BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang teori-teori yang digunakan dalam penelitian, temuan dan bahan penelitian terdahulu yang diperoleh dari beberapa referensi yang menunjang penelitian dalam penulisan skripsi. Serta teori-teori mengenai keamanan jaringan, IDS, *Naïve Bayes*, *SVM, SVM Linear, SVM Polynomial, SVM Sigmoid* dan Dataset ISCX

2.1 Tinjauan Pustaka

Tinjauan pustaka akan membahas mengenai penelitian yang pernah dilakukan sebelumnya yang terkait dan relevan dengan penelitian yang akan dilakukan oleh penulis.

Penelitian pertama adalah penelitian yang dilakukan oleh (Srinivas Mukkamala dan Andrew H. Sung , 2003) yang berjudul "*Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines"* dalam penelitian ini membahas tentang seleksi fitur yang digunakan untuk IDS dengan menggunakan metode ANN dan SVM agar performa yang dihasilkan pada IDS mencapai performa maksimal. Data yang digunakan dalam penelitian ini adalah dataset DARPA 1998. Menurutnya kinerja dari algoritma SVM lebih baik jika dibandingkan dengan ANN dalam hal solusi yang dicapai untuk kasus pengklasifikasian IDS.

Penelitian kedua adalah penelitian yang dilakukan oleh (Mrutyunjaya Panda dan Mana R. Patra , 2007) yang berjudul *"Network Intrusión Detection Using Naive Bayes"* penelitian ini menerapkan metode *Naive Bayes* pada deteksi intrusi berbasis anomali. Data yang digunakan pada penelitian ini yaitu dataset KDD Cup'99. Menurutnya kinerja algoritma *Naive Bayes* lebih efisien dalam melakukan klasifikasi *Network IDS (NIDS)* dibandingkan ANN. Metode klasifikasi *Naïve Bayes* atau *Naïve Bayes Classifier* (NBC) dikenal sebagai teknik yang paling baik dalam hal waktu komputasi dibandingkan teknik algoritma data mining lainnya.

Penelitian ketiga adalah penelitian yang dilakukan oleh (Dwi Widiastuti, 2012) yang berjudul "Analisa Perbandingan Algoritma SVM, *Naive Bayes* dan *Decision Tree* Dalam Mengklasifikasikan Serangan *(Attacks)* Pada Sistem Pendeteksi Intrusi" pada penelitian ini dilakukan perbandingan antara algoritma SVM, *Naive Bayes* dan *Decision Tree* dalam melakukan klasifikasi serangan pada sistem deteksi intrusi dimana data yang digunakan yaitu KDD Cup'99. Penelitian ini menyatakan kinerja algoritma *decision tree* lebih baik dibandingkan dengan algoritma SVM dan NBC. Salah satu faktornya yaitu kemampuannya yang secara sederhana *(simple)* mendefinisikan dan mengklasifikasikan masing-masing atribut ke setiap kelas. Namun untuk membangun sebuah model algoritma yang tercepat yaitu waktu komputasi *(running time),* yang terbaik adalah *naïve bayes*. Untuk membangun atau menambahkan teknik yang baik pada sistem IDS maka teknik algoritma yang dianjurkan adalah *Decision Tree*.

Tabel 2.1 Menjelaskan tentang kajian pustaka yang digunakan dalam penelitian.

NO	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Tugas Akhir Penulis
1	*Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines*	Srinivas Mukkamala dan Andrew H. Sung , 2003	Membahas tentang seleksi fitur yang digunakan untuk IDS dengan menggunakan metode ANN dan SVM agar IDS mencapai performa maksimal, data yang digunakan dataset DARPA 1998.	Menganalisa deteksi sistem dengan melakukan klasifikasi pada dataset menggunakan metode *Naive Bayes* dan SVM dengan tujuan mencari metode terbaik. Data yang digunakan yaitu ISCX 2012
2	*Network Intrusión Detection Using Naive Bayes*	Mrutyunjaya Panda dan Mana R. Patra , 2007	Menerapkan metode *Naive Bayes* pada deteksi intrusi berbasis anomali dimana data yang digunakan yaitu dataset KDD Cup'99	Menggunakan metode *Naive Bayes* dan SVM dalam klasifikasi serangan dimana dalam tahapannya menerapkan metode *behavior based*. Data yang digunakan ISCX 2012
3	Analisa Perbandingan Algoritma SVM, *Naive Bayes* dan *Decision Tree* Dalam Mengklasifikasikan Serangan (*Attacks*) Pada Sistem Pendeteksi Intrusi	Dwi Widiastuti, 2012	Melakukan perbandingan algoritma SVM, *Naive Bayes* dan *Decision Tree* dalam melakukan klasifikasi serangan pada sistem deteksi intrusi dimana data yang	Membandingkan algoritma *Naive Bayes*, SVM *Linear*, SVM *Polynomial*, dan SVM *Sigmoid* untuk mencari metode terbaik dalam proses klasifikasi. Dataset yang

			digunakan yaitu KDD Cup'99	digunakan ISCX 2012

Tabel 2.1 Kajian Pustaka

2.2 Dasar Teori

2.2.1 Keamanan Jaringan

Keamanan jaringan dapat dikatakan jika sebuah komputer yang terhubung dengan beberapa jaringan lain yang lebih banyak dapat memberikan ancaman keamanan daripada sebuah komputer yang sama sekali tidak terhubung pada sebuah jaringan (Ri2M, 2010). Dengan adanya sebuah pengendalian dan pendeteksian maka resiko keamanan jaringan yang tidak diinginkan dapat dilakukan pencegahan. Para pengguna jaringan komputer juga berharap dengan adanya keamanan jaringan saat ini, apabila dalam pengiriman pesan, maka dapat sampai ke tujuan yang tepat tanpa mengalami perubahan atau kekurangan saat diterima oleh penerima pesan. Sebuah jaringan apabila proses mengaksesnya mudah, maka keamanan jaringannya dapat dikatakan sedikit kurang aman dan rawan akan adanya sebuah serangan. Sementara itu jika sebuah jaringan proses mengaksesnya tidak mudah, maka kemanan jaringan yang diterapkan semakin baik. Beberapa faktor yang ada pada jaringan komputer mengakibatkan beberapa resiko dalam sebuah keamanan jaringan. Misalnya saja resiko yang biasanya terdapat pada jaringan komputer yaitu segala bentuk serangan baik dari segi fisik ataupun logic. Dapat berupa serangan langsung maupun tidak langsung yang dapat mengganggu segala aktivitas yang ada pada sebuah jaringan. Beberapa faktor yang mempengaruhi diantaranya:

- Kelemahan pada pengguna komputer
- Kelemahan pada *computer hardware*
- Kelemahan pada sistem operasi jaringan
- Kelemahan pada sistem jaringan komputer

2.2.2 Intrusion Detection System

Intrusion Detection System (IDS) merupakan sebuah perangkat keras maupun perangkat lunak yang mampu melakukan suatu deteksi pada sebuah aktivitas mencurigakan yang terjadi pada suatu jaringan komputer maupun sistem komputer yang mencurigakan dalam sebuah sistem atau jaringan (Monika Kusumawati, 2010).

