

**IMPLEMENTASI METODE NAIVE BAYES UNTUK  
INTRUSION DETECTION SYSTEM (IDS)**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV  
Politeknik Negeri Malang

Oleh:

**DEDI ARPANDI    NIM. 1641727017**



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**2017**

# **IMPLEMENTASI METODE NAIVE BAYES UNTUK INTRUSION DETECTION SYSTEM (IDS)**

## **SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV  
Politeknik Negeri Malang

Oleh:

**DEDI ARPANDI    NIM. 1641727017**



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**2017**

## HALAMAN PENGESAHAN

### IMPLEMENTASI METODE NAIVE BAYES UNTUK INTRUSION DETECTION SYSTEM (IDS)

Disusun oleh:

**DEDI ARPANDI     NIM. 1641727017**

**Laporan Skripsi ini telah diuji pada tanggal 11 Agustus 2017**

- |                  |   |  |       |
|------------------|---|--|-------|
| 1. Penguji I     | : | <u>DR.ENG. Cahya Rahmad, ST, M.Kom</u> | ..... |
|                  |   | NIP. 19720202 200501 1 002             |       |
| 2. Penguji II    | : | <u>Putra Prima Arhandi, ST, M.Kom</u>  | ..... |
|                  |   | NIP. 19861103 201404 1 001             |       |
| 3. Pembimbing I  | : | <u>Arief Prasetyo, S.Kom., M.Kom</u>   | ..... |
|                  |   | NIP. 19790313 200812 1 002             |       |
| 4. Pembimbing II | : | <u>Luqman Affandi, S.Kom, MMSI</u>     | ..... |
|                  |   | NIP. 19821130 201404 1 001             |       |

Mengetahui,

Ketua Jurusan  
Teknologi Informasi

Ketua Program Studi  
Teknik Informatika

Rudy Ariyanto, S.T., M.Cs.  
NIP. 19711110 199903 1 002

Ir. Deddy Kusbianto Purwoko Aji, MMKom  
NIP. 19621128 198811 1 001

## PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Dedi Arpandi

NIM : 1641727017

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam Skripsi saya yang berjudul “**Implementasi Metode Naive Bayes Untuk Intrusion Detection System (IDS)**” merupakan gagasan dan hasil karya saya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam daftar pustaka dibagian akhir Laporan Akhir.

Malang, Agustus 2017

Dedi Arpandi  
1641727017

## ABSTRAK

**Arpandi, Dedi.** “Implementasi Metode Naive Bayes Untuk Intrusion Detection System (IDS)”. Pembimbing: (1) Arief Prasetyo, S.Kom., M.Kom., (2) Luqman Affandi, S.Kom, MMSI.

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.**

IDS berfungsi untuk mengidentifikasi *traffic* atau lalu-lintas data pada sebuah jaringan komputer dimana IDS dapat menentukan apakah *traffic* aman, mencurigakan atau bahkan terindikasi merupakan serangan. Permasalahan muncul ketika ada aktifitas-aktifitas yang mencurigakan atau bahkan aktifitas tersebut merupakan serangan namun tidak terdaftar pada *rule* atau aturan yang diinputkan sehingga hal itu sangat membahayakan sebuah jaringan komputer.

Tujuan dari penelitian ini adalah membangun sistem deteksi pola serangan baru menggunakan metode *naive bayes* untuk mengatasi serangan-serangan baru yang muncul, dan yang belum terdaftar pada *signature* serta untuk meningkatkan akurasi pendeteksian serangan-serangan baru pada *Intruison Detection System* (IDS). Data yang digunakan pada penelitian ini adalah data NSL-KDD, NSL-KDD telah menyediakan *data training* dan *data testing* untuk proses penelitian klasifikasi serangan. Dari data NSL-KDD akan dilakukan klasifikasi serangan menggunakan metode *naive bayes* agar serangan-serangan baru dapat terklasifikasi.

Penelitian yang menggunakan metode *naive bayes* ini telah berhasil melakukan klasifikasi serangan-serangan baru dengan akurasi kebenaran adalah sebesar 81-84,67 %.

**Kata kunci :** *Intruison Detection System* (IDS), NSL-KDD, *Naive Bayes*

## ABSTRACT

**Arpandi, Dedi.** *“Implementation of Naive Bayes Methods For Intrusion Detection System (IDS)”*. Advisors: (1) Arief Prasetyo, S.Kom., M.Kom. (2) Lukman Affandi, S.Kom, MMSI.

*Undergraduate Thesis, Informatics Engineering Study Program, Department of Information Technology, State Polytechnic of Malang, 2017.*

*IDS servers to identify data traffic on computer networks where IDS can determine whether traffic is safe, suspicious or even indicated as an attack. The problem arises when there is a suspicious activity or even the activity as indicated an attack but not listed in the rules or regulations that is input so it is potentially dangerous for computer network.*

*The purpose of this research is to build a new attack pattern detection system using Naive Bayes method to overcome emerging new attacks, and those which are not yet listed in the signature and to improve the accuracy of detection of new attacks on Intrusion Detection System (IDS). The data used in this research is NSL-KDD data, NSL-KDD has provided data training and data testing for attacks classification process. From the NSL-KDD data then the attacks will be classified using Naive Bayes method so that new attacks can be clasified.*

*The research using Naive Bayes method has successfully classified new attacks with an accuracy of 81-84.67%.*

**Keywords :** *Intrusion Detection System (IDS), NSL-KDD, Naive Bayes*

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmat, hidayah, serta karunianya kepada penulis, sehingga penulis dapat menyelesaikan serangkaian proses untuk mendapatkan gelar Sarjana Sains Terapan Teknologi Informasi di Politeknik Negeri Malang.

Keberhasilan penulis dalam menyelesaikan Skripsi ini tidak lepas dari bantuan orang-orang yang dengan sepenuh hati memberikan doa, bimbingan dan dukungan. Oleh karena itu penulis mengucapkan terima kasih kepada :

1. Bapak Rudy Ariyanto, ST., M.Cs., selaku Ketua jurusan Teknologi Informasi
2. Bapak Arief Prasetyo, S.Kom., M.Kom., selaku Dosen Pembimbing I yang telah membimbing dan mengarahkan kami selama pengerjaan Tugas Akhir serta penyusunan Laporan Skripsi.
3. Bapak Luqman Affandi, S.Kom, MMSI selaku Dosen Pembimbing II yang telah membimbing dan mengarahkan kami selama pengerjaan Tugas Akhir serta penyusunan Laporan Skripsi.
4. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Skripsi dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Dalam penyusunan laporan ini penulis menyadari terdapat kekurangan dan keterbatasan, Penulis selalu menerima saran dan kritik membangun demi perbaikan di masa mendatang. Semoga laporan Skripsi ini dapat bermanfaat bagi pembaca.

Malang, Juni 2017

Penulis

## DAFTAR ISI

Halaman

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
PERNYATAAN.....	iii
ABSTRAK .....	v
<i>ABSTRACT</i> .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN .....	xiii
BAB I. PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan.....	3
1.5 Sistematika Penulisan.....	3
BAB II. TINJAUAN PUSTAKA.....	3
2.1 Keamanan Jaringan .....	5
2.2 <i>Intrusion Detection System</i> .....	6
2.3 <i>Naive Bayes</i> .....	7
2.4 Perhitungan Metode <i>Naive Bayes</i> .....	8
2.5 <i>My Sql</i> .....	12
2.6 PHP.....	13
2.7 Dataset NSL-KDD .....	13
2.8 Format Dataset KDD ‘99 .....	14
2.9 Perbandingan dengan Penelitian Terdahulu.....	16
BAB III. METODOLOGI PENELITIAN .....	17
3.1 Metodologi .....	17
BAB IV. ANALISA DAN PERANCANGAN.....	22
4.1 Pengambilan Data.....	22
4.1.1 Deskripsi Umum Sistem .....	23
4.1.2 Data Required .....	23



4.1.3	Analisis Kebutuhan Sistem.....	24
4.2	Analisis Permasalahan .....	24
4.3	Perancangan Desain dan Sistem .....	25
4.3.1	<i>Data Flow Diagram</i> Level 0.....	25
4.3.2	<i>Data Flow Diagram</i> Level 1.....	26
4.3.3	<i>Data Flow Diagram</i> Level 2.....	27
4.3.4	Rancangan <i>Database</i> .....	28
4.3.5	<i>Flowchart</i> .....	29
4.3.5.1	<i>Flowchart Sistem</i> .....	29
4.3.5.2	<i>Flowchart Metode Naive Bayes</i> .....	30
4.3.6	Perancangan Interface.....	31
4.4	Implementasi.....	35
4.5	Pengujian Sistem.....	36
BAB V. IMPLEMENTASI SISTEM.....		37
5.1	Pembuatan Data Training dan Testing .....	37
5.2	Pembuatan Database.....	38
5.3	Interface Aplikasi <i>Intrusion Detection System (IDS)</i> .....	39
BAB VI. PENGUJIAN DAN PEMBAHASAN .....		43
6.1	Pengujian Sistem .....	43
6.1.1	Pengujian Black Box .....	43
6.1.1.1	Uji Coba Login.....	43
6.1.1.2	Uji Coba Lihat Data Training.....	45
6.1.1.3	Uji Coba Import Data Traininig .....	47
6.1.1.4	Uji Coba Import Data Testing .....	50
6.1.1.5	Uji Coba Menu Hasil.....	52
6.2	Pengujian Akurasi .....	54
6.2.1	Pengujian Akurasi 1500 Data Training.....	55
6.2.2	Pengujian Akurasi 3000 Data Training.....	56
6.2.3	Pengujian Akurasi 5000 Data Training .....	58
6.3	Pembahasan .....	60
BAB VII. KESIMPULAN DAN SARAN .....		61
7.1	Kesimpulan.....	61
7.2	Saran .....	61
DAFTAR PUSTAKA .....		62

## DAFTAR GAMBAR

	Halaman
Gambar 3.1 Diagram Pengembangan Aplikasi .....	17
Gambar 3.2 Desain Sistem Klasifikasi Serangan Metode <i>Naive Bayes</i> .....	20
Gambar 4.1 <i>DFD</i> Level 0 .....	25
Gambar 4.2 <i>DFD</i> Level 1 .....	26
Gambar 4.3 <i>DFD</i> Level 2 .....	27
Gambar 4.4 <i>Flowchart Sistem</i> .....	29
Gambar 4.5 <i>Flowchart Sistem</i> metode <i>naive bayes</i> .....	30
Gambar 4.6 Halaman Utama Sistem .....	31
Gambar 4.7 Halaman Tentang .....	32
Gambar 4.8 Halaman Klasifikasi .....	32
Gambar 4.9 Halaman Form Login .....	33
Gambar 4.10 Halaman Metode .....	33
Gambar 4.11 Halaman Admin .....	34
Gambar 4.12 Halaman Import Data .....	34
Gambar 4.14 Halaman Hasil .....	35
Gambar 5.1 Data Training .....	37
Gambar 5.2 Data Testing .....	37
Gambar 5.3 Tabel Data Training .....	38
Gambar 5.4 Tabel Data Testing .....	38
Gambar 5.5 Halaman Utama .....	39
Gambar 5.6 Halaman Login .....	40

Gambar 5.7	Halaman Klasifikasi .....	41
Gambar 5.8	Halaman <i>Import data training</i> .....	41
Gambar 5.9	Halaman <i>import data testing</i> .....	42
Gambar 5.10	Halaman Hasil .....	42
Gambar 6.1	Gambar <i>login admin</i> berhasil .....	44
Gambar 6.2	Gambar <i>login admin</i> tidak berhasil .....	44
Gambar 6.3	Gambar uji coba menu data training .....	46
Gambar 6.4	Gambar data berhasil diimport .....	48
Gambar 6.5	Gambar data tidak sesuai dengan format.....	48
Gambar 6.6	Gambar data berhasil diimport .....	51
Gambar 6.7	Gambar data tidak sesuai dengan format.....	51
Gambar 6.8	Gambar hasil klasifikasi serangan .....	52
Gambar 6.9	Perbandingan Kebenaran Data Untuk Pengujian Akurasi 1500 Data Training.....	55
Gambar 6.10	Perbandingan Kebenaran Data Untuk Pengujian Akurasi 3000 Data Training .....	57
Gambar 6.11	Perbandingan Kebenaran Data Untuk Pengujian Akurasi 5000 Data Training .....	58

## DAFTAR TABEL

Halaman

Tabel 2.1	Perbedaan <i>Network Based</i> IDS dan <i>Client Based</i> IDS.....	7
Tabel 2.2	<i>Data Training</i> 3 Klasifikasi Serangan.....	9
Tabel 2.3	<i>Data Testing</i> .....	9
Tabel 2.4	<i>Data Testing</i> Terklasifikasi .....	11
Tabel 2.5	Atribut pada KDDCUP ‘99 .....	15
Tabel 2.6	Perbandingan Metode <i>Naive Bayes</i> dengan Metode Lain .....	16
Tabel 3.1	Pertanyaan Wawancara .....	19
Tabel 4.1	Deskripsi Field yang dibutuhkan.....	23
Tabel 4.2	Rancangan Database data testing dan data training .....	28
Tabel 4.3	Rincian Data Training dan Data Testing .....	36
Tabel 6.1	Skenario Uji Coba <i>Login</i> .....	44
Tabel 6.2	Skenario Uji Coba Menu Data Training.....	45
Tabel 6.3	Skenario Uji Coba Menu <i>Import Data Training</i> .....	47
Tabel 6.4	Skenario Uji Coba Menu <i>Import Data Testing</i> .....	50
Tabel 6.5	Skenario Uji Coba Menu Hasil.....	52
Tabel 6.6	Rincian Data Pengujian Akurasi .....	54
Tabel 6.7	Uji Coba Akurasi 1500 Data Training.....	56
Tabel 6.8	Uji Coba Akurasi 3000 Data Training.....	58
Tabel 6.9	Uji Coba Akurasi 5000 Data Training.....	59
Tabel 6.10	Perbandingan Hasil Akurasi Data Testing .....	60

## **DAFTAR LAMPIRAN**

Lampiran 1. Data Training dan Data Testing

Lampiran 2. Biodata Penulis

## BAB I. PENDAHULUAN

### 1.1 Latar Belakang Masalah

Kemanan adalah salah satu aspek yang penting dalam masalah internet khususnya jaringan komputer. Sebuah jaringan komputer harus mampu memberikan rasa aman terhadap akses yang dilakukan oleh seorang *user*, dengan memberikan jaminan informasi atau data pribadi aman dari pengaksesan seorang intruder (penyerang).

Namun dari tahun ke tahun serangan terhadap jaringan komputer khususnya internet mengalami peningkatan. Berdasarkan laporan dari Kaspersky Lab jumlah serangan melalui *browser* internet sejumlah 23.680.646 pada tahun 2007, meningkat menjadi 73.619.767 pada tahun 2009 dan meningkat lagi menjadi 580.371.937 pada tahun 2010. Internet *browser* menjadi alat utama dalam menyebarkan program-program *malicious* diantara sebagian besar pengguna komputer pada tahun 2010. Algoritma *Kaspersky Security Network* (KSN) hanya mampu mendeteksi serangan *web* sebesar 60 % [3].

Salah satu upaya melindungi jaringan dari ancaman-ancaman intruder (penyerang) adalah membangun sistem deteksi intrusi atau *Intrusion Detection System* (IDS). *Intrusion detection* adalah proses memonitor kejadian pada sistem komputer atau jaringan dan menganalisanya untuk memberikan tanda insiden yang mungkin, yang mana yang merupakan pelanggaran atau mendekati pelanggaran sebuah kebijakan keamanan komputer, kebijakan penggunaan yang disetujui atau praktik keamanan standar [7]. IDS berfungsi untuk mengidentifikasi *traffic* atau lalu-lintas data pada sebuah jaringan komputer dimana IDS dapat menentukan apakah *traffic* aman, mencurigakan atau bahkan terindikasi merupakan serangan. Aplikasi IDS yang ada saat ini bekerja menurut *rule-rule* atau aturan-aturan yang telah diinputkan oleh seorang *administrator* jaringan berdasarkan *rule-rule* atau aturan-aturan yang telah disediakan oleh masing-masing vendor yang bersangkutan.

Permasalahan muncul ketika ada aktifitas-aktifitas yang mencurigakan atau bahkan aktifitas tersebut merupakan serangan namun tidak terdaftar pada *rule* atau aturan yang diinputkan sehingga hal itu sangat membahayakan sebuah

jaringan komputer. Oleh karena itu dibutuhkan sebuah sistem klasifikasi serangan yang berfungsi untuk mengklasifikasi *traffic* jaringan yang ada dan dari klasifikasi tersebut akan diketahui apakah sebuah aktifitas pada sebuah *traffic* jaringan tersebut serangan atau bukan serangan. Dari hasil klasifikasi tersebut juga dapat digunakan menjadi dasar untuk membuat *rule* baru yang akan diinputkan menjadi aturan-aturan pada aplikasi IDS yang digunakan.

Menurut [11] *Naive Bayes* merupakan metode yang paling efektif dan efisien untuk mesin *learning* dan data mining. Selain itu kelebihan *naive bayes* adalah memiliki karakteristik asumsi yang sangat kuat dan independensi dari masing masing kondisi. Sehingga dengan tidak terkaitnya masing-masing kondisi pada *naive bayes* dapat meningkatkan hasil keputusan. Dari penelitian terdahulu yang dilakukan oleh [6], penelitian tersebut melakukan perbandingan akurasi metode klasifikasi *naive bayes* dan *k-nearest neighbor* menyimpulkan bahwa *naive bayes* memiliki akurasi yang lebih baik dibandingkan *k-nearest neighbor*. Penelitian juga dilakukan oleh [1], yang melakukan perbandingan klasifikasi *naive bayes* dan *k-nearest neighbor* dalam pengklasifikasian dokumen teks menyimpulkan bahwa *naive bayes* memiliki kinerja yang lebih baik dibandingkan *k-nearest neighbor*. Penelitian juga dilakukan oleh [10], yang melakukan analisa perbandingan algoritma *naive bayes* dan KNN untuk studi data “*Wisconsin Diagnosis Breast Cancer*” menyimpulkan bahwa *naive bayes* lebih cepat dalam mengklasifikasi data dibandingkan KNN. Dari kelebihan-kelebihan tersebut peneliti akan mencoba membuat sistem deteksi serangan baru menggunakan metode *naive bayes*.

