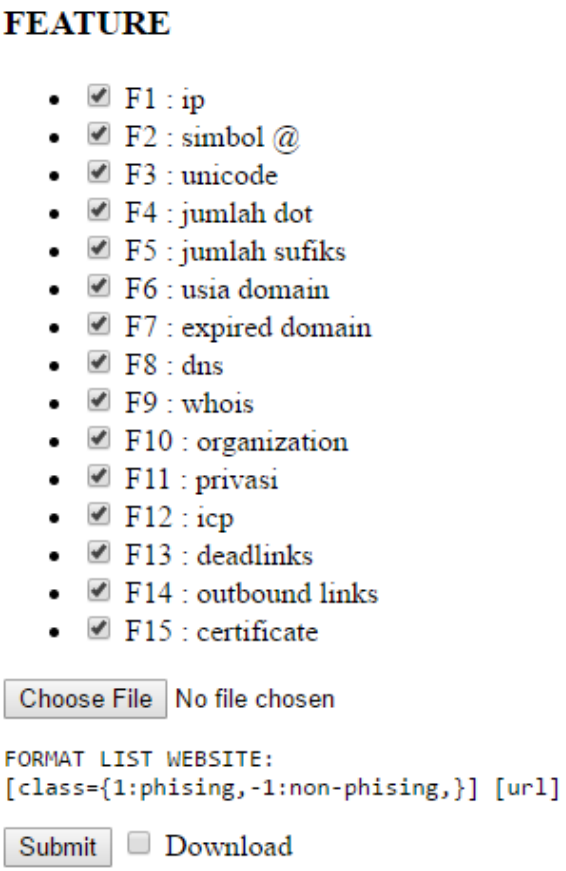
F1 : Apakah karakter dalam URL dikodekan ke dalam UNICODE?*?*

```
function unicode() {
        $this->attr[] = '@attribute unicode  { 1,-1 }';
        $this->real[] = false;
        $this->hasil[] = ((strlen($this->url) !=
        strlen(utf8_decode(urldecode($this->url)))) ? "1" : "-1");
}
```

F8 : Apakah domain didaftarkan oleh perusahaan?

```
function organization() {
        if ($this->whois == null) {
                $this->whois();
        }
        $status = (
                isset($this->whois['registrant_contact']['organization']) &&
                !in_array($this->whois['registrant_contact']['organization'],
                array('N/A','',' ')))?-1:1;
        $this->attr[] = '@attribute organization { 1,-1 }';
        $this->real[] = false;
        $this->hasil[] = $status;
}
```

Sedangkan di bawah ini adalah tampilan dari halaman *Web Crawler II* tersebut :



Gambar 4.2 Tampilan *Web Crawler II*

4.3.2 Hasil Prefiksasi

Pada tahap prefiksasi, semua list URL situs akan diberi imbuhan angka 1 dan -1 di depan URL sesuai dengan statusnya (1 untuk situs phising dan -1 untuk situs non-phising) agar mudah diolah dalam tahap selanjutnya yaitu ekstraksi fitur. Prefiksasi pada penelitian ini dilakukan manual oleh peneliti menggunakan 680 URL situs yang telah dikumpulkan sebelumnya. Output yang dihasilkan adalah file berformat .txt atau .csv berisi data hasil proses prefiksasi. Di bawah ini adalah sampelnya :

- *1 http://www.ebay.com.tw*
- *1 http://www.facebok.com*
- *1 http://hack-gmail-password.com*

- *1 http://ablytube.com/clip/Personal*
- *1 http://88.198.24.90/~consumired/pos/006b3/*
- *1 http://radeemnowevents.ye.vc*
- *-1 http://serbasepeda.com*
- *1 http://serba-sepeda.blogspot.co.id*

4.3.3 Hasil Ekstraksi Fitur

Seperti yang sudah dijelaskan sebelumnya, ekstraksi fitur pada penelitian ini berfungsi untuk mengekstrak fitur yang ada berdasarkan data hasil proses periksasi (.txt atau .csv) menjadi sebuah dataset berformat ARFF *(Attribute-Relation File Format)* berisi header (*relation* dan *attribute*) dan data (nilai fitur) agar bisa langsung diolah menggunakan software Weka. Untuk melakukan ekstraksi fitur, peneliti menggunakan *web crawler* berbasis PHP dan API yang telah dibuat sebelumnya. Pada penelitian ini ada 2 buah *web crawler* yaitu *Web Crawler I* untuk mengekstraksi fitur yang ada dalam penelitian ini dan *Web Crawler II* untuk mengekstraksi fitur yang ada dalam penelitian [2]. Di bawah ini adalah contoh hasil dari ekstraksi fitur menggunakan *Web Crawler I* :

```
@relation phishing

@attribute ip  { 1,-1 }
@attribute simbol_at  { 1,-1 }
@attribute jumlah_dot numeric
@attribute afiks numeric
@attribute usia_domain numeric
@attribute dns  { 1,-1 }
@attribute url_long numeric
@attribute url_https  { 1,0,-1 }
@attribute alexarank  { 1,0,-1 }
@attribute js numeric
@attribute page_score numeric
@attribute status  {'phising','non-phising'}

@data
-1,-1,0.0555555555556,0.0769230769231,0.165432948872,-
1,0.014175257732,1,-1,0.558558558559,0.73,'non-phising'
1,-1,0.166666666667,0.038461538461 5,0,-1,0.0425257731959,-1,-
1,0,0,'phising'
```

Sedangkan di bawah ini adalah contoh hasil ekstraksi fitur menggunakan *Web Crawler II* :

```
@relation phishing

@attribute ip  { 1,-1 }
@attribute simbol_at  { 1,-1 }
@attribute unicode  { 1,-1 }
@attribute jumlah_dot numeric
@attribute sufiks numeric
@attribute usia_domain numeric
@attribute expired_domain numeric
@attribute dns  { 1,-1 }
@attribute url_whois  { 1,-1 }
@attribute organization { 1,-1 }
@attribute privasi { 1,-1 }
@attribute icp { 1,-1 }
@attribute deadlinks numeric
@attribute outbound_links numeric
@attribute certificate  { 1,-1 }
@attribute status  {'phising','non-phising'}

@data
-1,-1,-1,0.0555555555556,0.5,0.165432948872,0.0766827605794,-1,-1,-1,-1,-
1,0.150554675119,0.148177496038,1,'non-phising'
1,-1,-1,0.166666666667,0.25,0,0,-1,1,1,1,-
1,0.00237717908082,0.00237717908082,1,'phising'
```

4.4 Uji Coba

Seperti yang sudah dijelaskan pada skenario uji coba, uji coba dalam penelitian ini dibagi menjadi 4 bagian yaitu uji coba algoritma klasifikasi, uji coba model klasifikasi pada penelitian sebelumnya, uji coba data baru dan uji coba data mining clustering.

4.4.1 Uji Coba Algoritma Klasifikasi

Pada penelitian ini uji coba algoritma klasifikasi dilakukan untuk mendapatkan algoritma klasifikasi dengan kinerja deteksi terbaik yang mana algoritma dengan kinerja deteksi terbaik akan digunakan dalam model klasifikasi pada penelitian ini untuk dibandingkan dengan model klasifikasi pada penelitian

[2]. Dataset yang digunakan dalam uji coba algoritma klasifikasi ini adalah data yang telah melalui tahap prefiksasi dan ekstraksi fitur menggunakan *Web Crawler I*. Sedangkan untuk uji coba algoritma klasifikasi ini dilakukan menggunakan software Weka 3.8 dengan metode *cross validation 10 folds* (interasi).

4.4.1.1 Uji Coba Algoritma SMO *(Sequential Minimal Optimization)*

Uji coba algoritma SMO pada penelitian ini dilakukan untuk mengukur kinerja dari algoritma SMO menggunakan dataset yang telah melalui tahap *pre-processing data* sekaligus untuk memastikan bahwa algoritma SMO mampu memecahkan masalah QP *(Quadratic Programming)* yang timbul selama pelatihan SVM (*Support Vector Machine*). Selain itu dataset yang digunakan dalam penelitian ini juga memiliki skala yang cukup besar, yang mana memungkinkan terjadinya kesalahan ketika memanipulasi matriks. Oleh sebab itu algoritma SMO diharapkan mampu memecahkan masalah-masalah tersebut sehingga menghasilkan kinerja deteksi yang baik. Di bawah ini adalah hasil uji coba algoritma SMO menggunakan software Weka yang disusun berdasarkan *confusion matrix* :

Tabel 4.3 *Confusion Matrix* Algoritma SMO

Prediksi	**Hasil**	
	Phishing	Non-Phising
Phishing	331	9
Non-Phising	19	321

Pada Tabel 4.3 terlihat bahwa algoritma SMO memprediksi 9 situs non-phising sebagai situs phising (FN) dan 19 situs phising sebagai situs non-phising (FP). Jika mengikuti keseluruhan data yang ada yaitu 680 situs yang terdiri dari 340 situs phising dan 340 situs non-phising, maka terdapat 321 prediksi benar terhadap situs non-phising (TN) dan 331 prediksi benar terhadap situs phising (TP). Selain itu algoritma SMO menghasilkan akurasi dan *training time* kurang lebih sekitar 95,88% dan 0,28 detik. Sedangkan untuk kinerja dari algoritma SMO berdasarkan P *(Precision)*, R *(Recall)* dan F *(F-Measure)* bisa dilihat pada Tabel 4.4.

Tabel 4.4 Kinerja Algoritma SMO

Class	P	R	F
Phishing	0,946	0,974	0,959
Non-Phising	0,973	0,944	0,958
Bobot	**0,959**	**0,959**	**0,959**

4.4.1.2 Uji Coba Algoritma Naive Bayes

Uji coba algoritma Naive Bayes pada penelitian ini dilakukan untuk mengukur kinerja dari algoritma Naive Bayes menggunakan dataset yang telah melalui tahap *pre-processing data* sekaligus untuk membuktikan bahwa prediksi yang dihasilkan oleh algoritma Naive Bayes cukup akurat. Karena pada dasarnya algoritma Naive Bayes dapat digunakan untuk memprediksi sesuatu berdasarkan data yang ada menggunakan metode probabilitas dan statistik. Di bawah ini adalah hasil uji coba algoritma Naive Bayes menggunakan software Weka yang telah yang disusun berdasarkan *confusion matrix* :

Tabel 4.5 *Confusion Matrix* Algoritma Naive Bayes

Prediksi	Hasil	
	Phishing	Non-Phising
Phishing	327	13
Non-Phising	8	332

Pada Tabel 4.5 terlihat bahwa algoritma Naive Bayes memprediksi 8 situs non-phising sebagai situs phising (FN) dan 13 situs phising sebagai situs non-phising (FP). Jika mengikuti keseluruhan data yang ada yaitu 680 situs yang terdiri dari 340 situs phising dan 340 situs non-phising, maka terdapat 332 prediksi benar terhadap situs non-phising (TN) dan 327 prediksi benar terhadap situs phising (TP). Selain itu algoritma Naive Bayes menghasilkan akurasi dan *training time* kurang lebih sekitar 96,91% dan 0,04 detik. Sedangkan untuk kinerja dari algoritma Naive Bayes berdasarkan P *(Precision),* R *(Recall)* dan F *(F-Measure)* bisa dilihat pada Tabel 4.6.

Tabel 4.6 Kinerja Algoritma Naive Bayes

Class	P	R	F
Phishing	0,976	0,962	0,969
Non-Phising	0,962	0,976	0,969
Bobot	**0,969**	**0,969**	**0,969**

4.4.1.3 Uji Coba Algoritma Bagging

Uji coba algoritma Bagging pada penelitian ini dilakukan untuk mengukur kinerja dari algoritma Bagging menggunakan dataset yang telah melalui tahap *pre-processing data* sekaligus untuk memastikan bahwa menggunakan beberapa suara yang digabung menjadi prediksi tunggal mampu menghasilkan kinerja deteksi yang baik. Dimana suara/diagnosa yang sering muncul dapat digunakan untuk memberi keputusan apakah situs tersebut termasuk situs phising atau non-phising. Di bawah ini adalah hasil uji coba algoritma Bagging menggunakan software Weka yang telah yang disusun berdasarkan *confusion matrix* :

Tabel 4.7 *Confusion Matrix* dari Algoritma Bagging

Prediksi	Hasil	
	Phishing	Non-Phising
Phishing	329	11
Non-Phising	7	333

Pada Tabel 4.7 terlihat bahwa algoritma Bagging memprediksi 7 situs non-phising sebagai situs phising (FN) dan 11 situs phising sebagai situs non-phising (FP). Jika mengikuti keseluruhan data yang ada yaitu 680 situs yang terdiri dari 340 situs phising dan 340 situs non-phising, maka terdapat 333 prediksi benar terhadap situs non-phising (TN) dan 329 prediksi benar terhadap situs phising (TP). Selain itu algoritma Bagging menghasilkan akurasi dan *training time* kurang lebih sekitar 97,35% dan 0,90 detik. Sedangkan untuk kinerja dari algoritma Bagging berdasarkan P *(Precision)*, R *(Recall)* dan F *(F-Measure)* bisa dilihat pada Tabel 4.8.

Tabel 4.8 Kinerja Algoritma Bagging

Class	P	R	F
Phishing	0,979	0,968	0,973
Non-Phising	0,968	0,979	0,974
Bobot	**0,974**	**0,974**	**0,974**

4.4.1.4 Uji Coba Algoritma Multilayer Perceptron

Uji coba algoritma Multilayer Perceptron pada penelitian ini dilakukan untuk mengukur kinerja dari algoritma Multilayer Perceptron menggunakan dataset yang telah melalui tahap *pre-processing data* sekaligus untuk memastikan bahwa algoritma berbasis Jaringan Syarat Tiruan (JST) yang diusulkan mampu menghasilkan kinerja deteksi yang baik. Pada penelitian yang dilakukan oleh Kyungro Lee dan kawan-kawan [9] digunakan algoritma sejenis berbasis JST yaitu NN *(Neural Network)* untuk membangun sebuah model klasifikasi yang mampu memprediksi *activator* pada CAR *(Constitutive Androstane Receptor)* dan menawarkan informasi struktural mengenai interaksi ligan/protein di dalam hati. Oleh sebab itu peneliti ingin mencoba menggunakan algoritma sejenis tetapi dengan studi kasus yang berbeda. Di bawah ini adalah hasil uji coba algoritma Multilayer Perceptron menggunakan software Weka yang telah yang disusun berdasarkan *confusion matrix* :

Tabel 4.9 *Confusion Matrix* dari Algoritma Multilayer Perceptron

Prediksi	Hasil	
	Phishing	Non-Phising
Phishing	329	11
Non-Phising	10	330

Pada Tabel 4.9 terlihat bahwa algoritma Multilayer Perceptron memprediksi 10 situs non-phising sebagai situs phising (FN) dan 11 situs phising sebagai situs non-phising (FP). Jika mengikuti keseluruhan data yang ada yaitu 680 situs yang terdiri dari 340 situs phising dan 340 situs non-phising, maka terdapat

330 prediksi benar terhadap situs non-phising (TN) dan 329 prediksi benar terhadap situs phising (TP). Selain itu algoritma Multilayer Perceptron menghasilkan akurasi dan *training time* kurang lebih sekitar 96,91% dan 5,58 detik. Sedangkan untuk kinerja dari algoritma Multilayer Perceptron berdasarkan P *(Precision)*, R *(Recall)* dan F *(F-Measure)* bisa dilihat pada Tabel 4.10.