Menurut (Wu, 2009*) Intrusion Detection System* (IDS) merupakan tindakan untuk melakukan sebuah deteksi dari beberapa *traffic* paket yang dalam prosesnya tidak sesuai atau tidak diharapkan terjadi pada sebuah jaringan.

Intrusion Detection System (IDS) dapat diimplementasikan melalui sebuah perangkat lunak yang sudah dilakukan instalasi pada *device*. IDS dalam proses kerjanya dapat memantau maupun melakukan deteksi dari beberapa paket yang mencurigakan yang dapat menyerang kemanan pada sebuah jaringan.

Pada dasarnya sebuah *Intrusion Detection System* (IDS) merupakan sistem yang dapat menganalisa serta memiliki kemampuan untuk melakukan deteksi serangan secara realtime, juga dapat mencatat *log*, maupun menghentikan sebuah serangan. IDS dapat dikatakan sebagai *security tools* dimana memiliki fungsi dalam menghadapi aktivitas mencurigakan yang dilakukan oleh *hackers*. Beberapa komponen yang terdapat pada IDS diantaranya:

a. S*ecurity events* yang dapat dikenali oleh sensor
b. Pengontrol sensor maupun pemonitor *events* atau *alerts* dilakukan oleh *Console*
c. Sensor melakukan penyimpanan *events logged* pada sebuah basis data dengan menggunakan *rules* kemanan yang dilakukan di *Central Engine*

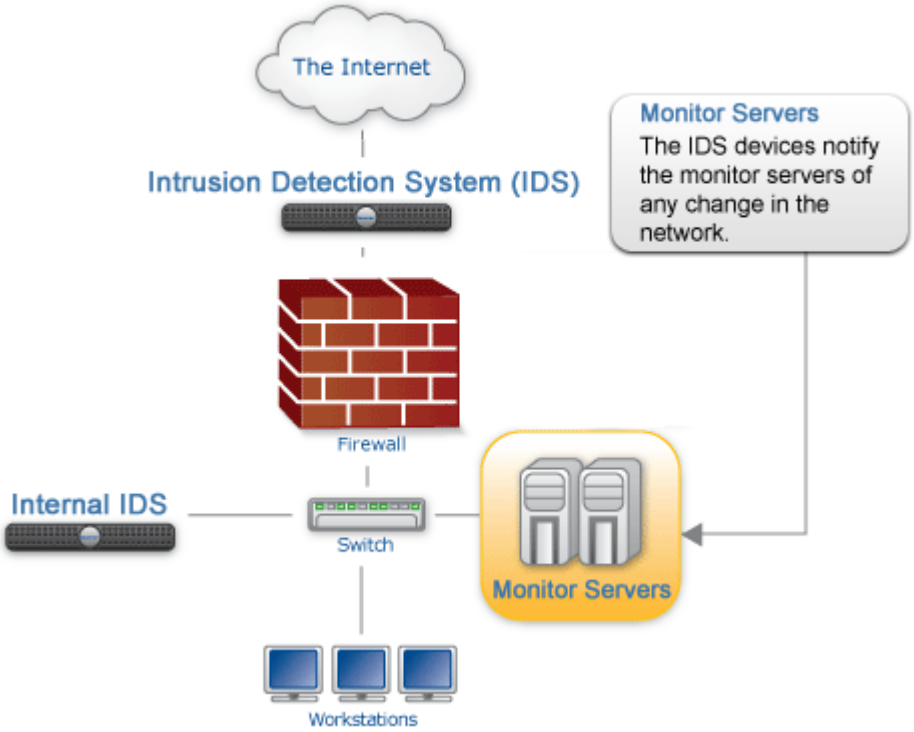
Akan tetapi sebuah *Intrusion Detection System* (IDS) memiliki kekurangan dalam melakukan sebuah pencegahan apabila terjadi intrusi (serangan) maupun penyusupan didalam sebuah jaringan yang terdapat aktivitas mencurigakan tersebut. IDS hanya dapat melakukan deteksi dan tidak dapat melakukan pencegahan. Namun IDS memiliki beberapa peranan penting dalam sebuah keamanan jaringan, berikut peranan penting yang ada pada IDS:

a. Melakukan pengamatan dengan aktif dalam setiap aktivitas mencurigakan.
b. Melakukan pemeriksaan secara teliti pada *audit logs.*
c. Melakukan pengiriman pesan atau *alerts* terhadap administrator jika terdapat aktivitas mencurigakan.
d. Melakukan pemberian *signal* atau tanda jika terdapat kerentanan.

IDS dalam kemampuannya melakukan deteksi intrusi ataupun serangan pada sebuah jaringan dapat dikategorikan menjadi 2 kategori yaitu sebagai berikut:

1. *Network-based Intrusion Detection System* (NIDS) yang dalam proses melakukan deteksi jika terdapat intrusi atau serangan, maka akan dilakukan analisis untuk seluruh lalu lintas yang terjadi pada jaringan tersebut. Pada dasarnya sebuah NIDS terdapat pada sebuah segmen penting yang ada pada jaringan, yaitu dapat dikatakan sebagai pintu masuk pada sebuah jaringan. Meskipun demikian kelemahan pada NIDS yaitu sedikit lebih rumit saat dilakukan implementasi pada sebuah jaringan yang menggunakan *switch Ethernet.* Namun saat ini beberapa dari *vendor switch Ethernet* sudah melakukan penerapan fungsi IDS pada *switch* yang telah dibuatnya supaya dapat memonitor koneksi port.

2. *Host-based Intrusion Detection System* (HIDS) merupakan IDS yang dalam proses kerjanya hanya dapat mendeteksi sebuah intrusi hanya pada *host* tempat dimana dilakukan implementasi IDS. HIDS melakukan pengamatan maupun pemantauan segala aktivitas hanya pada sebuah *host* jaringan individual apakah didalamnya terdapat aktivitas mencurigakan atau sebuah percobaan penyusupan. HIDS lebih sering diletakkan pada beberapa *server* penting, diantaranya pada *firewall*, *web server*, atau *server* yang terkoneksi ke internet.