## 1.2 Rumusan Masalah

Bagaimana menerapkan metode *naive bayes* untuk mengklasifikasi serangan yang mampu mengidentifikasi serangan-serangan baru yang belum terdaftar pada *signature* ?

### 1.3 Tujuan

Membangun sistem deteksi pola serangan baru menggunakan metode *naive bayes* untuk mengatasi serangan-serangan baru yang muncul, dan yang belum terdaftar pada *signature*.

### 1.4 Batasan Masalah

Pada penelitian ini ditentukan beberapa batasan masalah, yaitu sebagai berikut :

1. Data Training dan Data Testing berasal dari NSL-KDD ([nsl.cs.unb.ca/NSL-KDD/](http://nsl.cs.unb.ca/NSL-KDD/)).
2. Metode yang digunakan pada klasifikasi serangan adalah metode *naive bayes*.
3. Sistem ini dibuat untuk membantu adminintrator jaringan.

### 1.5 Sistematika Penulisan

Secara garis besar, materi Laporan Skripsi ini terbagi dalam beberapa bab yang tersusun sebagai berikut :

#### **BAB I PENDAHULUAN**

Bab ini berisikan latar belakang, rumusan masalah, tujuan, manfaat, dan sistematika penulisan laporan terkait dengan penelitian IDS menggunakan metode *naive bayes*.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini menguraikan tentang teori-teori yang berhubungan dengan IDS dan metode *naive bayes*.

#### **BAB III METODOLOGI PENELITIAN**

Bab ini menjelaskan tentang metode yang akan digunakan pada penelitian ini.



#### **BAB IV ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan mengenai analisa awal dari aplikasi IDS menggunakan metode *naive bayes*, rancangan awal sebelum pembuatan aplikasi, serta desain sistem yang akan digunakan.

#### **BAB V IMPLEMENTASI SISTEM**

Pada bab ini dijelaskan tentang implementasi metode *naive bayes* untuk klasifikasi serangan pada IDS.

#### **BAB VI PENGUJIAN DAN PEMBAHASAN**

Pada bab ini membahas mengenai hasil uji coba dari aplikasi IDS yang telah di buat secara keseluruhan.

#### **BAB VII KESIMPULAN DAN SARAN**

Berisi uraian singkat dan jelas tentang hasil laporan akhir yang dibuat yaitu implementasi metode *naive bayes* untuk *Intrusion Detection System* (IDS). Serta saran dari peneliti dalam mengembangkan penelitian-penelitian selanjutnya khususnya tentang IDS.

## BAB II. TINJAUAN PUSTAKA

### 2.1 Keamanan Jaringan

Komputer yang terhubung dengan jaringan memiliki resiko ancaman keamanan lebih besar daripada komputer yang tidak terhubung dengan jaringan. Dengan beberapa cara, jaringan komputer dapat lebih dioptimalkan dari resiko ancaman pihak yang tidak memiliki hak akses terhadap sumber yang ada pada jaringan tersebut, namun hal tersebut akan berbanding terbalik dengan kenyamanan akses pengguna, dimana tingkat keamanan yang tinggi akan membuat pengguna tidak nyaman, sedangkan tingkat keamanan yang rendah maka akses semakin nyaman.

Beberapa aspek yang perlu diperhatikan untuk merancang keamanan jaringan komputer, terdapat empat aspek yaitu privacy, integrity, authentication, dan availability [8].

#### a. *Privacy*

*Privacy* mencakup kerahasiaan data atau informasi. *Privacy* ini dimaksudkan agar data ataupun informasi tidak dilihat oleh orang yang tidak berhak. Salah satu cara usaha yang dapat dilakukan adalah dengan menggunakan enkripsi. Kita dapat menggunakan enkripsi tersebut pada informasi atau data yang kita anggap penting, dan terdapat metode atau cara untuk membuka sebuah kunci yang terenkripsi tersebut [8].

#### b. *Integrity*

Integritas mencakup keutuhan dari sebuah data atau informasi, yang dimaksudkan *integrity* adalah menjaga keutuhan dari informasi, agar tetap utuh tidak berubah tanpa sepengetahuan pemilik informasi atau data tersebut. Informasi tidak boleh mengalami perubahan atau penambahan dan pengurangan tanpa sepengetahuan pemilik informasi. Contoh dari gangguan yang mengancam integritas adalah serangan virus maupun Trojan. Salah satu usaha yang dapat dilakukan adalah membuat tanda tangan digital untuk mengurangi resiko ancaman perubahan informasi oleh pihak yang tidak berhak [8].

c. *Authentication*

*Authentication* mencakup keabsahan pemilik dari sebuah informasi atau data. Harus terdapat proses pencocokan antara data dan pemilik data, yang artinya hanya pemiliklah yang berhak mengakses dan memberikan hak akses terhadap data atau informasi tersebut. Usaha yang dilakukan untuk memenuhi aspek ini adalah dengan menggunakan akses kontrol dengan penggunaan user dan password [8].

d. *Availability*

Mencakup tentang ketersediaan sebuah informasi atau data. penyedia layanan harus memastikan pengguna selalu mendapatkan ketersediaan akses terhadap informasi atau data yang dimilikinya. Gangguan pada aspek ini seperti serangan *DDOS* ataupun *mailbomb* yang mengirim permintaan palsu secara bertubi tubi sehingga *server* tidak dapat melayani permintaan lain [8].

## 2.2 *Intrusion Detection System*

IDS merupakan sistem yang dapat melihat pola dari serangan-serangan yang terdapat pada jaringan komputer, pola tersebut berupa paket yang lewat yang teridentifikasi oleh IDS sebagai paket yang mengandung pola serangan. Terdapat dua katagori IDS yaitu *Network Based IDS* dimana jenis ini dapat menganalisa semua paket didalam jaringan dan yang kedua disebut *client based IDS* yang dapat menganalisa *logfile* yang berisi pola mencurigakan dari sebuah serangan terhadap suatu *client* yang lewat pada sebuah jaringan [5].

### 2.2.1. *Network Based IDS.*

*Network-Based Intrusion Detection System* (IDS) menggunakan paket yang ada didalam jaringan sebaai sumber dari data. Terdapat *promiscuous mode* pada IDS sehingga mampu melihat serta menganalisa semua paket yang lewat pada jaringan secara *realtime*. Terdapat empat teknik untuk menganali sebuah serangan yaitu :

- Pola, ekspresi atau pencocokan *bytecode*
- Frekuensi atau threshold paket yang lewat di dalam jaringan.
- Hubungan antara setiap *event*

- Statistik pendeteksi anomali paket

Ketika serangan teridentifikasi oleh IDS akan memberikan respon kepada modul untuk memberikan pilihan cara notifikasi, alarm ataupun mengambil sebuah tindakan terhadap serangan tersebut.

### 2.2.2. *Client Based IDS*

*Client based intrusion detection* merupakan sebuah IDS yang ruang lingkungannya hanya terbatas pada aktifitas *client* tertentu. Prinsipnya adalah pendeteksian hanya dilakukan pada *single client* yang diamati semua aktifitasnya.

Berikut adalah table 2.1 yaitu perbandingan antara *Network Based IDS* dengan *Client Based IDS*

Tabel 2.1 Perbedaan *Network Based IDS* dan *Client Based IDS*

<i>Network Based IDS</i>	<i>Client Based IDS</i>
Ruang lingkup luas ( Mengamati semua aktifitas dalam jaringan)	Mengamati aktivitas pada <i>client</i> tertentu
Lebih handal mendeteksi serangan dari jaringan luar	Lebih handal mendeteksi serangan yang beraasal dari jaringan
Lebih murah untuk diimplementasikan	Lebih mahal untuk diimplementasikan
Deteksi berdasarkan pada aktivitas jaringan	Pengujian berdasarkan hanya <i>client</i> yang ditentukan
Menguji paket header	Tidak ada pengujian paket header
Response yang realtime	Mendeteksi serangan local
OS-Independent	OS-Spesifik

## 2.3 *Naive bayes*

Menurut [9] algoritma *naive bayes* merupakan algoritma yang menggunakan pendekatan statistik dalam mengambil keputusan. Algoritma *naive bayes* berdasarkan teorema *bayes* bahwa semua atribut memberikan kontribusi yang sama penting dan saling bebas pada kelas tertentu.

Walaupun teori ini bertentangan dengan kehidupan nyata bahwa atribut tidak sama penting atau independen, tetapi *naive bayes* menunjukkan performa yang mampu bersaing dengan algoritma klasifikasi yang terkenal, *decision tree* dan *neural network* [4].

Algoritma *naive bayes* menggunakan perhitungan probabilitas dalam menentukan kelas. *Naive bayes* diterapkan pada *machine learning* dimana masing-masing *instance* dideskripsikan oleh konjungsi nilai atribut dan dimana fungsi target dapat mengambil nilai apapun dari beberapa perangkat kelas  $C$ . Seperangkat *training example* fungsi target disediakan dan *instance* baru dihadirkan, dideskripsikan oleh tuple nilai atribut  $(a_1, a_2, \dots, a_n)$ . Algoritma ditugaskan untuk memprediksi nilai target, atau klasifikasi untuk *instance* baru ini [4].

Seperti yang dikatakan [2], formulasi *naive bayes* adalah sebagai berikut :

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^q P(X_i|Y)}{P(X)}$$

Dimana untuk formula diatas :

- $P(Y|X)$  adalah probabilitas data dengan *vector*  $X$  pada kelas  $Y$ .
- $P(Y)$  adalah probabilitas awal kelas  $Y$ .
- $\prod_{i=1}^q P(X_i|Y)$  adalah probabilitas independen kelas  $Y$  dari semua fitur dalam *vector*  $X$ .
- Nilai  $P(X)$  selalu tetap sehingga dalam perhitungan prediksi nantinya tinggal menghitung bagian  $P(Y) \prod_{i=1}^q P(X_i|Y)$  dengan memilih yang terbesar sebagai kelas yang dipilih sebagai hasil prediksi.

*Naive Bayes Classifier (NBC)* membutuhkan jumlah *record* data yang sangat besar untuk mendapatkan hasil yang baik. Jika kategori prediktor tidak ada dalam *data training*, maka *naive bayes classifier* mengasumsikan bahwa *record* baru dengan kategori *predictor* memiliki probabilitas nol.

## 2.4 Perhitungan Metode Naive Bayes

Perhitungan metode *naive bayes* berfungsi untuk melihat cara menyelesaikan suatu masalah menggunakan metode *naive bayes*. Pada penelitian ini permasalahan dibagi menjadi 3 klasifikasi serangan yaitu *DOS (Denial Of Service)*, *Normal* (serangan normal), *Probe (scanning)*. Tabel 2.2 adalah tabel *data training* dari 3 klasifikasi serangan.

Tabel 2.2 *Data Training* 3 Klasifikasi Serangan

No	src _bytes	dst _bytes	count	srv _count	dst_host _count	dst_host _srv_count	dst_host _same_src _port_rate	dst_host _srv_diff _host_rate	Tipe _serangan
1	54540	8314	1	1	251	251	0	0	DOS
2	53776	7300	3	3	70	70	0.01	0	DOS
3	233	1075	1	2	249	223	0	0	NORMAL
4	349	343	19	22	255	255	0	0	NORMAL
5	18	0	1	1	1	113	1	1	PROBE
6	8	0	1	35	2	52	1	0.5	PROBE

Pada tabel 2.2 adalah *data training* dari 3 klasifikasi serangan yang terdiri dari 2 serangan *dos*, 2 normal (bukan serangan), 2 serangan *probe*. Setelah *data training* dibuat selanjutnya menyiapkan *data testing* yang akan diklasifikasi, nantinya *data training* tersebut akan terklasifikasi apakah data tersebut masuk kedalam serangan *dos*, normal atau *probe*. Berikut tabel 2.3 adalah tabel *data testing*.

Tabel 2.3 *Data Testing*

No	src _bytes	dst _bytes	count	srv _count	dst_host _count	dst_host _srv_count	dst_host _same_src _port_rate	dst_host _srv_diff _host_rate	Tipe _serangan
1	54540	0	1	2	255	255	0	0	

Dari data tabel 2.3 dapat dilihat bahwa tipe serangan pada data tersebut masih belum terklasifikasi untuk itu berikut adalah tahap-tahap penyelesaian menggunakan metode *naive bayes*.

#### 2.4.1 Tahap 1 Menghitung Jumlah Dari Masing-Masing Klasifikasi

$$P(C_i) = \text{Count}(C_i), C_i = \text{Dos}, \text{Normal}, \text{Probe}$$

- Dos = Jumlah dos / Jumlah data  
= 2/6  
= 0,333

- Normal = Jumlah normal / Jumlah data  
=  $2/6$   
= 0,333
- Probe = Jumlah probe / Jumlah data  
=  $2/6$   
= 0,333

#### 2.4.2 Tahap 2 Menghitung Jumlah Data Yang Sama Antara Field Terhadap Masing-Masing Klasifikasi

$$P(X|C_i) = \text{Count}(X) / \text{Count}(C_i)$$

##### 1. Dos

- |                            |                            |
|----------------------------|----------------------------|
| • 54540 DOS = $0/2$<br>= 0 | • 255 DOS = $1/2$<br>= 0,5 |
| • 0 DOS = $0/2$<br>= 0     | • 255 DOS = $1/2$<br>= 0,5 |
| • 1 DOS = $1/2$<br>= 0,5   | • 0 DOS = $2/2$<br>= 1     |
| • 2 DOS = $1/2$<br>= 0,5   | • 0 DOS = $2/2$<br>= 1     |

##### 2. Normal

- |                                 |                             |
|---------------------------------|-----------------------------|
| • 54540 Normal = $1/2$<br>= 0,5 | • 255 Normal = $0/2$<br>= 0 |
| • 0 Normal = $0/2$<br>= 0       | • 255 Normal = $0/2$<br>= 0 |
| • 1 Normal = $1/2$<br>= 0,5     | • 0 Normal = $1/2$<br>= 0,5 |
| • 2 Normal = $0/2$<br>= 0       | • 0 Normal = $2/2$<br>= 1   |

## 3. Probe

- $54540|Probe = 0/2$   
 $= 0$
- $0|Probe = 0/2$   
 $= 0$
- $1|Probe = 2/2$   
 $= 1$
- $2|Probe = 2/2$   
 $= 1$
- $255|Probe = 0/2$   
 $= 0$
- $255|Probe = 0/2$   
 $= 0$
- $0|Probe = 0/2$   
 $= 0$
- $0|Probe = 0/2$   
 $= 0$

### 2.4.3 Tahap 3 Melakukan Operasi Perkalian Terhadap Hasil Pada Tahap 2 Dengan Mengabaikan Nilai 0

1. Dos  $= 0,5 \times 0,5 \times 0,5 \times 0,5 \times 1 \times 1$   
 $= 0,0625$
2. Normal  $= 0,5 \times 0,5 \times 0,5 \times 1$   
 $= 0,125$
3. Probe  $= 1 \times 1$   
 $= 1$

### 2.4.4 Tahap 4 Mencari Nilai Maksimal Dari Setiap Kemungkinan Klasifikasi

Nilai Max (Dos, Normal, Probe) = 1 (Probe)

Dari hasil nilai max menunjukkan bahwa data testing telah terklasifikasi yaitu data terklasifikasi ke dalam serangan probe. Berikut tabel 2.4 adalah hasil klasifikasi dari data testing.

Tabel 2.4 *Data Testing* Terklasifikasi

No	src _bytes	dst _bytes	count	srv _count	dst_host _count	dst_host _srv_count	dst_host _same_src _port_rate	dst_host _srv_diff _host_rate	Tipe _serangan
1	54540	0	1	2	255	255	0	0	PROBE



## 2.5 Mysql

*Mysql* adalah salah satu aplikasi manajemen *database* SQL . aplikasi ini bersifat gratis dan *opensource* , dalam *database mysql* terdapat beberapa penggunaan data dan semuanya dapat terorganisir. *Mysql* tidak hanya tersedia pada sistem operasi *unix* atau *linux*, *mysql* juga dapat digunakan diatas *platform windows*.

*Mysql* mempunyai beberapa fasilitas dan kelebihan dibandingkan jenis manajemen *database* lainnya, untuk itu peneliti menggunakan *Mysql* sebagai aplikasi penyimpanan data. Diantaranya kelebihan itu adalah :

- *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan sistem operasi lainnya.
- *MySQL* didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
- *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah.
- *MySQL* memiliki kecepatan yang sangat baik dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- *MySQL* memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
- Perintah dan Fungsi. *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
- *MySQL* memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
- *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
- *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol *TCP/IP, Unix socket (UNIX)*, atau *Named Pipes (NT)*.

- *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
- *MySQL* memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- *MySQL* dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
- *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.

## 2.6 PHP

PHP adalah sebuah bahasa pemrograman yang dapat berfungsi untuk membuat sebuah *website* ataupun aplikasi *web* yang sifatnya dinamis. Kelebihan dari PHP adalah dapat berinteraksi dengan *database* seperti *mysql*, sehingga perintah SQL dapat dijalankan melalui pemrograman PHP. PHP disebut juga sebagai bahasa *scripting* sehingga semua proses dijalankan pada sisi *server*.

Peneliti menggunakan bahasa pemrograman PHP karena pada penelitian ini hasil klasifikasi serangan akan ditampilkan melalui web sehingga bahasa pemrograman PHP adalah bahasa yang paling tepat untuk digunakan pada penelitian ini. Selain itu PHP juga dapat langsung berinteraksi dengan *mysql*, sehingga mempermudah dalam menyelesaikan penelitian ini.

## 2.7 Dataset NSL-KDD

Pada penelitian digunakan dataset NSL-KDD sebagai data penelitian yang memiliki data serangan yang lengkap baik data training dan data testing. NSL-KDD menghasilkan empat kategori serangan yang sering terjadi yaitu :

- *DoS (Denial-of-Service)* - serangan yang berusaha menggagalkan layanan

server), termasuk di dalamnya : *Apache2, arppoison, back, Crashiis, dosnuke, Land, Mailbomb, SYN Flood, Neptune, Ping of Death (POD), Process Table, selfping, Smuff*.