Tabel 4.10 Kinerja Algoritma Multilayer Perceptron

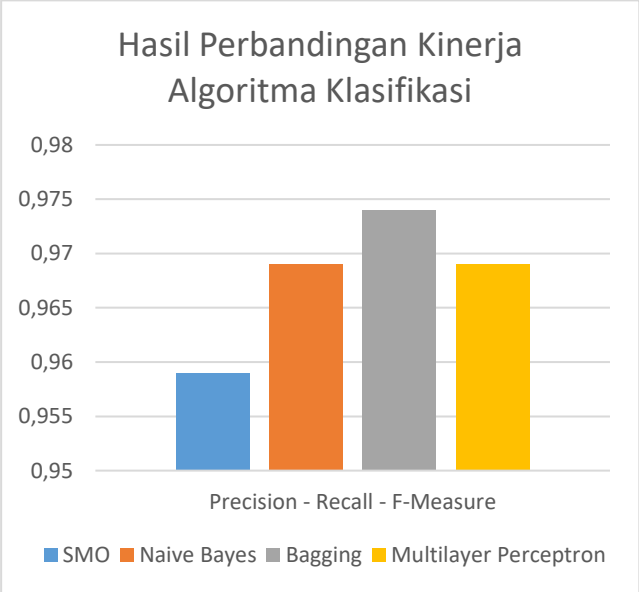
Class	P	R	F
Phishing	0,971	0,968	0,969
Non-Phising	0,968	0,971	0,969
Bobot	**0,969**	**0,969**	**0,969**

Pada uji coba algoritma klasifikasi pada penelitian ini, algoritma Naive Bayes menghasilkan *training time* tercepat yaitu hanya 0,04 detik, akan tetapi nilai FP, TN dan akurasi yang dihasilkan tidak begitu baik, sehingga tidak dipilih sebagai algoritma utama pada model klasifikasi dalam penelitian ini. Akurasi terbaik pada uji coba ini dihasilkan oleh algoritma Bagging yaitu kurang lebih sebesar 97,35%. Selain itu algoritma Bagging juga menghasilkan nilai FP dan TN terbaik yaitu 7 dan 333. Sehingga algoritma Bagging dipilih sebagai algoritma utama pada model klasifikasi pada penelitian ini. Algoritma Bagging dipilih bukan karena memiliki akurasi terbaik, akan tetapi karena algoritma Bagging memiliki nilai FP dan TN terbaik. Sebab seperti yang sudah dijelaskan sebelumnya, peneliti lebih memprioritaskan algoritma dengan nilai FP dan TN terbaik daripada algoritma yang memiliki nilai akurasi tertinggi ataupun *training time* tercepat. Hal ini dilakukan untuk menunjang asumsi dari penelitian ini yaitu lebih baik situs phising dianggap sebagai situs non-phising daripada situs non-phising dianggap sebagai situs phising. Untuk detail dari perbandingan *confusion matrix* dari masing-masing algoritma dapat dilihat pada Tabel 4.11.

Tabel 4.11 Perbandingan *Confusion Matrix* Tiap Algoritma

Algoritma	TP	FN	FP	TN	A	T
SMO	**331**	**9**	19	321	95,88%	0,28 s
Naive Bayes	327	13	8	332	96,91%	**0,04 s**
Bagging	329	11	**7**	**333**	**97,35%**	0,90 s
Multilayer Perceptron	329	11	10	330	96,91%	5,58 s

Sedangkan pada Gambar 4.3 adalah perbandingan kinerja algoritma klasifikasi berdasarkan *Precision*, *Recall* dan *F-Measure* yang mana algoritma Bagging menjadi algoritma penghasil nilai *Precision*, *Recall* dan *F-Measure* terbaik yaitu sebesar 0,974. Seperti yang sudah diketahui sebelumnya, nilai *Precision*, *Recall* dan *F-Measure* mengacu pada nilai TP, FN, FP dan TN, sehingga tak heran bila algoritma Bagging menghasilkan nilai *Precision*, *Recall* dan *F-Measure* yang sama dan tertinggi bila dibandingkan dengan lagoritma lainnya.



Gambar 4.3 Hasil Perbandingan Kinerja Algoritma Klasifikasi

Untuk mengetahui fitur-fitur mana yang paling berpengaruh terhadap kinerja deteksi situs phising, maka peneliti melakukan perankingan fitur pada model klasifikasi yang telah dibuat menggunakan algoritma IGAE *(Info Gain*

Attribute Eval) berbasis Ranker. Di bawah ini adalah hasil dari perankingan fitur yang telah dilakukan :

Tabel 4.12 Rank Fitur

No	Nama Fitur	No Fitur	Bobot
1	Afiks	3	0,87204
2	HTTPS	8	0,65667
3	Panjang URL	7	0,41695
4	Organisasi	5	0,41085
5	Alexa Rank	9	0,37622
6	Usia Domain	4	0,35850
7	Skor Halaman Web	11	0,10468
8	IP Address	1	0,03005
9	JS (JavaScript)	10	0,02207
10	Privasi	6	0,01602
11	Simbol @	2	0,00591

Modifikasi yang dilakukan pada fitur Prefiks (imbuhan di awal) dan Sufiks (imbuhan di akhir) menjadi Afiks (imbuhan di awal, akhir maupun tengah) terbukti membuahkan hasil. Modifikasi yang telah dilakukan membuat fitur ini berada pada posisi pertama. Jika prefiks dan sufiks hanya bisa digunakan untuk mendeteksi imbuhan di awal dan akhir pada sebuah domain, fitur afiks yang telah dimodikasi dapat digunakan untuk mendeteksi imbuhan karakter pada awal, tengah ataupun akhir dari sebuah domain. Berkat modifikasi yang yang telah dilakukan terhadap fitur HTTPS, fitur HTTPS menjadi salah satu fitur yang paling berpengaruh terhadap kinerja deteksi dan menempati posisi. Pada dasarnya situs phising dan non-phising dapat dibedakan dari protokol yang digunakan (HTTPS). Kebanyakan situs phising biasanya tidak memiliki protokol HTTPS, karena di dalam situs phising tersebut biasanya terdapat malware atau virus sehingga memungkinkan terjadinya bentrok dengan protokol HTTPS. Akan tetapi sebagian kecil situs phising memiliki HTTPS, walaupun demikian, belum tentu situs tersebut HTTPS-

nya disertai sertifikat SSL. Sedangkan pada penelitian ini, apabila ada situs yang memiliki HTTPS dan SSL, maka akan diberi label sebagai situs non-phising, sedangkan bila memiliki HTTPS dan tidak disertai SSL, maka akan diberi label situs suspicious, selain itu maka akan diberi label sebagai situs phising.

Panjang URL berada pada urutan ketiga, karena fitur tersebut dapat digunakan untuk menghitung jumlah karakter dari sebuah tautan/situs. Biasanya situs phising selalu menggunakan URL yang sangat panjang untuk mengelabui korbannya. Jadi tidak heran bila fitur panjang URL menjadi salah satu fitur yang paling berpengaruh terhadap deteksi. Fitur Organisasi menempati posisi selanjutnya. Biasanya setiap situs otentik atau situs terkenal mendaftarkan perusahaan/organisasinya ketika membeli domain atau start up bisnisnya, sedangkan untuk situs phising lebih banyak menggunakan nama palsu/tanpa nama agar pemiliknya tidak mudah dilacak oleh orang lain. Dengan demikian nama asli dari penjahat internet tidak akan tercemar bila situs tersebut sudah diketahui sebagai situs phising.

Biasanya sebuah website otentik memiliki pengunjung yang sangat banyak bila dibandingkan dengan situs phising yang hanya mengandalkan visitor dari hasil kesalahan pengguna internet atau email spam (jebakan) yang dibuat. Pada penelitian yang dilakukan oleh Neda Abdelhamid [4] digunakan sebuah fitur yang digunakan untuk mengukur pengunjung website, akan tetapi parameter yang digunakan adalah Alexa Rank. Oleh sebab itu peneliti pada penelitian ini juga menggunakan Alexa Rank menggunakan justifikasi yang telah dimodifikasi, hasilnya fitur ini menjadi salah satu fitur yang cukup berpengaruh terhadap deteksi situs phising dengan menempati posisi kelima. Sedangkan pada posisi keenam ditempati oleh fitur Usia Domain. Sejatinya situs phising biasanya memiliki umur yang jauh lebih muda bila dibandingkan dengan situs otentik, sehingga tak heran bila Usia Domain menjadi salah satu fitur yang cukup berguna bila digunakan untuk membedakan situs asli dengan situs phising.

Di lain sisi fitur baru yang diusulkan pada penelitian ini menempati posisi ketujuh. Fitur baru berbasis pendekatan konten dan URL yang dimaksud adalah fitur untuk mendeteksi skor halaman dari sebuah web. Fitur ini diusulkan karena pada penelitian [30] menyatakan bahwa situs yang memiliki loading yang cepat

pasti memiliki skor halaman web yang bagus. Biasanya situs phising memiliki loading yang sangat lambat karena terlalu banyak menyelundupkan file berupa virus, malware dan sejenisnya, sehingga situs phising kerap kali memiliki skor halaman web yang buruk. Walaupun fitur baru ini tidak masuk dalam kategori tiga besar, akan tetapi fitur baru ini memberikan kontribusi yang cukup baik untuk meningkatkan kinerja deteksi situs phising. Sehingga tidak sia-sia fitur ini disarankan untuk dimasukkan ke dalam penelitian ini.

Fitur IP Address menyusul di belakangnya. Fitur ini digunakan untuk memeriksa apakah situs menggunakan IP Address sebagai URL domainnya atau tidak. Pada dasarnya bila diamati, situs otentik memang jarang sekali menggunakan IP Address sebagai URL utamanya. Oleh sebab itu bila ada situs yang menggunakan IP Address sebagai URL utamanya, maka situs tersebut patut dicurigai. Pada penelitian ini, peneliti hanya menemukan beberapa situs phising yang menggunakan IP Address sebagai URL domainnya, sehingga membuat fitur ini tidak begitu mencolok kontribusinya. Pada posisi 3 dari bawah ada fitur JS (JavaScript).. Fitur ini dipilih karena pada penelitian [5], fitur ini digunakan untuk mendeteksi situs phising karena virus dan malware bermula dari sana. Pada penelitian ini, fitur JS yang semula menghitung rata2 JS pada keseluruhan halaman web dimodifikasi menjadi jumlah JS pada halaman web. Akan tetapi hasilnya kurang memuaskan, walaupun demikian, fitur yang tanpa modifikasi akan memakan waktu yang cukup lama ketika ekstraksi fitur menggunakan web crawler sebab harus mengumpulkan semua JS yang ada pada seluruh halaman website lalu membaginya berdasarkan jumlah halaman yang ada.

Pada posisi 10 atau 2 dari bawah ada fitur Privasi. Fitur ini digunakan untuk memeriksa apakah domain tersebut diprivasi atau tidak. Sebagian besar situs phising sengaja diprivasi agar orang lain tidak bisa memeriksa lebih dalam mengenai situs tersebut. Walaupun demikian adapula situs phising yang tidak diprivasi karena terkendala oleh biaya, contohnya adalah situs phising amatir, situs phising yang baru merintis karir, situs phising abal-abal dan sejenisnya. Kemungkinan kebanyakan situs yang digunakan sebagai data utama pada penelitian ini adalah situs-situs tersebut, sehingga hasil dari fitur Privasi tidak begitu terlihat. Posisi buncit ditempati oleh fitur yang digunakan untuk mendeteksi simbol @.

Apabila sebuah situs mengandung simbol @ di dalam URL, maka situs tersebut dapat dikatakan sebagai situs phising. Sama seperti fitur IP Address, kurangnya data situs phising yang mengandung simbol @ pada penelitian ini membuat fitur ini menduduki posisi paling buncit.

4.4.2 Uji Coba Model Klasifikasi Pada Penelitian Sebelumnya

Uji coba model klasifikasi ini dilakukan untuk mengukur kinerja model klasifikasi pada penelitian sebelumnya [2] menggunakan data yang sama dengan penelitian ini yang mana hasilnya nanti akan dibandingkan untuk mengetahui apakah model klasifikasi yang dibuat dalam penelitian ini mampu mengungguli model klasifikasi yang dibuat pada penelitian sebelumnya. Dataset yang digunakan dalam uji coba model klasifikasi ini adalah data yang telah melalui tahap prefiksasi dan ekstraksi fitur menggunakan *Web Crawler II* dimana algoritma yang digunakan dalam uji coba model klasifikasi ini adalah algoritma Sequential Minimal Optimization (SMO). Karena menurut Dongsong Zhang dan kawan-kawan [2], algoritma SMO bisa menghasilkan kinerja yang baik bila diterapkan ke dalam model klasifikasi pada penelitian tersebut. Sama seperti uji coba algoritma klasifikasi, pada uji coba model klasifikasi ini juga menggunakan software Weka 3.8 dengan metode *cross validation 10 folds* (interasi). Tabel 4.12 menunjukkan hasil dari uji coba model klasifikasi pada penelitian [2] menggunakan algoritma SMO dan dataset yang sama dengan penelitian ini yang disusun berdasarkan *confusion matrix*.

Tabel 4.13 *Confusion Matrix* Model Klasifikasi Pada Penelitian Dongsong Zhang (2014) Menggunakan Algoritma SMO

Prediksi	**Hasil**	
	Phishing	Non-Phising
Phishing	283	57
Non-Phising	75	265

Pada Tabel 4.13 terlihat bahwa model klasifikasi ini memprediksi 75 situs non-phising sebagai situs phising (FN) dan 57 situs phising sebagai situs non-phising (FP). Jika mengikuti keseluruhan data yang ada yaitu 680 situs yang terdiri dari 340 situs phising dan 340 situs non-phising, maka terdapat 265 prediksi benar terhadap situs non-phising (TN) dan 383 prediksi benar terhadap situs phising (TP). Selain itu model klasifikasi ini menghasilkan akurasi dan *training time* kurang lebih sekitar 80,59% dan 0,39 detik. Sedangkan untuk kinerja dari model klasifikasi ini berdasarkan P *(Precision)*, R *(Recall)* dan F *(F-Measure)* bisa dilihat pada Tabel 4.14.

Tabel 4.14 Kinerja Model Klasifikasi Pada Penelitian Dongsong Zhang (2014)

Class	P	R	F
Phishing	0,791	0,832	0,811
Non-Phising	0,823	0,779	0,801
Bobot	**0,807**	**0,806**	**0,806**

4.4.3 Uji Coba Data Baru

Uji coba ini dilakukan untuk menguji model klasifikasi yang telah dibuat dan model klasifikasi pada penelitian sebelumnya [2] apakah mampu membedakan situs phising dan situs non-phising secara akurat menggunakan data baru ataupun data yang sudah ada sebagai sampel. Hal ini dilakukan untuk memastikan bahwa model klasifikasi yang telah dibuat mampu mendeteksi situs phising dengan kinerja yang sangat baik bila dibandingkan dengan model klasifikasi pada penelitian sebelumnya [2] yang hanya menggunakan fitur dasar saja. Untuk menguji kedua model klasifikasi tersebut, peneliti mencoba untuk memasukkan data sampel kurang lebih sekitar 20 URL situs yang terdiri dari 80% situs yang sudah ada (8 situs phising dan 8 situs non-phising) dan 20% situs baru (2 situs phising dan 2 situs non-phising) menggunakan data training sebanyak 680 buah (340 situs phising dan 340 situs non phising). Detail dari data sampel tersebut bisa dilihat pada Tabel 4.15.

