ISSN : 2598-294X
E-ISSN : 2598-2958

PENELITIAN KOMPARASI ALGORITMA KLASIFIKASI DALAM MENENTUKAN WEBSITE PALSU

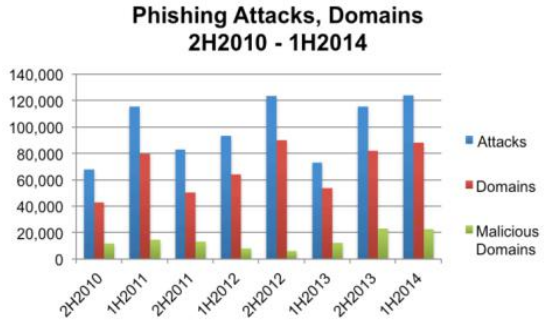
Sunaryono
Sekolah Tinggi manajemen Informatika dan Komputer Widya Utama
aryo.jateng@gmail.com

Abstract — *Website counterfeit or phishing website is a crime in the virtual world whose popularity is increasing even in Indonesia until now in 2015. In this study we take on phishing websites dataset from UCI Repository as many as 2546 data by 30 variables used to determine the website is a phising website or not. Having obtained the data, the authors conducted a study to determine the most appropriate algorithms. Determination of the algorithms with comparisons between algorithms classification techniques. Based on some related research and the advantages of the algorithm, the authors took five algorithms to be tested, the algorithm Decission Tree (C4.5), Naive Bayes, KNN, Support Vector Machine and Neural Network. This study using a test of accuracy and AUC as well as different test parametric T-test. In each model, the authors divide the main data into five sections, and on each of the training data validation was done using K-Fold Cross Validation. The results of this study demonstrate that Neural Network algorithm and SVM into the most appropriate algorithm used by the average value of accuracy is 94 and the value AUC 0.9.*

Keywords — Phising Website, Naive Bayes, KNN, Support Vector Machine, Neural Network

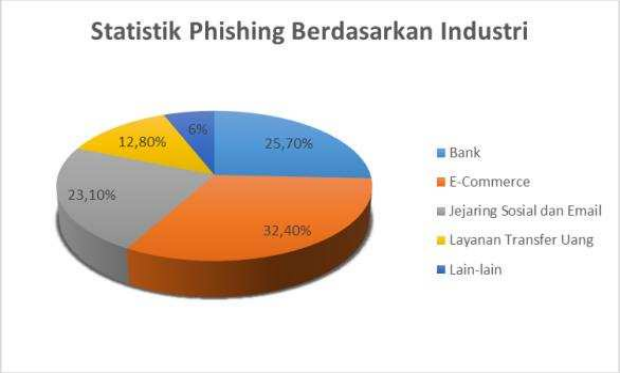
1. PENDAHULUAN

Website palsu atau biasa dikenal dengan *phising website* adalah salah satu kejahatan di dunia maya yang popularitas nya kian meningkat seiring dengan penambahan pengguna internet. Tercatat hingga pada akhir tahun 2014 pengguna internet sudah melebihi 3,079 Triliyun atau 42,4% populasi penduduk (internetworldstats.com) dan pertumbuhan website mencapai 1,010 Triliyun (Global Internet Report, 2014). Berdasarkan data insectpro.com *phishing website* merupakan kejahatan internet dengan prosentasi 22% dari total kejahatan internet, berada diatas *webbase attacks* dan *social engineering*.



Gambar 1 Basic Statistic Phishing Hingga Tahun 2014 (APWG Global Phishing Report 2014)

Disebutkan juga dalam APWG Global Phishing Report 2014 bahwa selain dari total statistik kejahatan phishing beradasarkan *attack, domain* dan *malcious domain* (Gambar 1) selama 20014 yaitu sekitar 234321 jenis serangan yang menempati urutan pertama untuk phishing adalah situs e-*commerce* selanjutnya situs bank.



Gambar 2 Basic Statistic Phishing Hingga Tahun 2014 (APWG Global Phishing Report 2014)

Di Indonesia sendiri beberapa kasus phishing terjadi yang salah satunya pada bank mandiri. Dikutip dari harian kompas *online* bulan April 2015 yang dengan *headline* nya : Kasus "Sinkronisasi Token" yang menimpa nasabah bank lewat layanan *internet banking* kembali muncul. Hal tersebut kali ini terjadi pada seorang nasabah Bank Mandiri asal Semarang, Wahab Yulfikar. dan ketika penulis telurusi banyak berita dari beberapa media tentang penipuan menggunakan *phishing website* pada bank mandiri atau pada situs bank atau *e-commerce* lain.

Penulis menyadari pentingnya *phishing* dan keamanan di dunia internet harus diketahui, maka penulis mengambil salah satu dataset dari UCI

ISSN : 2598-294X
E-ISSN : 2598-2958

Repository yaitu tentang kegiatan *phishing* khususnya untuk *website*. Data tersebut adalah "Phishing Websites Data Set" dataset ini merupakan sumbangan dari Rami Mustafa A Mohammad (University of Huddersfield), Lee McCluskey (University of Huddersfield) dan Fadi Thabtah (Canadian University of Dubai) pada tanggal 26 Maret 2015. Jumlah dataset ini adalah sebanyak 2456 data dengan jumlah variabel nya 30 variabel dalam menentukan website tersebut palsu atau tidak. Jenis data yang di olah adalah data klasifikasi dengan attribut nya berupa integer.

Banyak algortima dalam pengolahan data klasifikasi antara lain *decision tree* (C4.5, ID3, C5.0), SVM, *linear and quadratic discriminant analysis*, *neural network*, k-NN, *random forest*, CART, *naive bayes* dan lain sebagainya. Oleh karena itu perlu adanya penelitian komparasi data mining dalam menentukan website palsu dari *phishing website dataset.* Dalam penelitian ini penulis menggunakan 5 algoritma klasifikasi yaitu :