- *PROBING* (berusaha mencari kelemahan sistem yang ada), misal: *insidesniffer, Ipsweep, ls\_domain, Mscan, NTinfoScan, Nmap, queso, resetscan, Saint, Satan*.

- *R2L ( Remote To Local* - melakukan akses yang tidak bukan haknya dari jarak jauh) , termasuk dalam kategori ini : *Dictionary, Ftpwrite, Guest, Httptunnel, Imap, Named, ncftp, netbus, netcat, Phf, ppmacro, Sendmail, sshotrojan, Xlock, Xsnoop*.

- *U2R (User To Root* - melakukan akses yang bukan haknya ke *superuser* dari jaringan dalam), termasuk dalam kategori ini: *anypw, casesen, Eject, Ffbconfig, Fdformat, Loadmodule, ntfsdos, Perl, Ps, sechole, Xterm, yaga*.

*Denial of Service (DoS)* melakukan serangan dengan cara mencegah sistem menyediakan layanan ke *legitimate user*. Ketika berhasil sistem akan berhenti memberikan layanan atau hanya bisa menyediakan layanan yang terbatas. *DoS* mencapai tujuannya dengan cara membuat aplikasi *crash*, merusak data atau membuat *resource* menjadi cepat penuh. Hal ini bisa dilakukan dengan cara mengobservasi nilai – nilai yang tidak umum, misal: memanfaatkan *bug software*, menggunakan *bad checksum*, melakukan *spoofed addresses*, melakukan duplikasi paket *TCP* dengan payload berbeda, dsb.

*Probing* dilakukan untuk mencari kelemahan sistem yang ada. *Probing* sering dilakukan dengan cara melakukan *scanning* ke dalam sistem. Ciri dari serangan ini adalah melakukan koneksi ke sistem secara tidak penuh misal nmap dengan tipe *stealth scan* mengirim paket *TCP* tunggal tanpa *handshaking*.

## 2.8 Format Dataset KDD '99

KDDCUP '99 dataset ([kdd.ics.uci.edu](http://kdd.ics.uci.edu)) merupakan data hasil preprocessing yang berbasis pada data DARPA 1998 yang disediakan untuk perancangan sistem pendeteksian intrusi yang mana digunakan untuk melakukan evaluasi metodologi yang berbeda dari pendeteksian intrusi. Pada tahun 1999 dilakukan preprocessing pada data tcpdump ini untuk dimanfaatkan dalam

pendeteksian intrusi pada kegiatan “International Knowledge Discovery and Data Mining Tools Competition”. Atribut yang dihasilkan pada KDDCUP ’99 terdiri dari 41 atribut ditambah 1 untuk label. Berikut ini merupakan atribut yang terdapat pada data KDDCUP ’99 terlihat pada tabel 2.5 berikut :

Tabel 2.5 Atribut pada KDDCUP ’99

No	Nama Atribut	Tipe Data
1	duration	continuous
2	protocol_type	symbolic
3	service	symbolic
4	flag	symbolic
5	src_bytes	continuous
6	dst_bytes	continuous
7	land	continuous
8	wrong_fragment	continuous
9	urgent	continuous
10	hot	continuous
11	num_failed_logins	continuous
12	logged_in	continuous
13	num_compromised	continuous
14	root_shell	continuous
15	su_attempted	continuous
16	num_root	continuous
17	num_file_creations	continuous
18	num_shells	continuous
19	num_access_files	continuous
20	num_outbound_cmds	continuous
21	is_host_login	continuous
22	is_guest_login	continuous
23	count	continuous
24	srv_count	continuous
25	serror_rate	continuous
26	srv_serror_rate	continuous
27	rerror_rate	continuous
28	srv_rerror_rate	continuous
29	same_srv_rate	continuous
30	diff_srv_rate	continuous
31	srv_diff_host_rate	continuous
32	dst_host_count	continuous
33	dst_host_srv_count	continuous
34	dst_host_same_srv_rate	continuous
35	dst_host_diff_srv_rate	continuous

36	dst_host_same_src_port_rate	continuous
37	dst_host_srv_diff_host_rate	continuous
38	dst_host_serror_rate	continuous
39	dst_host_srv_serror_rate	continuous
40	dst_host_rerror_rate	continuous
41	dst_host_srv_rerror_rate	continuous

## 2.9 Perbandingan dengan Penelitian Terdahulu

Berikut data perbandingan metode *naive bayes* dan metode lain ditunjukkan pada tabel 2.6.

Tabel 2.6 Perbandingan Metode *Naive bayes* dengan Metode Lain

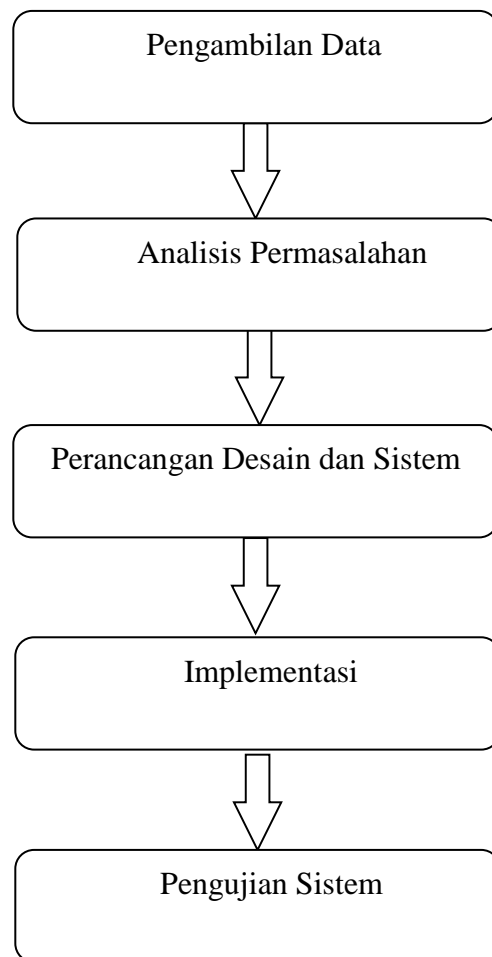
No	Kelebihan	Penelitian	Hasil	
			Naive bayes	KNN
1	Akurasi	Menurut [6], melakukan perbandingan akurasi metode klasifikasi <i>naive bayes</i> , <i>k-nearest neighbor</i>	Akurasi 98 %	Akurasi 96%
2	Kinerja	Menurut [1], melakukan perbandingan klasifikasi <i>naive bayes</i> dan <i>k-nearest neighbor</i> dalam pengklasifikasian dokumen teks	Kinerja lebih baik dari KNN	Kinerja tidak lebih baik dari <i>naive bayes</i>
3	Kecepatan	Menurut [10], analisa perbandingan algoritma <i>naive bayes</i> dan KNN untuk studi data “ <i>Wisconsin Diagnosis Breast Cancer</i> ”	Lebih cepat dari KNN	Tidak lebih cepat dari <i>naive bayes</i>

## BAB III. METODOLOGI PENELITIAN

### 3.1 Metodologi

Dalam pelaksanaan dan pengimplementasian sistem deteksi serangan menggunakan metode *naive bayes*, pengembangan aplikasi yang digunakan adalah *waterfall*.

Gambar 3.1 merupakan tahapan – tahapan yang dilakukan pada sistem deteksi serangan, terdapat 5 tahapan yang dilaksanakan untuk membangun sistem yaitu sebagai berikut.



Gambar 3.1 Diagram Pengembangan Aplikasi

### 3.1.1 Pengambilan Data

Pada metode ini, dilakukan pengumpulan data dan informasi dengan cara membaca referensi dari buku maupun dari internet, khususnya referensi-referensi yang terkait dengan IDS, serta melakukan wawancara kepada beberapa narasumber untuk menambah wawasan yang terkait dengan klasifikasi serangan pada IDS.

#### 3.1.1.1 Studi Literatur

Pada studi literatur ini digunakan untuk mengetahui klasifikasi serangan pada IDS dan metode-metode yang digunakan untuk klasifikasi serangan pada IDS. Berikut beberapa referensi yang digunakan :

1. *Network Intrusion Detection Using Naive Bayes* (Mrutyunjaya Pandadan Manas Ranjan Patra) metode *naive bayes* dibangun dengan pola layanan jaringan melalui kumpulan data yang diberi label, kemudian sistem mendeteksi serangan dari dataset yang ada. Sehingga metode *naive bayes* memiliki tingkat deteksi lebih tinggi, waktu proses yang lebih cepat serta biaya yang murah.
2. *Anomaly Detection pada Intrusion Detection System (IDS) Menggunakan Metode Bayesian Network* (Oktavia Ari Marlita, Adiwijaya, dan Angelina Prima Kurniati) dengan hasil Algoritma *Bayesian Network TAN Classifier* bisa diimplementasikan untuk *anomaly detection* pada IDS dengan performansi yang baik. Faktor yang mempengaruhi pembentukan model *TAN Classifier* yaitu jumlah *record*, jumlah atribut, dan jumlah *value* tiap atribut pada data *training* serta pengambilan sample untuk proporsi data normal dan data intrusi pada data *training* sangat berpengaruh terhadap pembentukan model *TAN Classifier*..

#### 3.1.1.2 Wawancara

Wawancara adalah salah satu cara yang digunakan untuk memperoleh data dan informasi mengenai klasifikasi serangan pada IDS. Wawancara dilakukan kepada bapak Akhmad Alimudin, S.Kom, M.Kom selaku narasumber yang pernah melakukan penelitian tentang klasifikasi serangan pada IDS. Berikut tabel 3.1 merupakan beberapa pertanyaan yang diajukan kepada narasumber.

Tabel 3.1 Pertanyaan Wawancara

No	Pertanyaan	Jawaban
1	Apa saja yang dibutuhkan untuk membangun sebuah sistem klasifikasi serangan pada IDS ?	<i>Data Raw KDDCUP '99</i> yang digunakan untuk data <i>training</i> dan data <i>testing</i> sistem klasifikasi, <i>Preprocessor</i> digunakan untuk menerapkan metode yang digunakan untuk klasifikasi
2	Metode apa yang bapak gunakan ?	<i>Demster-Shaper</i>
3	Apa saja yang dijadikan atribut pada sistem yang digunakan sebagai klasifikasi serangan ?	Ada 8 fieldd yang digunakan sebagai acuan yaitu : <i>src_bytes</i> , <i>dst_bytes</i> , <i>count</i> , <i>srv_count</i> , <i>dst_host_count</i> , <i>dst_host_srv_count</i> , <i>dst_host_same_src_port_rate</i> , <i>dst_host_srv_diff_host_rate</i> .
4	Bagaimana cara memperoleh data training dan testing ?	Dari raw data KDDCUP '99 yang kemudian dilakukan proses <i>preprocessor</i> yang berfungsi untuk mencari field-field yang diperlukan
5	Jenis serangan apa saja yang bapak teliti?	NORMAL, DOS, PROBE, R2L, U2R
6	Bagaimana hasil dari penelitian yang bapak lakukan ?	<ol style="list-style-type: none"> <li>1. <i>Dempster-Shafer</i> mampu mengolah dan menggabungkan beberapa sumber informasi yang berbeda menjadi suatu informasi yang lebih akurat</li> <li>2. <i>Dempster-Shafer</i> mampu mengintegrasikan beberapa metode klasifikasi serta meningkatkan akurasi dari klasifikasi KNN dan SVM</li> <li>3. Rata-rata waktu komputasi adalah 1,6 detik.</li> </ol>

Dari pertanyaan-pertanyaan pada tabel 3.1 dapat disimpulkan bahwa sistem klasifikasi serangan yang dibuat dengan menggunakan metode *dempster-shafer* mampu mengolah dan menggabungkan informasi lebih akurat serta memiliki rata-rata waktu komputasi yang cepat.



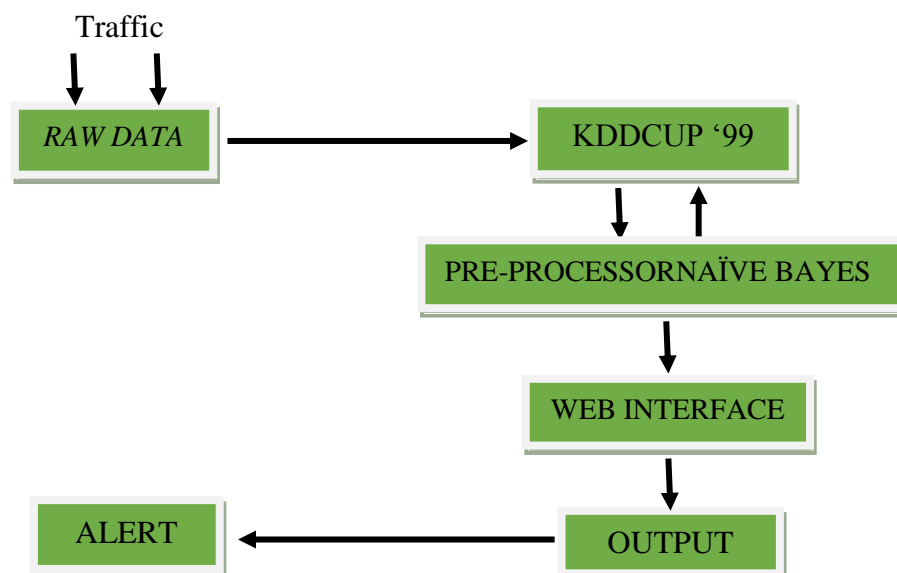
### 3.1.2 Analisis Permasalahan

Pada metode ini, dilakukan analisis terhadap masalah-masalah apa saja yang ada pada referensi tersebut dan berusaha untuk menyelesaikan permasalahan tersebut. Berikut beberapa permasalahan yang muncul :

1. Sistem klasifikasi membutuhkan data *training* dan *testing* yang diperoleh dari *raw data KDDCUP '99*
2. Sistem membutuhkan *preprocessor* yang digunakan untuk menerapkan metode dalam melakukan klasifikasi serangan
3. Atribut yang digunakan adalah *src\_bytes*, *dst\_bytes*, *count*, *srv\_count*, *dst\_host\_count*, *dst\_host\_srv\_count*, *dst\_host\_same\_src\_port\_rate*, dan *dst\_host\_srv\_diff\_host\_rate*.
4. Membutuhkan *web interface* yang digunakan untuk penyajian hasil dari klasifikasi serangan.

### 3.1.3 Perancangan Desain dan Sistem

Pada tahap ini dilakukan perancangan untuk pembuatan desain dan sistem klasifikasi serangan menggunakan metode *naive bayes*. Berikut desain sistem dari penelitian ini dapat dilihat pada gambar 3.2.



Gambar 3.2 Desain Sistem Klasifikasi Serangan Metode *Naive Bayes*

Pada gambar 3.2, *Raw data* diperoleh dari traffic atau aktifitas yang ada pada sebuah jaringan, kemudian *raw data* yang ada diubah menjadi format KDDCUP '99 agar dapat dilakukan proses *pre-processor*. Data yang telah diubah menjadi format KDDCUP '99 kemudian digunakan untuk data training pada *pre-processor naïve bayes*.

Hasil dari proses tersebut akan ditampilkan melalui *web interface* agar mudah dipahami oleh *user*. *Output* dari sistem berupa pemberitahuan atau *alert* yang menyatakan bahwa data *traffic* yang melewati sebuah jaringan tersebut merupakan serangan atau bukan serangan.

#### 3.1.4 Implementasi

Mengimplementasikan sistem klasifikasi serangan menggunakan metode *naive bayes* berdasarkan desain yang telah di buat. Data yang digunakan untuk training dan testing adalah data *traffic* jaringan yang telah diubah menjadi format KDDCUP '99, yang kemudian disimpan pada database untuk diolah menggunakan *php*.

Data yang telah tersedia kemudian akan diolah menggunakan metode *naive bayes* untuk melihat apakah data-data termasuk serangan atau bukan serangan berdasarkan klasifikasi yang telah dibuat. Hasil dari pemrosesan tersebut akan ditampilkan pada *web interface*.

#### 3.1.5 Pengujian Sistem

Menguji sistem klasifikasi serangan menggunakan metode *naive bayes* yang bertujuan untuk mengetahui apakah sistem yang dibuat dapat telah dapat mengklasifikasi serangan dan dapat membedakan jenis-jenis serangan.

Pengujian sistem dilakukan dengan cara melakukan *testing* terhadap metode dengan menginputkan data baru yang tidak terdapat dalam data klasifikasi sehingga hal ini bertujuan untuk melihat apakah metode yang digunakan dapat mengklasifikasikan serangan-serangan baru dengan acuan pola-pola serangan yang sudah ada sebelumnya.

## BAB IV. ANALISIS DAN PERANCANGAN

Bab ini membahas analisis sistem dan perancangan implementasi metode *naive bayes* untuk Intrusion Detection Sistem (IDS).

### 4.1 Pengambilan Data

Pada tahap ini melakukan pengumpulan data dan informasi dengan cara membaca referensi dari buku maupun dari internet, khususnya referensi-referensi yang terkait dengan IDS, serta melakukan wawancara kepada narasumber yang terkait untuk menambah wawasan tentang klasifikasi serangan pada IDS. Penelitian ini akan menggunakan data NSL-KDD sebagai *data testing* dan *data training*, data NSL-KDD dapat diperoleh di [nsl.cs.unb.ca/NSL-KDD/](http://nsl.cs.unb.ca/NSL-KDD/).