Tabel 4.15 Data Sampel

No	URL	Keterangan	Status
1	*https://ebay.com*	Sudah Ada	Non-Phising
2	*https://facebook.com*	Sudah Ada	Non-Phising
3	*https://gmail.com*	Sudah Ada	Non-Phising
4	*https://ibank.bankmandiri.co.id*	Sudah Ada	Non-Phising
5	*https://kaskus.co.id*	Sudah Ada	Non-Phising
6	*https://paypal.com*	Sudah Ada	Non-Phising
7	*https://pb.garena.co.id*	Sudah Ada	Non-Phising
8	*https://taobao.com*	Sudah Ada	Non-Phising
9	*http://www.ebay.com.tw*	Sudah Ada	Phising
10	*http://www.facebok.com*	Sudah Ada	Phising
11	*http://hack-gmail-password.com*	Sudah Ada	Phising
12	*http://ablytube.com/clip/Personal*	Sudah Ada	Phising
13	*https://kaskusbluemoviess.allalla.com*	Sudah Ada	Phising
14	*http://88.198.24.90/~consumired/pos/006b3/*	Sudah Ada	Phising
15	*http://radeemnowevents.ye.vc*	Sudah Ada	Phising
16	*https://taobaohacks.wordpress.com*	Sudah Ada	Phising
17	*https://integra.its.ac.id*	Baru	Non-Phising
18	*https://jalantikus.com*	Baru	Non-Phising
19	*http://robots.my/blog/cpp/cpp.php*	Baru	Phising
20	*http://boaliablhighschool.edu.bd/tint/T/Y1.html*	Baru	Phising

Agar bisa diolah menggunakan software Weka, maka data sampel tersebut harus melewati tahap *pre-processing data* terlebih dahulu yaitu prefiksasi dan ekstraksi fitur menggunakan *Web Crawler I* maupun *Web Crawler II*. Prefiksasi yang dilakukan pada data sampel ini cukup berbeda dengan data utama. Apabila data utama diberi prefiks 1 (phising) atau -1 (non-phising), maka pada data baru ini

diberi prefiks “?” (tanda tanya). Di bawah ini adalah hasil dari proses prefiksasi tersebut :

Tabel 4.16 Hasil Prefiksasi Data Sampel

Masukan	Hasil
https://ebay.com	*? https://ebay.com*
https://facebook.com	*? https://facebook.com*
https://gmail.com	*? https://gmail.com*
https://ibank.bankmandiri.co.id	*? https://ibank.bankmandiri.co.id*
https://kaskus.co.id	*? https://kaskus.co.id*
https://paypal.com	*? https://paypal.com*
https://pb.garena.co.id	*? https://pb.garena.co.id*
https://taobao.com	*? https://taobao.com*
http://www.ebay.com.tw	*? http://www.ebay.com.tw*
http://www.facebok.com	*? http://www.facebok.com*
http://hack-gmail-password.com	*? http://hack-gmail-password.com*
http://ablytube.com/clip/Personal	*? http://ablytube.com/clip/Personal*
https://kaskusbluemoviess.allalla.com	*? https://kaskusbluemoviess.allalla.com*
http://88.198.24.90/~consumired/pos/006b3/	*? http://88.198.24.90/~consumired/pos/006b3/*
http://radeemnowevents.ye.vc	*? http://radeemnowevents.ye.vc*
https://taobaohacks.wordpress.com	*? https://taobaohacks.wordpress.com*
https://integra.its.ac.id	*? https://integra.its.ac.id*
https://jalantikus.com	*? https://jalantikus.com*
http://robots.my/blog/cpp/cpp.php	*? http://robots.my/blog/cpp/cpp.php*
http://boaliablhighschool.edu.bd/tint/T/Y1.html	*? http://boaliablhighschool.edu.bd/tint/T/Y1.html*

Pemberian prefiks “?” pada data sampel dilakukan untuk mensimulasikan bahwa semua situs pada data sampel tersebut statusnya masih belum diketahui. Sedangkan prefiks 1 (phising) dan -1 (non-phising) diberikan pada data utama karena status pada data utama wajib diketahui terlebih dahulu agar bisa diolah menggunakan software Weka. Hasil dari proses prefiksasi tersebut adalah file berformat .txt atau .csv yang mana bila dilakukan ekstraksi fitur akan menghasilkan sebuah dataset berformat ARFF *(Attribute-Relation File Format)* berisi header (*relation* dan *attribute*) dan data (nilai fitur). Untuk menguji kedua model klasifikasi yang ada, maka peneliti melakukan ekstraksi fitur menggunakan *Web Crawler I* dan *Web Crawler II*. Di bawah ini adalah contoh hasil ekstraksi fitur menggunakan *Web Crawler I* :

```
@relation phishing

@attribute ip  { 1,-1 }
@attribute simbol_at  { 1,-1 }
@attribute afiks numeric
@attribute usia_domain numeric
@attribute organization { 1,-1 }
@attribute privasi { 1,-1 }
@attribute url_long numeric
@attribute url_https  { 1,0,-1 }
@attribute alexarank  { 1,0,-1 }
@attribute js numeric
@attribute page_score numeric
@attribute status  {'phising','non-phising'}

@data
-1,-1,0.0384615384615,0.681447502548,-1,-1,0.00259403372244,-1,-
1,0.157407407407,0.47,?
-1,-1,0.0384615384615,0.630224260958,-1,-1,0.00778210116732,-1,-
1,0.111111111111,0.88,?
-1,-1,0.0384615384615,0.680682976555,-1,-1,0.00389105058366,-1,-
1,0.0277777777778,0.8,?
-1,-1,0.0769230769231,0.568212708121,-1,-1,0.0220492866407,-1,-
1,0.166666666667,0.86,?
-1,-1,0.0769230769231,0.279561671764,-1,-1,0.00778210116732,-1,-
1,0.37037037037,0.66,?
```

Sedangkan di bawah ini adalah contoh hasil ekstraksi fitur menggunakan *Web Crawler II* :

```
@attribute jumlah_dot numeric
@attribute sufiks numeric
@attribute usia_domain numeric
@attribute expired_domain numeric
@attribute dns  { 1,-1 }
@attribute url_whois  { 1,-1 }
@attribute organization { 1,-1 }
@attribute privasi { 1,-1 }
@attribute icp { 1,-1 }
@attribute deadlinks numeric
@attribute outbound_links numeric
@attribute certificate  { 1,-1 }
@attribute status  {'phising','non-phising'}

@data
-1,-1,-1,0,0.25,0.681679129844,0.10652726759,1,-1,1,-1,-1,0.015,0.015,-1,?
-1,-1,-1,0,0.25,0.630438477226,0.793444475841,-1,-1,1,-1,-
1,0.0383333333333,0.0383333333333,-1,?
-1,-1,-1,0,0.25,0.680914343984,0.303758123764,-1,-1,1,-1,-
1,0.00583333333333,0.00583333333333,-1,?
-1,-1,-1,0.111111111111,0.5,0.568405846363,0.0912687199774,1,-1,1,-1,-
1,0,0,-1,?
-1,-1,-1,0.0555555555556,0.5,0.279656696125,0.207120655552,-1,-1,1,-1,-
1,0.144166666667,0.144166666667,-1,?
```

Hasil dari proses ekstraksi fitur tersebut kemudian diolah menggunakan software Weka berbasis *supplied test set* untuk mengetahui seberapa akurat prediksi deteksi dari masing-masing model klasifikasi. Tabel 4.17 menunjukkan hasil uji coba data baru pada model klasifikasi pada penelitian ini. Dari 20 sampel data yang diuji, model klasifikasi pada penelitian ini menghasilkan 19 prediksi benar setara 95% dan 1 prediksi salah setara dengan 5%. Tentunya hal ini mengindikasi bahwa model klasifikasi pada penelitian ini bekerja sangat baik walaupun menggunakan data yang bervariasi dan berkomposisi 50% situs non-phising dan 50% situs phising. Prediksi salah yang dihasilkan pada uji coba ini yaitu ketika memprediksi situs *https://taobaohacks.wordpress.com*.

Tabel 4.17 Hasil Uji Coba Data Baru Pada Model Klasifikasi Di Dalam Penelitian Ini

No	URL	Realita	Prediksi
1	*https://ebay.com*	Non-Phising	Non-Phising
2	*https://facebook.com*	Non-Phising	Non-Phising
3	*https://gmail.com*	Non-Phising	Non-Phising
4	*https://ibank.bankmandiri.co.id*	Non-Phising	Non-Phising
5	*https://kaskus.co.id*	Non-Phising	Non-Phising
6	*https://paypal.com*	Non-Phising	Non-Phising
7	*https://pb.garena.co.id*	Non-Phising	Non-Phising
8	*https://taobao.com*	Non-Phising	Non-Phising
9	*http://www.ebay.com.tw*	Phising	Phising
10	*http://www.facebok.com*	Phising	Phising
11	*http://hack-gmail-password.com*	Phising	Phising
12	*http://ablytube.com/clip/Personal*	Phising	Phising
13	*https://kaskusbluemoviess.allalla.com*	Phising	Phising
14	*http://88.198.24.90/~consumired/pos/006b3/*	Phising	Phising
15	*http://radeemnowevents.ye.vc*	Phising	Phising
16	*https://taobaohacks.wordpress.com*	Phising	Non-Phising
17	*https://integra.its.ac.id*	Non-Phising	Non-Phising
18	*https://jalantikus.com*	Non-Phising	Non-Phising
19	*http://robots.my/blog/cpp/cpp.php*	Phising	Phising
20	*http://boaliablhighschool.edu.bd/tint/T/Y1.html*	Phising	Phising

Ada beberapa faktor yang menyebabkan prediksi yang dilakukan oleh model klasifikasi pada penelitian ini terhadap situs *https://taobaohacks.wordpress.com* menjadi salah antara lain :

a. Data Utama

Perlu diketahui bahwa proporsi jumlah data utama yang dijadikan data training dapat mempengaruhi hasil prediksi terhadap situs *https://taobaohacks.wordpress.com*, karena berdasarkan uji coba yang telah dilakukan semakin banyak data training yang ada, semakin baik pula prediksi yang dihasilkan.

b. Class

Pada model klasifikasi ini peneliti hanya menggunakan 2 class yaitu phising dan non-phising. Apabila menggunakan 3 class bisa jadi situs *https://taobaohacks.wordpress.com* tidak langsung diprediksi sebagai situs non-phising akan tetapi diprediksi sebagai situs suspicious (situs yang dicurigai sebagai situs phising). Peneliti pada penelitian ini tidak menggunakan class suspicious karena data yang digunakan pada penelitian ini hanya terbagi menjadi 2 class saja yaitu situs phising dan non-phising. Apabila peneliti ingin menggunakan 3 class (phising, non-phising dan suspicious) sebagai hasil prediksi, maka peneliti harus menggunakan data mining clustering dengan asumsi class yang ada masih belum diketahui. Apabila menggunakan data mining clustering belum tentu hasil yang didapatkan bisa sebaik klasifikasi, karena algoritma, data yang ada dan fitur yang digunakan juga dapat mempengaruhi kinerja dari data mining clustering. Oleh sebab itu peneliti juga melakukan uji coba data mining clustering pada penelitian ini untuk melihat kecenderungan data dan memastikan bahwa model klasifikasi memang sangat cocok digunakan untuk membedakan situs phising dengan situs non-phising.

c. Akurasi Model Klasifikasi

Akurasi dari model klasifikasi pada penelitian ini tidak 100% akurat yaitu hanya sekitar 97,35%. Tentunya hal ini dapat mempengaruhi hasil prediksi, akan tetapi walaupun demikian model klasifikasi pada penelitian ini memberikan hasil yang cukup baik yaitu hanya menghasilkan 1 prediksi salah dari 20 situs sampel yang diuji coba atau setara 95%. Selain itu, peneliti juga lebih mementingkan nilai FP dan TN daripada nilai akurasi, sehingga tak masalah apabila situs phising diprediksi sebagai situs non-phising daripada situs non-phising diprediksi sebagai situs phising. Dalam uji coba ini ada 1 situs phising yang diprediksi sebagai situs non-phising.

d. Fitur

Walaupun situs ini dilabeli sebagai situs phising, situs ini justru memiliki umur yang sangat tua (lahir tahun 2000). Hal ini yang menyebabkan model klasifikasi pada penelitian ini tak mampu memprediksi secara benar. Selain itu situs ini juga tidak mengandung IP Address dan simbol @ di dalam URL-nya, sehingga membuat website ini condong menjadi situs non-phising

e. Time

Waktu dimana situs tersebut diuji pada model klasifikasi pada penelitian ini juga mempengaruhi hasil dari prediksi, karena ada beberapa fitur yang menggunakan *real time* data pada saat proses ekstraksi fitur contohnya umur, organisasi, privasi, https, Alexa Rank, JS dan skor halaman web yang mana bisa berubah sewaktu-waktu karena pada dasarnya peneliti tidak tahu kapan pemilik web tersebut akan melakukan update pada sisi internal.

Sedangkan pada di bawah ini adalah hasil uji coba data pada model klasifikasi pada penelitian sebelumnya [2].

Tabel 4.18 Hasil Uji Coba Data Baru Pada Model Klasifikasi Di Dalam Penelitian Dongsong Zhang (2014)

No	URL	Realita	Prediksi
1	*https://ebay.com*	Non-Phising	Non-Phising
2	*https://facebook.com*	Non-Phising	Non-Phising
3	*https://gmail.com*	Non-Phising	Non-Phising
4	*https://ibank.bankmandiri.co.id*	Non-Phising	Non-Phising
5	*https://kaskus.co.id*	Non-Phising	Phising
6	*https://paypal.com*	Non-Phising	Non-Phising
7	*https://pb.garena.co.id*	Non-Phising	Phising
8	*https://taobao.com*	Non-Phising	Non-Phising
9	*http://www.ebay.com.tw*	Phising	Non-Phising
10	*http://www.facebok.com*	Phising	Non-Phising
11	*http://hack-gmail-password.com*	Phising	Phising

12	*http://ablytube.com/clip/Personal*	Phising	Phising
13	*https://kaskusbluemoviess.allalla.com*	Phising	Phising
14	*http://88.198.24.90/~consumired/pos/006b3/*	Phising	Phising
15	*http://radeemnowevents.ye.vc*	Phising	Phising
16	*https://taobaohacks.wordpress.com*	Phising	Non-Phising
17	*https://integra.its.ac.id*	Non-Phising	Non-Phising
18	*https://jalantikus.com*	Non-Phising	Phising
19	*http://robots.my/blog/cpp/cpp.php*	Phising	Phising
20	*http://boaliablhighschool.edu.bd/tint/T/Y1.html*	Phising	Phising

Pada Tabel 4.18 terlihat bahwa model klasifikasi pada penelitian sebelumnya [2] menghasilkan 14 prediksi benar dan 6 prediksi salah (3 situs non phising diprediksi sebagai situs phising dan 3 situs phising diprediksi sebagai situs non-phising) dari 20 sampel data yang diuji. Dengan demikian dapat ditarik kesimpulan bahwa kinerja dari model klasifikasi pada penlitian ini jauh lebih baik daripada kinerja model klasifikasi pada penelitian sebelumnya [2].