Gambar 2.1 Ilustrasi Jaringan yang terpasang IDS (Interactive Systems, 2010)

Pada Gambar 2.1 merupakan diagram tentang ilustrasi jaringan yang terpasang IDS. Pada gambar terlihat ada dua sistem IDS, satu didalam jaringan dan yang lainnya diluar. Perangkat IDS tetap berhubungan konstan dengan monitor server dan menginformasikan mereka tentang perubahan infrastruktur jaringan. Tujuan *Intrusion Detection Systems* (IDS) adalah untuk memantau keefektifan sistem kontrol dengan memantau bukti adanya serangan. Langkah-langkah deteksi intrusi sering diperlukan untuk membantu mengendalikan risiko yang terkait dengan kerentanan umum seperti virus yang ditularkan melalui *e-mail*, laptop yang terinfeksi dan faktor manusia. Agar efektif, solusi IDS memerlukan proses disiplin dan staf ahli, serta konfigurasi pemantauan yang dikonfigurasi dengan hati-hati. Sistem Interaktif dapat mengambil perkiraan pekerjaan dari *Intrusion Detection* dengan menyediakan solusi terkelola dengan biaya efektif.

2.2.2.2 Metode IDS Mengenali *Intruder*

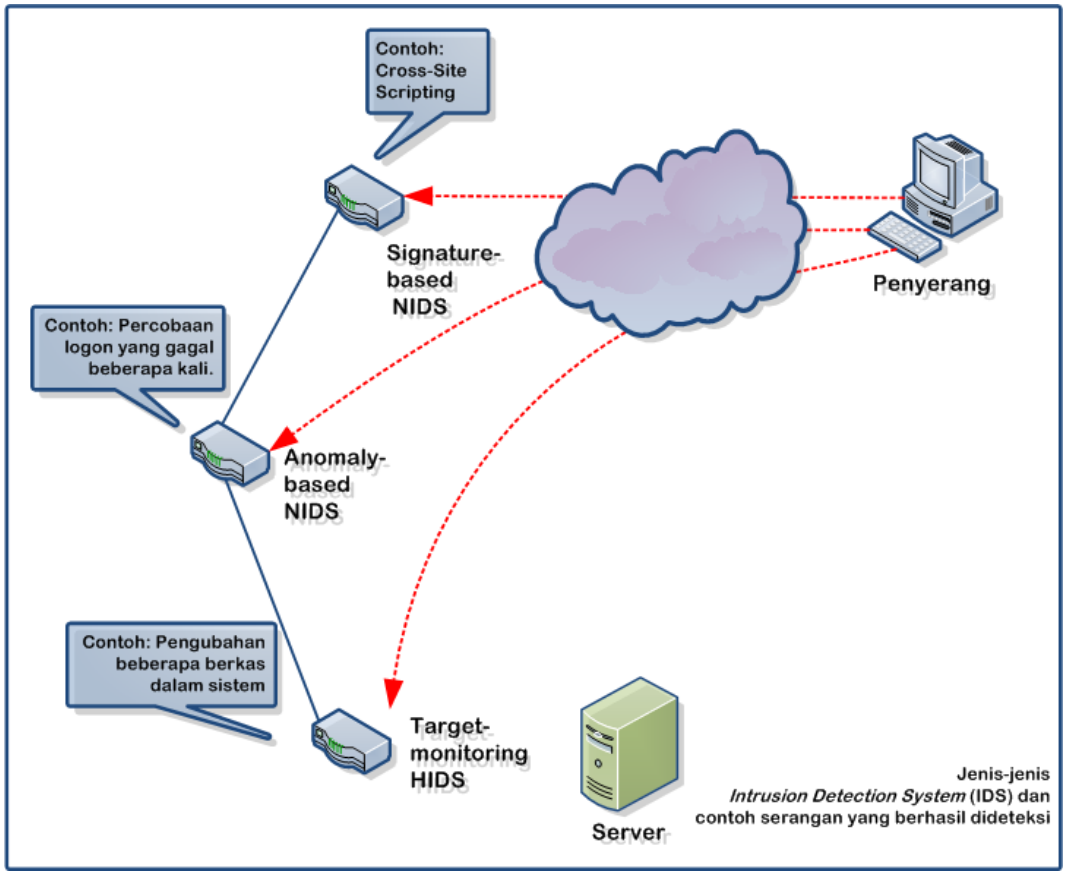
Dalam mengenali sebuah intruder, sebuah IDS terbagi dalam dua bagian. Yang pertama *Knowledge based* dan yang kedua *Behavior based.* Sebuah *Intrusion* atau serangan dapat dikatakan sebagai sebuah kegiatan yang bersifat *anomaly*, *incorrect* atau *inappropriate* yang terjadi pada sebuah *host* ataupun jaringan (Monika Kusumawati, 2010).

1. *Knowlege based* merupakan metode dalam mengenali adanya sebuah *intruder* dengan cara melakukan penyadapan pada sebuah paket data, setelah itu dilakukan perbandingan dengan *rules* yang ada pada *database*. *Database* berisi *signature-signature* atau pola serangan yang sudah dikenali oleh administrator. Jika paket data tersebut memiliki *signature* yang sama dengan yang ada pada *database*, maka dapat dianggap sebagai atau serangan.
2. *Behavior based* merupakan metode dalam mengenali adanya sebuah *intruder* dengan cara melakukan perbandingan pola atau aktivitas yang ada pada sebuah dataset, kemudian dilakukan klasifikasi dengan sebuah metode dan menghasilkan sebuah model. Dari model yang sudah dibangun tersebut diuji dengan data *testing* menghasilkan sebuah output untuk melihat akurasi apakah sebuah *traffic* yang ada dapat dikategorikan sebagai intrusi atau bukan. Maka dalam hal ini dibutuhkan sebuah metode yang digunakan untuk proses klasifikasi untuk menghasilkan akurasi yang akurat.

2.2.2.3 Cara Kerja IDS

Cara Kerja yang paling sering dilakukan oleh IDS yaitu dengan melakukan deteksi berbasis *signature*. Cara pendeteksian ini dilakukan dengan membandingkan kecocokan lalu lintas yang mengalir pada jaringan dengan *database* yang berisi pola-pola serangan yang telah dikenali sebelumnya karena sering dilakukan oleh penyerang. Untuk itu, dalam hal ini selalu dibutuhkan pembaharuan terhadap *rules* database IDS yang bersangkutan.

Metode lain yang dilakukan IDS dalam proses kerjanya yaitu dengan melakukan deteksi apakah terdapat anomali. Dalam hal ini disebut dengan *anomaly-based* IDS. Metode ini dilakukan dengan cara melibatkan pola lalu lintas yang mengalir pada jaringan apakah terdapat kemungkinan adanya sebuah serangan yang sedang dilakukan. Secara umum, metode jenis ini dilakukan dengan melakukan perbandingan pada sebuah lalu lintas yang sedang mengalir pada jaringan dengan lalu lintas normal yang terjadi seperti biasanya. Metode jenis ini terdapat kelebihan dibandingkan dengan metode *signature-based,* yaitu bisa melakukan deteksi jenis serangan baru yang sebelumnya tidak pernah terdeteksi pada pola yang ada pada *database*. Akan tetapi metode jenis ini memiliki kelemahan yaitu seringnya mengirimkan pesan *false positive*.



