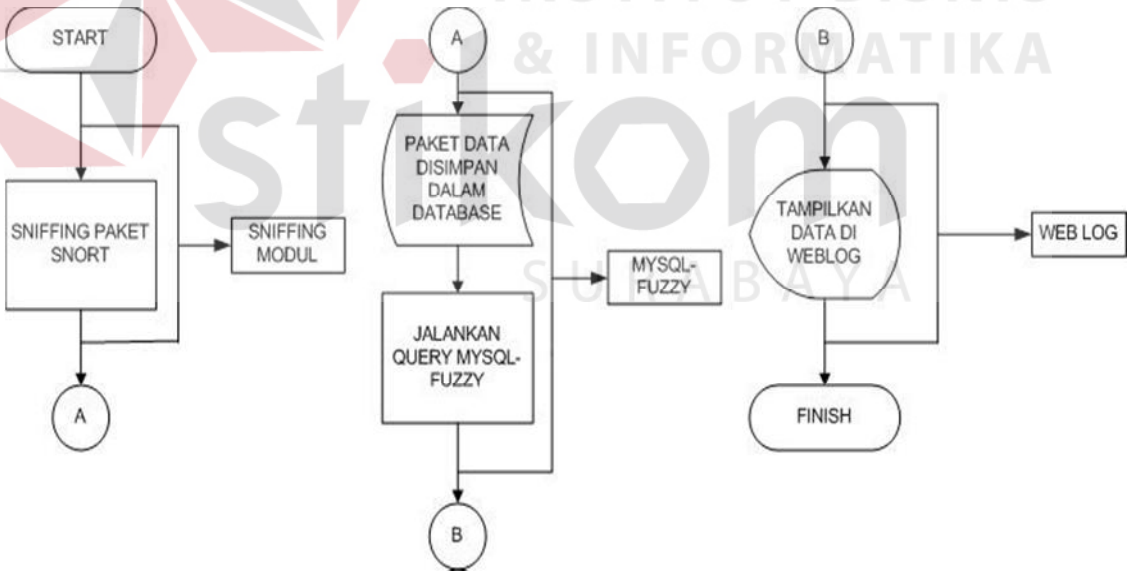
BAB III

METODE PENELITIAN

3.1 Perancangan Sistem Dan Blok Diagram Sistem

Didalam perancangan aplikasi IDS dengan menggunakan metode fuzzy terdiri dari 2 modul utama, yaitu modul *packet sniffing,* dan modul fuzzy dimana masing-masing modul mempunyai fungsi yang berbeda-beda. Modul paket *sniffing* berfungsi untuk mengumpulkan data dan modul fuzzy berfungsi untuk mengklasifikasikan serangan berdasarkan metode fuzzy Sugeno. *Log* serangan ditampilkan bentuk web.

Penyusunan dua modul diatas berdasarkan atas diagram alur sistem aplikasi IDS sebagai berikut:

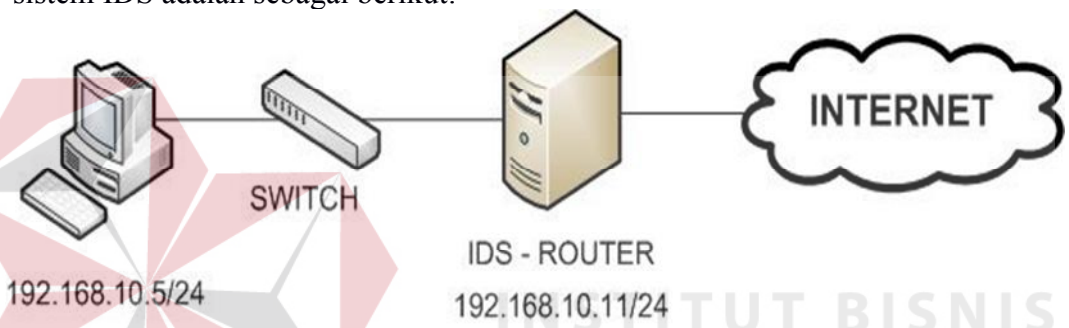


Gambar 3.1. Alur Kerja Sistem IDS

Gambar 3.1 menunjukkan alur kerja sistem IDS secara umum. Sistem mulai bekerja dengan menjalankan modul *sniffing paket* yang berfungsi untuk memonitoring semua paket data yang melalui *interface ethernet* router. Aplikasi yang digunakan untuk memantau paket data adalah snort. Semua paket data yang