Dari tahap penilaian keadaan ini diperoleh beberapa hal mengenai latar belakang dibangunnya system deteksi pola serangan-serangan baru diantaranya yaitu :

1. Permasalahan yang terjadi saat sistem deteksi klasifikasi pola serangan baru belum dibuat :
  - Seorang *administrator* akan ketergantungan pada aplikasi yang berasal dari *vendor* untuk mengamankan jaringan yang dimiliki.
  - *Administrator* akan selalu menunggu update rule-rule atau aturan-aturan baru yang disediakan *vendor* sehingga hal ini kurang efektif dan efisien.
  - Ketika ada serangan baru yang tidak terdapat pada rule atau aturan yang disediakan hal ini sangat membahayakan bagi keamanan jaringan tersebut.
2. Alasan peneliti menggunakan metode *naive bayes* yaitu :
  - Penelitian terdahulu membuktikan bahwa *naive bayes* adalah metode yang paling efektif dan efisien untuk mesin *learning* dan data mining.
  - Metode *naive bayes* juga memiliki akurasi, kinerja dan kecepatan lebih baik dibandingkan metode KNN.

#### 4.1.1 Deskripsi Umum Sistem

Sistem deteksi pola serangan baru menggunakan metode *naive bayes* adalah sebuah sistem yang bertujuan mengklasifikasikan serangan-serangan baru yang muncul pada sebuah jaringan. Seorang *administrator* jaringan saat ini sangat ketergantungan dengan aplikasi yang disediakan *vendor*, faktanya *vendor* tidak melakukan *update* rule atau aturan setiap hari. Hal ini sangat membahayakan ketika ada serangan baru yang muncul dan serangan tersebut tidak ada dalam rule atau aturan sehingga dibuatlah sebuah sistem yang dapat menangani permasalahan tersebut.

Pada sistem ini data yang digunakan baik *data training* maupun *data testing* berasal dari NSL-KDD yang kemudian diubah ke dalam format KDDCUP '99. Dari format KDDCUP '99 di ambil 8 *field* yang dijadikan sebagai acuan klasifikasi yaitu *src\_bytes*, *dst\_bytes*, *count*, *srv\_count*, *dst\_host\_count*, *dst\_host\_srv\_count*, *dst\_host\_same\_src\_port\_rate*, *dst\_host\_srv\_diff\_host\_rate*.

#### 4.1.2 Data Required

Sistem deteksi serangan ini membutuhkan *data training* yang digunakan sebagai data acuan klasifikasi serta *data testing* yang digunakan sebagai data pengujian untuk melihat apakah sistem dapat mengklasifikasikan serangan-serangan baru. Data yang digunakan berasal dari NSL-KDD ([nsl.cs.unb.ca/NSL-KDD/](http://nsl.cs.unb.ca/NSL-KDD/)) yang kemudian diubah ke dalam format KDDCUP '99, dari format KDDCUP '99 di ambil 8 *field* yang dijadikan sebagai acuan klasifikasi. Berikut tabel 4.1 yang merupakan 8 field yang digunakan sebagai data dalam penelitian ini :

Tabel 4.1 Deskripsi Field yang dibutuhkan

No	Nama Field	Deskripsi
1	<i>src_bytes</i>	<i>bytes sent in one connection</i>
2	<i>dst_bytes</i>	<i>bytes received in one connection</i>
3	<i>Count</i>	<i>sum of connections to the same destination IP address</i>
4	<i>srv_count</i>	<i>sum of connections to the same destination port number</i>

5	<i>dst_host_count</i>	<i>sum of connections to the same destination IP address</i>
6	<i>dst_host_srv_count</i>	<i>sum of connections to the same destination port number</i>
7	<i>dst_host_same_src_port_rate</i>	<i>the percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count</i>
8	<i>dst_host_srv_diff_host_rate</i>	<i>the percentage of connections that were to different destination machines, among the connections aggregated in dst_host_srv_count</i>

#### 4.1.3 Analisis Kebutuhan Sistem

Dalam analisis kebutuhan sistem terdapat kebutuhan perangkat keras dan kebutuhan perangkat lunak yang digunakan untuk membangun aplikasi ini.

1. Kebutuhan perangkat keras laptop dengan spesifikasi :
  - Processor Intel Core i3 2310M 2,10GHz
  - Intel HD Graphics
  - 3GB DDR3 Memory
  - Hard Disk 500 Gb
  - Mouse Optik
2. Kebutuhan perangkat lunak.
  - Sistem Operasi *Windows 7 Ultimate*
  - *Microsoft Office 2007*
  - *Dia Diagram Editor*

#### 4.2 Analisis Permasalahan

Pada tahap ini dilakukan analisis permasalahan yang berasal dari pengambilan data baik studi literatur maupun wawancara kemudian permasalahan yang ada akan dijadikan landasan dan acuan dalam membangun sistem ini. Beberapa permasalahan yang telah diselesaikan yaitu :

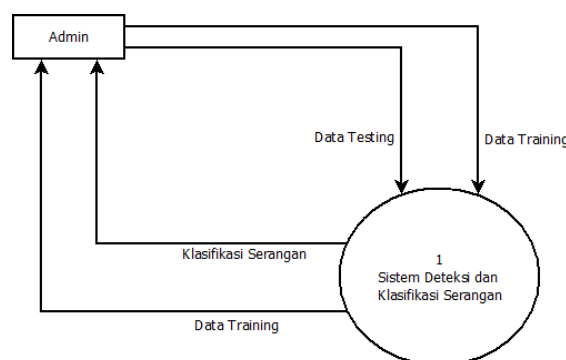
1. Sistem klasifikasi membutuhkan data *training* dan *testing* yang diperoleh dari *raw data KDDCUP '99*
2. Sistem membutuhkan *preprocessor* yang digunakan untuk menerapkan metode dalam melakukan klasifikasi serangan
3. Atribut yang digunakan adalah *src\_bytes*, *dst\_bytes*, *count*, *srv\_count*, *dst\_host\_count*, *dst\_host\_srv\_count*, *dst\_host\_same\_src\_port\_rate*, dan *dst\_host\_srv\_diff\_host\_rate*.
4. Membutuhkan *web interface* yang digunakan untuk penyajian hasil dari klasifikasi serangan.
5. Aplikasi dibuat untuk membantu administrator jaringan dalam mengamankan sistem jaringan terhadap serangan-serangan baru.

### 4.3 Perancangan Desain dan Sistem

Tahap perancangan desain dan sistem akan digambarkan dengan DFD (*Data Flow Diagram*), Rancangan *Database* dan *Flowchat*, untuk mempermudah membaca dan menerapkan sistem yang digunakan.

#### 4.3.1 DFD (*Data Flow Diagram*) Level 0

Membuat *Data Flow Diagram* yang berfungsi untuk memberikan informasi tentang hak akses yang diberikan kepada masing-masing pengguna baik *admin* maupun *user* biasa. *Admin* dapat mengubah *data training* dan menginputkan *data testing* serta dapat melihat hasil klasifikasi serangan. Berikut gambar 4.1 adalah *Data Flow Diagram* level 0 dari sistem deteksi dan klasifikasi serangan.

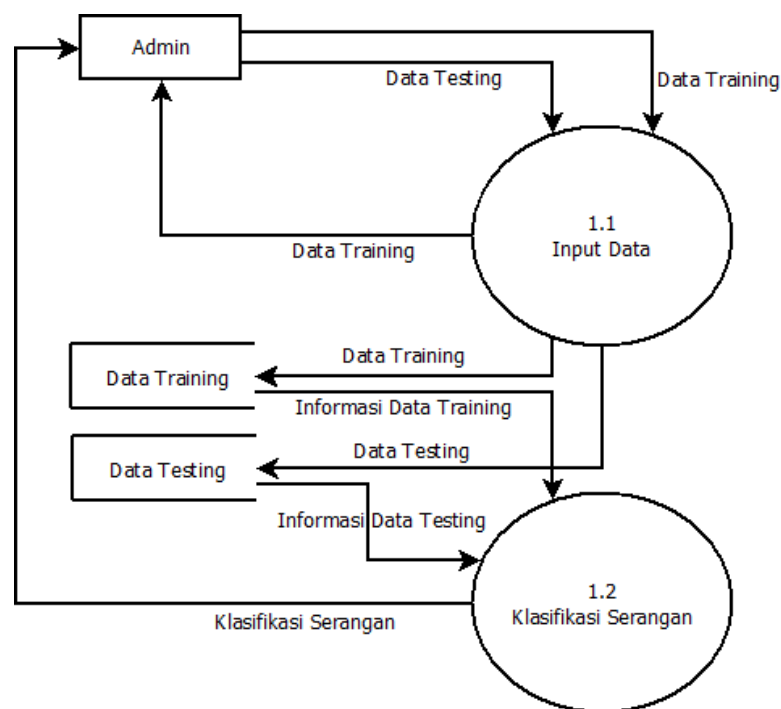


Gambar 4.1 DFD Level 0 sistem deteksi dan klasifikasi serangan

### 4.3.2 DFD (Data Flow Diagram) Level 1

*Data Flow Diagram* (DFD) level 1 akan memperlihatkan proses jalannya sistem secara lebih rinci, dari data diinputkan hingga seorang pengguna mendapatkan hasil yang diharapkan. Pada DFD level 1 hak akses *admin* akan menginputkan 3 data ke proses *input data* yaitu *data testing*, *user* dan *data training*. Kemudian hak akses *admin* akan mendapatkan 3 *output* yaitu *data user*, *data training* dan klasifikasi serangan.

DFD level 1 memiliki 2 proses yaitu proses *input data* yang berfungsi untuk menangani proses inputan data, serta proses klasifikasi data yang berfungsi mengolah data dari *data training* dan *data testing* menjadi data klasifikasi serangan yang dibutuhkan oleh pengguna. Berikut gambar 4.2 adalah *Data Flow Diagram* level 1 dari sistem deteksi dan klasifikasi serangan.

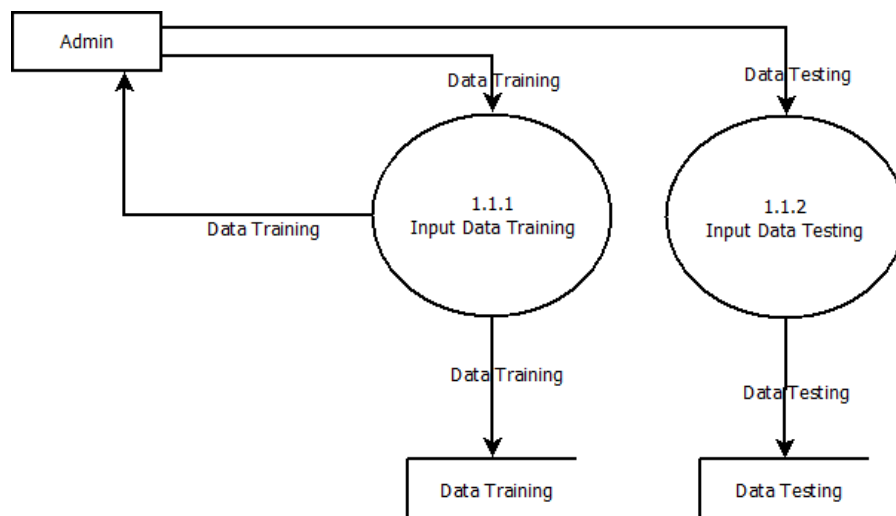


Gambar 4.2 DFD Level 1 sistem deteksi dan klasifikasi serangan

### 4.3.3 DFD (Data Flow Diagram) Level 2

*Data Flow Diagram* (DFD) level 2 akan memperlihatkan jalannya sistem secara lebih rinci dari proses *input data*. Ada dua proses dalam *input data* yaitu proses *input data training* dan proses *input data testing*. Dalam proses ini data diinputkan oleh *admin* sebagai pengelola sistem deteksi dan klasifikasi serangan. Berikut gambar 4.3 adalah *Data Flow Diagram* level 2 dari sistem deteksi dan klasifikasi serangan.

*Admin* dapat memodifikasi *data training* dengan cara menginputkan *data training* kemudian data inputan akan disimpan di *database data training*, *admin* juga dapat menginputkan *data testing* sebagai data yang akan dilakukan klasifikasi serangan kemudian inputan *data testing* akan disimpan di *database data testing*. *Admin* akan mendapatkan informasi mengenai *data training* yang ada, agar *admin* dapat melakukan pemantauan *data training* yang sesuai dengan kebutuhan klasifikasi serangan. Berikut gambar 4.3 adalah *Data Flow Diagram* dari sistem deteksi dan klasifikasi serangan.



Gambar 4.3 DFD Level 2 sistem deteksi dan klasifikasi serangan



#### 4.3.4 Rancangan Database

Rancangan *database* berfungsi untuk memberikan informasi tentang kebutuhan-kebutuhan apa saja yang terkait dengan pembuatan aplikasi. Pada penelitian ini ada 3 tabel yang digunakan oleh peneliti yaitu tabel data yang berfungsi menyimpan *data training*, *tabel testing* yang berfungsi menyimpan *data testing* serta tabel *user* yang berfungsi menyimpan informasi tentang user.

Pada *data training* dan *data testing* memiliki 10 *field* yang akan digunakan dalam proses klasifikasi serangan oleh sistem. 10 *field* tersebut yaitu *no\_id*, *src\_bytes*, *dst\_bytes*, *count*, *srv\_count*, *dst\_host\_count*, *dst\_host\_srv\_count*, *dst\_host\_same\_src\_port\_rate*, dan *dst\_host\_srv\_diff\_host\_rate*, *tipe\_serangan*. Berikut tabel 4.2 adalah tabel rancangan *database* dari *data training* dan *data testing* yang digunakan pada penellitian ini.

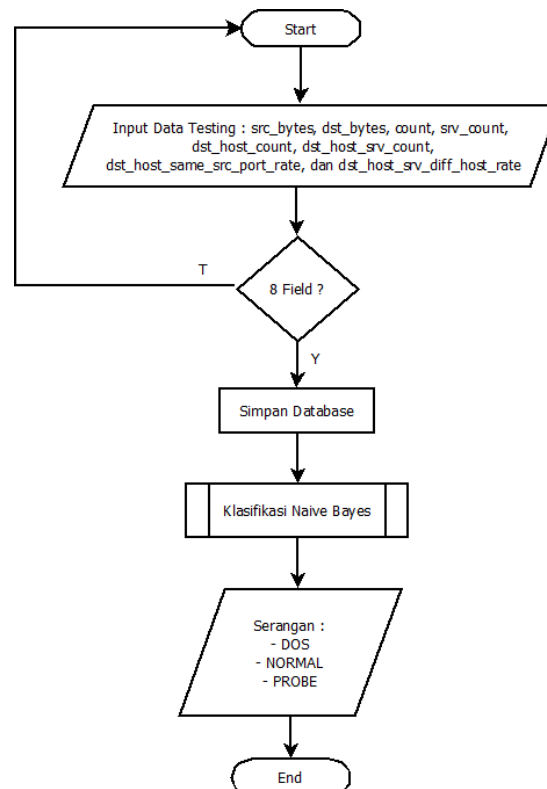
Tabel 4.2 Rancangan *Database* data training dan data testing

Nama Tabel	Jenis
<i>no_id</i>	Integer
<i>src_bytes</i>	Float
<i>dst_bytes</i>	Float
<i>Count</i>	Float
<i>srv_count</i>	Float
<i>dst_host_count</i>	Float
<i>dst_host_srv_count</i>	Float
<i>dst_host_same_src_port_rate</i>	Float
<i>dst_host_srv_diff_host_rate</i>	Float
<i>tipe_serangan</i>	Varchar

### 4.3.5 Flowchart

#### 4.3.5.1 Flowchart Sistem

Untuk mengetahui gambaran sistem maka dibuatlah sebuah *flowchart* dalam desain alur aplikasi, berikut gambar 4.4 menunjukkan *flowchart* sistem penelitian ini.

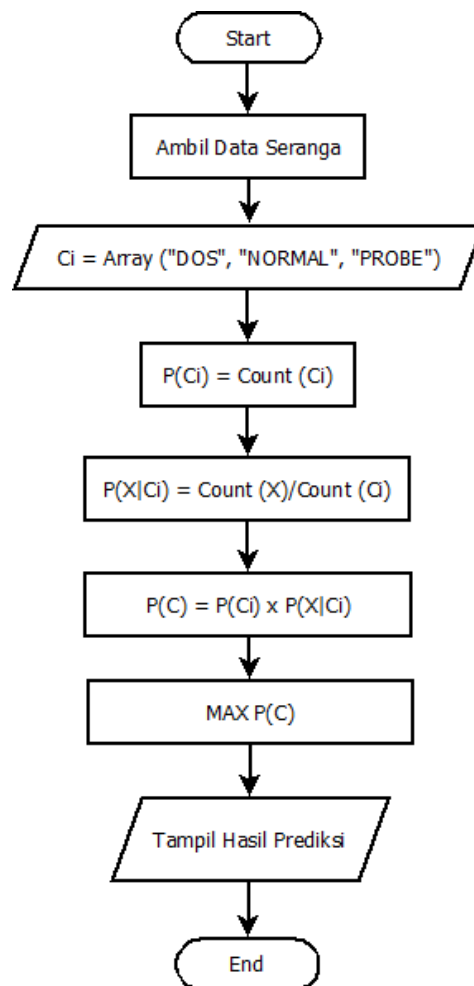


Gambar 4.4 *Flowchart* Sistem

Pada gambar 4.4 proses diawali dengan menginputkan *data testing* yang digunakan untuk data klasifikasi serangan, *data testing* berisi data jenis serangan yang belum terklasifikasi. *Data testing* akan diperiksa apakah sesuai dengan *field* data yang dibutuhkan oleh sistem yaitu 8 *field*, jika tidak cocok maka proses tidak dapat berlanjut dan pengguna dapat menginputkan data yang sesuai dengan kebutuhan sistem. Selanjutnya proses klasifikasi akan dilakukan dengan cara membandingkan pola-pola *data testing* dan pola-pola *data training* yang ada, setelah membandingkan pola-pola tersebut akan muncul pilihan klasifikasi apakah data akan masuk dalam klasifikasi DOS, NORMAL, atau PROBE.