Gambar 2.2 Jenis IDS dan contoh serangan yang berhasil dideteksi (Willy Saefurrahman, 2007)

Pada Gambar 2.2 merupakan jenis-jenis *Intrusion Detection System* (IDS) dan contoh serangan yang berhasil dideteksi. Dari gambar terlihat penyerang memasuki *cloud* dengan tiga metode penyerangan yaitu dengan metode *signature-based* NIDS, *anomaly-based* NIDS, dan *target-monitoring* HIDS. Contoh serangan pertama yaitu *cross-site scripting* yaitu satu jenis serangan injeksi code *(code injection attack)* yang dilakukan oleh penyerang dengan cara memasukkan kode HTML atau *client script code* lainnya ke suatu situs. Contoh serangan kedua menyebabkan kegagalan percobaan saat *login* berkali-kali. Dan contoh serangan ketiga yang berhasil dideteksi yaitu pengubahan beberapa berkas dalam sistem.

2.2.2.4 Kelebihan dan Kekurangan IDS

Berikut ini kelebihan serta kekurangan yang terdapat pada IDS:

- Kelebihan
  1. Dapat melakukan deteksi serangan pada jaringan internal maupun *external hackers.*
  2. Dapat secara mudah menyesuaikan dalam melakukan perlindungan untuk keseluruhan jaringan.
  3. Dapat menangani serangan yang menyebar dengan pengelolaan terpusat.

4. Mampu memberikan pertahanan pada bagian dalam.
5. Mampu memberikan layer perlindungan tambahan.
6. Melakukan deteksi serangan dengan memonitor internet.
7. Memberikan bantuan pada organisasi dalam upaya memberikan kebijakan keamanan yang efisien dan efektif.
8. Memberikan pengelolaan keamanan secara menyeluruh terhadap anggota *non-technical*.
9. Selalu melakukan pengecekan jika ada perubahan pada *file* data serta pemeriksaan integritas data.
10. Melaukan pelacakan seluruh aktivitas *user.*
11. Memberikan penyederhanaan pada sistem yang kompleks.

- Kekurangan
  1. Hanya dapat mendeteksi tidak mampu mencegah.
  2. Memberikan hasil data yang besar untuk dilakukan analisa
  3. Lebih rawan terhadap jenis serangan yang lambat dan rendah
  4. Belum mampu memberikan penanganan pada trafik jaringan yang dienkripsi.
  5. Hanya mampu memberikan perlindungan dari karakteristik yang ada sebelumnya.
  6. Belum mampu memberikan penyediaan terhadap penanganan kecelakaan.
  7. Belum mampu melakukan identifikasi serangan berasal.
  8. *Network*-IDS lebih rentan pada *overload*
  9. *Network*-IDS masih sering menyalahartikan hasil dari lalu lintas jaringan yang mencurigakan.

2.2.3 Naïve Bayes

Naive Bayes menurut Xhemali, et al (2009) adalah sebuah metode atau algoritma klasifikasi sederhana *(simple),* yang mampu berkontribusi pada keputusan akhir dan pada setiap atributnya memiliki sifat *independent*. Menurut Hang dkk (2006) *Naive Bayes* merupakan proses klasifikasi statistik yang bisa digunakan dalam melakukan prediksi suatu probabilitas pada keanggotaan sebuah *class.* Sedangkan menurut Bustami (2013) *Naive Bayes* merupakan metode pengklasifikasian suatu probabilitas dan statistik yang diperoleh *Thomas Bayes* seorang ilmuwan Inggris dengan cara melakukan prediksi peluang di masa depan berdasarkan pengalaman pada masa sebelumnya. Definisi lain yang dikemukakan oleh Kusrini (2009) bahwa *Naive Bayes Classification* merupakan sebuah teorema *Bayes* dimana kemampuan yang dimilikinya mampu memberikan klasifikasi yang serupa dengan metode *neural network* dan *decision tree*. *Naive bayes* dalam proses klasifikasinya mampu memberikan hasil akurasi serta kecepatan yang tinggi untuk waktu komputasi saat diimplementasikan ke dalam sistem *database* dengan data yang besar.

Teori probabilitas *Bayesian* merupakan salah satu teori *statistik matematik* yang memberikan kemungkinan dalam membuat suatu model ketidakpastian dari suatu kejadian dengan cara menggabungkan pengetahuan umum dengan fakta hasil pengamatan. Beberapa kelebihan yang ada pada Teori *Bayesian* menurut Grainner (1998) diantaranya sebagai berikut:

1. Mampu dipahami dengan mudah.
2. Mampu dilakukan hanya dengan kode sederhana.
3. Mampu melakukan perhitungan dengan cepat.

Akan tetapi Teori Bayesian juga memiliki kelemahan yaitu pada teori ini, dalam satu probabilitas belum mampu mengukur seberapa dalam tingkat keakuratannya. Dapat dikatakan bahwa belum mampu menghasilkan bukti kebenaran atas jawaban yang dihasilkan dari teori bayesian ini.

Teorema *Bayes* yang biasanya dipakai dalam pemrograman memiliki bentuk umum sebagai berikut (Han, Kamber, & Pei, 2011):

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \qquad (1)$$

Dari persamaan rumus di atas, dapat disimpulkan bahwa H merepresentasikan suatu kelas, sedangkan X merepresentasikan suatu atribut. P(H) dalam persamaan rumus diatas merupakan *prior probability* H sedangkan P(X) merupakan *prior probability* X.