4.4.4 Uji Coba Data Mining Clustering

Peneliti melakukan uji coba data mining clustering untuk menganalisis kecenderungan data, sekaligus untuk memastikan bahwa data mining klasifikasi memang sangat cocok digunakan untuk membedakan situs phising dan situs non phising bila dibandingkan dengan data mining clustering. Pada uji coba ini peneliti menggunakan dataset (data utama) hasil ekstraksi fitur dari *Web Crawler I* dan data baru (data sampel) yang sama pada uji coba data baru, sedangkan software yang digunakan adalah Tanagra 1.4. Peneliti memakai algoritma K-Means dengan 3 cluster (phising, non-phising dan suspicious), 40 interasi dan 20 trial error menggunakan *Mc Queen average computation* dan *standard seed random generator* tanpa *ditance normalization* pada software Tanagra. Algoritma K-Means digunakan pada data mining clustering ini karena mampu mengkategorikan data

berdasarkan centroid (titik tengah) dari nilai fitur yang bersangkutan sehingga diharapkan mampu membagi data secara adil. Hasil uji coba yang telah dilakukan dapat dilihat pada tabel di bawah ini :

Tabel 4.19 Hasil Uji Coba Data Baru Menggunakan Data Mining Clustering

No	URL	Realita	Prediksi
1	*https://ebay.com*	Non-Phising	Non-Phising
2	*https://facebook.com*	Non-Phising	Non-Phising
3	*https://gmail.com*	Non-Phising	Non-Phising
4	*https://ibank.bankmandiri.co.id*	Non-Phising	Non-Phising
5	*https://kaskus.co.id*	Non-Phising	Non-Phising
6	*https://paypal.com*	Non-Phising	Non-Phising
7	*https://pb.garena.co.id*	Non-Phising	Non-Phising
8	*https://taobao.com*	Non-Phising	Non-Phising
9	*http://www.ebay.com.tw*	Phising	Suspicious
10	*http://www.facebok.com*	Phising	Phising
11	*http://hack-gmail-password.com*	Phising	Suspicious
12	*http://ablytube.com/clip/Personal*	Phising	Suspicious
13	*https://kaskusbluemoviess.allalla.com*	Phising	Phising
14	*http://88.198.24.90/~consumired/pos/006b3/*	Phising	Suspicious
15	*http://radeemnowevents.ye.vc*	Phising	Non-Phising
16	*https://taobaohacks.wordpress.com*	Phising	Suspicious
17	*https://integra.its.ac.id*	Non-Phising	Non-Phising
18	*https://jalantikus.com*	Non-Phising	Phising
19	*http://robots.my/blog/cpp/cpp.php*	Phising	Suspicious
20	*http://boaliablhighschool.edu.bd/tint/T/Y1.html*	Phising	Suspicious

Seperti yang sudah diketahui sebelumnya, pada uji coba ini peneliti menggunakan data mining clustering dengan 3 cluster (phising, non-phising dan

suspicious). Dari 484 data yang diuji (termasuk data sampel), data mining clustering menghasilkan 374 prediksi benar, 229 prediksi hampir benar dan 81 prediksi salah, bila dikalkulasi uji coba ini menghasilkan akurasi kurang lebih 71,42% atau -25,93% dari akurasi model klasifikasi yang telah dibuat pada penelitian ini. Sedangkan dari 20 data sampel yang diuji, data mining clustering pada uji coba ini menghasilkan 2 prediksi salah (masing-masing ada 1 situs non-phising diprediksi sebagai situs phising dan situs phising diprediksi sebagai situs non-phising), 7 prediksi hampir benar (suspicious/dicurigai sebagai situs phising) dan 11 prediksi benar atau setara 72,5%. Ada beberapa hal yang menyebabkan kinerja dari data mining clustering tidak begitu bagus antara lain sebagai berikut :

a. Data Utama

   Perlu diketahui bahwa proporsi jumlah data yang diolah dapat mempengaruhi clusterisasi dan kecenderungan data. Semakin kompleks data yang digunakan semakin baik pula prediksi yang dihasilkan oleh data mining clustering.

b. Penelitian Sebelumnya

   Beberapa penelitian terdahulu lebih banyak menggunakan data mining klasifikasi untuk mendeteksi situs phising daripada menggunakan data mining clustering, karena pada dasarnya fitur-fitur berbasis pendekatan konten dan URL yang digunakan memang khusus dibuat untuk data mining klasifikasi bukan untuk data mining clustering, sehingga bila ingin menggunakan data mining clustering, maka harus melakukan riset dan pemilihan fitur ulang untuk mendapatkan hasil yang memuaskan.

c. Akurasi

   Dari hasil uji coba data mining clustering yang telah dilakukan, terlihat akurasinya tidak begitu baik yaitu hanya 71,42%. Tentunya hal ini sangat berpengaruh terhadap kinerja dari deteksi situs phising itu sendiri.

d. Fitur

   Fitur yang memiliki andil besar dalam prediksi salah (situs phising diprediksi sebagai situs non-phising) pada uji coba data mining clustering ini adalah fitur Alexa Rank. Bila melihat dari hasil uji coba yang telah dilakukan, situs phising *http://radeemnowevents.ye.vc* justru memiliki Alexa Rank yang sangat baik, sehingga clusterisasi yang dilakukan oleh data mining clustering menjadi salah.

Sedangkan untuk situs non-phising *https://jalantikus.com* dimasukkan ke dalam cluster situs phising oleh data mining clustering pada uji coba ini karena situs tersebut informasi domainnya diprivasi oleh sang pemilik. Itu terlihat pada fitur Privasi yang mana situs tersebut memiliki nilai fitur 1.

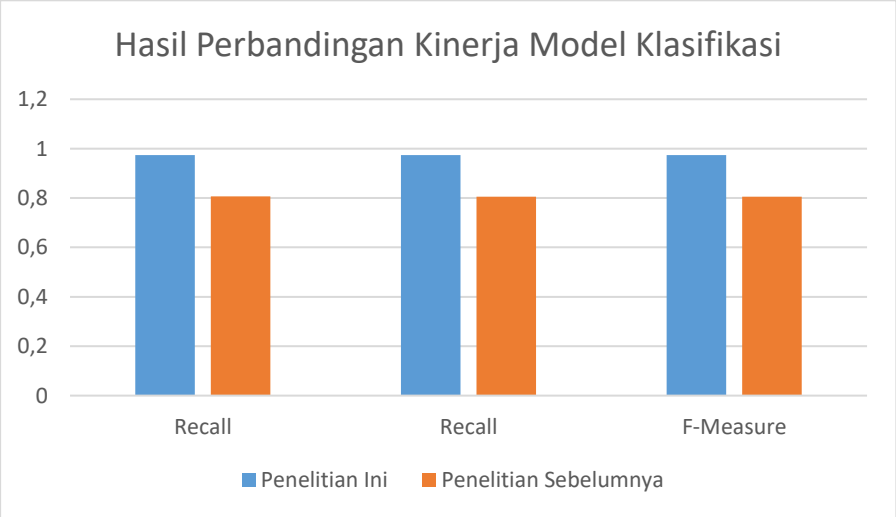
Akan tetapi walaupun demikian, data mining clustering yang telah dilakukan pada penelitian ini tidak langsung memprediksi *https://taobaohacks.wordpress.com* sebagai situs non-phising, akan tetapi memprediksinya sebagai situs suspicious (yang dicurigai sebagai situs phising). Bila pada model klasifikasi pada penelitian ini situs tersebut cenderung dianggap sebagai situs non-phising yang seharusnya adalah situs phising, maka pada data mining clustering yang telah dilakukan pada penelitian ini, situs tersebut menempati cluster 3 (suspicious). Dengan demikian prediksi yang dilakukan oleh data mining clustering kepada situs tersebut adalah hampir benar.

4.5 Analisis Hasil

Dari hasil uji coba algoritma klasifikasi pada penelitian ini diputuskan bahwa algoritma Bagging berhak menjadi algoritma utama pada model klasifikasi pada penlitian ini karena algoritma Bagging memliki nilai FP *(False Positive)*, TN *(True Negative)*, A (akurasi), P *(Precision)*, R *(Recall)* dan F *(F-Measure)* terbaik bila dibandingkan dengan algoritma lainnya. Akan tetapi nilai yang paling diprioritaskan adalah nilai FP dan TP sebab sesuai dengan asumsi yang dilontarkan oleh penulis yaitu bahwa lebih baik situs phising diprediksi sebagai situs non-phising daripada situs non-phising diprediksi sebagai situs phising. Makna tersebut sepenuhnya terkandung di dalam nilai FP dan TP, sehingga algoritma Bagging dipilih untuk dijadikan algoritma utama dalam model klasifikasi pada penelitian ini agar bisa dibandingkan dengan model klasifikasi pada penelitian sebelumnya [2].

Walaupun algoritma Bagging kalah dalam aspek lain seperti TP *(True Positive)*, FN *(False Negative)* dan T *(training time)* oleh algoritma SMO *(Sequential Minimal Optimization)* dan Naive Bayes masing-masing kalah 2 point dan 0,86 detik, algoritma Bagging tetap dipilih dalam penelitian ini karena sudah memenuhi asumsi dan kontribusi teoritis dari penelitian ini yaitu menciptakan

model klasifikasi deteksi situs phising di Indonesia dengan kinerja baik. Sedangkan untuk fitur yang paling berpengaruh terhadap deteksi situs phising adalah Fitur Afiks. Itu semua tak lepas dari modifikasi yang dilakukan oleh peneliti terhadap fitur tersebut. Di lain sisi, fitur baru berbasis pendekatan konten dan URL yang ditambahkan pada penelitian ini juga memberikan kontribusi yang lumayan bagus. Hal itu terlihat ketika fitur baru tersebut menempati posisi ke 7 dari 11 fitur yang ada. Tentunya hal ini menjadi kredit positif tersendiri pada penelitian ini, karena modifikasi dan penambahan fitur baru mampu meningkatkan kinerja model klasifikasi pada penelitian sebelumnya.



Gambar 4.4 Hasil Perbandingan Kinerja Model Klasifikasi

Untuk memastikan bahwa model klasifikasi tersebut telah berhasil meningkat kinerjanya, maka peneliti mencoba untuk membandingkan kinerja dari model klasifikasi pada penelitian ini dengan model klasifikasi pada penelitian sebelumnya [2] yang hanya menggunakan fitur dasar saja. Gambar 4.4 dan Tabel 4.20 menunjukkan bahwa model klasifikasi yang dibuat dalam penelitian ini mengungguli model klasifikasi pada penelitian sebelumnya berdasarkan TP *(True Positive)*, FN *(False Negative)*, FP *(False Positive)*, TN *(True Negative)*, A (akurasi), P *(Precision)*, R *(Recall)* dan F *(F-Measure)* kecuali T *(training time)* masing-masing unggul 46, 46, 68, 68, 16,76%, 0,167, 0,168, 0,168 dan -0,51 detik.

Tabel 4.20 Hasil Perbandingan Kinerja Model Klasifikasi

Model Kalsifikasi	TP	FN	FP	TN	A	T
Penelitian Ini	**329**	**11**	**7**	**333**	**97,35%**	0,90 s
Penelitian Sebelumnya	283	57	75	265	80,59%	**0,39 s**

Terbukti model klasifikasi pada penelitian ini mampu menghasilkan kinerja yang sangat baik dengan prediksi benar 95% ketika dimasukkan data baru (sampel) sebanyak 20 buah. Hasilnya hanya ada 1 prediksi salah yaitu ketika memprediksi situs phising *https://taobaohacks.wordpress.com*. Situs phising tersebut justru diprediksi sebagai situs non-phising, tapi walaupun demikian tidak menjadi masalah karena berdasarkan asumsi yang ada lebih baik situs phising diprediksi sebagai situs non-phising daripada situs non-phising diprediksi sebagai situs phising dan hal itu terjadi pada uji coba data baru menggunakan model klasifikasi pada penelitian sebelumnya [2] dan data mining clustering. Sehingga dapat dikatakan bahwa kinerja dari model klasifikasi pada penlitian ini lebih baik daripada model klasifikasi pada penelitian sebelumnya [2] dan model klasifikasi ini sangat cocok digunakan untuk membedakan situs phising dan non-phising daripada data mining clustering.

4.6 Kontribusi

Bila diamati, hasil dari penelitian ini menghasilkan dua jenis kontribusi yaitu kontribusi secara teoritis dan kontribusi secara praktis. Di bawah ini adalah analisa dari masing-masing kontribusi yang dihasilkan oleh penelitian ini :

4.6.1 Kontribusi Secara Teoritis

Sebelum membahas kontribusi teoritis yang dihasilkan oleh penelitian ini. Alangkah baiknya jika peneliti menjabarkan secara ringkas alasan dari pembuatan penelitian ini. Perlu diketahui bahwa semakin bertambah banyaknya situs phising di dunia dan minimnya penelitian mengenai situs phising di Indonesia menjadi awal mula dari pembuatan penelitian ini. Pada dasarnya orang awam yang baru memulai debutnya di internet atau transaksi secara online sangat rentan terhadap serangan situs phising yang dilakukan oleh penjahat internet bila dibandingkan dengan

pengguna internet yang sudah lama berkecimpung di dunia maya. Karena pada dasarnya situs phising bisa mengelabui korbannya dengan cara menyamar seolah-olah menjadi situs otentik (situs asli) dari sumber yang sah.

Selain itu situs phising juga memberikan informasi atau petunjuk palsu yang menyesatkan yang mana apabila pengguna internet melakukan perintah/mengikuti petunjuk yang diberikan, maka data dari pengguna internet bisa dicuri oleh penjahat internet. Bahkan pengguna internet yang tergolong veteran di dunia maya, juga masih berpeluang terkena jebakan dari situs phising, apabila tampilan dari situs phising benar-benar sangat mirip dengan situs aslinya atau informasi/petunjuk yang diberikan situs phising sangat menggiurkan. Tentunya hal ini akan merugikan pengguna internet dan memberikan rasa cemas bagi orang awam ketika bertransaksi secara online khususnya di Indonesia yang mayoritas penduduknya masih baru mengenal internet. Kerugian yang ditimbulkan oleh situs phising bisa berupa kerugian secara finansial hingga *data loss*.

Oleh sebab itu peneliti pada penelitian melakukan penelitian yang mana pada intinya bisa membedakan situs phising dan situs non-phising khususnya di Indonesia secara akurat. Dengan menggunakan data training situs berbahasa Indonesia, bersever di Indonesia dan sering di akses oleh pengguna internet dari Indonesia, peneliti mencoba membuat sebuah sistem/model klasifikasi yang mampu mendeteksi situs phising secara akurat. Model klasifikasi dipilih dalam penelitian ini karena pada penelitian sejenis [2], [4]-[5], juga digunakan model serupa untuk mendeteksi situs phising.

Menurut [2], ada empat jenis pendekatan yang dapat digunakan untuk mendeteksi situs phising antara lain pendekatan berbasis *blacklist*, *visual similarity*, fitur konten dan URL, dan *third-party search engine*. Pada penelitian ini digunakan pendekatan berbasis fitur konten dan URL. Pendekatan berbasis fitur konten dan URL berfokus pada analisis karakteristik konten dan URL dari situs target. Yang menjadi tantangan utama agar penelitian ini mampu memberikan kontribusi secara praktis bila menggunakan pendekatan berbasis fitur konten dan URL adalah kesulitan dalam menentukan fitur konten dan URL yang berpengaruh terhadap deteksi itu sendiri. Oleh karena itu peneliti melakukan studi literatur pada beberapa

penelitian terdahulu yang sejenis [2], [4]-[5] dan [30] untuk mendapatkan fitur yang relevan dan mampu menghasilkan kinerja deteksi yang baik.

Hasilnya peneliti mendapatkan fitur-fitur yang wajib dimasukkan ke dalam model klasifikasi, karena rasio yang dihasilkan pada penelitian sebelumnya cukup tinggi sehingga sangat berpengaruh terhadap hasil klasifikasi. Selain itu peneliti juga memodifikasi fitur-fitur yang sudah ada dan menambahkan fitur baru berbasis pendekatan konten dan URL sehingga diharapkan mampu menghasilkan model klasifikasi dengan kinerja deteksi yang baik. Fitur baru yang dimaksud adalah F11 (skor halaman web), sedangkan fitur yang dimodifikasi adalah F4 (afiks), F8 (HTTPS), F9 (Alexa Rank) dan F10 (JS).