#### 4.3.5.2 Flowchart Metode Naive Bayes

Untuk menggambarkan jalannya metode *naive bayes* pada sistem deteksi dan klasifikasi serangan maka dibuat sebuah *flowchart*. *Flowchart* menggambarkan proses dari input data serangan hingga didapatkan sebuah prediksi atau kemungkinan data tersebut masuk ke dalam klasifikasi yang telah dibuat sebelumnya. Berikut gambar 4.5 adalah flowchart metode *naive bayes*



Gambar 4.5 *Flowchart* metode *naive bayes*

Pada gambar 4.5 proses pertama adalah mengambil *data training* dari *database* yang tersedia, kemudian sistem akan menghitung *prior probability* ( $P(C_i)$ ) yaitu menghitung jumlah masing-masing klasifikasi yang ada pada *data training*, dimana  $C_i$  terdiri dari 3 klasifikasi yaitu serangan *dos*, *normal*, dan *probe*. Selanjutnya menghitung *probability atribut* serangan terhadap masing-

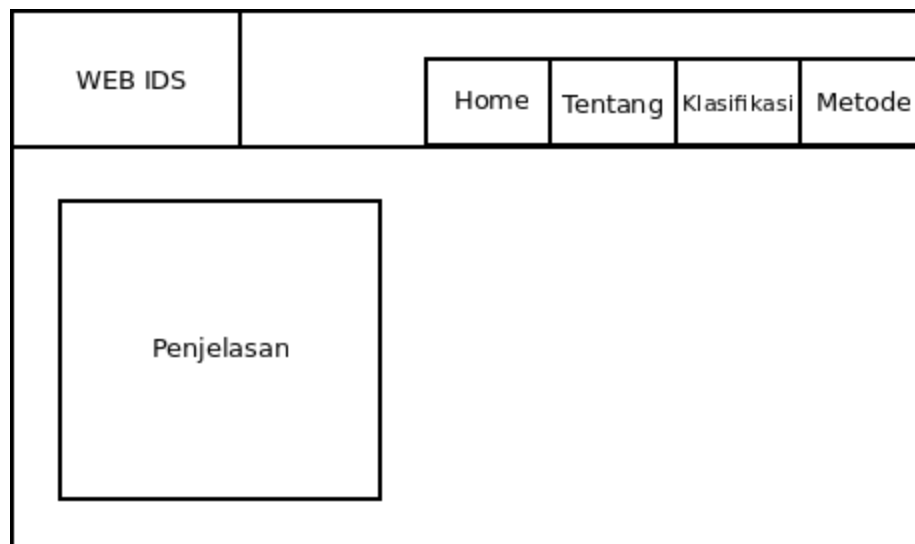
masing *class* ( $P(X|C_i)$ ) yaitu menghitung jumlah data yang sama dari setiap *class* atau *field* terhadap masing-masing klasifikasi. Setelah itu melakukan operasi perkalian antara prior *probability* ( $P(C_i)$ ) dengan *probability atribut* serangan terhadap masing-masing *class* ( $P(X|C_i)$ ). Hasil dari operasi perkalian akan dicari nilai tertinggi dari setiap kemungkinan klasifikasi, jenis klasifikasi yang memiliki nilai tertinggi akan dijadikan hasil prediksi dari data yang belum terklasifikasi tersebut.

#### 4.3.6 Perancangan Interface

Interface merupakan tampilan yang digunakan sebagai mediator interaksi antara sistem yang dibuat atau sebuah perangkat dengan pengguna (*user*). *Interface* sangat berpengaruh terhadap penggunaan aplikasi untuk memberikan kemudahan terhadap *user*. Dibawah ini adalah perancangan *interface* aplikasi deteksi pola serangan baru.

##### 4.3.6.1 Halaman Menu Utama

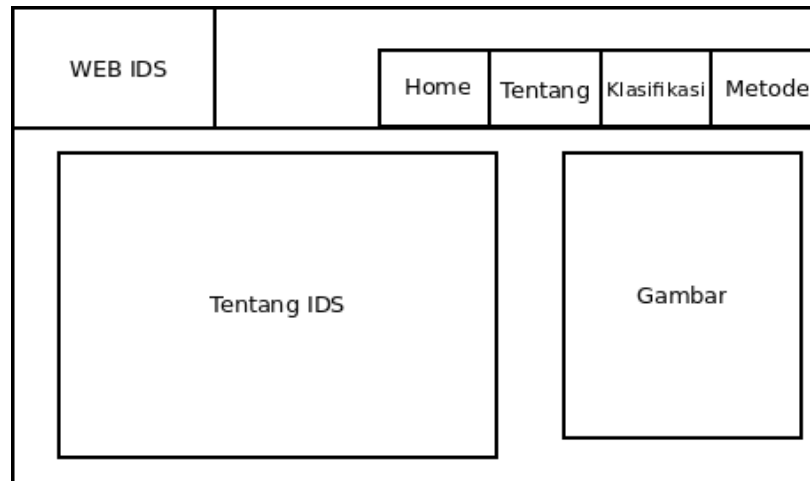
Halaman menu utama adalah halaman awal dari aplikasi yang dibuat. Halaman ini yang menampung menu-menu utama yang terdapat dalam aplikasi ini. Berikut gambar 4.6 adalah halaman utama *web* deteksi pola serangan baru.



Gambar 4.6 Halaman utama

#### 4.3.6.2 Halaman Tentang

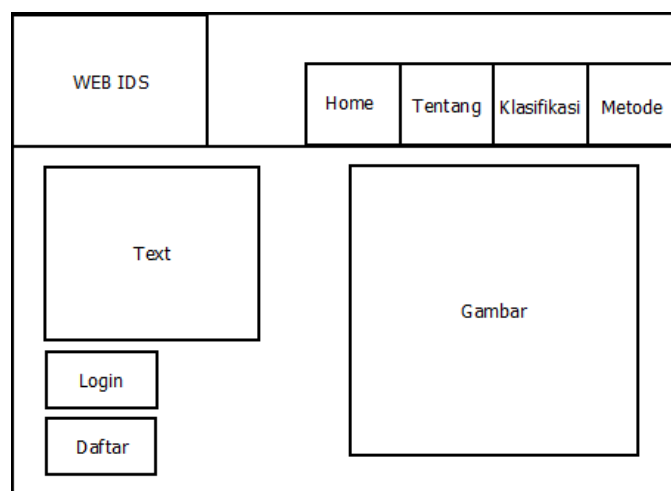
Halaman tentang adalah halaman yang menjelaskan apa itu Intrusion Detection System (IDS) dan yang semua yang terkait tentang IDS. Berikut gambar 4.7 adalah halaman tentang dari *web* deteksi pola serangan baru.



Gambar 4.7 Halaman tentang

#### 4.3.6.3 Halaman Klasifikasi

Halaman klasifikasi adalah halaman yang berfungsi untuk melakukan klasifikasi pada data baru yang akan diinputkan. Berikut gambar 4.8 adalah halaman klasifikasi dari *web* deteksi pola serangan baru.



Gambar 4.8 Halaman klasifikasi

#### 4.3.6.4 Halaman Form Login

Halaman login adalah halaman yang berfungsi untuk berpindah hak akses dari user biasa ke admin. Berikut gambar 4.9 adalah halaman login dari *web* deteksi pola serangan baru.

```

graph TD
    subgraph LoginForm [ ]
        direction TB
        Text1[Text]
        User[User]
        Password[Password]
        subgraph Buttons
            direction LR
            Login[Login]
            Close[Close]
        end
    end

```

Gambar 4.9 Halaman form *login*

#### 4.3.6.5 Halaman Metode

Halaman metode adalah halaman yang berisi tentang penjelasan metode yang digunakan pada penelitian ini. Berikut gambar 4.10 adalah halaman metode dari *web* deteksi pola serangan baru.

```

graph TD
    subgraph Header
        direction LR
        WEB_IDS[WEB IDS]
        subgraph Menu
            direction LR
            Home[Home]
            Tentang[Tentang]
            Klasifikasi[Klasifikasi]
            Metode[Metode]
        end
    end
    subgraph Content
        direction TB
        Text1[Text]
        subgraph Boxes
            direction LR
            Box1[Text]
            Box2[Text]
            Box3[Text]
        end
    end

```

Gambar 4.10 Halaman metode

#### 4.3.6.6 Halaman *Admin*

Halaman *admin* adalah halaman khusus yang hanya dapat diakses oleh *admin*. Berikut gambar 4.11 adalah gambar halaman *admin* dari *web* deteksi pola serangan baru.

WEB IDS		
Gambar	Title	Data
Data Training		
Import Data Training		
Import Data Testing		
Hasil		

Gambar 4.11 Halaman *admin*

#### 4.3.6.7 Halaman *Import Data*

Halaman import data adalah halaman yang berfungsi untuk menambahkan *data training* dan *data testing*. Berikut gambar 4.13 adalah halaman tambah data dari *data training*.

WEB IDS		
Gambar	Title	Text
Data Training		
Import Data Training		
Import Data Testing		
Hasil		
	Import	

Gambar 4.13 Halaman *import data*

#### 4.3.6.8 Halaman Hasil

Halaman hasil adalah halaman yang berfungsi untuk menampilkan hasil dari klasifikasi serangan. Berikut gambar 4.14 adalah halaman hasil.

WEB IDS		
Gambar	Title	
Data Training	Data	
Import Data Training		
Import Data Testing		
Hasil		

Gambar 4.14 Halaman *import data*

### 4.4 Implementasi

Pada tahap ini dilakukan implementasi sistem yang terdiri dari membuat *data training* dan *data testing*, menerapkan metode *naive bayes*, membuat *web interface*.

Sistem menggunakan data NSL-KDD sebagai data mentah, NSL-KDD telah menyediakan *data training* dan *data testing* untuk penelitian terkait dengan *Intrusion Detection System (IDS)*. *Raw data* yang diperoleh dari NSL-KDD kemudian diubah menjadi format KDDCUP '99 yaitu sebanyak 41 field atau fitur, dari 41 *field* tersebut dipilih 8 *field* yang akan digunakan sebagai acuan data training dan data testing. 8 *field* tersebut yaitu, *src\_bytes*, *dst\_bytes*, *count*, *srv\_count*, *dst\_host\_count*, *dst\_host\_srv\_count*, *dst\_host\_same\_src\_port\_rate*, dan *dst\_host\_srv\_diff\_host\_rate*.

Penelitian ini berfokus pada serangan Normal, Dos, dan Probe, peneliti menggunakan 1500, 3000, dan 5000 *data training* yang akan dijadikan acuan dalam proses klasifikasi data serangan baru. Sedangkan untuk *data testing* peneliti



menggunakan 100, 150, dan 200 data. Berikut rincian data training dan data testing dapat dilihat pada tabel 4.3

Tabel 4.3 Rincian data training dan data testing

No	Data	Jumlah Data	Rincian		
			Normal	Dos	Probe
1	Data Training	1500	500	500	500
		3000	500	1500	1000
		5000	700	2300	2000
2	Data Testing	100	9	51	40
		150	14	78	58
		200	24	90	86

Penelitian ini menggunakan metode *naive bayes* yang berfungsi untuk mengklasifikasikan pola serangan baru berdasarkan pola-pola serangan yang sudah ada pada data training. Metode *naive bayes* diterjemahkan menggunakan pemrograman *php* yang kemudian hasil klasifikasi tersebut akan ditampilkan melalui *web interface*.

#### 4.5 Pengujian Sistem

Menguji sistem klasifikasi serangan menggunakan metode *naive bayes* yang bertujuan untuk mengetahui apakah sistem yang dibuat dapat telah dapat mengklasifikasi serangan dan dapat membedakan jenis-jenis serangan.

Pengujian sistem dilakukan dengan cara melakukan *testing* terhadap metode dengan menginputkan data baru yang tidak terdapat dalam data klasifikasi sehingga hal ini bertujuan untuk melihat apakah metode yang digunakan dapat mengklasifikasikan serangan-serangan baru dengan acuan pola-pola serangan yang sudah ada sebelumnya.

## BAB V. IMPLEMENTASI SISTEM

Implementasi sistem merupakan proses pembuatan aplikasi berdasarkan analisis dan perancangan yang telah dilakukan sebelumnya. Implementasi sistem berisi uraian pembuatan interface dari sistem deteksi pola serangan baru menggunakan metode *naive bayes*.

### 5.1 Pembuatan Data Training dan Testing

Data adalah salah satu hal yang paling penting pada penelitian ini, data yang digunakan sebagai objek penelitian adalah data dari NSL-KDD. NSL-KDD telah menyediakan *data training* dan *data testing* sebagai bahan penelitian khusus tentang *IDS*. *Raw data* yang berasal dari NSL-KDD kemudian diubah menjadi format KDDCUP '99 41 fitur atau field. Dari 41 fitur tersebut akan di gunakan 8 field sebagai acuan dalam penelitian ini.

#### 5.1.1 Data Training

Pada penelitian ini data yang digunakan sebagai *data training* adalah sebanyak 1500, 3000 dan 5000 data. *Data training* digunakan sebagai klasifikasi serangan yang akan dijadikan bahan learning bagi sistem yang kemudian akan dijadikan acuan dalam proses klasifikasi serangan baru. Berikut gambar 5.1 adalah data training yang digunakan pada penelitian ini.

491	1480	0	7	7	91	7	0.08	0	DOS
492	1480	0	10	10	255	12	0.05	0	DOS
493	1480	0	1	3	2	154	0.5	0.02	DOS
494	1480	0	1	3	3	151	1	0.06	DOS
495	1480	0	2	4	2	22	1	0.5	DOS
496	1480	0	1	3	3	151	1	0.07	DOS
497	1480	0	1	3	1	27	1	0.52	DOS
498	1480	0	1	1	4	155	0.75	0.01	DOS
499	1480	0	1	1	3	5	0.33	0.6	DOS
500	1480	0	2	4	2	12	1	0.5	DOS
501	491	0	2	2	150	25	0.17	0	NORMAL
502	146	0	13	1	255	1	0.88	0	NORMAL
503	232	8153	5	5	30	255	0.03	0.04	NORMAL
504	199	420	30	32	255	255	0	0	NORMAL
505	287	2251	3	7	8	219	0.12	0.03	NORMAL
506	300	13788	8	9	91	255	0.01	0.02	NORMAL
507	233	616	3	3	66	255	0.02	0.03	NORMAL
508	343	1178	9	10	157	255	0.01	0.04	NORMAL
509	253	11905	8	10	87	255	0.01	0.02	NORMAL
510	147	105	1	1	255	1	1	0	NORMAL
511	437	14421	1	1	255	25	0	0	NORMAL
512	227	6588	5	22	43	255	0.02	0.14	NORMAL
513	215	10499	14	14	255	255	0	0	NORMAL

Gambar 5.1 Data Training

### 5.1.2 Data Testing

Pada penelitian ini data testing yang akan digunakan adalah sebanyak 100, 125, dan 150 data Berikut gambar 5.2 adalah *data testing* yang digunakan pada penelitian ini.

no	src_bytes	dst_bytes	count	srv_count	dst_host_	dst_host_	dst_host_	dst_host_	tipe_serangan
1	76944	1	12	12	241	238	0	0	
2	72564	0	11	11	255	237	0	0	
3	69644	0	12	12	255	237	0	0	
4	0	0	110	110	255	255	0	0	
5	0	0	118	118	255	255	0	0	
6	74024	0	10	10	255	244	0	0	
7	0	0	34	34	255	255	0	0	
8	29824	0	44	44	255	253	0	0	
9	54540	8314	4	4	255	254	0	0	
10	54540	8314	3	4	255	255	0	0	
11	54540	8314	4	10	255	254	0	0	
12	54540	8314	3	10	255	251	0	0	
13	33580	2920	4	4	255	254	0	0	
14	45260	2920	3	3	255	254	0	0	
15	33580	7300	3	3	255	254	0	0	
16	8	0	1	14	2	98	1	0.5	
17	18	0	1	1	1	50	1	1	
18	8	0	1	34	2	51	1	0.51	
19	18	0	1	1	1	211	1	1	
20	8	0	1	24	2	8	1	0.5	

Gambar 5.2 Data Testing

## 5.2 Pembuatan Database

Database dari sistem ini dibuat untuk menyimpan data-data yang diperlukan dan juga berguna untuk mempercepat proses pemanggilan data yang dibutuhkan oleh aplikasi. Pada penelitian ini menggunakan *PHP Myadmin* sebagai tempat penyimpanan data.

### 5.2.1 Tabel Data Training

Pada gambar 5.3 adalah implementasi tabel *data training* yang digunakan untuk menyimpan *data training*.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
<input type="checkbox"/>	1 no_id 🗨	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/>	2 src_bytes	float			Tidak	Tidak ada		
<input type="checkbox"/>	3 dst_bytes	float			Tidak	Tidak ada		
<input type="checkbox"/>	4 count	float			Tidak	Tidak ada		
<input type="checkbox"/>	5 srv_count	float			Tidak	Tidak ada		
<input type="checkbox"/>	6 dst_host_count	float			Tidak	Tidak ada		
<input type="checkbox"/>	7 dst_host_srv_count	float			Tidak	Tidak ada		
<input type="checkbox"/>	8 dst_host_same_src_port_rate	float			Tidak	Tidak ada		
<input type="checkbox"/>	9 dst_host_srv_diff_host_rate	float			Tidak	Tidak ada		
<input type="checkbox"/>	10 tipe_serangan	varchar(1000) latin1_swedish_ci			Tidak	Tidak ada		

Gambar 5.3 Tabel Data Training

### 5.2.2 Tabel Data Testing

Pada gambar 5.4 adalah implementasi tabel *data testing* yang digunakan untuk menyimpan *data testing*.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
1	no_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
2	src_bytes	float			Tidak	Tidak ada		
3	dst_bytes	float			Tidak	Tidak ada		
4	count	float			Tidak	Tidak ada		
5	srv_count	float			Tidak	Tidak ada		
6	dst_host_count	float			Tidak	Tidak ada		
7	dst_host_srv_count	float			Tidak	Tidak ada		
8	dst_host_same_src_port_rate	float			Tidak	Tidak ada		
9	dst_host_srv_diff_host_rate	float			Tidak	Tidak ada		
10	tipe_serangan	varchar(1000)	latin1_swedish_ci		Tidak	Tidak ada		

Gambar 5.4 Tabel *Data Testing*

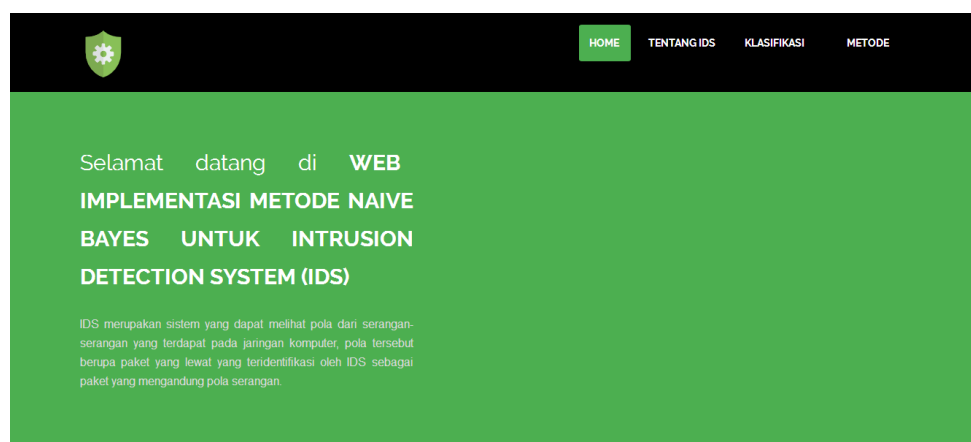
## 5.3 Interface Aplikasi Intrusion Detection System (IDS)

Aplikasi *Intrusion Detection System* (IDS) adalah aplikasi yang berfungsi melakukan klasifikasi serangan-serangan baru berdasarkan pola-pola serangan yang telah ada pada *data training*.