Tabel 2.2 Penjelasan atribut rumus klasifikasi *naive bayes*

Atribut	Keterangan
X	Sebuah data yang belum diketahui class-nya
H	Suatu *class* yang spesifik pada hipotesis data X
P(H|X)	Probabilitas hipotesis H berdasarkan kondisi X *(posterior probability)*
P(H)	*Prior probability* H, dapat disebut probabilitas dari hipotesis H
P(X|H)	Probabilitas hipotesis X berdasarkan kondisi H *(posterior probability)*
P(X)	*Prior probability* X, dapat disebut probabilitas dari hipotesis X

Pada *Teorema Bayes*, klasifikasi *naive bayes* mempunyai persamaan berikut:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \qquad (2)$$

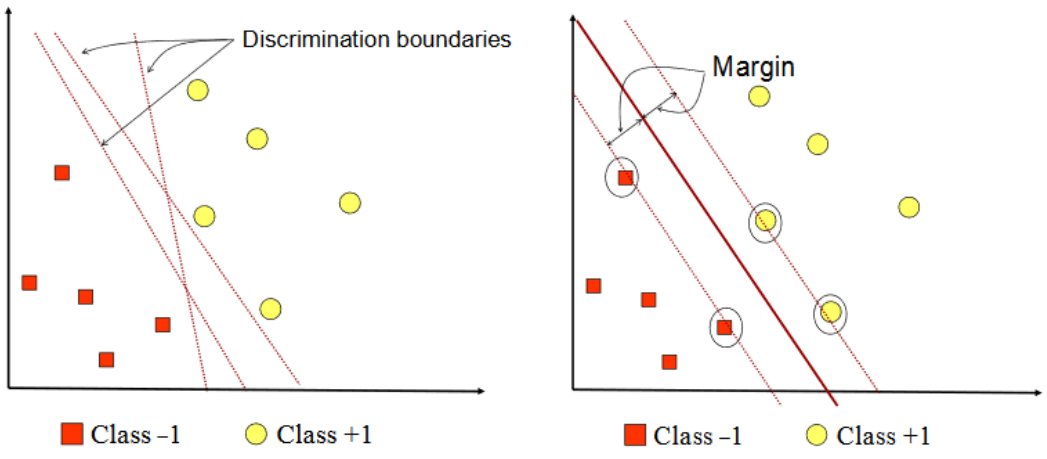
2.2.4 *Support Vector Machine* (SVM)

*Support Vector Machine (*SVM) dikembangkan oleh Boser, Guyon, dan Vapnik yang dikemukakan pada tahun 1992 di *Annual Workshop on Computational Learning Theory.* Konsep dasar yang ada pada SVM merupakan beberapa teori komputasi yang sudah ada puluhan tahun sebelumnya, misalnya saja *margin hyperplane*. Pada tahun 1950 seorang Aronszajn memperkenalkan konsep kernel SVM serta kosep-konsep pendukung yang lain. Beberapa komponen yang ada tersebut belum terdapat upaya untuk dirangkai hingga tahun 1992.

Terdapat beberapa pilihan fungsi Kernel dari SVM diantranya (Suykens, Gestel, Brabanter, Moor, & Vandewalle, 2002):

- SVM Linear
- SVM Polynomial
- Kernel RBF (Radial Basis Function)
- Kernel MLP (Multi Layer Perceptron)
- Tangent Hyperbolic (sigmoid)

Secara sederhana, konsep SVM dapat didefinisikan dengan cara mencari sebuah *hyperplane* terbaik yang memiliki fungsi dalam memisahkan dua buah kelas pada *input space*. Terdapat istilah diantaranya *Pattern* dan *Margin*. *Pattern* merupakan anggota dari dua buah kelas yaitu +1 dan -1 dan *discrimination boundaries* yang disebut sebagai alternatif garis. Sedangkan *Margin* merupakan jarak antara *hyperplane* dengan *support vector*. *Support vector* merupakan *pattern* terdekat. Proses pembelajaran pada SVM intinya hanya usaha dalam mencari lokasi *hyperplane* (Cristianini dan Tayor, 2000).



Gambar 2.3 SVM berusaha menemukan *hyperplane* terbaik yang memisahkan kedua *class* –1 dan +1 (Anto Satriyo, 2003)

Gambar 2.3 menggambarkan dua buah *class* yaitu +1 dan -1. Anggota dari dua buah class tersebut terdapat beberapa *pattern*. *Pattern* yang terdapat pada *class* –1 digambarkan dengan bentuk kotak warna merah, sementara itu *pattern* pada *class* +1, disgambarkan dengan bentuk lingkaran warna kuning. Dalam proses klasifikasi dilakukan dengan usaha untuk menemukan garis (*hyperplane*) yang memisahkan antara kedua *class* tersebut. *Discrimination boundaries* atau disebut juga sebagai alternatif garis pemisah telah ditunjukkan pada gambar 2.3. untuk menemukan *Hyperplane* pemisah terbaik antara kedua *class* dapat dilakukan dengan cara mencari titik maksimal serta mengukur *margin hyperplane*-nya. *Margin* merupakan jarak antara *hyperplane* dari masing-masing *class* dengan *support vector*. *Support vector* merupakan *pattern* terdekat. Sedangkan garis solid yang ada pada gambar sebelah kanan menunjukkan *hyperplane* yang terbaik, yaitu pada gambar ditunjukkan tepat pada tengah-tengah kedua *class*, sedangkan *support vector* digambarkan dengan titik merah dan kuning yang berada dalam lingkaran hitam.

2.2.4.1 SVM Linear

Tiap data (*example*) dinotasikan sebagai $\mathrm{xi}\ 2 <\mathrm{D}, \mathrm{i} = 1, 2, \cdots ,\mathrm{N}$. N adalah banyaknya data. Positive *class* dinotasikan sebagai +1, dan negative *class* sebagai −1. Dengan demikian, tiap data dan label *class*-nya dinotasikan sebagai : yi 2 {−1,+1}. Diasumsikan bahwa kedua *class* tersebut dapat dipisahkan secara sempurna oleh *hyperplane* di D-*dimensional feature space*. *Hyperplane* tersebut didefenisikan sebagi berikut:

$$w.xi + b = 0 \quad (3)$$

Data x_i yang tergolong ke dalam *negative class* adalah mereka yang memenuhi pertidaksamaan berikut:

$$w.xi - b = 0 \quad (4)$$

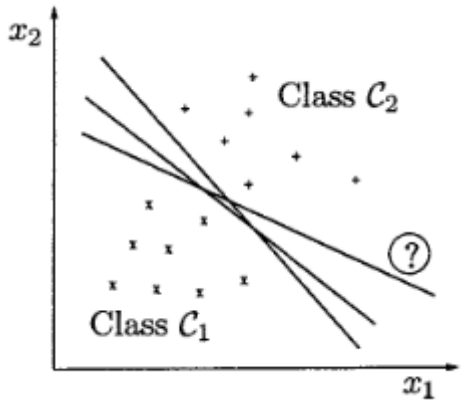
Adapun data x_i yang tergolong ke dalam *positive class*, adalah mereka yang memenuhi pertidaksamaan:

$$w.xi + b = 1 \quad (5)$$

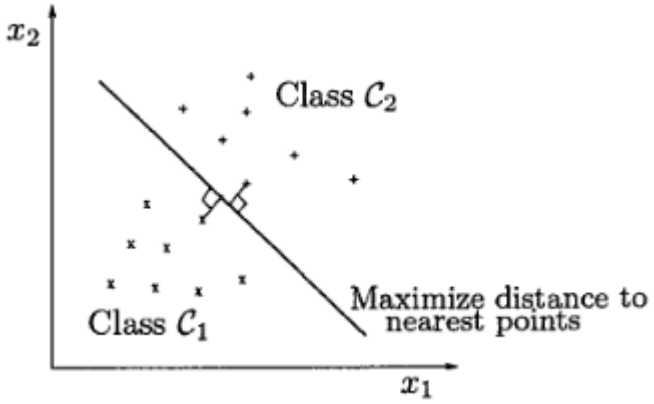
Optimal margin dihitung dengan memaksimalkan jarak antara *hyperplane* dan pattern terdekat. Jarak ini dirumuskan sebagai 1/kwk (kwk adalah norm dari weight vector w). Selanjutnya, masalah ini diformulasikan ke dalam *Quadratic Programming* (QP) *problem*, dengan meminimalkan Eq.4 dibawah *constraint* Eq.5. Minimize [21]:

$$/w /^2 \quad (6)$$

Untuk suatu training set , bisa diperoleh lebih dari satu *hyperplane* seperti contoh pada Gambar 2.1 di bawah.



Gambar 2.4 Contoh Banyaknya Alternatif *Hyperplane* untuk Suatu Masalah Klasifikasi (Suykens et al., 2002)



Gambar 2.5 Pendefinisian *Hyperplane* yang Unik Berdasarkan Konsep Optimal *Hyperplane* (Suykens et al., 2002)

2.2.4.2 SVM Polynomial

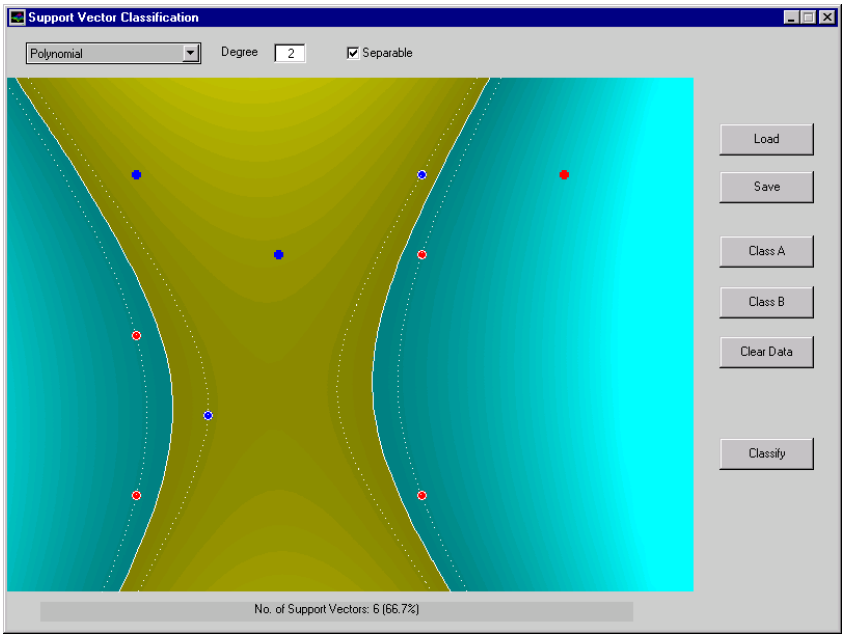
Polynomial merupakan salah satu kernel dari SVM yang sering digunakan untuk klasifikasi gambar.

$$K(x, x0) = (hx, x0i + 1)^2$$

Tabel 2.3 *Non-Linearly Separable Classification* Data (Gunn SR, 1998)

X_1	X_2	*Class*
1	1	-1
3	3	1
1	3	1
3	1	-1
2	2.5	1
3	2.5	-1
4	3	-1

1.5	1.5	1
1	2	-1



Gambar 2.6 Memetakan ruang masukan ke ruang fitur *Polynomial* (Gunn SR, 1998)

Yang memetakan vektor masukan dua dimensi ke dalam ruang fitur enam dimensi. Menerapkan SVC *non linier* ke data pelatihan *linier* non-terpisah dari Tabel 2.3, menghasilkan klasifikasi yang digambarkan pada Gambar 2.3. Margin tidak lagi lebar konstan karena proyeksi non linier ke ruang input. Namun, walaupun SVM menerapkan prinsip SRM dan karenanya dapat menggeneralisasi dengan baik, pemilihan fungsi kernel yang hati-hati diperlukan untuk menghasilkan batas klasifikasi yang sesuai secara topologi. Selalu memungkinkan untuk memetakan ruang masukan ke dimensi yang lebih besar daripada jumlah titik pelatihan dan menghasilkan klasifikasi tanpa kesalahan klasifikasi pada rangkaian pelatihan (Gunn SR, 1998).

2.2.4.3 SVM Sigmoid

The Hyperbolic Tangent Kernel juga dikenal sebagai *Sigmoid* Kernel dan sebagai *Multilayer Perceptron* (MLP) kernel. Kernel *Sigmoid* berasal dari *Neural Networks*, dimana fungsi *sigmoid bipolar* sering digunakan sebagai fungsi aktivasi untuk *neuron* buatan. Perlu dicatat bahwa model SVM menggunakan fungsi kernel sigmoid setara dengan dua lapis, jaringan syaraf *perceptron*. Kernel ini cukup populer untuk mendukung mesin vektor karena berasal dari teori jaringan syaraf tiruan. Selain itu, meski hanya positif bersyarat positif, telah terbukti berhasil

dengan baik dalam praktiknya. Ada dua parameter yang dapat disesuaikan pada kernel *sigmoid*, *alpha* kemiringan dan konstanta *intercept* c. Nilai umum untuk alpha adalah 1 / N, di mana N adalah dimensi data. Studi lebih rinci tentang kernel *sigmoid* dapat ditemukan dalam karya-karya Hsuan-Tien dan Chih-Jen.

Kernel *sigmoid* $K\ (x, y) = \tanh(\ ax^T y + c)$

2.2.5 Evaluasi Kinerja *Classifier*

Percobaan dari penelitian dapat dilakukan sebuah evaluasi dengan pengukuran nilai *accuracy, precision, recall* dan *f-score*. Menurut Xhemali, et al (2009) *Confusion Matrix* dapat dilakukan pengukuran dengan cara menggunakan tabel klasifikasi yang bersifat prediktif.

2.2.5.1 Confusion Matrix

Confusion matrix menurut Han dan Kamber (2011) dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali *tuple* dari kelas yang berbeda. Nilai dari *True-Positive* dan *True-Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False-Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data.