Di lain sisi, classifier/algoritma klasifikasi juga menjadi kunci untuk menghasilkan kinerja deteksi yang baik. Pada penelitian sebelumnya [2], [4]-[5] untuk pemilihan algoritma klasifikasi dilakukan dengan cara membandingkan algoritam satu dengan lainnya, yang mana algoritma klasifikasi dengan kinerja yang baik dipilih dalam model klasifikasi tersebut. Penelitian [2] dan [4] memilih algoritma dengan akurasi terbaik, sedangkan penelitian [5] memilih algoritma dengan *training time* terbaik. Untuk mewujudkan asumsi dan kontribusi teoritis pada penelitian ini, maka peneliti lebih condong untuk memilih algoritma klasifikasi dengan nilai FP *(False Positive)*, TN *(True Positive)* dan akurasi terbaik, karena peneliti berasumsi bahwa lebih baik situs phising dianggap sebagai situs non-phising daripada situs non-phising dianggap sebagai situs phising.

Algoritma yang digunakan pada penelitian ini antara lain yaitu SMO *(Sequential Minimal Optimization)*, Naive Bayes, Bagging dan Multilayer Perceptron. Algoritma SMO dipakai karena dapat memecahkan masalah QP *(Quadratic Programming)* yang timbul selama pelatihan SVM (*Support Vector Machine*) dimana pada penelitian ini akan digunakan data dalam skala besar yang mana memungkinkan terjadinya kesalahan ketika memanipulasi matriks. Naive Bayes digunakan dalam penelitian ini karena algoritma tersebut adalah algoritma yang paling sering dipakai dalam penelitian sejenis. Bagging diusulkan dalam model tersebut karena mampu memberikan sebuah keputusan menggunakan beberapa suara yang digabung menjadi prediksi tunggal. Sedangkan Multilayer Peceptron diusulkan karena pada penelitian [9] digunakan algoritma sejenis

berbasis JST yaitu NN *(Neural Network)* untuk membangun sebuah model klasifikasi yang mampu memprediksi *activator* pada CAR *(Constitutive Androstane Receptor)* dan menawarkan informasi struktural mengenai interaksi ligan/protein di dalam hati.

Diharapkan dari keempat algoritma tersebut ada algoritma yang mampu menghasilkan akurasi yang sangat baik sehingga sesuai dengan kontribusi praktis dan tujuan awal dari pembuatan penelitian ini yaitu membuat model klasifikasi untuk deteksi situs phising dengan kinerja baik. Hasilnya, algoritma Bagging mampu menghasilkan nilai FP, TN dan akurasi tertinggi yaitu 7, 333 dan 97,35%. Untuk memastikan bahwa model klasifikasi yang dibuat kinerjanya sudah meningkat, maka peneliti membandingkan hasil uji coba model klasifikasi ini dengan model klasifikasi pada penelitian sebelumnya [2]. Hasilnya, model klasifikasi pada penelitian ini mengungguli model klasifikasi pada penelitian sebelumnya dalam beberapa aspek antara lain yaitu TP *(True Positive)*, FN *(False Negative)*, FP *(False Positive)*, TN *(True Negative)*, A (akurasi), P *(Precision)*, R *(Recall)* dan F *(F-Measure)* kecuali T *(training time)* yang mana masing-masing unggul 46, 46, 68, 68, 16,76%, 0,167, 0,168, 0,168 dan -0,51 detik.

Dengan hasil kinerja deteksi yang sangat baik secara teoritis model klasifikasi ini dapat digunakan untuk membuat/menunjang penelitian lain yang sejenis dan lebih spesifik lagi seperti sistem deteksi bank online phising, sistem deteksi sosial media phising, sistem deteksi situs jual beli phising di Indonesia atau sejenisnya. Sebagai contoh untuk sistem deteksi situs jual beli phising di Indonesia, bisa menggunakan model klasifikasi yang ada pada penelitian ini dan menambahkan fitur khusus untuk situs jual beli yang bisa didapatkan pada penelitian [2], sedangkan untuk sistem deteksi sosial media phising atau sistem deteksi bank online phising bisa menambahkan fitur baru berbasis konten dan URL seperti fitur untuk mendeteksi mata uang yang digunakan, logo kartu kredit ataupun menghitung jumlah form dan *input box*.

4.6.2 Kontribusi Secara Praktis

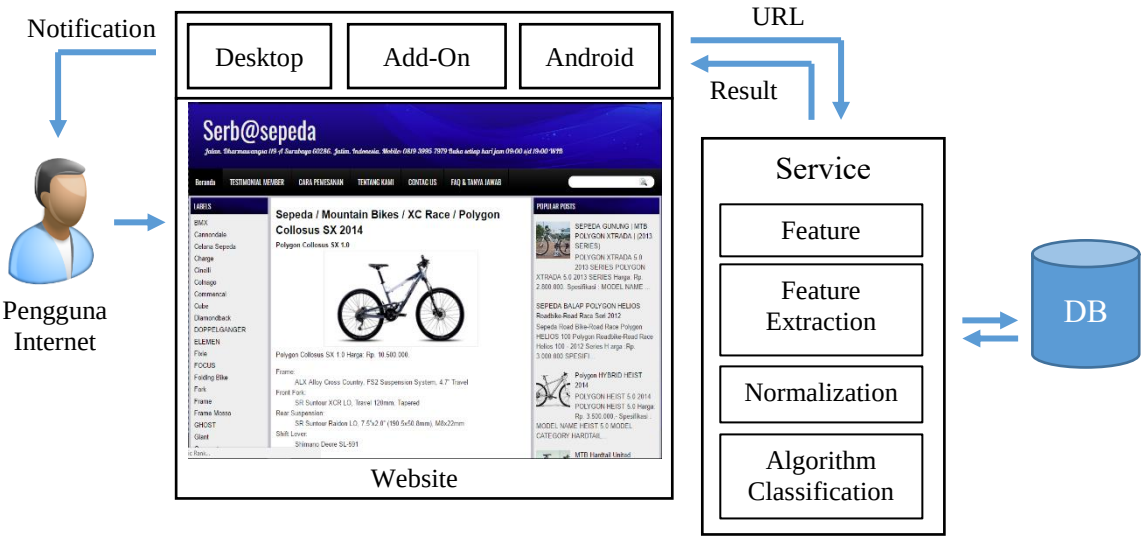
Seperti yang sudah dijelaskan sebelumnya bahwa model klasifikasi ini bisa dikembangkan pada penelitian selanjutnya untuk membuat sistem/model sejenis

atau yang lebih spesifik lagi seperti sistem deteksi bank online phising, sistem deteksi sosial media phising, sistem deteksi situs jual beli phising di Indonesia atau sejenisnya. Dimana model-model tersebut dapat diimplementasikan ke dalam sebuah *service*. Kontribusi praktis dari penelitian ini adalah membantu dan mempermudah peneliti pada penelitian selanjutnya dalam pengembangan sistem deteksi situs phising menggunakan model klasifikasi yang telah dibuat yang diimplementasikan menjadi sebuah s*ervice*. Pada penelitian ini, *service* ini nantinya bisa diimplementasikan pada dua jenis teknologi yaitu *standalone website* atau API *(Application Programming Interface)*.

API adalah sebuah *function* berupa *service* (model klasifikasi) yang telah dibuat yang mampu mendeteksi situs phising dan situs non-phising secara *real time* dan akurat yang dapat digunakan pada platform tertentu. Rancangan implementasi API pada penelitian ini dibagi menjadi tiga jenis berdasarkan platformnya yaitu desktop, android dan add-on pada *web browser*. Aplikasi/software berbasis desktop yang ditanami oleh *service* ini dapat mendeteksi situs phising dan memberikan infromasi *warning* ketika pengguna mengakses situs tertentu menggunakan *web browser* atau membuka email yang mengandung URL situs menggunakan MUA *(Mail User Agent)* seperti Thunderbird melalui laptop/komputer desktop. Untuk *service* yang ditanam pada aplikasi android dikhususkan agar bisa mendeteksi situs phising melalui *handphone* yang memiliki OS *(Operating System)* android ketika pengguna membuka situs tertentu, sedangkan *service* yang ditanam pada *web browser* berupa add-on akan memberikan informasi warning pada *web browser* jika pengguna mengakses situs phising.

Pada dasarnya rancangan dari *service* yang ditanam pada aplikasi/software berbasis desktop, android atau add-on pada *web browser* memiliki alur yang sama. Gambar 4.5 menunjukkan alur dan rancangan ringkas dari *service* yang ditanam pada aplikasi/software berbasis desktop, android atau add-on pada *web browser*. Apabila pengguna mengakses halaman situs tertentu, maka secara otomatis URL dari situs tersebut akan diolah oleh *service* bersama database yang ada pada platform desktop, android atau add-on pada *web browser* untuk diklasifikasikan menjadi situs phising atau phising non-phising. Tahap awal yang dilakukan oleh *service* adalah melakukan ekstraksi fitur untuk studi kasus (URL + database) yang

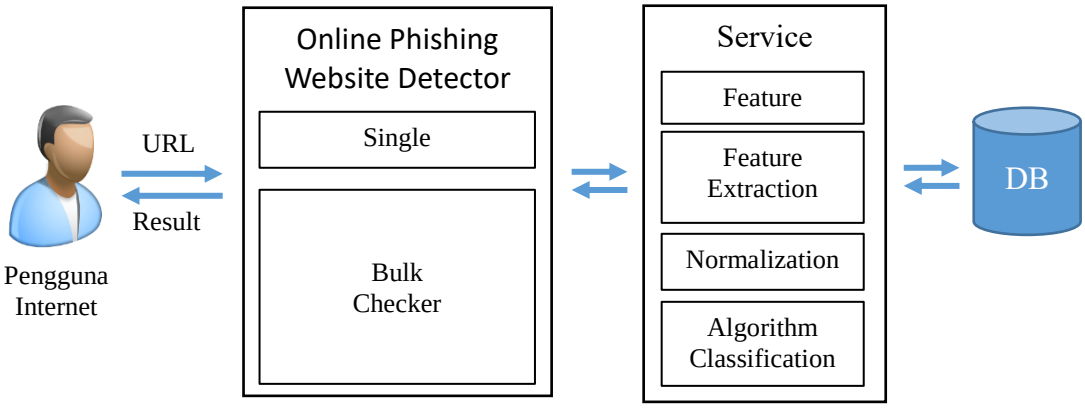
ada berdasarkan fitur-fitur yang tersedia menggunakan *web crawler* yang sudah dirancang pada penelitian ini. Langkah kedua, *service* akan melakukan normalisasi pada fitur yang memiliki tipe data *numeric* untuk menghasilkan deteksi situs phising yang akurat. Langkah ketiga, dataset dari hasil ekstraksi fitur akan diuji menggunakan algoritma Bagging untuk menentukan apakah situs yang diakses termasuk situs phising atau situs non-phising. Hasilnya berupa *output* dalam bentuk informasi *warning* bila ternyata situs yang diakses adalah situs phising, sedangkan bila situs yang diakses bukanlah situs phising, maka *service* tidak akan memberikan informasi apa-apa.



Gambar 4.5 Rancangan Implementasi *Service* Menjadi API

Sedangkan *standalone website* diartikan sebagai situs yang berdiri sendiri yang dapat digunakan untuk mengecek status sebuah situs apakah termasuk situs phising atau situs non phising. *Standalone website* tidak dapat mendeteksi situs phising secara otomatis, karena *standalone website* bersifat pasif. Harus ada pengguna internet yang memberikan input agar *standalone website* bisa aktif/berkerja. Karena pada dasarnya *standalone website* hanya digunakan untuk memeriksa apakah situs tertentu termasuk situs phising atau situs non-phising. Salah satu contoh *standalone website* yang dapat digunakan untuk memeriksa apakah sebuah website termasuk situs phising atau situs non-phising adalah *phishtank.com*. Tapi sayangnya *standalone website* tersebut tidak bisa melakukan

bulk check (cek secara massal). Oleh sebab itu, hal tersebut bisa menjadi peluang pada penelitian ke depannya agar model klasifikasi pada penelitian ini bisa dirancang untuk mendeteksi situs phising secara massal.



Gambar 4.6 Rancangan Implementasi *Service* Menjadi *Standalone Website*

Pada Gambar 4.6 adalah rancangan dari *service* (model klasifikasi) bila diimplementasikan menjadi *standalone website* yang bisa melakukan *bulk check*. Untuk menjalankan *service* yang ada pada *standalone website*, maka pertama-tama pengguna internet harus mengakses *standalone website* dan memasukkan URL situs yang ingin diperiksa statusnya pada *input box* yang tersedia (bisa *single* atau massal). Secara otomatis, *service* akan bekerja sesuai dengan prosedur yang ada. Tahap awal yang dilakukan oleh *service* adalah melakukan ekstraksi fitur untuk studi kasus (URL/*bulk URL* + database) yang ada berdasarkan fitur-fitur yang tersedia menggunakan *web crawler* yang sudah dirancang pada penelitian ini. Langkah kedua, *service* akan melakukan normalisasi pada fitur yang memiliki tipe data *numeric* untuk menghasilkan deteksi situs phising yang akurat. Langkah ketiga, dataset dari hasil ekstraksi fitur akan diuji menggunakan algoritma Bagging untuk menentukan apakah situs yang diakses termasuk situs phising atau situs non-phising. Hasilnya berupa *output* dalam bentuk informasi berupa status situs yang telah diinputkan pada website *standalone website*.

Dengan demikian informasi yang diberikan oleh sistem pada *standalone website* dapat menjadi acuan atau pertimbangan bagi pengguna internet agar tidak mengunjungi situs dengan hasil positif terindikasi sebagai situs phising. Sedangkan untuk *service* yang ditanamkan pada platform desktop, android atau add-on pada

web browser menggunakan API, maka informasi yang diberikan oleh sistem berupa *warning* dapat menghindarkan pengguna internet terkena serangan *malware* atau *hijacking* dari situs phising maupun mengurangi resiko kerugian finansial dan *data loss* yang ditimbulkan oleh dari situs phising itu sendiri.

(Halaman ini sengaja dikosongkan)

BAB 5
KESIMPULAN DAN SARAN

Bab ini menjelaskan mengenai kesimpulan dari penelitian yang telah dilakukan dan saran untuk menunjang penelitian selanjutnya yang mungkin bisa dilakukan.

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan dapat dipetik beberapa kesimpulan antara lain sebagai berikut :

a. Bila dilihat dari hasil uji coba pada penelitian ini, model klasifikasi memang sangat cocok digunakan untuk membedakan situs phising dan non phising di Indonesia daripada menggunakan data mining clustering, karena sejatinya model klasifikasi mampu melakukan pengklasifikasian menggunakan data ada sebagai data training untuk menentukan dan memastikan apakah situs-situs yang dimaksud termasuk situs phising atau non-phising berdasarkan persyaratan-persyaratan (fitur-fitur) yang telah ditentukan sebelumnya.
b. Untuk menghasilkan kinerja deteksi yang baik, peneliti menggunakan teknik deteksi berdasarkan analisa situs berbasis pendekatan fitur konten dan URL yang ada pada penelitian sebelumnya untuk mendapatkan fitur-fitur yang revelan dan cocok digunakan dalam model klasifikasi pada penelitian ini.
c. Pada penelitian ini SMO *(Sequential Minimal Optimization)* adalah algoritma klasifikasi yang menghasilkan TN *(True Negative)* dan FN *(False Negative)* terbaik yaitu 331 dan 9 dibandingkan dengan algoritma lainnya. Hal ini mengindikasikan bahwa prediksi yang dihasilkan oleh algoritma SMO sangat baik terhadap situs phising. Untuk *training time* tercepat dimiliki oleh algoritma Naive Bayes yaitu kurang lebih sekitar 0,04 detik lebih cepat 0,24, 0,86 dan 5,54 detik dari algoritma SMO, Bagging dan Multilayer Perceptron.
d. Sedangkan Bagging adalah algoritma yang memiliki kinerja klasifikasi terbaik bila dibandingkan dengan algoritma lainnya. Algoritma Bagging unggul dalam beberapa aspek antara lain FP *(False Positive)*, TN *(True Negative)*, A (akurasi),

P *(Precision)*, R *(Recall)* dan F *(F-Measure)* dari algoritma lainnya, sehingga algoritma Bagging digunakan dalam model klasifikasi pada penelitian ini untuk dibandingkan dengan model klasifikasi pada penelitian sebelumnya. Kinerja klasifikasi yang baik ini tak lepas dari normalisasi yang dilakukan ketika ekstraksi fitur.

e. Ketika dibandingkan dengan model klasifikasi pada penelitian sebelumnya yang hanya menggunakan fitur dasar saja, model klasifikasi yang telah dibuat pada penelitian ini unggul di beberapa aspek seperti TP *(True Positive)*, FN *(False Negative)*, FP *(False Positive)*, TN *(True Negative)*, A (akurasi), P *(Precision)*, R *(Recall)* dan F *(F-Measure)* kecuali T *(training time)* yang mana masing-masing unggul 46, 46, 68, 68, 16,76%, 0,167, 0,168, 0,168 dan -0,51 detik. Modifikasi fitur lama dan penambahan fitur baru yang diusulkan berdasarkan pendekatan konten dan URL terbukti mampu meningkatkan kinerja klasifikasi pada penelitian ini. Fitur-fitur yang dimaksud antara lain adalah skor halaman web, JS, Alexa Rank, HTTPS dan afiks.