Dibawah ini adalah interface dari aplikasi *Intrusion Detection System* (IDS) menggunakan metode *naive bayes*.

### 5.3.1 Halaman Utama

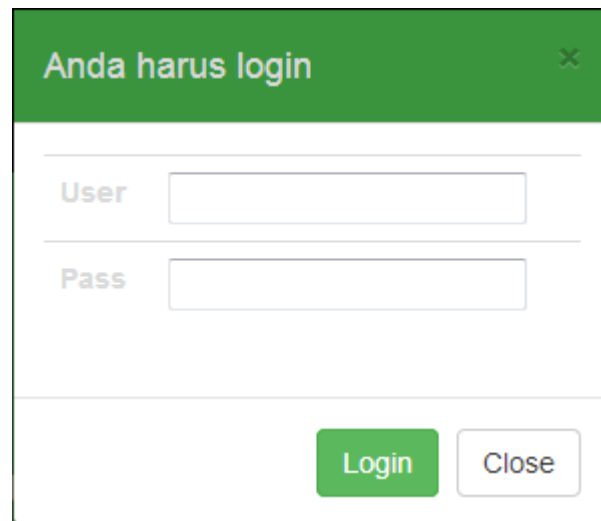
Halaman utama adalah halaman awal dari aplikasi yang berisi penjelasan singkat tentang penelitian. Berikut gambar 5.5 adalah gambar halaman utama



Gambar 5.5 Halaman Utama

### 5.3.2 Halaman *Login*

Setelah masuk ke dalam sistem pilih menu klasifikasi dan klik halaman *login*, admin akan menginputkan username dan password agar dapat mengklasifikasikan serangan baru. Berikut gambar 5.6 adalah gambar *login*.

The image shows a login dialog box with a green header bar containing the text "Anda harus login" and a close button (X). Below the header, there are two input fields: one labeled "User" and another labeled "Pass". At the bottom of the dialog, there are two buttons: a green "Login" button and a white "Close" button with a green border.

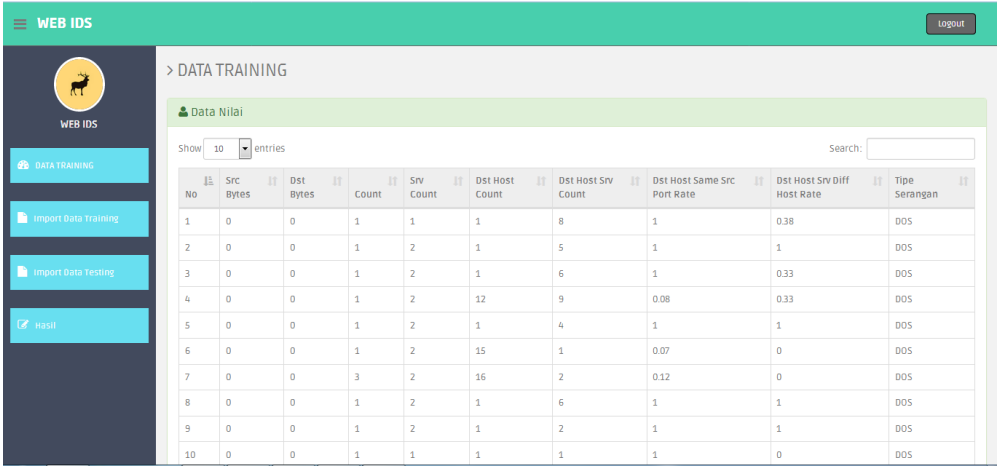
Gambar 5.6 Halaman *Login*

### 5.3.4 Halaman Klasifikasi

Pada halaman *admin* terdapat 4 menu yaitu menu *data training*, menu import *data training*, menu import *data testing*, menu hasil.

Menu *data training* yang berfungsi untuk melihat *data training* yang ada, menu *import data training* berfungsi untuk memodifikasi *data training* baik mengubah ataupun menambah data, menu *import data testing* berfungsi untuk menginputkan *data testing* yang nantinya data tersebut akan diklasifikasi, menu hasil berfungsi untuk melihat hasil dari klasifikasi *data testing* dan melihat apakah data termasuk serangan atau bukan.

Terdapat juga menu *logout* yang berfungsi untuk keluar dari sistem klasifikasi deteksi serangan-serangan baru. Berikut gambar 5.7 adalah halaman klasifikasi *admin*.



The screenshot shows the 'DATA TRAINING' section of the WEB IDS application. It features a sidebar with navigation links: DATA TRAINING, Import Data Training, Import Data Testing, and Hasil. The main content area displays a table titled 'Data Nilai' with 10 entries. The table has columns for No, Src Bytes, Dst Bytes, Count, Srv Count, Dst Host Count, Dst Host Srv Count, Dst Host Same Src Port Rate, Dst Host Srv Diff Host Rate, and Tipe Serangan. All 'Tipe Serangan' values are 'DOS'.

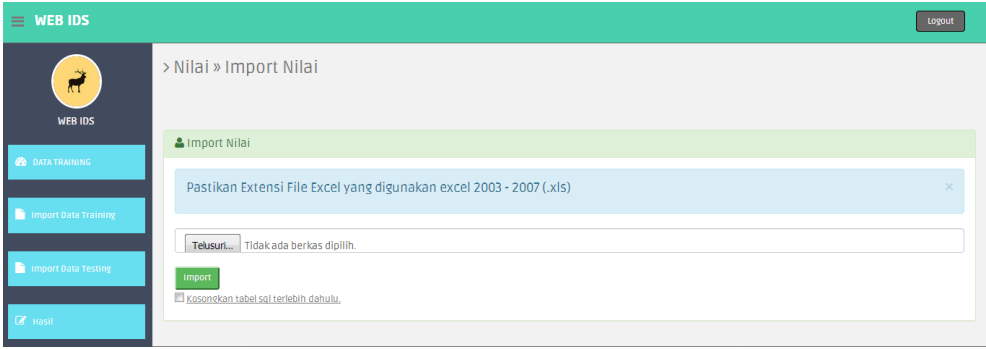
No	Src Bytes	Dst Bytes	Count	Srv Count	Dst Host Count	Dst Host Srv Count	Dst Host Same Src Port Rate	Dst Host Srv Diff Host Rate	Tipe Serangan
1	0	0	1	1	1	8	1	0.38	DOS
2	0	0	1	2	1	5	1	1	DOS
3	0	0	1	2	1	6	1	0.33	DOS
4	0	0	1	2	12	9	0.08	0.33	DOS
5	0	0	1	2	1	4	1	1	DOS
6	0	0	1	2	15	1	0.07	0	DOS
7	0	0	3	2	16	2	0.12	0	DOS
8	0	0	1	2	1	6	1	1	DOS
9	0	0	1	2	1	2	1	1	DOS
10	0	0	1	1	1	1	1	0	DOS

Gambar 5.7 Halaman Klasifikasi

### 5.3.4.1 Halaman *Import Data Training*

Halaman *import data training admin* adalah halaman yang berfungsi untuk menginputkan *data training* yang akan digunakan acuan dalam klasifikasi. Pada halaman ini kita dapat memilih *file* yang ada pada komputer, namun pada halaman ini hanya dapat menginputkan data yang berformat *.xls* (microsoft excel 2003-2007).

Saat ingin melakukan import kita juga dapat mengosongkan data terlebih dahulu atau tidak, jika tidak maka data akan langsung menambahkan data baru tanpa menghapusnya. Berikut gambar 5.8 adalah halaman *import data training admin*



The screenshot shows the 'Import Nilai' page in the WEB IDS application. It includes a sidebar with navigation links: DATA TRAINING, Import Data Training, Import Data Testing, and Hasil. The main content area has a heading 'Nilai » Import Nilai' and a section titled 'Import Nilai'. A message box states: 'Pastikan Extensi File Excel yang digunakan excel 2003 - 2007 (.xls)'. Below this is a file selection area with a 'Telusuri...' button and the text 'Tidak ada berkas dipilih.'. There is an 'import' button and a checkbox labeled 'Kosongkan tabel sql terlebih dahulu.'.

Gambar 5.8 Halaman *Import Data Training*

### 5.3.4.2 Halaman *Import Data Testing*

Halaman *import data testing admin* adalah halaman yang berfungsi untuk menginputkan data testing kemudian akan dilakukan klasifikasi serangan oleh sistem. Pada halaman ini kita dapat memilih *file* yang ada pada komputer, namun pada halaman ini hanya dapat menginputkan data yang berformat *.xls* (microsoft excel 2003-2007). Berikut gambar 5.9 adalah halaman *import data testing admin*



Gambar 5.9 Halaman *Import Data Testing*

### 5.3.4.3 Halaman Hasil

Halaman hasil *admin* adalah halaman yang berfungsi untuk menampilkan hasil dari klasifikasi serangan. Pada halaman hasil tipe serangan telah terisi hal ini menjelaskan bahwa data telah terklasifikasi berdasarkan *data training* yang ada. Berikut gambar 5.10 adalah halaman hasil *admin*

No	Src Bytes	Dst Bytes	Count	Srv Count	Dst Host Count	Dst Host Srv Count	Dst Host Same Src Port Rate	Dst Host Srv Diff Host Rate	Tipe Serangan
1	76964	1	12	12	241	238	0	0	DOS
2	72564	0	11	11	255	237	0	0	DOS
3	69644	0	12	12	255	237	0	0	DOS
4	0	0	110	110	255	255	0	0	PROBE
5	0	0	118	118	255	255	0	0	DOS
6	74024	0	10	10	255	244	0	0	DOS
7	0	0	34	34	255	255	0	0	DOS
8	29824	0	44	44	255	253	0	0	DOS
9	54540	8314	4	4	255	254	0	0	DOS
10	54540	8314	3	4	255	255	0	0	NORMAL

Gambar 5.10 Halaman Hasil

## BAB VI. PENGUJIAN DAN PEMBAHASAN

### 6.1 Pengujian Sistem

Uji coba perlu dilakukan untuk mengukur tingkat keberhasilan sistem aplikasi yang telah dibuat. Parameter yang dijadikan acuan pengukuran keberhasilan adalah pengujian sistem, pengujian akurasi, dan pembahasan.

#### 6.1.1 Pengujian *Black Box*

##### 6.1.1.1 Uji Coba *Login*

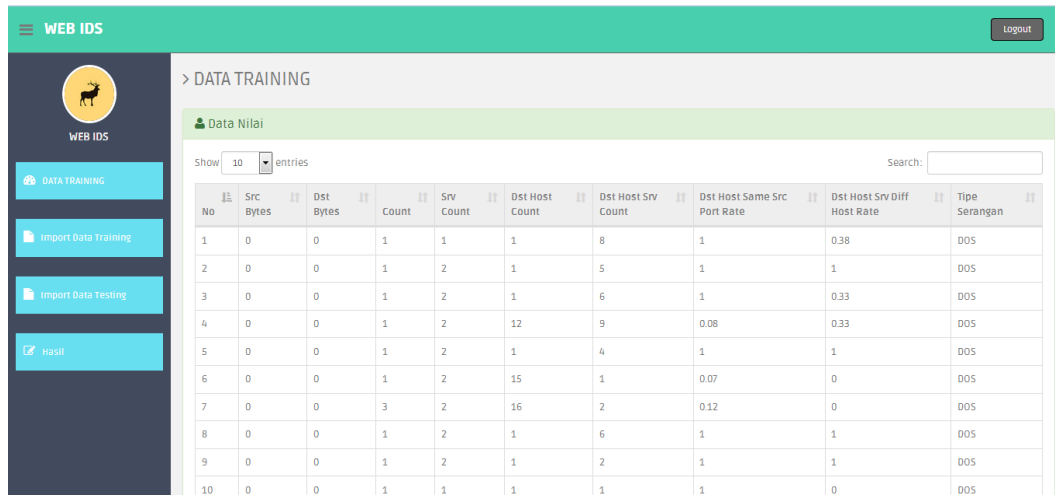
Fungsi dari tombol *login* adalah memeriksa jika *username* dan *password* inputan sama dengan yang ada pada sistem maka *admin* akan masuk kedalam halaman utama dari aplikasi klasifikasi. Sedangkan jika *username* dan *password* tidak sesuai dengan apa yang ada pada sistem maka *admin* tidak dapat masuk kedalam halaman utama dari aplikasi klasifikasi dan akan muncul pemberitahuan bahwa *username* dan *password* salah. Serta jika *username* dan *password* yang diminta tidak diisi atau dikosongkan maka *admin* juga tidak dapat masuk ke dalam sistem aplikasi klasifikasi. Skenario pengujian dapat dilihat pada tabel 6.1

Tabel 6.1 Skenario Uji Coba *Login*

No	Proses	Hasil	Keterangan
1	<i>Input username : admin</i> <i>Input password : admin</i>	<i>Admin</i> dapat masuk ke dalam sistem	Sukses
2	<i>Input username : 123</i> <i>Input password : 123</i>	<i>Admin</i> tidak dapat masuk ke dalam sistem	Sukses
3	<i>Input username : (kosong)</i> <i>Input password : (kosong)</i>	<i>Admin</i> tidak dapat masuk ke dalam sistem	Sukses



Apabila *admin* menginputkan *username* dan *password* yang benar sesuai dengan sistem maka *admin* akan diarahkan ke halaman utama dari aplikasi. Berikut gambar 6.1 adalah gambar *login admin* berhasil.



The screenshot shows the WEB IDS dashboard. The left sidebar contains a logo and navigation links: DATA TRAINING, Import Data Training, Import Data Testing, and Hasil. The main content area is titled > DATA TRAINING and shows a 'Data Nilai' section with a table of 10 entries. The table has columns for No, Src Bytes, Dst Bytes, Count, Srv Count, Dst Host Count, Dst Host Srv Count, Dst Host Same Src Port Rate, Dst Host Srv Diff Host Rate, and Tipe Serangan.

No	Src Bytes	Dst Bytes	Count	Srv Count	Dst Host Count	Dst Host Srv Count	Dst Host Same Src Port Rate	Dst Host Srv Diff Host Rate	Tipe Serangan
1	0	0	1	1	1	8	1	0.38	DOS
2	0	0	1	2	1	5	1	1	DOS
3	0	0	1	2	1	6	1	0.33	DOS
4	0	0	1	2	12	9	0.08	0.33	DOS
5	0	0	1	2	1	4	1	1	DOS
6	0	0	1	2	15	1	0.07	0	DOS
7	0	0	3	2	16	2	0.12	0	DOS
8	0	0	1	2	1	6	1	1	DOS
9	0	0	1	2	1	2	1	1	DOS
10	0	0	1	1	1	1	1	0	DOS

Gambar 6.1 gambar *login admin* berhasil

Ketika *admin* memasukkan *username* dan *password* yang tidak sesuai dengan apa yang ada pada sistem maka aplikasi akan mengeluarkan peringatan yaitu PERHATIAN! User ID atau Password Salah. Hanya User yang terdaftar yang dapat mengakses layanan. Berikut gambar 6.2 adalah login admin tidak berhasil.



Gambar 6.2 gambar *login admin* tidak berhasil

Berikut adalah *source code* yang digunakan pada sistem saat login.

**user-login.php**

```
<?php
if (isset($_POST['text_name']) && isset($_POST['pass'])) {

    $valid = 0;
    $name = $_POST['text_name'];
    $pass = $_POST['pass'];

    if ($name=="admin" && $pass=="admin") {
        header('location:dashboard.php');
    }else{

        header ('location:index.php?valid=0');
    }
}

?>
```

#### 6.1.1.2 Uji Coba Lihat *Data Training*

Menu *data training* pada aplikasi berfungsi untuk melihat *data training* yang ada pada *database*, ketika data yang diinputkan pada *database* benar maka data tersebut akan ditampilkan pada menu *data training*. Skenario pengujian pada menu *data training* dapat dilihat pada tabel 6.2

Tabel 6.2 Skenario Uji Coba Menu *Data Training*

No	Proses	Hasil	Keterangan
1	Melakukan cek jumlah data di aplikasi dan di <i>database</i>	Jumlah data yang ada dan ditampilkan sama	Sukses

Uji coba pada menu *data training* akan dilakukan dengan cara melihat apakah jumlah data yang ditampilkan pada aplikasi sama dengan jumlah data yang ada pada *database*. Berikut gambar 6.3 adalah gambar uji coba menu *data training*.