1) *Decision Tree (C4.5)*
   Alasan penggunaan algoritma ini adalah karena Pohon Keputusan (Decision Tree) merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Aturan ini juga dapat diekspresikan dalam bentuk bahasa basis data seperti SQL untuk mencari record pada kategori tertentu. Pohon keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target. Karena pohon keputusan memadukan antara eksplorasi data dan pemodelan, pohon keputusan ini sangat bagus sebagai langkah awal dalam proses pemodelan bahkan ketika dijadikan sebagai model akhir dari beberapa teknik lain (J R Quinlan, 1993).
2) *Naive Bayes*
   Alasan pemilihan algoritma karena Algoritma ini memanfaatkan teori probabilitas yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya. Dua kelompok peneliti, satu oleh Pantel dan Lin, dan yang lain oleh Microsoft Research memperkenalkan metode statistik bayesian ini pada teknologi anti spam filter. Tetapi yang membuat naive bayesian filtering ini popular adalah pendekatan yang dilakukan oleh Paul Graham. (Stuart J.Russel and Peter Narvig,1995,h426)
3) *K-Nearest Neighbor*
   Alasan pemilihan algoritma ini karena KNN memiliki beberapa kelebihan yaitu bahwa dia tangguh terhadap training data yang noisy dan efektif apabila training data-nya besar. Sedangkan kelemahan dari KNN adalah KNN perlu menentukan nilai dari parameter K (jumlah dari tetangga terdekat), pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap query instance pada keseluruhan training sample (Wakhidah, 2012)
4) *Neural Network*
   Alasan pemilihan algoritma ini karena neural network terletak pada kemampuan non linear mapping yang dimiliki, disamping kemudahannya dalam mendesain solusi bagi problema non linear. Model pemetaan didesain lewat pemakaian data pada training set dalam proses pembelajaran, tanpa harus mengasumsikan distribusi statistik data yang diolah. Selain itu Kemampuan mengakuisisi pengetahuan walaupun dalam keadaan ketidakpastian, mampu menciptakan sendiri representasi melalui kemampuan belajar dan Kemampuan untuk memberikan toleransi atas suatu distorsi dimana gangguan kecil pada data dianggap sebagai gangguan (noise) (Zurada, J.M 1992)
5) *SVM*
   Alasan pemilihan algoritma ini karena Kemampuan generalisasi SVM untuk mengklasifikasikan suatu pattern, yang tidak termasuk data yang dipakai dalam fase pembelajaran metode. Vapnik menjelaskan bahwa generalization error dipengaruhi oleh dua faktor: error terhadap training set, dan satu faktor lagi yang dipengaruhi oleh dimensi VC (Vapnik-Chervokinensis) terlebih Curse of dimensionality sering dialami dalam aplikasi di bidang biomedical engineering, karena biasanya data biologi yang tersedia sangat terbatas, dan penyediaannya memerlukan biaya tinggi. Vapnik membuktikan bahwa tingkat generalisasi yang diperoleh oleh SVM tidak dipengaruhi oleh dimensi dari input vector. Hal ini merupakan alasan mengapa SVM merupakan salah satu metode yang tepat dipakai untuk memecahkan masalah berdimensi tinggi, dalam keterbatasan sampel data yang ada. Terakhir SVM dapat diimplementasikan relative mudah, karena proses penentuan support vector dapat dirumuskan dalam QP problem. Dengan demikian jika kita memiliki library untuk menyelesaikan QP problem, dengan sendirinya SVM dapat diimplementasikan dengan mudah. Selain itu dapat diselesaikan dengan metode sekuensial sebagaimana penjelasan sebelumnya. (Vapnik, 1979)

Tool yang dipergunakan dalam penelitian ini adalah RapidMiner. Rapidminer merupakan perangkat lunak yang bersifat terbuka (*open source*). RapidMiner

ISSN : 2598-294X
E-ISSN : 2598-2958

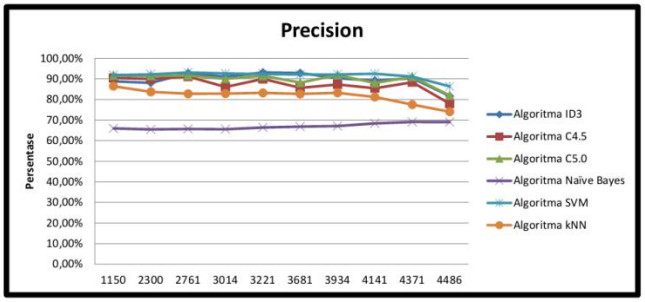
adalah sebuah solusi untuk melakukan analisis terhadap data mining, text mining dan analisis prediksi. RapidMiner menggunakan berbagai teknik deskriptif dan prediksi dalam memberikan wawasan kepada pengguna sehingga dapat membuat keputusan yang paling baik. RapidMiner memiliki kurang lebih 500 operator data mining, termasuk operator untuk input, output, data preprocessing dan visualisasi. RapidMiner merupakan software yang berdiri sendiri untuk analisis data dan sebagai mesin data mining yang dapat diintegrasikan pada produknya sendiri. RapidMiner ditulis dengan munggunakan bahasa java sehingga dapat bekerja di semua sistem operasi.

2. TINJAUAN PUSTAKA

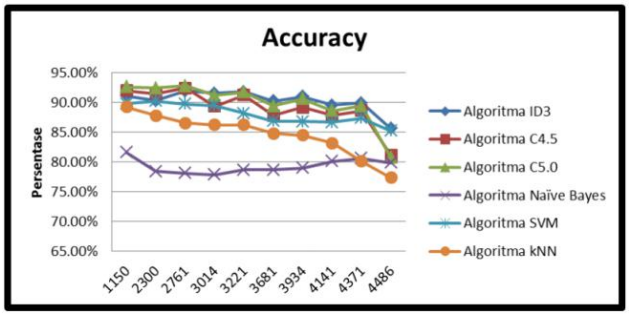
Penelitian tentang komparasi algoritma klasifikasi sudah sangat banyak, salah satunya tentang Kinerja Algoritma Data Mining Decision Tree (ID3, C4.5, C5.0), Naïve Bayes, SVM dan kNN untuk klasifikasi email Spam dan Non-Spam yang dilakukan oleh Wawan dan M. Hendayun (Universitas Langlang Buana Bandung 2013) yang dipublikasikan oleh http://idsirtii.or.id Kementrian Komunikasi dan Informatika Republik Indonesia.

Penelitian tersebut menggunakan database spam mail : UCI Machine Learning Repository http://www.ics.uci.edu/mlearn /MLRepository.html (sumbangan dari George Forman dari laboratorium Hewlett-Packard (HP). Sebanyak 4601 email, dimana 1813 (39,4 %) adalah spam dan 2788 (60.6 %) non spam. Kemudian data dikelompokkan menjadi 10 bagian persentase 25%, 50%, 60%, 65,5%, 70%, 80%, 85,5%, 90%, 95% dan 97,5% dengan jumlah data 1150, 2300, 2761, 3014, 3221, 3682, 3934, 4141, 4371 dan 4486.

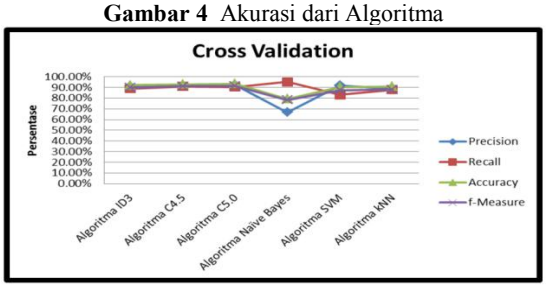
Proses penelitian komparasi tersebut adalah dengan pembentukan Confusion Matrix atau tabel penilaian yang digunakan untuk menghitung Precision, Recall dan Accuracy. Dari hasil Precision dan Recall, diperoleh nilai f-Measure dari setiap Algoritma. Hasil dari penelitian tersebut penulis tampilkan pada gambar 3, gambar 4 dan gambar 5.