5.2 Saran

Untuk penelitian selanjutnya yang mungkin bisa dilakukan adalah bagaimana cara mengoptimasi nilai FP *(False Positive)*, TN *(True Negative)* dan *training time*. Cara yang dapat dilakukan untuk mengotimasi nilai FP dan TN adalah memperkaya data utama, memodifikasi fitur atau menemukan fitur baru yang dinggap cukup berpengaruh untuk membedakan situs phising dan non-phising. Sedangkan untuk mengoptimasi *training time*, cara yang dapat dilakukan adalah menggunakan algoritma sejenis Naive Bayes atau menemukan algoritma baru. Pada penelitian ini, algoritma Naive Bayes memiliki *training time* tercepat yaitu hanya 0,04 detik, namun algoritma Naive Bayes hanya memiliki nilai FP, TN dan akurasi masing-masing sebesar 8, 332 dan 96,91% kalah 1 point dan 0,44% dari algoritma Bagging.

Perlu diketahui bahwa model klasifikasi yang telah dibuat dapat digunakan dalam penelitian lain untuk membuat sistem deteksi situs phishing yang lebih spesifik lagi (misalnya sistem deteksi bank online phising, sistem deteksi sosial media phising, sistem deteksi situs jual beli phising di Indonesia atau sejenisnya). Karena pada dasarnya hasil dari penelitian ini memang dapat dikembangkan pada

penelitian selanjutnya untuk menciptakan jenis sistem deteksi situs phising lainnya. Selain itu, hal lain yang mungkin bisa dilakukan pada penelitian selanjutnya adalah mengimplementasikan model klasifikasi yang telah dibuat menjadi *service* pada *standalone website* atau platform tertentu (desktop, android, add-on pada *web browser* atau sejenisnya) yang mana informasi yang diberikan oleh model klasifikasi yang telah dibuat dapat menghindarkan dan mengurangi resiko pengguna internet dijarah oleh penjahat internet maupun terkena serangan malware atau *hijacking* dari situs phising.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

[1] IdWebHost, "Mengenal Phishing," *Blog IDWebHost*, 04-Jan-2010. .

[2] D. Zhang, Z. Yan, H. Jiang, dan T. Kim, "A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites," *Inf. Manage.*, vol. 51, no. 7, hal. 845–853, Nov 2014.

[3] APWG, "Phishing Activity Trends Report, 4th Quarter 2016." 2017.

[4] N. Abdelhamid, A. Ayesh, dan F. Thabtah, "Phishing detection based Associative Classification data mining," *Expert Syst. Appl.*, vol. 41, no. 13, hal. 5948–5959, Okt 2014.

[5] Y. Li, L. Yang, dan J. Ding, "A minimum enclosing ball-based support vector machine approach for detection of phishing websites," *Opt. - Int. J. Light Electron Opt.*, vol. 127, no. 1, hal. 345–351, Jan 2016.

[6] Erdi Susanto, "Data Mining Menggunakan Weka," *Erdi Susanto*, Jun-2012. .

[7] Jiawei Han, M. K. Micheline Kamber, dan Jian Pei, *Data Mining : Concepts and Techniques*. 2006.

[8] C. Catal dan M. Nangir, "A sentiment classification model based on multiple classifiers," *Appl. Soft Comput.*, vol. 50, hal. 135–141, Jan 2017.

[9] K. Lee, H. You, J. Choi, dan K. T. No, "Development of pharmacophore-based classification model for activators of constitutive androstane receptor," *Drug Metab. Pharmacokinet.*

[10] K. Thirumala, A. C. Umarikar, dan T. Jain, "A new classification model based on SVM for single and combined power quality disturbances," *IEEE*, Feb 2017.

[11] H. Zhang, G. Liu, T. W. S. Chow, dan W. Liu, "Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, hal. 1532–1546, Okt 2011.

[12] M. J. HASUGIAN, *Menguasai analisis kompleks dalam matematika teknik*, vol. 5. Bandung : Rekayasa Sains, 2006.

[13] G. G. Chowdhury, *Introduction to Modern Information Retrieval*, vol. 205. Library Association Publishing, 1999.

[14] Christopher D. Manning, Prabhakar Raghavan, dan Hinrich Schütze, *An Introduction to Information Retrieval*. Cambridge University Press, 2009.

[15] A. Abbasi, Z. Zhang, D. Zimbra, H. Chen, dan Nunamaker, *Detecting fake websites: The contribution of statistical learning theory*. MIS Quarterly: Management Information Systems, 2010.

[16] J. F.-T. HE Gao-Hui, "Phishing Detection System Based on SVM Active Learning Algorithm," *Comput. Eng.*, vol. 37, no. 19, hal. 126–128, 2011.

[17] F. THABTAH, W. HADI, N. ABDELHAMID, dan A. ISSA, "PREDICTION PHASE IN ASSOCIATIVE CLASSIFICATION MINING," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 21, no. 06, hal. 855–876, Sep 2011.

[18] Aditya Yessika Alana, W. H. Wahyu Hidayat, dan Handoyo Djoko W., "Pengaruh Citra Merek, Desain, dan Fitur Produk terhadap Keputusan Pembelian Handphone Nokia (Studi Kasus pada Mahasiswa Universitas Diponegoro)," *Univ. Diponegoro*, 2013.

[19] Tita Tjahyati, "Analisis Perbandingan Metode Certainty Factor dan Naive Bayesian Dalam Mendeteksi Kemungkinan Anak Terkena Disleksia," UNIKOM, 20014.

[20] Hida Nur Firqiani, "Seleksi Fitur Menggunakan Fast Correlation Based Filter Pada Algoritma Voting Feature Intervals 5," Institut Pertanian Bogor, 2007.

[21] Nurlaila, *Manajemen Sumber Daya Manusia I*. LepKhair.

[22] Fitria Sridianti, "Perbedaan Akurasi dan Presisi dalam Pengukuran," *IlmuAlam*, 02-Apr-2016. .

[23] William S Noble, "What is A Support Vector Machines?," *Nat. Biotechnol.*, vol. 24, hal. 1565–1567, 2006.

[24] Steve Gunn, "Support Vector Machines for Classification and Regression," *Tech. Rep. Fac. Eng.*, 1998.

[25] J. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," Apr 1998.

[26] Willy Sutina, I. A. Imelda Atastina, dan A. A. S. Arie Ardiyanti Suryani, "Pengaruh Algoritma Sequential Minimal Optimization Pada Support Vector Machine Untuk Klasifikasi Data (Influence Of Sequential Minimal

Optimization Algorithm On Support Vector Machine For Data Classification)," 2010.

[27] Ian H Witten, Eibe Frank, dan Mark A Hall, *Data Mining Practical Machine Learning Tools and Techniques*, 3 ed. USA: Morgan Kaufmann Publishers, 2011.

[28] Jiawei Han dan M. K. Micheline Kamber, *Data Mining Concepts and Techniques*, 2 ed. San Francisco, United State America, 2007.

[29] Michael Negnevitsky, *A Guide to Intelligent Systems*, 3 ed. .

[30] M. Almulla, H. Yahyaoui, dan K. Al-Matori, "A new fuzzy hybrid technique for ranking real world Web services," *Knowl.-Based Syst.*, vol. 77, hal. 1–15, Mar 2015.

[31] Bowo, "Data Preparation - Pengertian, Alasan dan Langkah-Langkah yang Dilakukan," *Bow's Blog*, Nov-2010. .

[32] Indah Purnama Sari, "Proses Morfologis 1," *Proses Morfologis*, 01-Okt-2013. .

[33] Alfian, "Fiture Extraction," *Berpacu menjadi yang terbaik*, 31-Mar-2013. .

[34] Dea Chintia, "Meaning Application Programming Interface," *All about Operating System*, 20-Okt-2012. .

[35] Bonda Sisephaputra, "Pemanfaatan Filter Dalam Object-Based Opinion Mining Pada Review Produk Pariwisata," Institut Teknologi Sepuluh Nopember Surabaya, 2016.

(Halaman ini sengaja dikosongkan)

LAMPIRAN A

Lampiran ini berisi *source code* dari *web crawler* yang digunakan dalam penelitian ini. Seperti yang sudah diketahui sebelumnya, pada penelitian ini terdapat 2 buah *web crawler* yaitu *Web Crawler I* dan *Web Crawler II*. Di bawah ini adalah rincian dari *source code*-nya.

1. *Web Crawler I*

Susunan dari *Web Crawler I* terbagi menjadi 2 bagian yaitu halaman depan dan *function*. Di bawah ini adalah detail *source code* dari masing-masing bagian yang ada :

1.1 *Source Code* Halaman Depan *Web Crawler I*

Halaman depan *Web Crawler I* berisi kode CSS dan PHP yang membentuk tampilan dari *Web Crawler I* itu sendiri, sehingga pengguna dapat memanipulasi pemilihan fitur maupun mengupload data yang ingin diekstraksi fiturnya. Di bawah ini adalah *source code*-nya :

```
<?php
ini_set('memory_limit', '-1');
if (isset($_FILES['url']) && isset($_POST['feature'])) {
   include 'function.php';
   $file = fopen($_FILES['url']['tmp_name'], "r");
   $max=array();
   $min=array();
   $data_temp=array();
   if ($file) {
      $i=0;
      while (($url = fgets($file)) !== false) {
         $data = new data($url,$i);
         foreach($_POST['feature'] as $feature){
            $data->$feature();
         }
         $temp=$data->extract();
         foreach($temp[2] as $idx=>$val){
               if($val){
                        $max[$idx]=(@$max[$idx]>$temp[0][$idx])?@$max[$idx]:$temp[0][$idx];
                        $min[$idx]=(isset($min[$idx]) &&
$min[$idx]<$temp[0][$idx])?$min[$idx]:$temp[0][$idx];
               }
         }
         $data_temp[$i]=$temp;
         $i++;
      }
      echo "\n".str_pad('',50,'-')."\n";
      echo "@relation phishing\n";
      foreach($data_temp[0][1] as $idx=>$val){
```

```
                echo "\n".$val;
                if(isset($min[$idx])){
                        echo " [min:".$min[$idx]." | max:".$max[$idx]."]";
                }
        }
        echo "\n@attribute status  {'phising','non-phising'}\n\n@data\n";
        foreach($data_temp as $idx=>$val){
          foreach($val[2] as $idx2=>$val2){
                        if($val2){
                        $val[0][$idx2]=floor(($val[0][$idx2] - $min[$idx2]) / ($max[$idx2] - $min[$idx2])*10);
                }
          }
                echo implode(",", $val[0]) . "\n";
        }
        fclose($file);
    }
    die();
}
?>

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <style>
            body{
                padding: 1em;
            }
        </style>
    </head>
    <body>
        <form method="post" enctype="multipart/form-data" target="_blank">
            <div>
                <h3>FEATURE</h3>
                <ul>
                    <li><input name="feature[]" type="checkbox" value="ip" checked> F1 : ip</li>
                    <li><input name="feature[]" type="checkbox" value="simbol_at" checked> F2 : simbol @</li>
                    <li><input name="feature[]" type="checkbox" value="afiks" checked> F3 : jumlah afiks</li>
                    <li><input name="feature[]" type="checkbox" value="usia_domain" checked> F4 : usia
domain</li>
                    <li><input name="feature[]" type="checkbox" value="organization" checked> F5 : organisasi</li>
                    <li><input name="feature[]" type="checkbox" value="privasi" checked> F6 : privasi</li>
                    <li><input name="feature[]" type="checkbox" value="url_long" checked> F7 : panjang url</li>
                    <li><input name="feature[]" type="checkbox" value="https" checked> F8 : https</li>
                    <li><input name="feature[]" type="checkbox" value="alexarank" checked> F9 : alexarank</li>
                    <li><input name="feature[]" type="checkbox" value="js" checked> F10 : javascript</li>
                    <li><input name="feature[]" type="checkbox" value="page_score" checked> F11 : score
halaman</li>
                </ul>
            </div>
            <div>
                <input name="url" type="file" required="">
<pre>
FORMAT LIST WEBSITE:
[class={1:phising,-1:non-phising,}] [url]
</pre>
            </div>
            <div>
                <input type="submit">
                <label><input name="download" type="checkbox" value="1"> Download</label>
            </div>
        </form>
    </body>
</html>
```