**Data Nilai**

Show  entries Search:

No	Src Bytes	Dst Bytes	Count	Srv Count	Dst Host Count	Dst Host Srv Count	Dst Host Same Src Port Rate	Dst Host Srv Diff Host Rate	Tipe Serangan
1	0	0	1	1	1	8	1	0.38	DOS
2	0	0	1	2	1	5	1	1	DOS
3	0	0	1	2	1	6	1	0.33	DOS
4	0	0	1	2	12	9	0.08	0.33	DOS
5	0	0	1	2	1	4	1	1	DOS
6	0	0	1	2	15	1	0.07	0	DOS
7	0	0	3	2	16	2	0.12	0	DOS
8	0	0	1	2	1	6	1	1	DOS
9	0	0	1	2	1	2	1	1	DOS
10	0	0	1	1	1	1	1	0	DOS

Showing 1 to 10 of 1,500 entries

Previous **1** 2 3 4 5 ... 150 Next

tb\_data ★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus 1,500 InnoDB latin1\_swedish\_ci 128 KB

Gambar 6.3 gambar uji coba menu *data training*

Berikut *source code* yang digunakan pada menu *data training*.

#### ajax-data.php

```
$sql = "SELECT no_id, src_bytes, dst_bytes, count, srv_count,
dst_host_count,      dst_host_srv_count,dst_host_same_src_port_rate,
dst_host_srv_diff_host_rate, tipe_serangan";
    $sql.=" FROM tb_data";
    $query=mysqli_query($conn, $sql) or die("ajax-data.php: get
Nilai");
    $totalData = mysqli_num_rows($query);
    $totalFiltered = $totalData;
    if( !empty($requestData['search']['value']) ) {
        $sql = "SELECT no_id, src_bytes, dst_bytes, count, srv_count,
dst_host_count,      dst_host_srv_count,      dst_host_same_src_port_rate,
dst_host_srv_diff_host_rate, tipe_serangan";
        $sql.=" FROM tb_data";
        $sql.="
                                WHERE                                no_id                                LIKE
'".$requestData['search']['value']."%' ";
        //
        $requestData['search']['value'] contains search parameter
        $sql.=" OR src_bytes LIKE '".$requestData['search']['value']."%' ";
        $sql.=" OR dst_bytes LIKE '".$requestData['search']['value']."%' ";
        $sql.=" OR count LIKE '".$requestData['search']['value']."%' ";
        $sql.="
                                OR                                dst_host_count                                LIKE
'".$requestData['search']['value']."%' ";
```

```

    $sql.="          OR          dst_host_count          LIKE
    '". $requestData['search']['value']. "%' ";
    $sql.="          OR          dst_host_srv_count        LIKE
    '". $requestData['search']['value']. "%' ";
    $sql.="          OR          dst_host_same_src_port_rate  LIKE
    '". $requestData['search']['value']. "%' ";
    $sql.="          OR          dst_host_srv_diff_host_rate  LIKE
    '". $requestData['search']['value']. "%' ";
    $sql.="          OR          tipe_serangan              LIKE
    '". $requestData['search']['value']. "%' ";
    $query=mysqli_query($conn, $sql) or die("ajax-data.php: get
    Siswa");
    $totalFiltered = mysqli_num_rows($query);

```

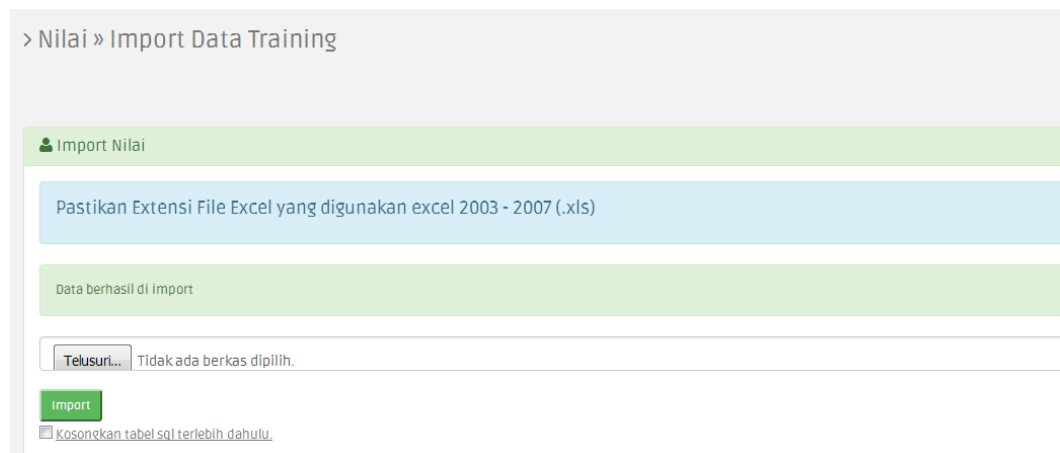
#### 6.1.1.3 Uji Coba *Import Data Training*

Menu *import data training* pada aplikasi berfungsi untuk memodifikasi *data training* baik itu menambah *data training* maupun mengubah *data training*. Ketika *data training* yang akan diimport sesuai dengan format yaitu (.xls) maka data tersebut akan disimpan pada *database* dan akan dijadikan data acuan dari klasifikasi. Skenario pengujian pada menu *import data training* dapat dilihat pada tabel 6.3

Tabel 6.3 Skenario Uji Coba Menu *Import Data Training*

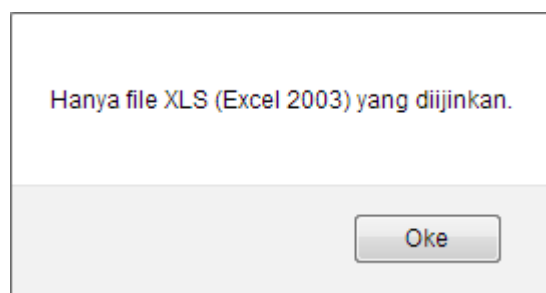
No	Proses	Hasil	Keterangan
1	Import data training sesuai dengan format (.xls)	Data berhasil diimport	Sukses
2	Import data training tidak sesuai dengan format	Hanya file XLS (Excel 2003) yang diijinkan	Sukses

Apabila *admin* melakukan *import data training* sesuai dengan format yang diinginkan maka pada sistem akan tertulis bahwa data berhasil diimport dan data tersebut akan disimpan pada *database data training* yang nantinya akan dijadikan sebagai acuan dalam klasifikasi. Berikut gambar 6.4 adalah gambar data berhasil diimport.



Gambar 6.4 gambar data berhasil diimport pada menu *import data training*

Ketika *admin* melakukan import data yang tidak sesuai dengan format yang diinginkan sistem maka akan muncul notifikasi bahwa hanya file XLS (excel 2003) yang diijinkan. Sehingga *admin* harus melakukan import ulang data yang berformat excel 2003 (.xls). berikut gambar 6.5 adalah gambar data yang tidak sesuai dengan format.



Gambar 6.5 gambar data tidak sesuai dengan format

Berikut *source code* yang digunakan pada menu *import data training*.

### nilai\_import.php

```
<div class="row mt">

    <div class="col-lg-12">
        <div class="panel panel-success">
            <div class="panel-heading">
                <h3 class="panel-title"><i class="fa fa-
user"></i> Import Nilai</h3>

            </div>

            <div class="panel-body">
                <div class="alert alert-info alert-
dismissable"><button type="button" class="close" data-
dismiss="alert" aria-hidden="true">&times;</button><h4>Pastikan
Extensi File Excel yang digunakan excel 2003 - 2007 (.xls) </div>
                if(isset($_POST['submit'])) {

                    $target = basename($_FILES['filepegawaiaiall']['name']) ;

                    move_uploaded_file($_FILES['filepegawaiaiall']['tmp_name'],
$target);

                    $data = new
Spreadsheet_Excel_Reader($_FILES['filepegawaiaiall']['name'], false);

                    // menghitung jumlah baris file xls
                    $baris = $data->rowcount($sheet_index=0);

                    // jika kosongkan data dicentang jalankan kode berikut
                    $drop = isset($_POST["drop"]) ? $_POST["drop"] : 0 ;
                    if($drop == 1){

                        // kosongkan tabel pegawai
                        $truncate = "TRUNCATE TABLE tb_data";
                        mysqli_query($conn, $truncate);

                    };

                    for ($i=1; $i<=$baris; $i++)

                    {

                        // membaca data (kolom ke-1 sd terakhir)

                        $src_bytes = $data->val($i, 2);
                        $dst_bytes = $data->val($i, 3);
                        $count = $data->val($i, 4);
                        $srv_count = $data->val($i, 5);
                        $dst_host_count= $data->val($i, 6);
                        $dst_host_srv_count = $data->val($i, 7);
                        $dst_host_same_src_port_rate = $data->val($i, 8);
                        $dst_host_srv_diff_host_rate = $data->val($i, 9);
                        $tipe_serangan = $data->val($i, 10);
```

```

$query = "INSERT INTO tb_data (src_bytes, dst_bytes, count,
srv_count,          dst_host_count,          dst_host_srv_count,
dst_host_same_src_port_rate,          dst_host_srv_diff_host_rate,
tipe_serangan) VALUES
('$src_bytes',          '$dst_bytes',          '$count',          '$srv_count',
'$dst_host_count',          '$dst_host_srv_count',
'$dst_host_same_src_port_rate',          '$dst_host_srv_diff_host_rate',
'$tipe_serangan')";

ini_set('max_execution_time', 600);
$hasil = mysqli_query($conn, $query);
}

```

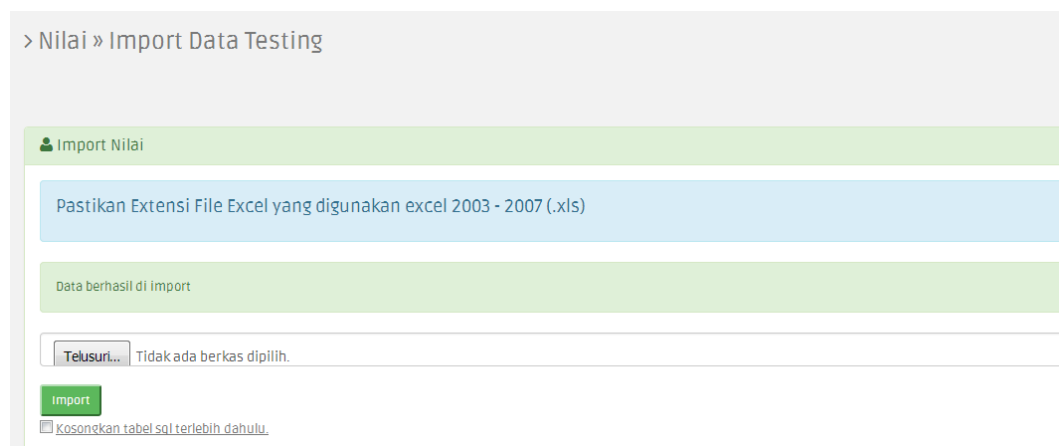
#### 6.1.1.4 Uji Coba *Import Data Testing*

Menu *import data training* pada aplikasi berfungsi untuk menginputkan data yang akan dilakukan proses klasifikasi. Ketika *data testing* yang akan diimport sesuai dengan format yaitu (.xls) maka data tersebut akan disimpan pada *database* dan data telah siap digunakan untuk klasifikasi serangan. Skenario pengujian pada menu *import data testing* dapat dilihat pada tabel 6.4

Tabel 6.4 Skenario Uji Coba Menu *Import Data Testing*

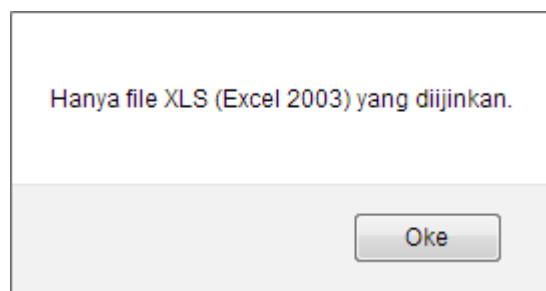
No	Proses	Hasil	Keterangan
1	Import data training sesuai dengan format (.xls)	Data berhasil diimport	Sukses
2	Import data training tidak sesuai dengan format	Hanya file XLS (Excel 2003) yang diijinkan	Sukses

Apabila *admin* melakukan *import data testing* sesuai dengan format yang diinginkan maka pada sistem akan tertulis bahwa data berhasil diimport dan data tersebut akan disimpan pada *database data testing* yang nantinya akan dilakukan proses klasifikasi serangan. Berikut gambar 6.6 adalah gambar data berhasil diimport.



Gambar 6.6 gambar data berhasil diimport pada menu *import data testing*

Ketika *admin* melakukan import data yang tidak sesuai dengan format yang diinginkan sistem maka akan muncul notifikasi bahwa hanya file XLS (excel 2003) yang diijinkan. Sehingga *admin* harus melakukan import ulang data yang berformat excel 2003 (.xls). berikut gambar 6.7 adalah gambar data yang tidak sesuai dengan format.



Gambar 6.7 gambar data tidak sesuai dengan format



#### 6.1.1.4 Uji Coba Menu Hasil

Menu hasil pada aplikasi berfungsi untuk melakukan klasifikasi dan melihat hasil dari klasifikasi serangan. Klasifikasi serangan berasal dari data *import data testing*, data yang diimport pada menu *import data testing* masih belum terklasifikasi apakah serangan tersebut *dos*, *normal* maupun *probe*. Sehingga pada menu hasil pada aplikasi telah terlihat serangan-serangan yang masuk kedalam klasifikasi-klasifikasi serangan. Skenario pengujian pada menu hasil dapat dilihat pada 6.8

Tabel 6.5 Skenario Uji Coba Menu Hasil

No	Proses	Hasil	Keterangan
1	Melakukan klasifikasi data	Data telah terklasifikasi	Sukses

Uji coba dilakukan dengan cara melihat hasil klasifikasi, apakah data yang diimport pada menu *import data testing* telah terklasifikasi pada menu hasil. Data yang diimport pada menu *import data testing* belum terklasifikasi untuk itu pada menu hasil akan mengklasifikasi data tersebut. Berikut gambar 6.8 adalah gambar data sebelum dan sesudah klasifikasi.

no	src_bytes	dst_bytes	count	srv_count	dst_host	dst_host_count	dst_host_diff	tipe_serangan
1	76944	1	12	12	241	238	0	0
2	72564	0	11	11	255	237	0	0
3	69644	0	12	12	255	237	0	0
4	0	0	110	110	255	255	0	0
5	0	0	118	118	255	255	0	0
6	74024	0	10	10	255	244	0	0
7	0	0	34	34	255	255	0	0
8	29824	0	44	44	255	253	0	0
9	54540	8314	4	4	255	254	0	0
10	54540	8314	3	4	255	255	0	0
11	54540	8314	4	10	255	254	0	0
12	54540	8314	3	10	255	251	0	0
13	33580	2920	4	4	255	254	0	0
14	45260	2920	3	3	255	254	0	0
15	33580	7300	3	3	255	254	0	0
16	8	0	1	14	2	98	1 0.5	1
17	18	0	1	1	1	50	1	1
18	8	0	1	34	2	51	1 0.51	1
19	18	0	1	1	1	211	1	1
20	8	0	1	24	2	8	1 0.5	1

no	src_bytes	dst_bytes	count	srv_count	dst_host	dst_host_count	dst_host_diff	tipe_serangan
1	76944	1	12	12	241	238	0	DOS
2	72564	0	11	11	255	237	0	DOS
3	69644	0	12	12	255	237	0	DOS
4	0	0	110	110	255	255	0	PROBE
5	0	0	118	118	255	255	0	DOS
6	74024	0	10	10	255	244	0	DOS
7	0	0	34	34	255	255	0	DOS
8	29824	0	44	44	255	253	0	DOS
9	54540	8314	4	4	255	254	0	DOS
10	54540	8314	3	4	255	255	0	DOS
11	54540	8314	4	10	255	254	0	DOS
12	54540	8314	3	10	255	251	0	DOS
13	33580	2920	4	4	255	254	0	DOS
14	45260	2920	3	3	255	254	0	DOS
15	33580	7300	3	3	255	254	0	DOS
16	8	0	1	14	2	98	1 0.5	1
17	18	0	1	1	1	50	1	1
18	8	0	1	34	2	51	1 0.51	1
19	18	0	1	1	1	211	1	1
20	8	0	1	24	2	8	1 0.5	1

Gambar 6.8 gambar hasil klasifikasi serangan

Berikut *source code* yang digunakan pada menu hasil.

### **bayes.php**

```

$table="tb_data";
$sql = "SELECT * FROM $table";
$query = mysqli_query($conn, $sql);
$count = mysqli_num_rows($query);
#echo "<br>Jumlah data dengan mysql_num_rows: $count
<br/>";

    $sql      =      "SELECT      *      FROM      $table      where
tipe_serangan='DOS'";

    $query = mysqli_query($conn, $sql);
    $dos1 = mysqli_num_rows($query);
    $dos2 = $dos1 / $count;
    $dos2 = round($dos2,3);
    $serangan=array("DOS","NORMAL","PROBE");
    $nilaikriteria = array($dos1,$normal1,$probe1);

    $field=array("src_bytes","dst_bytes","count","srv_cou
nt","dst_host_count","dst_host_srv_count","dst_host_same_sr
c_port_rate","dst_host_srv_diff_host_rate");

    for ($h=0; $h<count($serangan);$h++){

        for ($i=0; $i<count($field); $i++){

            $nfield = $field[$i];

            $sql = "Select * from tb_data where
$nfield=$data2[$nfield] and tipe_serangan='$serangan[$h] '";
            $query = mysqli_query($conn, $sql);
            $dosa = mysqli_num_rows($query);
            $value=$dosa / $nilaikriteria[$h];
            $nilai[$h][$i]=$value;
        }

    $hasil=max($fix);
    $key=array_search($hasil, $fix);
    $key_id = $data2['no_id'];
    $hasil2 = $serangan[$key];

```

## 6.2 Pengujian Akurasi

Salah satu pengukur kinerja klasifikasi adalah tingkat akurasi. Sebuah sistem dalam melakukan klasifikasi diharapkan dapat mengklasifikasi semua set data dengan benar, tetapi tidak dipungkiri bahwa kinerja suatu sistem tidak bisa 100% akurat. Untuk pada penelitian ini akan dilakukan uji akurasi yang berfungsi melihat seberapa baik akurasi klasifikasi serangan menggunakan metode *naive bayes*. Berikut adalah formula untuk menghitung akurasi

$$\text{Akurasi} = \frac{\text{jumlah data benar}}{\text{jumlah data}} \times 100 \%$$

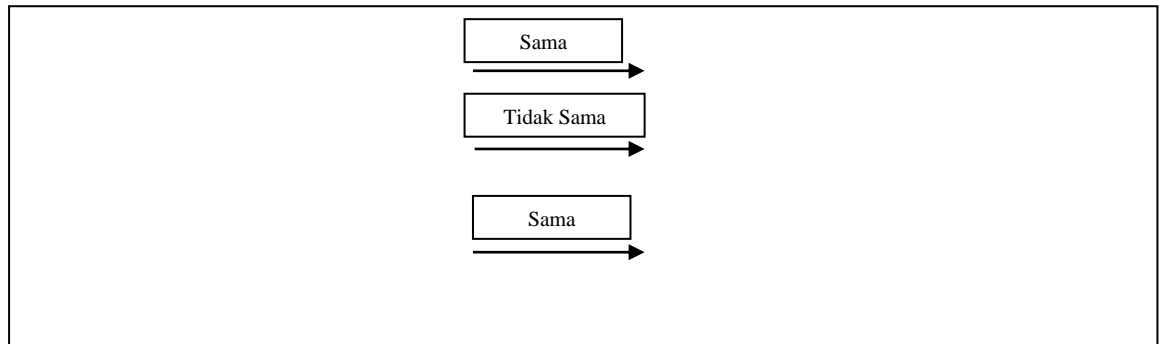
*Data training* yang akan digunakan pada pengujian akurasi ini adalah sebanyak 1500, 3000, 5000 data sedangkan untuk *data testing* sebanyak 100, 150, 200 data. Masing-masing *data testing* akan diuji pada masing-masing *data training*, berikut tabel 6.9 adalah tabel rincian pembagian data yang digunakan pada penelitian ini.