Gambar 3 Precision dari Algoritma



Gambar 4 Akurasi dari Algoritma



Gambar 5 Kinerja Algoritma dengan Cross Validation

Kesimpulan dari penelitian tersebut adalah Berdasarkan jumlah data dan cross validation, untuk Decision tree khusus Algoritma ID3 dan C5.0 mempunyai kinerja keseluruhan yang sangat bagus dan lebih dalam melakukan akurasi klasifikasi data, dilihat dari nilai Recall ataupun Accuracy. Algoritma ID3 merupakan algoritma yang terbaik dan akurat dibandingkan kelima algoritma yang lain. Rata – rata keberhasilan algoritma Decision Tree (ID3,C4.5, C5.0), SVM dan kNN dalam melakukan klasifikasi data mencapai akurasi di atas 90%, terkecuali Algoritma Naïve Bayes dibawah 80%.

Penelitian selanjutnya tentang *phishing*. Penelitian ini dilakukan oleh Ningxia Zhang dan YongqingYuan dengan judul "Phishing Detection Using Neural Network" penelitian ini dipublikasikan oleh bagian ilmu komputer dan bagian statistik dari Universitas Stanford pada tahun 2012.

Tujuan dari penelitian ini adalah untuk menerapkan jaringan saraf multilayer feedforward dan merancang set fitur, proses dataset phishing, dan menerapkan jaringan saraf (NN). Peneliti menggunakan cross validation untuk mengevaluasi kinerja NN dengan nilai yang berbeda dari lapisan tersembunyi dan fungsi aktivasi. Peneliti juga membandingkan kinerja NN dengan algoritma utama lainnya terutama SVM. Dari analisis statistik, kami menyimpulkan bahwa NN dengan sesuai jumlah lapisan tersembunyi dapat mencapai akurasi yang memuaskan bahkan ketika contoh pelatihan tidak dikenal. Selain itu, fitur di temukan efektif dalam menangkap karakteristik email phishing.

Dataset Pelatihan *phishing* email yang di dapatkan dari contoh nyata dalam MIME standar Format. Ada sejumlah total 4202 email sah dan 4560 email phishing, dipisahkan dalam 7 folder, 3 folder untuk email sah dan 4 folder untuk phishing email. Setiap file teks berisi email MIME tunggal. Peneliti kemudian

ISSN : 2598-294X
E-ISSN : 2598-2958

melakukan analisis data. Analisis data tersebut juga membandingkan dengan beberapa algoritma. Hasil dari penelitian ini penulis tampilkan pada tabel 1 dan tabel 2.

Tabel 1 Evaluasi Neural Network dengan 2 Fungsi Aktivasi

Activation	*Accu*	W_{accu}	Precision	Recall	F_1
HT	0.9551	0.9494	0.9525	0.9618	0.9571
sigmoid	0.9516	0.9417	0.9450	0.9630	0.9539

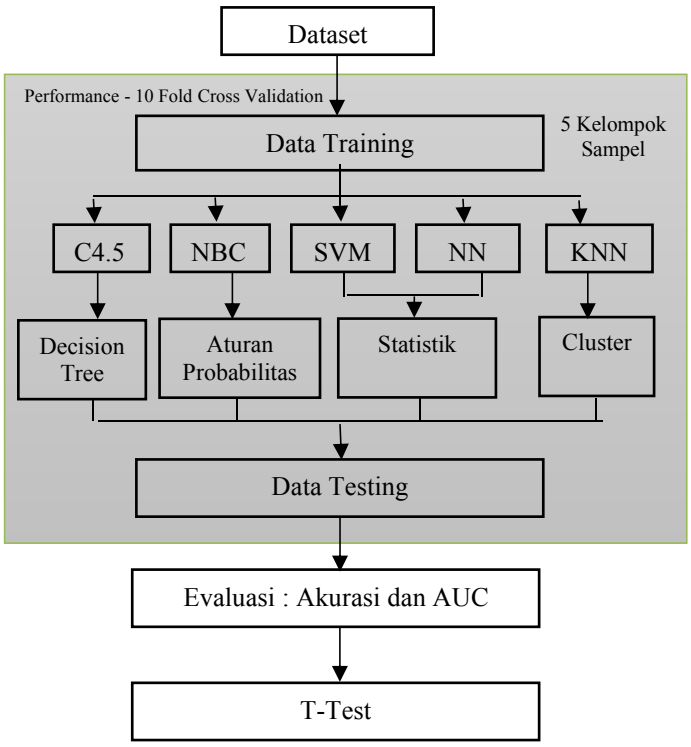
Tabel 2 Evaluasi Neural Network dibandingkan dengan algoritma lain

Method	*Accu*	W_{accu}	Precision	Recall	F_1
DT	0.9658	0.9742	0.9778	0.9561	0.9668
SVM1	0.9218	0.8929	0.9022	0.9555	0.9275
SVM2	0.9579	0.9654	0.9693	0.9491	0.9591
NB	0.9278	0.9370	0.9460	0.9173	0.9367
K-nearest	0.9558	0.9536	0.9585	0.9583	0.9579
NN	0.9551	0.9494	0.9525	0.9618	0.9571

Kesimpulan dari penelitian ini adalah seperti terlihat pada tabel 1 ketika NN dengan dua fungsi aktivasi maka fungsi HT sedikit lebih baik dalam semua penilaian kecuali nilai recall. Hal tersebut menunjukkan bahwafungsi HT lebih baik untuk menghindari salah pengklasifikasian email sah ke email phishing. Sedangkan pada tabel 2 merangkum performace untuk NN dan algoritma pembelajaran lainnya. Seperti ditunjukkan dalam tabel, DT memberikan kinerja terbaik secara keseluruhan. NN memberi recall tertinggi namun nilai masih nilai akurasi > 95% dari presisi, hal tersebut menunjukkan bahwa NN sangat baik dalam mendeteksi phishing email saat salah klasifikasi hanya sebagian kecil dari email yang sah.

3. METODE PENELITIAN

Dalam memulai penelitian penulis melakukan pencarian dan studi dari dataset, kemudian dari dataset tersebut penulis lakukan proses KDD Data Mining, selanjutnya penulis melakukan X-Fold Cross Validation, dalam hal ini penulis membuat 10x percobaan validasi dari hasil data training yaitu menjadi data testing dari masing-masing sampel yaitu 20%, 40%, 60%, 80% dan 100%. Proses validasi tersebut penulis menggunakan 5 algoritma yaitu C4.5, Naive Bayes, KNN, SVM dan NN. Selanjutnya penulis mengambil nilai akurasi dan AUC dari masing-masing algoritma dan membandingkan. Penulis juga menggunakan operator T-Test dalam proses perbandingan algoritma, sehingga diharapkan hasil uji dan penelitian penulis dapat menjadi referensi dalam teknik penentuan website palsu dengan hasil yang akurat dan memuaskan. Berikut Kerangka kerja yang penulis lakukan dalam penelitian ini dapat dilihat pada gambar 6.