1.2 Source Code Function Web Crawler I

Function Web Crawler I berisi kode untuk *grabbing* nilai fitur menggunakan API *(Application Programming Interface)*, normalisasi dan ekstraksi fitur. Di bawah ini adalah *source code*-nya :

```
<?php
require_once "lib/phpQuery.php";
error_reporting(0);
if(isset($_POST['download'])){
   header("Content-Type: application/text");
   header("Content-disposition: attachment; filename=\"dataset_" . time() . ".arff\"");
}else{
   header("Content-Type: text/plain");
}
class data {
   private $url;
   private $no;
   private $class;
   private $attr = array();
   private $real = array();
   private $hasil = array();
   private $sufiks = array('-', '&', '*', '_', '20%', '.aspx', '.php', '.html', '.ga', '.website', '.name', '.in', '.space',
'.asia', '.co', '.my', '.id', '.web', '.or', '.co', '.sch', '.us', '.xyz', '.me', '.ws', '.tv', '.mobi', '.sg', '.bz', '.cd', '.de', '.hk',
'.gen', '.firm', '.jp', '.kr', '.la', '.li', '.mn', '.xxx', '.nz', '.ph', '.pk', '.sg', '.tw', '.uk', '.vn', '.cc', '.biz');
   private $whois = null;
   private $content = null;
   private $array_links = null;

   function __construct($url, $no) {
      $url = explode(" ", $url);
      if ($url[0]==1){
            $this->class = "'phising'";
      } else if ($url[0]==-1){
         $this->class = "'non-phising'";
      } else {
            $this->class = "?";
      }
      unset($url[0]);
      $this->url = trim(implode(" ", $url));
      $this->no = $no;
   }

   function extract() {
      if ($this->no == 0) {
         echo "@relation phishing\n\n";
         echo implode("\n", $this->attr) . "\n";
         echo "@attribute status  {'phising','non-phising'}\n\n@data\n";
      }
      $this->hasil[] = $this->class;
      echo implode(",", $this->hasil) . "\n";
      return array($this->hasil,$this->attr,$this->real);
   }

   function ip() {
      $this->attr[] = '@attribute ip  { 1,-1 }';
      $this->real[] = false;
      $this->hasil[] = (filter_var($this->base_url(), FILTER_VALIDATE_IP) ? "1" : "-1");
   }

   function simbol_at() {
      $this->attr[] = '@attribute simbol_at  { 1,-1 }';
      $this->real[] = false;
      $this->hasil[] = ((strpos($this->url, '@') === FALSE) ? "-1" : "1");
```

```
}

    function afiks() {
        $this->attr[] = '@attribute afiks numeric';
        $url = str_replace($this->sufiks, ';sufiks;', $this->url);
        $jumlah = substr_count($url, ';sufiks;');
        $this->real[] = true;
        $afiks = (($jumlah-0)/(26-0));
        $this->hasil[] = $afiks;
    }

    function usia_domain() {
        if ($this->whois == null) {
            $this->whois();
        }
        if (isset($this->whois['created_on'])) {
            $create = explode(" ", $this->whois['created_on']);
            $date1 = date_create(date('Y-m-d'));
            $date2 = date_create($create[0]);
            $diff = date_diff($date1, $date2);
            $usia = $diff->format("%a");
        } else {
            $usia = 0;
        }
        $this->attr[] = '@attribute usia_domain numeric';
        $this->real[] = true;
        $usia_domain = (($usia-0)/(11772-0));
        $this->hasil[] = $usia_domain;
    }

    function organization() {
        if ($this->whois == null) {
            $this->whois();
        }
        $status = (
                        isset($this->whois['registrant_contact']['organization']) &&
                        !in_array($this->whois['registrant_contact']['organization'],array('N/A','',' '))
                )?-1:1;
        $this->attr[] = '@attribute organization { 1,-1 }';
        $this->real[] = false;
        $this->hasil[] = $status;
    }

    function privasi() {
        if ($this->whois == null) {
            $this->whois();
        }
        $status = (($this->whois['registrant_contact']['name']=='Registration Private') || ($this-
>whois['registrant_contact']['name']=='')?1:-1);
        $this->attr[] = '@attribute privasi { 1,-1 }';
        $this->real[] = false;
        $this->hasil[] = $status;
    }

    function url_long() {
                $url_long=strlen($this->url);
            $this->attr[] = '@attribute url_long numeric';
            $this->real[] = true;
            $panjang_url = (($url_long-14)/(785-14));
        $this->hasil[] = ($panjang_url);
    }

    function https() {
        $this->attr[] = '@attribute url_https  { 1,0,-1 }';
        $this->real[] = false;
        $status=((strpos($this->url, 'https://') === FALSE) ? "1" : "0");
        if($status=='0'){
            $ch = curl_init();
            curl_setopt($ch, CURLOPT_URL, $this->url);
            curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
            curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
            curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, true);
```

```
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, true);
        $content = curl_exec($ch);
        curl_close($ch);
        if($content){
           $status='-1';
        }
      }
      $this->hasil[] = $status;
   }

   function alexarank() {

        $curl = curl_init();
        curl_setopt_array($curl, array(
           CURLOPT_URL => 'http://data.alexa.com/data?cli=10&dat=snbamz&url='.$this->url,
           CURLOPT_RETURNTRANSFER => true,
           CURLOPT_CUSTOMREQUEST => "GET"
        ));
        $response = curl_exec($curl);
        curl_close($curl);
        $xml=simplexml_load_string($response);

      $alexarank = isset($xml->SD[1]->POPULARITY)?$xml->SD[1]->POPULARITY->attributes()-
>TEXT:0;

      $this->attr[] = '@attribute alexarank  { 1,0,-1 }';
      $this->real[] = true;
      $this->hasil[] = ($alexarank == 0 ? "1" : ($alexarank > 10000000 ? "0" : "-1"));;
   }

   function js() {
                        $myKEY = "AIzaSyCaeEua3aAdR3oaBsu4Uv9ACtmXsUbQcZc";
      $url = $this->url;
                        $url_req =
'https://www.googleapis.com/pagespeedonline/v1/runPagespeed?url='.$url.'&screenshot=true&key='.$myK
EY;
                        $result = @file_get_contents($url_req);
                        if ($result == '') {
                                 $ch = curl_init();
                                 $timeout = 60;
                                 curl_setopt($ch, CURLOPT_URL, $url_req);
                                 curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
                                 curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
                                 curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
                                 curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
                                 curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, $timeout);
                                 $result = curl_exec($ch);
                                 curl_close($ch);
                        }
                        $r = json_decode($result, true);
                        $js = @$r['pageStats']['numberJsResources'];
            $this->attr[] = '@attribute js numeric';
               $this->real[] = true;
               $jumlah_js = (($js-0)/(108-0));
      $this->hasil[] = $jumlah_js;

   }

   function page_score() {
                        $myKEY = "AIzaSyCaeEua3aAdR3oaBsu4Uv9ACtmXsUbQcZc";
      $url = $this->url;
                        $url_req =
'https://www.googleapis.com/pagespeedonline/v1/runPagespeed?url='.$url.'&screenshot=true&key='.$myK
EY;
                        $result = @file_get_contents($url_req);
                        if ($result == '') {
                                 $ch = curl_init();
                                 $timeout = 60;
                                 curl_setopt($ch, CURLOPT_URL, $url_req);
                                 curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
                                 curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
                                 curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
```

```
                                    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
                                    curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, $timeout);
                                    $result = curl_exec($ch);
                                    curl_close($ch);
                            }
                            $r = json_decode($result, true);
                            $score = @$r['score'];
                    $this->attr[] = '@attribute page_score numeric';
                        $this->real[] = true;
        $page_score = (($score-0)/(100-0));
        $this->hasil[] = $page_score;

    }

    private function whois() {
        $url = $this->base_url(FALSE);
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL,
"https://www.enclout.com/api/v1/whois/show.json?auth_token=9njdyLasx1jBRogfgLtw&url=$url");
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
        $res = curl_exec($ch);
        curl_close($ch);
        $this->whois = json_decode($res, TRUE);
    }

    private function content() {
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $this->url);
        curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
        $this->content = curl_exec($ch);
        curl_close($ch);
    }

    function is_available($url) {
        $headers=get_headers($url);
        return is_array($headers)?0:1;
    }

    private function base_url($status = true) {
        if ($status) {
            $PARSED_URL = parse_url($this->url);
            return $PARSED_URL['host'];
        } else {
            $url = explode("/", str_replace(array('http://', 'https://', 'www.'), '', $this->url));
            return $url[0];
        }
    }

    private function pure_url($url,$status = true) {
        if ($status) {
            $PARSED_URL = parse_url($url);
            return $PARSED_URL['host'];
        } else {
            $url = explode("/", str_replace(array('http://', 'https://', 'www.'), '', $url));
            return $url[0];
        }
    }
}
?>
```

2. *Web Crawler II*

Sama seperti Web Crawler I, susunan dari *Web Crawler II* juga terbagi menjadi 2 bagian yaitu halaman depan dan *function*. Pada halaman berikut ini adalah detail *source code* dari masing-masing bagian yang ada :

2.1 Source Code Halaman Depan *Web Crawler II*

Halaman depan *Web Crawler II* berisi kode CSS dan PHP yang membentuk tampilan dari *Web Crawler II* itu sendiri, sehingga pengguna dapat memanipulasi pemilihan fitur maupun mengupload data yang ingin diekstraksi fiturnya. Di bawah ini adalah *source code*-nya :

```
<?php
ini_set('memory_limit', '-1');
if (isset($_FILES['url']) && isset($_POST['feature'])) {
   include 'function.php';
   $file = fopen($_FILES['url']['tmp_name'], "r");
   $max=array();
   $min=array();
   $data_temp=array();
   if ($file) {
      $i=0;
      while (($url = fgets($file)) !== false) {
         $data = new data($url,$i);
         foreach($_POST['feature'] as $feature){
            $data->$feature();
         }
         $temp=$data->extract();
         foreach($temp[2] as $idx=>$val){
               if($val){
                        $max[$idx]=(@$max[$idx]>$temp[0][$idx])?@$max[$idx]:$temp[0][$idx];
                        $min[$idx]=(isset($min[$idx]) &&
$min[$idx]<$temp[0][$idx])?$min[$idx]:$temp[0][$idx];
               }
         }
         $data_temp[$i]=$temp;
         $i++;
      }
      echo "\n".str_pad('',50,'-')."\n";
      echo "@relation phishing\n";
      foreach($data_temp[0][1] as $idx=>$val){
               echo "\n".$val;
               if(isset($min[$idx])){
                        echo " [min:".$min[$idx]." | max:".$max[$idx]."]";
               }
      }
      echo "\n@attribute status  {'phising','non-phising'}\n\n@data\n";
      foreach($data_temp as $idx=>$val){
         foreach($val[2] as $idx2=>$val2){
                        if($val2){
                        $val[0][$idx2]=floor(($val[0][$idx2] - $min[$idx2]) / ($max[$idx2] - $min[$idx2])*10);
               }
         }
                  echo implode(",", $val[0]) . "\n";
      }
      fclose($file);
```

```
    }
    die();
}
?>

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <style>
            body{
                padding: 1em;
            }
        </style>
    </head>
    <body>
        <form method="post" enctype="multipart/form-data" target="_blank">
            <div>
                <h3>FEATURE</h3>
                <ul>
                    <li><input name="feature[]" type="checkbox" value="ip" checked> F1 : ip</li>
                    <li><input name="feature[]" type="checkbox" value="simbol_at" checked> F2 : simbol @</li>
                    <li><input name="feature[]" type="checkbox" value="unicode" checked> F3 : unicode</li>
                    <li><input name="feature[]" type="checkbox" value="jumlah_dot" checked> F4 : jumlah dot</li>
                    <li><input name="feature[]" type="checkbox" value="sufiks" checked> F5 : jumlah sufiks</li>
                    <li><input name="feature[]" type="checkbox" value="usia_domain" checked> F6 : usia
domain</li>
                    <li><input name="feature[]" type="checkbox" value="expired_domain" checked> F7 : expired
domain</li>
                    <li><input name="feature[]" type="checkbox" value="dns" checked> F8 : dns</li>
                    <li><input name="feature[]" type="checkbox" value="url_whois" checked> F9 : whois</li>
                    <li><input name="feature[]" type="checkbox" value="organization" checked> F10 :
organization</li>
                    <li><input name="feature[]" type="checkbox" value="privasi" checked> F11 : privasi</li>
                    <li><input name="feature[]" type="checkbox" value="icp" checked> F12 : icp</li>
                    <li><input name="feature[]" type="checkbox" value="deadlinks" checked> F13 : deadlinks</li>
                    <li><input name="feature[]" type="checkbox" value="outbound_links" checked> F14 : outbound
links</li>
                    <li><input name="feature[]" type="checkbox" value="certificate" checked> F15 : certificate</li>
                </ul>
            </div>
            <div>
                <input name="url" type="file" required="">
<pre>
FORMAT LIST WEBSITE:
[class={1:phising,-1:non-phising,}] [url]
</pre>
            </div>
            <div>
                <input type="submit">
                <label><input name="download" type="checkbox" value="1"> Download</label>
            </div>
        </form>
    </body>
</html>
```