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P + N$

Gambar 2.7 *Confusion Matrix* menampilkan total *positive* dan *negative tuple* (Han dan Kamber, 2011)

TP *(True Positive)* → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif

FP *(False Positive)* → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif

FN *(False Negative)* → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi negatif

TN *(True Negative)* → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif

Adapun perhitungannya adalah sebagai berikut:

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \text{x}100\% \quad (1)$$

$$Precision = \frac{\text{TP}}{\text{FP+TP}} \text{x}100\% \quad (2)$$

$$Recall = \frac{\text{TP}}{\text{FN+TP}} \text{x}100\% \quad (3)$$

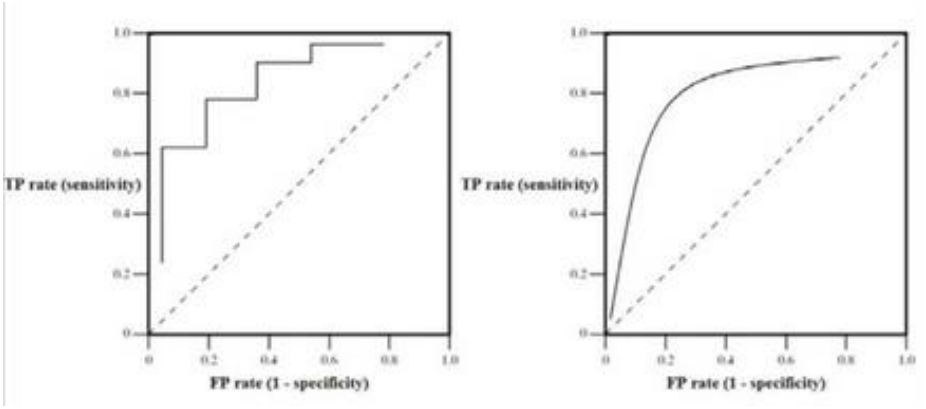
Dari *Confusion Matrix* yang dihasilkan dapat diukur nilai *accuracy, precision* dan *recall* untuk dilakukan analisa dari kinerja masing-masing algoritma dalam melakukan klasifikasi. *Accuracy* merupakan persentase dalam proses klasifikasi yang dihasilkan merupakan prediksi yang benar. Sementara itu *Precision* merupakan ukuran dari nilai akurasi yang telah diprediksi sebelumnya dari suatu kelas. Sedangkan *recall* merupakan persentase dari hasil klasifikasi data yang bernilai positif serta nilai prediksinya juga bernilai positif. Ukuran besaran *precision* , *recall* , dan *accuracy* pada umumnya berbentuk presentase antara 1 sampai 100%. Sebuah sistem akan dianggap baik jika tingkat *accuracy, precision*, dan *recall*-nya tinggi.

2.2.5.2 Kurva ROC *(Receiver Operating Characteristic)*

Kurva ROC menunjukan visualisasi dari akurasi model dan membandingkan perbedaan antar model klasifikasi. *Receiver Operating Characteristic* (ROC) mengekspresikan *confusion matrix* (Vercellis, 2009). ROC merupakan grafik dua dimensi dimana *false positives* sebagai garis horizontal sedangkan *true positives* untuk mengukur perbedaaan performasi metode yang digunakan. Kurva ROC merupakan teknik untuk memvisualisasi dan menguji kinerja pengklasifikasian berdasarkan performanya (Gorunescu, 2011). Model klasifikasi yang lebih baik adalah yang mempunyai kurva ROC lebih besar (Vercellis, 2009).

Kurva ROC merupakan bentuk plot dua dimensi dimana *false-positive* terdapat pada sumbu X sedangkan *true-positive* terdapat pada sumbu Y. Jika nilai *false-positive* bernilai 0 dan nilai *true-positive* bernilai 1 memberikan titik (0,1) yang berarti menghasilkan klasifikasi yang sempurna. Jika nilai *false-positive* bernilai 0 dan nilai *true-positive* bernilai 0 memberikan titik (0,0) maka menjadikan klasifikasi dalam memprediksi setiap kasus menjadi negatif {-1}. Jika nilai *false-positive* bernilai 1 dan nilai *true-positive* bernilai 1 memberikan titik (1,1) maka menjadikan klasifikasi dalam memprediksi setiap kasus menjadi positif {1}. Kurva ROC menggambarkan *trade-off* antara *true-positive* dan *false-positive.*

Berikut ini merupakan gambaran dari dua jenis kurva ROC yaitu discrete dan continous:



Gambar 2.8 Grafik ROC Discrete dan Continous (Gorunescu, 2011)

Pada Gambar 2.5 merupakan garis diagonal yang membagi ruang ROC, yaitu dapat dijelaskan sebagai berikut:

1. Pada gambar a, terlihat jika garis terletak diatas garis *threshold* atau ambang batas (diagonal) maka dapat dikatakan hasil klasifikasinya yang baik.

2. Pada gambar b, terlihat jika garis terletak diatas garis *threshold* atau ambang batas (diagonal) tetapi sedikit lebih kebawah daripada gambar a, maka masih dapat dikatakan hasil klasifikasinya yang baik

Dapat ditarik kesimpulan dari dua gambar di atas, bahwa salah satu gambar pada kurva ROC adalah lebih baik dari pada yang lainnya jika arah garis melintang dari kiri bawah ke kanan atas di dalam kurva.

Performa keakurasian AUC dapat diklasifikasikan menjadi lima kelompok yaitu (Gorunescu, 2011):

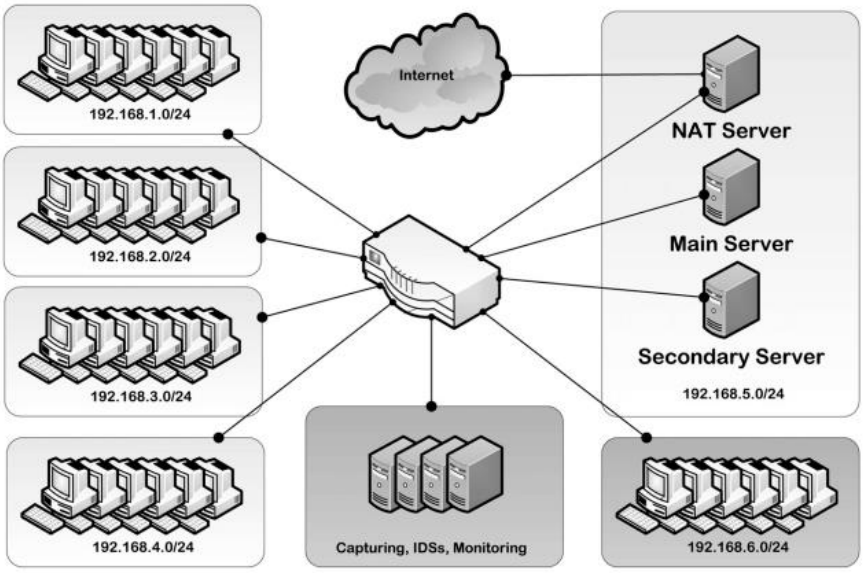
1. Akurasi 0,90 – 1.00 = *Excellent Classification* (Unggul)
2. Akurasi 0,80 – 0,90 = *Good Classification* (Baik)
3. Akurasi 0,70 – 0,80 = *Fair Classification* (Cukup)
4. Akurasi 0,60 – 0,70 = *Poor Classification* (Kurang)
5. Akurasi 0,50 – 0,60 = *Failure* (Gagal)