Gambar 6 Kerangka Kerja Penelitian

3.1 Dataset

Dataset penelitian adalah data phishing website yang diambil dari UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Phishing+Websites) sejumlah 2456 data dengan 30 variabel dengan keseluruhan nilai dari variabel adalah integer. Struktur data tersebut dapat dilihat pada tabel 3.

Tabel 3 Struktur Dataset Phishing Website

No	Nama Variabel	Tipe Data	Keterangan
1	having_IP_Address	Integer	Attribut
2	URL_Length	Integer	Attribut
3	Shortining_Service	Integer	Attribut
4	having_At_Symbol	Integer	Attribut
5	double_slash_redirecting	Integer	Attribut
6	Prefix_Suffix	Integer	Attribut
7	having_Sub_Domain	Integer	Attribut
8	SSLfinal_State	Integer	Attribut
9	Domain_registeration_length	Integer	Attribut
10	Favicon	Integer	Attribut
11	Port	Integer	Attribut
12	HTTPS_token	Integer	Attribut
13	Request_URL	Integer	Attribut
14	URL_of_Anchor	Integer	Attribut
15	Links_in_tags	Integer	Attribut
16	SFH	Integer	Attribut
17	Submitting_to_email	Integer	Attribut
18	Abnormal_URL	Integer	Attribut
19	Redirect	Integer	Attribut
20	on_mouseover	Integer	Attribut
21	RightClick	Integer	Attribut
22	popUpWidnow	Integer	Attribut
23	Iframe	Integer	Attribut
24	age_of_domain	Integer	Attribut
25	DNSRecord	Integer	Attribut
26	web_traffic	Integer	Attribut
27	Page_Rank	Integer	Attribut
28	Google_Index	Integer	Attribut
29	Links_pointing_to_page	Integer	Attribut
30	Statistical_report	Integer	Attribut
31	Result	Binominal	Label

ISSN : 2598-294X
E-ISSN : 2598-2958

3.2 Metode Validasi

Dalam penelitian digunakan metode validasi 10-fold cross validation untuk training dan testing dataset. Yang berarti dataset dibagi menjadi sepuluh bagian, sembilan bagian digunakan untuk data training dan satu bagian digunakan untuk data testing. Proses validasi dilakukan sebanyak sepuluh kali. Peneliti menggunakan metode validasi 10-fold cross validation karena metode ini menjadi standar metode validasi dari penelitian yang telah dilakukan sebelumnya.

3.3 Metode Evaluasi

Peneliti menentukan nilai accuracy dari confusion matrix dan nilai area under curve (AUC) dari ROC curve sebagai indikator tingkat akurasi performansi dari algoritma klasifikasi. Istilah accuracy sering digunakan dalam konteks metode klasifikasi. Accuracy mengacu pada pengukuran tingkat keakuratan atau prediksi dari suatu model atau metode klasifikasi (Sammut, 2011). Nilai accuracy dalam penelitian ini diperoleh dari tabel confusion matrix RapidMiner. AUC mengukur performansi metode klasifikasi berdasarkan ROC curve. Nilai AUC ditunjukkan dalam skala 0 sampai 1 dimana angka 0 menunjukkan tingkat negatif dan angka 1 menunjukkan tingkat positif (Sammut, 2011).

3.4 Metode Perbandingan

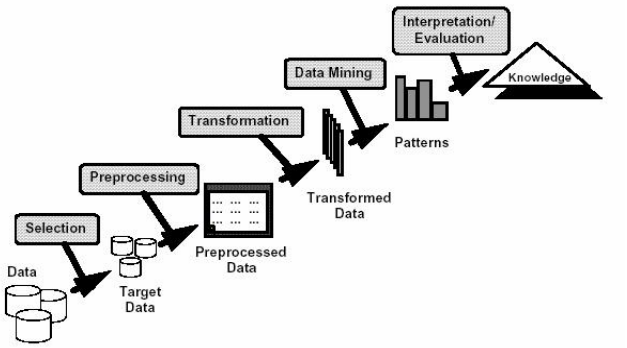
Peneliti menggunakan metode perbandingan uji beda parametrik t-test untuk membandingkan akurasi algoritma klasifikasi. Nilai akurasi yang diperoleh dibandingkan menggunakan t-test untuk memastikan apakah ada perbedaan signifikan pada akurasi algoritma. Jika perbedaan antara dua rata-rata akurasi tidak signifikan, dapat dikatakan bahwa akurasi algoritma tidak dapat dibedakan dan jika perbedaannya signifikan, maka salah satu algoritma memiliki akurasi yang tidak bagus dibandingkan algoritma yang lain (Crc & Hofmann, 2014).

4. TINJAUAN PUSTAKA

4.1 Data Mining

Data mining merupakan proses untuk menemukan pola (*pattern*) dari suatu data. Pola (*pattern*) yang ditemukan harus memiliki arti atau mengandung informasi penting (Witten, Frank, & Hall, 2011).

Istilah *data mining* dan *knowledge discovery in databases* (KDD) sering digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Kedua istilah tersebut memiliki konsep yang berbeda, tetapi saling berkaitan. Salah satu tahapan dalam keseluruhan proses KDD adalah *data mining*. Berikut adalah penjelasan proses KDD secara garis besar:



Gambar 7 Tahapan Data Mining

1. *Data Selection*
   Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional
2. *Pre-processing / Cleanin*
   Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.
3. *Transformation*
   Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.
4. *Data mining*
   Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.
5. *Interpretation / Evaluation*
   Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut dengan *interpretation.* Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

ISSN : 2598-294X
E-ISSN : 2598-2958

4.2 Algoritma Klasifikasi

4.2.1 Algoritma C4.5

Algoritma C 4.5 adalah salah satu metode untuk membuat *decision tree* berdasarkan *training data* yang telah disediakan. Algoritma C 4.5 merupakan pengembangan dari ID3. Beberapa pengembangan yang dilakukan pada C 4.5 adalah bisa mengatasi *missing value*, bisa mengatasi c*ontinue data*, dan *pruning*.

Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Dan mereka juga dapat diekspresikan dalam bentuk bahasa basis data seperti *Structured Query Language* untuk mencari *record* pada kategori tertentu. Pohon keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel *input* dengan sebuah variabel target.

Karena pohon keputusan memadukan antara eksplorasi data dan pemodelan, pohon keputusan sangat bagus sebagai langkah awal dalam proses pemodelan bahkan ketika dijadikan sebagai model akhir dari beberapa teknik lain. Sebuah pohon keputusan adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan keputusan. Dengan masingmasing rangkaian pembagian, anggota himpunan hasil menjadi mirip satu dengan yang lain *(Berry dan Linoff, 2004)*.

Sebuah model pohon keputusan terdiri dari sekumpulan aturan untuk membagi sejumlah populasi yang heterogen menjadi lebih kecil, lebih homogen dengan memperhatikan pada variabel tujuannya. Sebuah pohon keputusan mungkin dibangun dengan seksama secara manual atau dapat tumbuh secara otomatis dengan menerapkan salah satu atau beberapa algoritma pohon keputusan untuk memodelkan himpunan data yang belum terklasifikasi.