Tabel 6.6 Rincian Data Pengujian Akurasi

No	Jumlah Data Training	Jumlah Data Testing
1	1500 data	100 data
		150 data
		200 data
2	3000 data	100 data
		150 data
		200 data
3	5000 data	100 data
		150 data
		200 data

### 6.2.1 Pengujian Akurasi 1500 Data Training

Uji coba pertama dilakukan dengan *data training* sebanyak 1500 data, untuk *data testing* yang digunakan sebanyak 100, 150, dan 200 data. Pada tahap ini akan dilakukan uji coba untuk melihat tingkat akurasi kebenaran dari klasifikasi serangan menggunakan metode *naive bayes*. Berikut gambar 6.9 adalah perbandingan kebenaran data.



Gambar 6.9 Perbandingan Kebenaran Data Untuk Pengujian Akurasi 1500 Data Training

Pada gambar 6.9 menunjukkan data hasil klasifikasi dan data testing yang telah terklasifikasi oleh NSL-KDD. Hal ini bertujuan untuk melihat apakah klasifikasi berhasil dilakukan dengan cara membandingkan hasil dari keduanya serta akan dapat menentukan tingkat ketepatan akurasi dari sistem yang telah dibuat.

Rincian data :

- 100 serangan = serangan benar (sama) 81 serangan, 19 serangan salah (tidak sama)
- 150 serangan = serangan benar (sama) 122 serangan, 28 serangan salah (tidak sama)
- 200 serangan = serangan benar (sama) 165 serangan, 35 serangan salah (tidak sama)

Perhitungan :

$$\text{Akurasi} = \frac{\text{jumlah data benar}}{\text{jumlah data}} \times 100 \%$$

- Akurasi 100 serangan =  $\frac{81}{100} \times 100 \% = 81 \%$
- Akurasi 150 serangan =  $\frac{122}{150} \times 100 \% = 81,33 \%$
- Akurasi 200 serangan =  $\frac{165}{200} \times 100 \% = 82,5 \%$

Berikut tabel 6.10 adalah hasil dari pengujian akurasi untuk *data training* 1500 dan *data testing* 100, 150, 200.

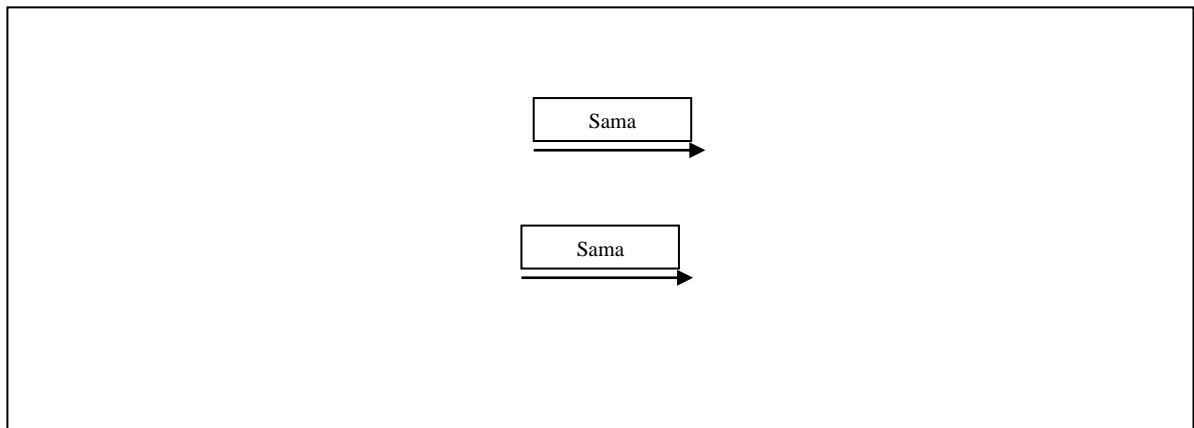
Tabel 6.7 Uji Coba Akurasi 1500 *Data Training*

No	Jumlah Data Training	Jumlah Data Testing	Akurasi
1	1500 data	100 data	81 %
		150 data	81,33 %
		200 data	82,5 %

Dari tabel 6.10 dapat dilihat bahwa jumlah *data testing* sebanyak 100 data memiliki tingkat akurasi yang lebih kecil yaitu 81 % sedangkan akurasi yang paling baik adalah untuk *data testing* sebanyak 200 data yaitu 82,5 %.

### 6.2.2 Pengujian Akurasi 3000 *Data Training*

Uji coba pertama dilakukan dengan *data training* sebanyak 3000 data, untuk *data testing* yang digunakan sebanyak 100, 150, dan 200 data. Pada tahap ini akan dilakukan uji coba untuk melihat tingkat akurasi kebenaran dari klasifikasi serangan menggunakan metode *naive bayes*. Berikut gambar 6.10 adalah perbandingan kebenaran data.



Gambar 6.10 Perbandingan Kebenaran Data Untuk Pengujian Akurasi  
3000 Data Training

Pada gambar 6.10 menunjukkan data hasil klasifikasi dan data testing yang telah terklasifikasi oleh NSL-KDD. Hal ini bertujuan untuk melihat apakah klasifikasi berhasil dilakukan dengan cara membandingkan hasil dari keduanya serta akan dapat menentukan tingkat ketepatan akurasi dari sistem yang telah dibuat.

Rincian data :

- 100 serangan = serangan benar (sama) 82 serangan, 18 serangan salah (tidak sama)
- 150 serangan = serangan benar (sama) 124 serangan, 26 serangan salah (tidak sama)
- 200 serangan = serangan benar (sama) 166 serangan, 34 serangan salah (tidak sama)

Perhitungan :

$$\text{Akurasi} = \frac{\text{jumlah data benar}}{\text{jumlah data}} \times 100 \%$$

- Akurasi 100 serangan =  $\frac{82}{100} \times 100 \% = 82 \%$
- Akurasi 150 serangan =  $\frac{124}{150} \times 100 \% = 82,67 \%$

- Akurasi 200 serangan =  $\frac{166}{200} \times 100 \% = 83 \%$

Berikut tabel 6.11 adalah hasil dari pengujian akurasi untuk *data training* 3000 dan *data testing* 100, 150, 200.

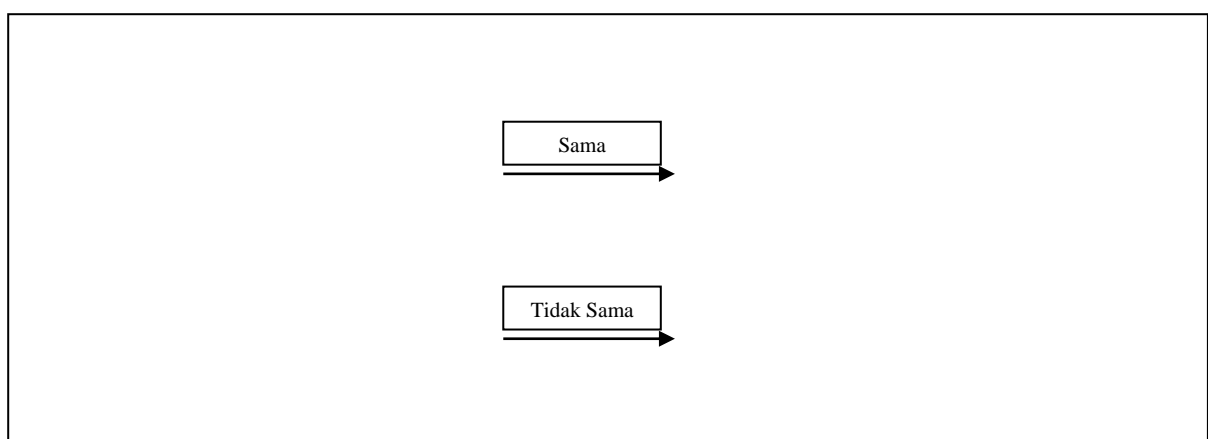
Tabel 6.8 Uji Coba Akurasi 3000 *Data Training*

No	Jumlah Data Training	Jumlah Data Testing	Akurasi
1	3000 data	100 data	82 %
		150 data	82,67 %
		200 data	83 %

Dari tabel 6.11 dapat dilihat bahwa jumlah *data testing* sebanyak 100 data memiliki tingkat akurasi yang lebih kecil yaitu 82 % sedangkan akurasi yang paling baik adalah untuk *data testing* sebanyak 200 data yaitu 83 %.

### 6.2.3 Pengujian Akurasi 5000 *Data Training*

Uji coba pertama dilakukan dengan *data training* sebanyak 5000 data, untuk *data testing* yang digunakan sebanyak 100, 150, dan 200 data. Pada tahap ini akan dilakukan uji coba untuk melihat tingkat akurasi kebenaran dari klasifikasi serangan menggunakan metode *naive bayes*. Berikut gambar 6.11 adalah perbandingan kebenaran data.



Gambar 6.11 Perbandingan Kebenaran Data Untuk Pengujian Akurasi 5000 *Data Training*

Pada gambar 6.11 menunjukkan data hasil klasifikasi dan data testing yang telah terklasifikasi oleh NSL-KDD. Hal ini bertujuan untuk melihat apakah klasifikasi berhasil dilakukan dengan cara membandingkan hasil dari keduanya serta akan dapat menentukan tingkat ketepatan akurasi dari sistem yang telah dibuat.

Rincian data :

- 100 serangan = serangan benar (sama) 84 serangan, 16 serangan salah (tidak sama)
- 150 serangan = serangan benar (sama) 127 serangan, 23 serangan salah (tidak sama)
- 200 serangan = serangan benar (sama) 168 serangan, 32 serangan salah (tidak sama)

Perhitungan :

$$\text{Akurasi} = \frac{\text{jumlah data benar}}{\text{jumlah data}} \times 100 \%$$

- Akurasi 100 serangan =  $\frac{84}{100} \times 100 \% = 84 \%$
- Akurasi 150 serangan =  $\frac{127}{150} \times 100 \% = 84,67 \%$
- Akurasi 200 serangan =  $\frac{168}{200} \times 100 \% = 84 \%$

Berikut tabel 6.12 adalah hasil dari pengujian akurasi untuk *data training* 5000 dan *data testing* 100, 150, 200.

Tabel 6.9 Uji Coba Akurasi 5000 *Data Training*

No	Jumlah Data Training	Jumlah Data Testing	Akurasi
1	5000 data	100 data	84 %
		150 data	84,67 %
		200 data	84 %



Dari tabel 6.12 dapat dilihat bahwa jumlah *data testing* sebanyak 100 data memiliki tingkat akurasi yang lebih kecil yaitu 82,5 % sedangkan akurasi yang paling baik adalah untuk *data testing* sebanyak 200 data yaitu 84,67 %.

### 6.3 Pembahasan

Setelah dilakukan uji coba akurasi menggunakan *data training* sebanyak 1500, 3000, 5000 data serta menggunakan data testing sebanyak 100, 150, 200 data, hasil akurasi menunjukkan bahwa semakin banyak *data training* maka hasil akurasi kebenaran akan semakin baik. Berikut tabel 6.13 perbandingan banyaknya data training dan hasil akurasi yang diperoleh.

Tabel 6.10 Perbandingan Hasil Akurasi *Data Testing*

No	Jumlah Data Testing	Jumlah Data Training	Akurasi
1	100 data	1500 data	81 %
		3000 data	82 %
		5000 data	84 %
2	150 data	1500 data	81,33 %
		3000 data	82,67 %
		5000 data	84,67 %
3	200 data	1500 data	82,5 %
		3000 data	83 %
		5000 data	84 %

Dari tabel 6.13 dapat dilihat bahwa semakin banyak *data training* yang digunakan dalam proses klasifikasi serangan maka akurasi kebenaran dalam menentukan serangan akan semakin baik. Pada penelitian klasifikasi serangan menggunakan metode *naive bayes* ini telah sesuai dengan apa yang diharapkan yaitu dapat mengklasifikasikan serangan-serangan baru dengan acuan *data training* yang ada.

## **BAB VII. KESIMPULAN DAN SARAN**

### **7.1 Kesimpulan**

Dari beberapa uji coba yang didapatkan dari penelitian ini, dapat disimpulkan beberapa hal sebagai berikut :

1. Metode *naive bayes* dapat digunakan sebagai klasifikasi serangan-serangan baru pada Intrusion Detection System (IDS).
2. Semakin banyak *data training* yang digunakan maka hasil dari akurasi kebenaran akan semakin baik.
3. Akurasi kebenaran pada klasifikasi serangan menggunakan metode *naive bayes* adalah 81-84,67 %.

### **7.2 Saran**

Dari apa yang telah dilakukan dan diujicobakan dalam penelitian ini, dari segi akurasi yang telah didapatkan sudah menunjukkan hasil positif. Namun untuk lebih menyempurnakan penelitian ini, mungkin diperlukan beberapa masukan dari peneliti yaitu :

1. Menggunakan *field-field* tambahan untuk proses klasifikasi serangan memungkinkan untuk dapat meningkatkan hasil keakuratan.
2. Menggunakan metode-metode klasifikasi lain dan membandingkan tingkat akurasi kebenaran dalam melakukan klasifikasi serangan.

## DAFTAR PUSTAKA

- [1] Darujati, C. (2010). Perbandingan Klasifikasi Dokumen Teks Menggunakan Metode Naive Bayes dan K-Nearest Neighbor .
- [2] Prasetyo, E. 2012. *Data Mining Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta : Andi.
- [3] Gostev, A. & Namestnikov, Y., 2011. Kaspersky Security Bulletin 2010. Statistics, 2010. [Online] Available at:[http://www.securelist.com/en/analysis/204792162/Kaspersky\\_Security\\_Bulletin\\_2010\\_Statistics\\_2010](http://www.securelist.com/en/analysis/204792162/Kaspersky_Security_Bulletin_2010_Statistics_2010) [Accessed 28 Desember 2016].
- [4] Mitchell, T., 1997. *Machine Learning*. New York: McGraw Hill.
- [5] Rafiudin, R. 2012. *Mengganyang Hacker Dengan Snort*. Yogyakarta:Penerbit Andi
- [6] Santoso, D. (2013). Perbandingan Kinerja Metode Naive Bayes, K-Nearest Neighbor dan Metode Gabungan K-Means dan LVQ dalam Pengkategorian Buku Komputer Bahasa Indonesia Berdasarkan Judul dan Sinopsis.
- [7] Scarfone, K. & Mell, P., Februari, 2007. *Special Publication 800-94: Guide To Intrusion Detection and Prevention Systems*. Gaithersburg, Maryland: National Institute Standard and Technology.
- [8] Sofana, I. 2012. *Cisco CCNA dan Jaringan Komputer*. Bandung: Penerbit Informatika.
- [9] Witten, I.H., Frank, E. & Hall, M.A., 2011. *Data Mining Practical Machine Learning Tools and Technique Third Edition*. New York: Morgan Kaufmann.
- [10] Wicaksana, P. D. (2015). *PERBANDINGAN ALGORITMA K-NEAREST NEIGHBOR DAN NAIVE BAYES UNTUK STUDI DATA "WISCONSIN DIAGNOSIS BREAST CANCER"*. Yogyakarta.
- [11] Ying Yang, G. I. (2002). A Comparative Study of Discretization Methods for Naive-Bayes Classifiers.



## Lampiran 2. Biodata Penulis



### 1. DATA PRIBADI

Nama : Dedi Arpandi  
Tempat, Tanggal Lahir : Teluk Bayur, 13 Mei 1995  
Jenis Kelamin : Laki-Laki  
Agama : Islam  
Kewarganegaraan : Indonesia  
Alamat : Jalan Pinang Hijau RT 10

RW 5 Stasiun 1, Teluk Bayur, Berau

Status Perkawinan : Belum Menikah  
No. Telepon : 085246660348  
Email : [dediarpandi@gmail.com](mailto:dediarpandi@gmail.com)

### 2. RIWAYAT PENDIDIKAN

- Tahun 2001 s/d 2007 : SD Negeri 002 Berau
- Tahun 2007 s/d 2010 : SMP Negeri 8 Berau
- Tahun 2010 s/d 2013 : SMA Negeri 2 Berau
- Tahun 2013 s/d 2016 : D3 Politeknik Negeri Jember  
(Teknik Komputer)
- Tahun 2016 s/d 2017 : D4 Politeknik Negeri Malang  
(Teknik Informatika)