2.2 *Source Code Function Web Crawler II*

Function Web Crawler II berisi kode untuk *grabbing* nilai fitur menggunakan API *(Application Programming Interface)*, normalisasi dan ekstraksi fitur. Di bawah ini adalah *source code*-nya :

```
<?php
require_once "lib/phpQuery.php";
error_reporting(0);
if(isset($_POST['download'])){
   header("Content-Type: application/text");
   header("Content-disposition: attachment; filename=\"dataset_" . time() . ".arff\"");
}else{
   header("Content-Type: text/plain");
}

class data {
   private $url;
   private $no;
   private $class;
   private $attr = array();
   private $real = array();
   private $hasil = array();
   private $sufiks = array('.aspx', '.php', '.html', '.ga', '.website', '.name', '.in', '.space', '.asia', '.co', '.my', '.id',
'.web', '.or', '.co', '.sch', '.us', '.xyz', '.me', '.ws', '.tv', '.mobi', '.sg', '.bz', '.cd', '.de', '.hk', '.gen', '.firm', '.jp', '.kr',
'.la', '.li', '.mn', '.xxx', '.nz', '.ph', '.pk', '.sg', '.tw', '.uk', '.vn', '.cc', '.biz');
   private $whois = null;
   private $content = null;
   private $array_links = null;

   function __construct($url, $no) {
      $url = explode(" ", $url);
      if ($url[0]==1){
               $this->class = "'phising'";
      } else if ($url[0]==-1){
         $this->class = "'non-phising'";
      } else {
               $this->class = "?";
      }
      unset($url[0]);
      $this->url = trim(implode(" ", $url));
      $this->no = $no;
   }

   function extract() {
      if ($this->no == 0) {
         echo "@relation phishing\n\n";
         echo implode("\n", $this->attr) . "\n";
         echo "@attribute status  {'phising','non-phising'}\n\n@data\n";
      }
      $this->hasil[] = $this->class;
      echo implode(",", $this->hasil) . "\n";
      return array($this->hasil,$this->attr,$this->real);
   }

   function ip() {
      $this->attr[] = '@attribute ip  { 1,-1 }';
      $this->real[] = false;
      $this->hasil[] = (filter_var($this->base_url(), FILTER_VALIDATE_IP) ? "1" : "-1");
   }

   function simbol_at() {
      $this->attr[] = '@attribute simbol_at  { 1,-1 }';
```

```
    $this->real[] = false;
    $this->hasil[] = ((strpos($this->url, '@') === FALSE) ? "-1" : "1");
}

function unicode() {
    $this->attr[] = '@attribute unicode  { 1,-1 }';
    $this->real[] = false;
    $this->hasil[] = ((strlen($this->url) != strlen(utf8_decode(urldecode($this->url)))) ? "1" : "-1");
}

function jumlah_dot() {
    $this->attr[] = '@attribute jumlah_dot numeric';
    $jumlah = count(explode(".", $this->url))-1;
    $this->real[] = true;
    $jumlah_dot = (($jumlah-1)/(19-1));
    $this->hasil[] = $jumlah_dot;
}

function sufiks() {
    $this->attr[] = '@attribute sufiks numeric';
    $url = str_replace($this->sufiks, ';sufiks;', $this->url);
    $jumlah = substr_count($url, ';sufiks;');
    $this->real[] = true;
    $jumlah_sufiks = (($jumlah-0)/(4-0));
    $this->hasil[] = $jumlah_sufiks;
}

function usia_domain() {
    if ($this->whois == null) {
        $this->whois();
    }
    if (isset($this->whois['created_on'])) {
        $create = explode(" ", $this->whois['created_on']);
        $date1 = date_create(date('Y-m-d'));
        $date2 = date_create($create[0]);
        $diff = date_diff($date1, $date2);
        $usia = $diff->format("%a");
    } else {
        $usia = 0;
    }
    $this->attr[] = '@attribute usia_domain numeric';
    $this->real[] = true;
    $usia_domain = (($usia-0)/(11768-0));
    $this->hasil[] = $usia_domain;
}

function expired_domain() {
    if ($this->whois == null) {
        $this->whois();
    }
    if (isset($this->whois['expires_on'])) {
        $create = explode(" ", $this->whois['expires_on']);
        $date1 = date_create(date('Y-m-d'));
        $date2 = date_create($create[0]);
        $diff = date_diff($date1, $date2);
        $sisa = $diff->format("%a");
    } else {
        $sisa = 0;
    }
    $this->attr[] = '@attribute expired_domain numeric';
    $this->real[] = true;
    $expired_domain = (($sisa-0)/(3539-0));
    $this->hasil[] = $expired_domain;
}

function dns() {
    if ($this->whois == null) {
```

```
        $this->whois();
      }
      $status = 1;
      if (isset($this->whois['name_servers']['hosts'])) {
        $hosts = explode(",", $this->whois['name_servers']['hosts']);
        $dns = dns_get_record($this->base_url(FALSE), DNS_NS);
        foreach ($dns as $d) {
          foreach ($hosts as $h) {
            if (strtolower($d['target']) == strtolower($h)) {
              $status = -1;
              break;
            }
          }
        }
      }
      $this->attr[] = '@attribute dns  { 1,-1 }';
      $this->real[] = false;
      $this->hasil[] = $status;
    }

    function url_whois() {
      if ($this->whois == null) {
        $this->whois();
      }
      $status = -1;
      if (isset($this->whois['error'])) {
        $status = 1;
      }
      $this->attr[] = '@attribute url_whois  { 1,-1 }';
      $this->real[] = false;
      $this->hasil[] = $status;
    }

    function organization() {
      if ($this->whois == null) {
        $this->whois();
      }
      $status = (
                  isset($this->whois['registrant_contact']['organization']) &&
                  !in_array($this->whois['registrant_contact']['organization'],array('N/A','',' '))
          )?1:-1;
      $this->attr[] = '@attribute organization { 1,-1 }';
      $this->real[] = false;
      $this->hasil[] = $status;
    }

    function privasi() {
      if ($this->whois == null) {
        $this->whois();
      }
      $status = (($this->whois['registrant_contact']['name']=='Registration Private') || ($this-
>whois['registrant_contact']['name']=='')?1:-1);
      $this->attr[] = '@attribute privasi { 1,-1 }';
      $this->real[] = false;
      $this->hasil[] = $status;
    }

    function icp() {
      if ($this->whois == null) {
        $this->whois();
      }
      $lokasi = (($this->whois['registrant_contact']['country']=='CN')?1:-1);
      $this->attr[] = '@attribute icp { 1,-1 }';
      $this->real[] = false;
      $this->hasil[] = $lokasi;
    }
```

```
function deadlinks() {
   if ($this->content == null) {
      $this->content();
   }
   $deadlinks = 0;
   if ($this->array_links == null) {
      $links = phpQuery::newDocument($this->content)->find('a');
      $array_links = array();
      foreach ($links as $r) {
         $link = pq($r)->attr('href');
         if (!in_array($link, array('#', '')) && !in_array($link, $array_links)) {
            $array_links[] = $link;
         }
      }
      $this->array_links=$array_links;
   }
   foreach ($this->array_links as $link) {
      $deadlinks+=$this->is_available($link);
   }
   $this->attr[] = '@attribute deadlinks numeric';
   $this->real[] = true;
   $dead = (($deadlinks-0)/(1200-0));
   $this->hasil[] = $dead;
}

function outbound_links() {
   if ($this->content == null) {
      $this->content();
   }
   $outlinks = 0;
   if ($this->array_links == null) {
      $links = phpQuery::newDocument($this->content)->find('a');
      $array_links = array();
      foreach ($links as $r) {
         $link = pq($r)->attr('href');
         if (!in_array($link, array('#', '')) && !in_array($link, $array_links)) {
            $array_links[] = $link;
         }
      }
      $this->array_links=$array_links;
   }
   foreach ($this->array_links as $link) {
      if($this->base_url()!=$this->pure_url($link)){
         $outlinks++;
      }
   }
   $this->attr[] = '@attribute outbound_links numeric';
   $this->real[] = true;
   $out = (($outlinks-0)/(1200-0));
   $this->hasil[] = $out;
}

function certificate() {
   if ($this->content == null) {
      $this->content();
   }
   $cert = 0;
   if ($this->array_links == null) {
      $links = phpQuery::newDocument($this->content)->find('a');
      $array_links = array();
      foreach ($links as $r) {
         $link = pq($r)->attr('href');
         if (!in_array($link, array('#', '')) && !in_array($link, $array_links)) {
            $array_links[] = $link;
         }
```

```
}
        $this->array_links=$array_links;
    }
    foreach ($this->array_links as $link) {
            if(strpos(strtolower($link),'cert')){
        $cert++;
        }
    }
    $this->attr[] = '@attribute certificate  { 1,-1 }';
    $this->hasil[] = ($cert>0? "1" : "-1");
  }

  function form_external_url() {
    if ($this->content == null) {
      $this->content();
    }
    $outform = 0;
    $links = phpQuery::newDocument($this->content)->find('form');
    $array_links = array();
    foreach ($links as $r) {
      $link = pq($r)->attr('actin');
      if (!in_array($link, array('#', '','?'))) {
        $array_links[] = $link;
      }
    }
    foreach ($array_links as $link) {
      if($this->base_url()!=$this->pure_url($link)){
        $outform++;
      }
    }
    $this->attr[] = '@attribute form_external_url real';
    $this->real[] = true;
    $this->hasil[] = $outform;
  }

  private function whois() {
    $url = $this->base_url(FALSE);
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,
"https://www.enclout.com/api/v1/whois/show.json?auth_token=ALFCTzCPftbBrRJFYtJe&url=$url");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    $res = curl_exec($ch);
    curl_close($ch);
    $this->whois = json_decode($res, TRUE);
  }

  private function content() {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $this->url);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    $this->content = curl_exec($ch);
    curl_close($ch);
  }

  function is_available($url) {
    $headers=get_headers($url);
    return is_array($headers)?0:1;
  }

  private function base_url($status = true) {
    if ($status) {
      $PARSED_URL = parse_url($this->url);
```

```
            return $PARSED_URL['host'];
        } else {
            $url = explode("/", str_replace(array('http://', 'https://', 'www.'), '', $this->url));
            return $url[0];
        }
    }

    private function pure_url($url,$status = true) {
        if ($status) {
            $PARSED_URL = parse_url($url);
            return $PARSED_URL['host'];
        } else {
            $url = explode("/", str_replace(array('http://', 'https://', 'www.'), '', $url));
            return $url[0];
        }
    }
}
```

LAMPIRAN B

Lampiran ini berisi screenshot hasil dari uji coba pada penelitian ini. Diharapkan dengan adanya screenshot ini dapat membuktikan bahwa penelitian ini memang dilakukan secara real dan sejujur-jujurnya. Di bawah ini adalah beberapa screenshot dari hasil uji coba yang telah dilakukan :

1. **Screenshot Hasil Uji Coba Algoritma SMO *(Sequential Minimal Optimization)* Pada Penelitian Ini**

   Di bawah ini adalah bukti screenshot dari hasil uji coba algoritma SMO yang telah dilakukan pada penelitian ini :

```
Time taken to build model: 0.28 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         652               95.8824 %
Incorrectly Classified Instances        28                4.1176 %
Kappa statistic                          0.9176
Mean absolute error                      0.0412
Root mean squared error                  0.2029
Relative absolute error                  8.2353 %
Root relative squared error             40.584  %
Total Number of Instances              680

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,974    0,056    0,946      0,974    0,959      0,918    0,959     0,934     phising
                 0,944    0,026    0,973      0,944    0,958      0,918    0,959     0,946     non-phising
Weighted Avg.    0,959    0,041    0,959      0,959    0,959      0,918    0,959     0,940

=== Confusion Matrix ===

   a   b   <-- classified as
 331   9 |   a = phising
  19 321 |   b = non-phising
```

2. Screenshot Hasil Uji Coba Algoritma Naive Bayes Pada Penelitian Ini

Di bawah ini adalah bukti screenshot dari hasil uji coba algoritma Naive Bayes yang telah dilakukan pada penelitian ini :

```
Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         659               96.9118 %
Incorrectly Classified Instances        21                3.0882 %
Kappa statistic                          0.9382
Mean absolute error                      0.0331
Root mean squared error                  0.1489
Relative absolute error                  6.6253 %
Root relative squared error             29.7789 %
Total Number of Instances              680

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,962    0,024    0,976      0,962    0,969      0,938    0,996     0,995     phising
                 0,976    0,038    0,962      0,976    0,969      0,938    0,996     0,996     non-phising
Weighted Avg.    0,969    0,031    0,969      0,969    0,969      0,938    0,996     0,996

=== Confusion Matrix ===

   a   b   <-- classified as
 327  13 |   a = phising
   8 332 |   b = non-phising
```

3. Screenshot Hasil Uji Coba Algoritma Bagging Pada Penelitian Ini

Di bawah ini adalah bukti screenshot dari hasil uji coba algoritma Bagging yang telah dilakukan pada penelitian ini :

```
Time taken to build model: 0.9 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         662               97.3529 %
Incorrectly Classified Instances        18                2.6471 %
Kappa statistic                          0.9471
Mean absolute error                      0.0491
Root mean squared error                  0.1421
Relative absolute error                  9.8249 %
Root relative squared error             28.4196 %
Total Number of Instances              680

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,968    0,021    0,979      0,968    0,973      0,947    0,990     0,993     phising
                 0,979    0,032    0,968      0,979    0,974      0,947    0,990     0,982     non-phising
Weighted Avg.    0,974    0,026    0,974      0,974    0,974      0,947    0,990     0,988

=== Confusion Matrix ===

   a   b   <-- classified as
 329  11 |   a = phising
   7 333 |   b = non-phising
```

4. Screenshot Hasil Uji Coba Algoritma Multilayer Perceptron Pada Penelitian Ini

Di bawah ini adalah bukti screenshot dari hasil uji coba algoritma Multilayer Perceptron yang telah dilakukan pada penelitian ini :

```
Time taken to build model: 5.58 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         659               96.9118 %
Incorrectly Classified Instances        21                3.0882 %
Kappa statistic                          0.9382
Mean absolute error                      0.0323
Root mean squared error                  0.1611
Relative absolute error                  6.4685 %
Root relative squared error             32.2182 %
Total Number of Instances              680

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,968    0,029    0,971      0,968    0,969      0,938    0,990     0,986     phising
                 0,971    0,032    0,968      0,971    0,969      0,938    0,990     0,991     non-phising
Weighted Avg.    0,969    0,031    0,969      0,969    0,969      0,938    0,990     0,988

=== Confusion Matrix ===

   a   b   <-- classified as
 329  11 |   a = phising
  10 330 |   b = non-phising
```

5. Screenshot Hasil Uji Coba Model Klasifikasi Pada Penelitian Dongsong Zhang (2014) Menggunakan Algoritma SMO

Di bawah ini adalah bukti screenshot dari hasil uji coba model klasifikasi pada penelitian Dongsong Zhang (2014) menggunakan algoritma SMO :

```
Time taken to build model: 0.39 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         548               80.5882 %
Incorrectly Classified Instances       132               19.4118 %
Kappa statistic                          0.6118
Mean absolute error                      0.1941
Root mean squared error                  0.4406
Relative absolute error                 38.8235 %
Root relative squared error             88.1176 %
Total Number of Instances              680

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,832    0,221    0,791      0,832    0,811      0,613    0,806     0,742     phising
                 0,779    0,168    0,823      0,779    0,801      0,613    0,806     0,752     non-phising
Weighted Avg.    0,806    0,194    0,807      0,806    0,806      0,613    0,806     0,747

=== Confusion Matrix ===

   a   b   <-- classified as
 283  57 |   a = phising
  75 265 |   b = non-phising
```

6. **Screenshot Hasil Uji Coba Data Baru Pada Model Klasifikasi Di Dalam Penelitian Ini**

   Di bawah ini adalah bukti screenshot dari hasil uji coba menggunakan data baru yang telah dilakukan pada model klasifikasi di dalam penelitian ini :

inst#	actual	predicted	error	prediction
1	1:?	2:non-phising		0.982
2	1:?	2:non-phising		0.982
3	1:?	2:non-phising		0.982
4	1:?	2:non-phising		0.814
5	1:?	2:non-phising		0.982
6	1:?	2:non-phising		0.982
7	1:?	2:non-phising		0.982
8	1:?	2:non-phising		0.982
9	1:?	1:phising		0.873
10	1:?	1:phising		0.973
11	1:?	1:phising		0.823
12	1:?	1:phising		0.996
13	1:?	1:phising		0.527
14	1:?	1:phising		0.998
15	1:?	1:phising		0.989
16	1:?	2:non-phising		0.728
17	1:?	2:non-phising		0.982
18	1:?	2:non-phising		0.982
19	1:?	1:phising		0.896
20	1:?	1:phising		0.896

7. **Screenshot Hasil Uji Coba Data Baru Pada Model Klasifikasi Di Dalam Penelitian Dongsong Zhang (2014)**

   Pada halaman berikut ini adalah bukti screenshot dari hasil uji coba menggunakan data baru yang telah dilakukan pada model klasifikasi di dalam penelitian Dongsong Zhang (2014) :

```
inst#     actual  predicted error prediction
    1        1:? 2:non-phising       1
    2        1:? 2:non-phising       1
    3        1:? 2:non-phising       1
    4        1:? 2:non-phising       1
    5        1:? 2:non-phising       1
    6        1:? 2:non-phising       1
    7        1:?  1:phising       1
    8        1:? 2:non-phising       1
    9        1:? 2:non-phising       1
   10        1:? 2:non-phising       1
   11        1:?  1:phising       1
   12        1:?  1:phising       1
   13        1:?  1:phising       1
   14        1:?  1:phising       1
   15        1:?  1:phising       1
   16        1:? 2:non-phising       1
   17        1:? 2:non-phising       1
   18        1:?  1:phising       1
   19        1:?  1:phising       1
   20        1:?  1:phising       1
```

8. Screenshot Hasil Perankingan Fitur

Di bawah ini adalah bukti screenshot dari hasil perankingan fitur yang telah dilakukan pada penelitian ini :

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 12 status):
        Information Gain Ranking Filter

Ranked attributes:
 0.87204    3 afiks
 0.65667    8 url_https
 0.41695    7 url_long
 0.41085    5 organization
 0.37622    9 alexarank
 0.3585     4 usia_domain
 0.10468   11 page_score
 0.03005    1 ip
 0.02207   10 js
 0.01602    6 privasi
 0.00591    2 simbol_at

Selected attributes: 3,8,7,5,9,4,11,1,10,6,2 : 11
```

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Febry Eka Purwiantono lahir di Surabaya tanggal 23 Februari 1992. Pendidikan formal dari penulis dimulai dari SD YBPK Tempursari Lumajang, kemudian lanjut ke SMP YBPK Tempursari Lumajang, SMKN 1 Gempol Pasuruan, S1 di STIKI Malang dan Pasca Sarjana di ITS Surabaya. Pada saat SMK, penulis mengambil jurusan TKJ (Teknik Komputer dan Jaringan), sedangkan pada saat S1 mengambil jurusan Teknik Informatika dengan kosentrasi mata kuliah E-Business. Pada saat menempuh pendidikan SMK, penulis pernah mengikuti diklat komputer di UNESA dan lomba komputer tingkat Jawa Timur di Ponorogo, sedangkan pada saat S1, penulis pernah menjadi asisten dosen selama 1 tahun di STIKI dan mengikuti lomba INAICTA tingkat Nasional di Jakarta. Di luar akademik, penulis adalah seorang pebisnis online. Penulis memulai karirnya sejak SMK yaitu tahun 2008 dengan modal komputer pinjaman. Walau dengan modal komputer hasil pijaman, penulis berkeyakinan kuat mampu mengembangkan bisnisnya. Alhasil pada tahun 2017 bisnisnya sudah berkembang pesat dan beliau memiliki beberapa aset sekaligus bercita-cita melanjutkan pendidikan ke jenjang S3. Jenis bisnis online yang diikuti oleh penulis antara lain Google Adsense, Freelancer, Fiverr, SEOClerks dan masih banyak yang lainnya yang tidak bisa disebutkan satu per satu.

(Halaman ini sengaja dikosongkan)