Variabel tujuan biasanya dikelompokkan dengan pasti dan model pohon keputusan lebih mengarah pada perhitungan *probability* dari tiap-tiap *record* terhadap kategori-kategori tersebut atau untuk mengklasifikasi *record* dengan mengelompokkannya dalam satu kelas. Pohon keputusan juga dapat digunakan untuk mengestimasi nilai dari variabel *continue* meskipun ada beberapa teknik yang lebih sesuai untuk kasus ini. Banyak algoritma yang dapat dipakai dalam pembentukan pohon keputusan,antara lain ID3, CART, dan C4.5 *(Larose, 2006)*. Berikut ini algoritma dasar dari C4.5:

Input : sampel *training*, label *training*, atribut

1. Membuat simpul akar untuk pohon yang dibuat
2. Jika semua sampel positif, berhenti dengan suatu pohon dengan satu simpul akar, beri tanda (+)
3. Jika semua sampel negatif, berhenti dengan suatu pohon dan satu simpul akar, beri tanda (-)
4. Jika atribut kosong, berhenti dengan suatu pohon dengan suatu simpul akar, dengan label sesuai nilai yang terbanyak yang ada pada label training
5. Untuk yang lain, Mulai
   a. A ----- atribut yang mengklasifikasikan sampel dengan hasil terbaik (berdasarkan *Gain* rasio)
   b. Atribut keputusan untuk simpul akar ---- A
   c. Untuk setiap nilai, *vi*, yang mungkin untuk A
      1) Tambahkan cabang di bawah akar yang berhubungan dengan A= *vi*
      2) Tentukan sampel Svi sebagai subset dari sampel yang mempunyai nilai vi untuk atrribut A
      3) Jika sampel Svi kosong *:* Di bawah cabang tambahkan simpul daun dengan label = nilai yang terbanyak yang ada pada label training dan yang lain tambah cabang baru di bawah cabang yang sekarang C4.5 (sampel *training*, label *training*, atribut-[A])
   d. Berhenti

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut:

1. Pilih atribut sebagai akar
2. Buat cabang untuk masing-masing nilai
3. Bagi kasus dalam cabang
4. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *Gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *Gain* digunakan rumus seperti tertera dalam Rumus 1 *(Craw, 2005)*.

ISSN : 2598-294X
E-ISSN : 2598-2958

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy(Si)$$

Dengan

S : Himpunan Kasus

A : Atribut

N : Jumlah partisi atribut A

|Si| : Jumlah kasus pada partisi ke i

|S| : Jumlah kasus dalam S

Sedangkan perhitungan nilai *Entropy* dapat dilihat pada rumus 2 berikut *(Craw, 2005)*:

$$Entropy(A) = \sum_{i=1}^{n} - pi * \log_2 pi$$

Dengan

S : Himpunan Kasus

A : Fitur

n : Jumlah partisi S

pi : Proporsi dari *Si* terhadap *S*

4.2.2 Naive Bayes

Algoritma *Naïve Bayes* akan mengevaluasi setiap atribut yang mengkontribusi prediksi pada atribut target. *Naïve Bayes* tidak memperhitungkan relasi antar atribut-atribut kontributor prediksi, tidak seperti *Decision Tree* yang memperhitungkan relasi antara atribut. Bentuk tugas dasar yang dilakukan oleh algoritma *Naïve Bayes* adalah hanyalah klasifikasi (*ZhaoHui & MacLennan, 2005, p.132*). Naïve Bayes merupakan teknik *data mining* dengan pendekatan teori probabilitas untuk membangun sebuah model klasifikasi berdasarkan pada kejadian masa lalu yang mempunyai potensi membentuk sebuah objek baru yang dikategorikan sebagai kelas yang memiliki probabilitas terbaik (*Turban et all, 2011, p.220*).

Teorema Bayes memiliki bentuk umum sebagai berikut:

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

dengan

X = Data dengan *class* yang belum diketahui

H = Hipotesis data X merupakan suatu *class* spesifik

P(H|X) = Probabilitas hipotesis H berdasarkan kondisi x (posteriori prob.)

P(H) = Probabilitas hipotesis H (prior prob.)

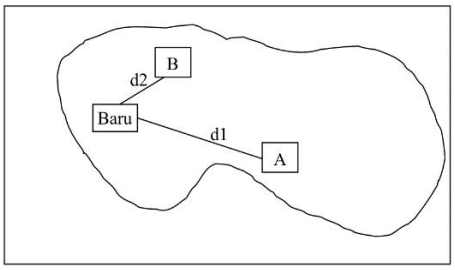
P(X|H) = Probabilitas X berdasarkan kondisi tersebut

P(X) = Probabilitas dari X

Berdasarkan rumus di atas kejadian H merepresentasikan sebuah kelas dan X merepresentasikan sebuah atribut. P(H) disebut *prior probability* H, contoh dalam kasus ini adalah probabilitas kelas yang mendeklarasikan normal. P(X) merupakan prior probability X, contoh untuk probabilitas sebuah atribut protocol_type. P(H|X) adalah *posterior probability* yang merefleksikan probabilitas munculnya kelas normal terhadap data atribut protocol_type. P(X|H) menunjukkan kemungkinan munculnya prediktor X (protocol_type) pada kelas normal. Dan begitu juga seterusnya untuk proses menghitung probabilitas keempat kelas lainnya.

4.2.3 K-Nearest Neighbor

Nearest Neighbor adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. Misalkan diinginkan untuk mencari solusi terhadap seorang pasien baru dengan menggunakan solusi dari pasien terdahulu. Untuk mencari kasus pasien mana yang akan digunakan maka dihitung kedekatan kasus pasien baru dengan semua kasus pasien lama. Kasus pasien lama dengan kedekatan terbesar-lah yang akan diambil solusinya untuk digunakan pada kasus pasien baru.