2.2.6 Dataset ISCX

Dataset *Information Security Center of eXcellence* (ISCX) merupakan data yang dikembangkan oleh Fakultas Ilmu Komputer, *Universitas New Brunswick* dari tahun 2009 sampai tahun 2012. Penelitian yang dilakukan oleh (Shiravi, Tavallaee dan Ghorbani, 2011) menjelaskan tentang ISCX dan jenis pendekatan yang digunakan dalam mengembangkan dataset. Seluruh dataset berlabel ISCX terdiri dari 157867 paket dengan 19 *features* dan mengumpulkan lebih dari tujuh hari aktivitas jaringan (yaitu normal dan intrusi). Berikut kumpulan data simulasi ISCX berdasarkan skenario serangan:

1. Infiltrasi jaringan dari dalam

2. HTTP *Denial of Service*
3. *Distributed DoS* menggunakan IRC *Botnet*
4. *Brute-force SSH*



Gambar 2.9 Arsitektur Jaringan *Testbed* ISCX oleh (Shiravi, Tavallaee dan Ghorbani, 2011)

Dataset ISCX 2012 merupakan arsitektur yang dibentuk dengan jaringan *testbed*. Terlihat pada (Gambar 2.9) dataset ISCX dibentuk dari 21 *windows workstation* yang saling terhubung satu sama lain. Terdapat dua mesin berbasis *linux Ubuntu* dan satu mesin yang telah dilakukan instalasi *windows server* 2003. Terdapat 4 elemen multi tahap dalam menghasilkan dataset ISCX diantaranya:

1. *Probe,* yang bertujuan untuk mengumpulkan informasi
2. Identifikasi *vulnerability*
3. Penciptaan *backdoors* untuk mempertahankan akses
4. Secara efektif memiliki kemampuan untuk menutupi jalur penyerangan.

Pada penelitian ini digunakan dataset ISCX 2012 hasil *testbed* pada tanggal 14 juni 2012 yang merupakan serangan HTTP *Denial of Service*. Skenario serangan HTTP *Denial of Service* dirancang tanpa membanjiri jaringan, sehingga *bandwitdth* yang dibutuhkan rendah. Dataset ISCX 2012 memanfaatkan *slowloris* sebagai alat utama dalam skenario serangan *HTTPDoS*.

Data yang didapatkan dari jaringan *testbed* tersebut adalah data dari jaringan *testbed* dengan format *Pcap* yang di arsip dalam 7 zip pengarsipan dan 1 arsip dengan format *xml* yaitu *labelled flows xml*,yang mencakup keseluruhan data *testbed* tanggal 11 juni hingga 17 juni. Dari 7 zip data tersebut kemudian dipecah menjadi beberapa folder data. Data itu kemudian dipecah menjadi beberapa

tahap, yang menjadi aktivitas normal, infiltrasi jaringan dari dalam, *HTTP denial of Service, DDoS* menggunakan *botnet* dan *Brute Force SSH*, 19 fitur diekstraksi dan disusun menggunakan metode Universitas *New Brunswick*. Paket diekstraksi berdasarkan ukuran *packet header* dari protocol, jumlah paket per aliran, pola-pola tertentu dalam *payload* dan ukuran keseluruhan setiap *payload*. Real trace dataset dianalisis untuk membuat profil agen yang menghasilkan traffic secara *real* untuk HTTP, SMT, SSH, IMAP, POP3 dan FTP. Berikut tahap evaluasi *Intrusion Detection System* dataset ISCX 2012:

Tabel 2.4 Evaluasi IDS Dataset ISCX 2012 (Normal dan Attack)

Hari	Tanggal	Deskripsi	Size (GB)
Jumat	11/06/2010	Aktifitas Normal	16.1
Sabtu	12/06/2010	Aktifitas Normal	4.22
Minggu	13/06/2010	*Infiltrating the network from inside* dan aktifitas normal	3.95
Senin	14/06/2010	*HTTP Denial of Service* dan aktifitas normal	6.85
Selasa	15/06/2010	*DDoS IRC Botnet*	23.4
Rabu	16/06/2010	Aktifitas Normal	17.6
Kamis	17/06/2010	*Brute Force SSH* + aktifitas normal	12.3

Tabel 2.5 Daftar *Features* pada Dataset ISCX

No.	Nama	No.	Nama
1	*appName*	11	*Src TCPFlagsDescription*
2	*totalSourceBytes*	12	*Dst TCPFlagsDescription*
3	*totalDestinationBytes*	13	*Source*
4	*totalDestinationPacket*	14	*protocolName*
5	*totalSourcePacket*	15	*sourcePort*
6	*Src PayloadAsBase64*	16	*Destination*
7	*Src PayloadAsBaseUTF*	17	*destinationPort*
8	*Dst PayloadAsBase64*	18	*startDateTime*
9	*Dst PayloadAsBaseUTF*	19	*stopDateTime*
10	*direction*		

Fitur-fitur diatas diperoleh dari *extraction raw* paket data kedalam bentuk *comma separated value (*csv).* Tujuan perubahan *extention* untuk memudahkan dalam pengolahan data.

2.2.7 *Spyder*

Spyder (Scientific PYthon Development EnviRonment) merupakan sebuah *open source cross-platform integrated development environment* (IDE) untuk bahasa pemrograman *Python*. *Software* ini dilengkapi dengan pengeditan lanjutan, pengujian interaktif, *debugging* dan fitur introspeksi. Serta dilengkapi juga lingkungan komputasi numerik yang didukung *IPython (enhanced interactive Python interpreter)* dan *library Python* yang populer seperti *NumPy* (aljabar linier), *SciPy* (pemrosesan sinyal dan gambar) atau *matplotlib* (plot 2D / 3D interaktif). *Spyder* juga dapat digunakan sebagai *library* yang menyediakan *widget* terkait konsol untuk aplikasi berbasis *PyQt*. Misalnya digunakan untuk mengintegrasikan konsol *debug* langsung di tata letak *user interface.* Fitur yang ada pada *spyder* diantaranya:

- *Interactive console*
- *Documentation viewer*
- *Variable explorer*
- *Find in filfes*
- *File explorer*
- *History log*