Gambar 8 Ilustrasi kedekatan kasus

Seperti tampak pada Gambar 8 Ada 2 pasien lama A dan B. Ketika ada pasien Baru, maka solusi yang akan diambil adalah solusi dari pasien terdekat dari pasien Baru. Seandainya d1 adalah kedekatan antara pasien Baru dan pasien A, sedangkan d2 adalah kedekatan antara pasien Baru dengan pasien B. Karena d2 lebih dekat dari d1 maka solusi dari pasien B lah yang akan digunakan untuk memberikan solusi pasien Baru. Adapun rumus untuk melakukan penghitungan kedekatan antara 2 kasus adalah sebagai berikut:

$$similarity(T,S) = \frac{\sum_{i=1}^{n} f(T_i,S_i)\ x\ w_i}{w_i}$$

dengan

T : kasus baru

S : kasus yang ada dalam penyimpanan

n : jumlah atribut dalam masing-masing kasus

ISSN : 2598-294X
E-ISSN : 2598-2958

i : atribut individu antara 1 s/d n
f : fungsi similarity atribut i antara kasus T dan kasus S
w : bobot yang diberikan pada atribut ke i

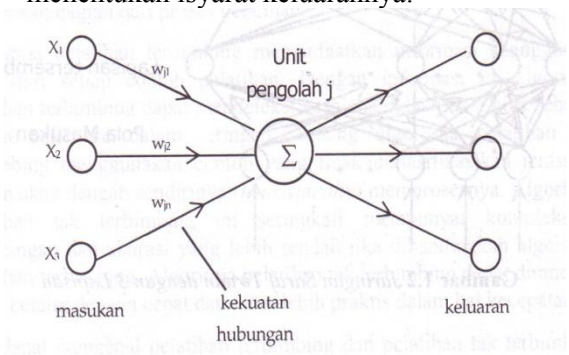
Kedekatan biasanya berada pada nilai antara 0 s/d 1. Nilai 0 artinya kedua kasus mutlak tidak mirip, sebaliknya untuk nilai 1 kasus mirip dengan mutlak.

4.2.4 Neural Network

Jaringan Saraf Tiruan atau Neural Network adalah sistem komputasi di mana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologi di dalam otak (Andri Kristanto. 2004 : 2). Nama jaringan saraf tiruan merupakan terjemahan dari *"Artificial Neural Network"*. Terjemahan yang diambil bukan jaringan saraf buatan seperti dalam menterjemahkan *Artificial Intelligent (AI)*.

Jaringan saraf tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi :

1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut *neuron*.
2. Isyarat mengalir di antara sel saraf (*neuron*) melalui suatu sambungan penghubung.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini akan digunakan untuk menggandakan atau mengalikan isyarat yang dikirim melaluinya.
4. Setiap sel saraf akan menerapkan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.



Gambar 9 Struktur Unit Jaringan Saraf Tiruan

Perhitungan kesalahan merupakan pengukuran bagaimana jaringan dapat belajar dengan baik sehingga jika dibandingkan dengan pola yang baru akan dengan mudah dikenali. Kesalahan pada keluaran jaringan merupakan selisih antara keluaran sebenarnya (*current output*) dan keluaran yang diinginkan (*desired output*). Selisih yang dihasilkan antara keduanya biasanya ditentukan dengan cara dihitung menggunakan suatu persamaan.

Sum Square Error (SSE) dihitung sebagai berikut :

1) Hitung keluaran jaringan saraf untuk masukan pertama.
2) Hitung selisih antara nilai keluaran jaringan saraf dan nilai target atau yang diinginkan untuk setiap keluaran.
3) Kuadratkan setiap keluaran kemudian hitung seluruhnya. Ini merupakan kuadrat kesalahan untuk contoh latihan.

Adapun rumusnya adalah :

$$SSE = \sum_{p} \sum_{j} \left(T_{jp} - X_{jp}\right)^2$$

dengan

T_{jp} : nilai keluaran jaringan saraf
X_{jp} : nilai target atau yang diinginkan untuk setiap keluaran.

Root Mean Square Error (RMS Error) :

1) Hitung SSE.
2) Hasilnya dibagi dengan perkalian antara banyaknya data pada pelatihan dan banyaknya keluaran, kemudian diakarkan.

Rumusnya adalah :

$$RMS\ Error = \sqrt{\frac{\sum_{p} \sum_{j} \left(T_{jp} - X_{jp}\right)^2}{n_p\, n_o}}$$

dengan

n_p : jumlah seluruh pola
p_o : jumlah keluaran.

4.2.5 Support Vector Machine (SVM)

Menurut Santoso (2007) Support vector machine (SVM) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM berada dalam satu kelas dengan Artificial Neural Network (ANN) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *supervised learning*.

Teori SVM dimulai dengan kasus klasifikasi yang secara linier bisa dipisahkan. Dalam hal ini fungsi pemisah yang dicari adalah fungsi linier. Fungsi ini bisa didefinisikan sebagai Persamaan Hyperplane (garis) :

G: $w_1x_1 + w_2x_2 + b = 0$

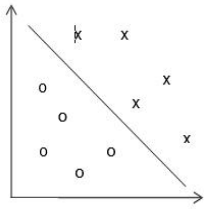
Sebagai contoh pada dataset iris ada label santosa dan versi color maka, agar G memisahkan kelas Sentosa dan Versicolor dapat dipilih w1,w2 dan b sehingga :

$w_1x_1^{(i)} + w_2x_2^{(i)} + b > 0$ utk x1(i),y1(i) Anggota Bilangan Setosa

$w_1x_1^{(i)} + w_2x_2^{(i)} + b < 0$ utk x1(i),y1(i) Anggota Bilangan Semicolor

Di sini, w merupakan vektor di ruang dimensi *n*, sehingga w_i ($w1$ sampai w_n) merupakan komponen-komponen nilai penyusun vektor *w* di ruang dimensi *n*, dan nilai dari vektor *w* di sinilah yang akan menjadi *classifier* bagi *support vector*

machines.



Gambar 10 Proses Support Vector Machine memisahkan menjadi 2 bagian

4.3 Fold Cross Validation

Cross validation merupakan salah satu teknik untuk menilai / validasi keakuratan sebuah model berdasarkan dataset tertentu. Dalam pengujian menggunakan K-fold cross validation disebut data traininig, sedangkan data yang digunakan untuk validasi model disebut data test. Data set dibagi menjadi sejumlah K-buah partisi secara acak. Kemudian dilakukan sejumlah K-kali eksperimen, dimana setiap eksmerimen menggunakan data partisi ke-K sebagai data testing dan memanfaatkan sisa partisi lainya sebagai data training.

Tabel 4 Stratified 10-Fold Cross Validation

Test	Dataset									
1	■									
2		■								
3			■							
4				■						
5					■					
6						■				
7							■			
8								■		
9									■	
10										■

4.4 Uji Beda Parametrik

Uji beda parametrik yaitu metode pengujian yang mempertimbangkan jenis sebaran atau distribusi data, yaitu apakah data menyebar secara normal atau tidak. Dengan kata lain, data yang akan dianalisis menggunakan uji parametrik harus memenuhi asumsi normalitas. Pada umumnya, jika data tidak menyebar normal, maka data seharusnya dikerjakan dengan metode non parametrik, atau setidak-tidaknya dilakukan transformasi terlebih dahulu agar data mengikuti sebaran normal, sehingga bisa dikerjakan dengan statistik parametrik. Uji parametrik membutuhkan input dari user, seperti jumlah interval jumlah maksimum (Maimon, O., & Rokach, L., 2010). Ciri-ciri uji parametrik:

a. Data dengan skala interval dan rasio,
b. Data menyebar atau berdistribusi normal.

Keunggulan uji parametrik:

a. Syarat-syarat parameter dari suatu populasi yang menjadi sampel biasanya tidak diuji dan dianggap memenuhi syarat
b. Pengukuran terhadap data dilakukan dengan kuat,

Kelemahan uji parametrik:

a. Variabel-variabel yang diteliti harus diukur setidaknya dalam skala interval.
b. Dalam analisis varian ditambahkan persyaratan rata-rata populasi harus normal dan bervarian sama dan harus merupakan kombinasi linear dari efek-efek yang ditimbulkan.

Dalam pengklasifikasian keakuratan dari tes diagnostik menggunakan Area Under Curve (AUC), sebuah sistem nilai yang disajikan (Gorunescu, 2011).

Tabel 5 Keterangan Nilai AUC

AUC	Keterangan
0.90 - 1.00	Exellent classification
0.80 - 0.90	Good classification
0.70 - 0.80	Fair classification
0.60 - 0.70	Poor classification
< 0.60	Failure

4.5 T-test

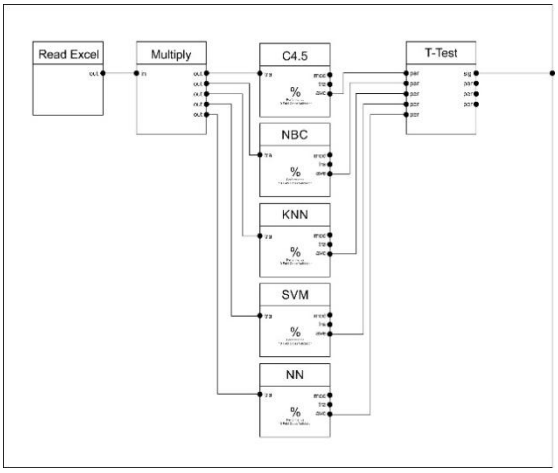
T-Test adalah salah satu bentuk metode yang digunakan untuk menguji kebenaran atau kepalsuan hipotesis yang menyatakan bahwa diantara kedua buah mean sampel yang diambil secara random dari popolusai yang sama dan tidak ada perbedaan signifikan.

Syarat untuk melakukan T-test ada 2 :

1. Sampel yang diambil secara acak dari populasi yang sama.
2. Data Skala interval atau rasio

4. HASIL DAN PEMBAHASAN

Dalam penelitian ini penulis menggunakan Processor Intel Core i5 2.53 GHz CPU, 3 GB RAM, dan sitem operasinya Microsoft Windows 8 Entrerprise 32-bit. Aplikasi yang digunakan yaitu RapidMiner 6.0. Proses penelitian dapat dilihat pada gambar 10.



Gambar 11 Skema Proses Penelitian

Selanjutnya dalam proses penelitian penulis membagi data pelatihan (Dataset) menjadi 5 bagian

ISSN : 2598-294X
E-ISSN : 2598-2958

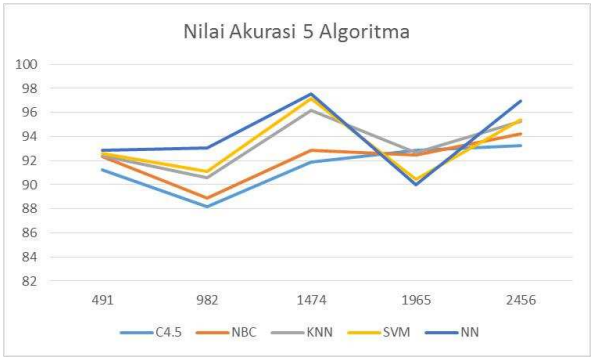
data training dan testing, yaitu : 20%, 40%, 60%, 80% dan 100% data. Atau sejumlah 491, 982, 1474, 1965 dan 2456.

4.1 Hasil Akurasi Algoritma

Hasil akurasi algoritma setelah dilakukan pengujian dapat dilihat pada tabel 6 dan grafiknya pada gambar 12.

Tabel 6 Nilai Akurasi Pengujian Algoritma

No	Jumlah Data	C4.5	NBC	KNN	SVM	NN
1	491	91.24	88.18	91.86	92.87	93.28
2	982	92.36	88.90	92.88	92.46	94.20
3	1474	92.40	90.57	96.16	92.67	95.25
4	1965	92.57	91.09	97.16	90.45	95.37
5	2456	92.88	93.05	97.54	89.97	96.98
6	Rata-rata	91.48	92.16	93.41	93.32	94.08



Gambar 12 Grafik Nilai Akurasi 5 Algoritma

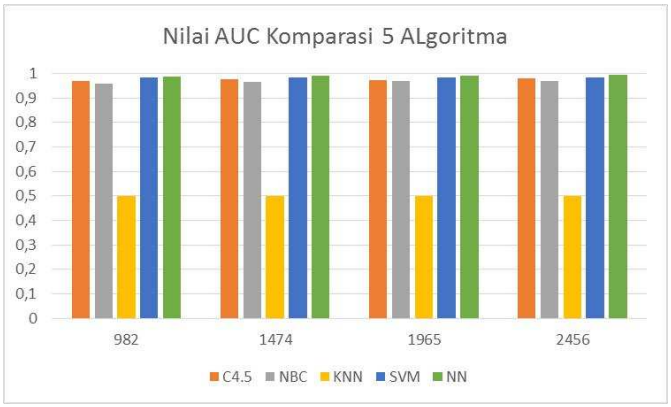
Jika dilihat baik pada tabel 6 atau pada gambar 12 maka nilai akurasi akan semakin tinggi apabila jumlah data semakin lengkap yaitu 100%, nilai tertinggi diperoleh oleh algoritma Neural Network dengan nilai 97,54 dan nilai rata-rata tertinggi yaitu 94,08 juga diperoleh oleh algoritma yang sama.

4.2 Hasil Nilai AUC

Hasil nilai AUC algoritma setelah dilakukan pengujian dapat dilihat pada tabel 7 dan grafiknya pada gambar 12.

Tabel 7 Nilai AUC Pengujian 5 Algoritma

No	Jumlah Data	C4.5	NBC	KNN	SVM	NN
1	491	0.932	0.956	0.500	0.982	0.982
2	982	0.968	0.957	0.500	0.983	0.986
3	1474	0.976	0.967	0.500	0.985	0.990
4	1965	0.972	0.970	0.500	0.985	0.990
5	2456	0.979	0.970	0.500	0.985	0.994
6	Rata-rata	0.965	0.964	0.500	0.984	0.988



Gambar 13 Grafik Nilai AUC 5 Algoritma

Jika dilihat baik pada tabel 7 atau pada gambar 13 maka nilai AUC juga akan semakin tinggi apabila jumlah data semakin lengkap yaitu 100%, nilai tertinggi diperoleh oleh algoritma Neural Network dengan nilai 0,994 yang artinya *excellent classification* diperoleh oleh algoritma Neural Network. Nilai *failure* atau gagal dilakukan oleh algoritma KNN, dimana dari mulai pengujian 20% s/d 100% menunjukkan nilai yang sama yaitu 0.500 yang artinya *failure*. Untuk nilai rata-rata tertinggi yaitu 94,08 juga diperoleh oleh algoritma yang sama.

4.3 Hasil T-Test Algoritma

Pada penelitian ketiga penulis menggunakan T-Test untuk membandingkan 5 algoritma. Pengujian T-Test ini akan menguji agar mendapatkan nilai yang terbaik, dimana dalam pengujian tersebut sampai mendapatkan nilai terkecil ≤ 0,05 dinyatakan sebagai hasil uji yang terbaik (Santoso. S, 2010). Hasil proses T-Test dapat dilihat pada tabel 8.

Tabel 8 T-Test 5 Algoritma

	C4.5	**NBC**	**KNN**	**SVM**	**NN**
C4.5 0.917 +/- 0.036		0.118	0.920	0.449	0.752
NBC 0.882 +/- 0.051			0.229	**0.040**	**0.041**
KNN 0.915 +/- 0.052				0.470	0.731
SVM 0.931 +/- 0.044					0.516
NN 0.921 +/- 0.017					

Nilai yang dicetak tebal berarti lebih kecil dari alpha = 0.050 yang mengindikasikan adanya perbedaan signifikan diantara nilai rata-rata aktual. Dari Tabel 8 dapat ditarik kesimpulan, algoritma Neural Network dan SVM memiliki akurasi paling bagus (dominan) terhadap algoritma yang lain. Berikutnya ada algoritma

ISSN : 2598-294X
E-ISSN : 2598-2958

Naive Bayes, C4.5 dan KNN tidak ada perbedaan signifikan diantara algoritma tersebut.

5. KESIMPULAN

Penelitian dengan membandingkan lima algoritma klasifikasi *decision tree* (C4.5), *naive bayes*, k-*nearest neighbor*, *support vector machine*, dan *neural network* untuk memprediksi website palsu, menggunakan metode validasi k-*fold cross validation* dan sampel (*stratifield*), serta dilakukan uji beda terhadap akurasi masing-masing algoritma dengan uji beda parametrik *t-test* menunjukkan bahwa algoritma Neural Network dan Support Vector Machine memiliki akurasi tertinggi, dan yang menjadi pilihan utama adalah Algoritma Neural Network. Penulis mendukung pemilihan algoritma Neural Network yang digunakan dalam penelitian yang dilakukan oleh Ningxia Zhang dan YongqingYuan dengan judul "Phishing Detection Using Neural Network". Sedangkan penulis kurang merekomendasikan algoritma KNN untuk digunakan dalam penentuan website palsu meskipun akurasinya cukup tinggi tetapi nilai AUC nya hanya 0,50 yang artinya algoritma tersebut gagal dalam pengambilan nilai AUC. Selanjutnya untuk algoritma decission tree (C4.5) dan Naive Bayes merupakan algoritma dengan tingkat akurasi yang lebih rendah dibandingkan dengan algoritma SVM dan Neural Network namun nilai AUC nya berada pada *Excellent Classification*.

DAFTAR PUSTAKA

[1] Breiman, L.1996. Bagging predictors. *Machine Learning*, 24(2): 123–140.

[2] Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. 2011. Comparing Boosting and Bagging Techniques With Noisy and Imbalanced Data. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 41(3): 552–568.

[3] Ko, Y.D., Moon, P., Kim, C. E., Ham, M.H., Myoung, J.M., & Yun, I. 2009. Modeling and Optimization of the Growth Rate for ZnO thin Films using neural Networks and genetic Algorithms. *Expert Systems with Applications*, 36(2): 4061–4066.

[4] Lee, J., & Kang, S. 2007. GA Based Meta-Modeling of BPN Architecture for Constrained Approximate Optimization. *International Journal of Solids and Structures*, 44(18-19): 5980–5993.

[5] Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. 2008. Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings. *IEEE Transactions on Software Engineering*, 34(4): 485–496.

[6] Lin, S.W., Chen, S.C., Wu, W.J., & Chen, C.H. 2009. Parameter Determination and Feature Selection for Back-Propagation Network by particle Swarm Optimization. *Knowledge and Information Systems*, 21(2): 249–266.

[7] Menzies, T., Greenwald, J., & Frank, A. 2007. Data Mining Static Code Attributes to Learn Defect Predictors. *IEEE Transactions on Software Engineering*, 33(1): 2–13.

[8] Menzies, T., Milton, Z., Turhan, B., Cukic, B., Jiang, Y., & Bener, A. 2010. Defect Prediction from Static Code Features: Current Results, Limitations, New Approaches. *Automated Software Engineering*, 17(4): 375–407.

[9] Seiffert, C., Khoshgoftaar, T. M., & Van Hulse, J. 2009. Improving Software-Quality Predictions With Data Sampling and Boosting. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 39(6): 1283–1294.

[10] Shull, F., Basili, V., Boehm, B., Brown, A. W., Costa, P., Lindvall, M.,Zelkowitz, M. 2002. What We Have Learned about Fighting Defects. In *Proceedings Eighth IEEE Symposium on Software Metrics 2002,* 249–258.

[11] Tony Hou, T.H., Su, C.H., & Chang, H.Z. 2008. Using Neural Networks and Immune Algorithms to Find the Optimal Parameters for an IC Wire Bonding Process. *Expert Systems with Applications*, 34(1): 427–436.

[12] Wahono, R. S., & Herman, N. S. 2014. Genetic Feature Selection for Software Defect Prediction. *Advanced Science Letters*, 20(1): 239–244.

[13] Wahono, R. S., & Suryana, N. 2013. Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for Software Defect Prediction. *International Journal of Software Engineering and Its Applications*, 7(5): 153–166.

[14] Wang, S., & Yao, X. 2013. Using Class Imbalance Learning for Software Defect Prediction. *IEEE Transactions on Reliability*, 62(2): 434–443.

[15] Wang, T.Y., & Huang, C.Y. 2007. Applying optimized BPN to A Chaotic Time Series Problem. *Expert Systems with Applications*, 32(1): 193–200.

[16] Witten, I. H., Frank, E., & Hall, M. A. 2011. *Data Mining Third Edition*. Elsevier Inc.

[17] Yusta, S. C. 2009. Different Metaheuristic Strategies to Solve the Feature Selection Problem. *Pattern Recognition Letters*, 30(5): 525–534.

[18] Zheng, J. 2010. Cost-Sensitive Boosting Neural Networks for Software Defect Prediction. *Expert Systems with Applications*, 37(6): 4537–4543.

BIOGRAFI PENULIS

Sunaryono. Mendapatkan gelar S1 dan S2 di bidang TI. Selain berprofesi sebagai Dosen di STMIK Widya Utama juga merupakan pendiri Dieng Cyber yaitu salah satu jasa layanan IT dan pembuatan perangkat lunak. Memiliki minat pada pengembangan website dan perangkat lunak bergerak (Android). Saat ini sedang melakukan riset tentang data mining dan kecerdasan buatan.

ISSN : 2598-294X
E-ISSN : 2598-2958