termonitor oleh router akan disimpan dalam bentuk *database*. Dari *database* tersebut diolah untuk membuat keputusan fuzzy melalui *query* sql.

Pengolahan data dilakukan berdasarkan aturan-aturan fuzzy yang telah dibentuk sesuai dengan karakter serangan. Setelah melalui proses klasifikasi serangan oleh modul fuzzy selanjutnya data-data yang terindikasi serangan ditampilkan dalam bentuk web *log*. Sistem IDS ini diterapkan pada router yang berfungsi sebagai keluar masuknya trafik jaringan. Adapun diagram blok topologi sistem IDS adalah sebagai berikut:



Gambar 3.2. IDS Topologi

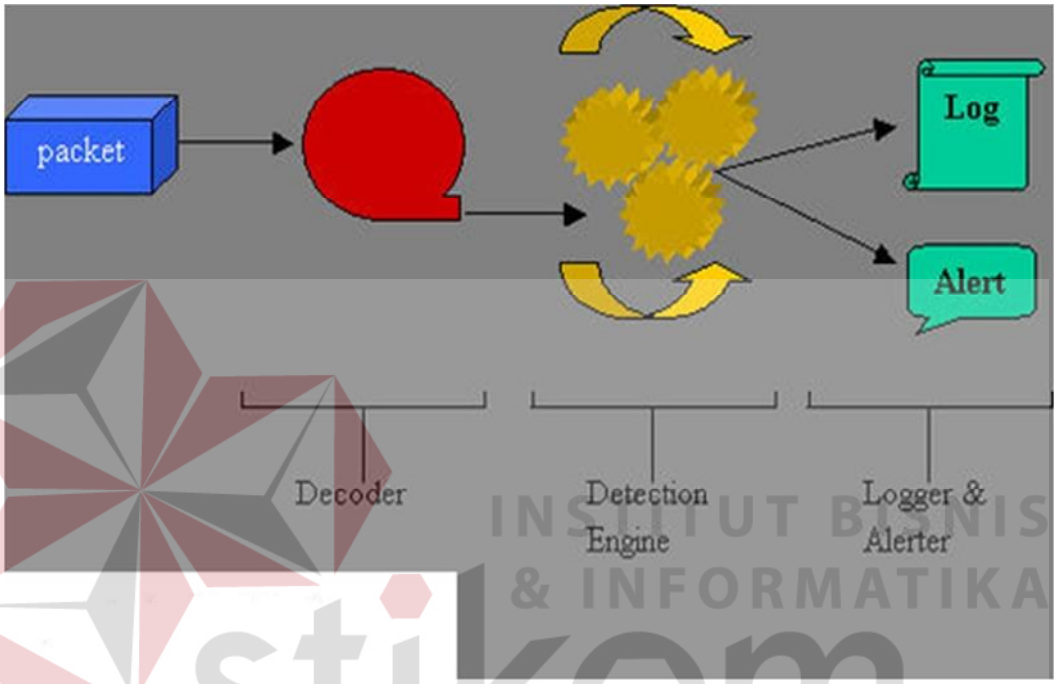
Pada Gambar 3.2 IDS akan memonitor paket data dari *workstation 192.168.10.5/24*, data yang terindikasi serangan akan disimpan dan ditampilkan dalam *history log* router. Penampilan *log* sistem IDS berbasis web dengan memanfaatkan pemrograman php. Pada web *log* data serangan ditampilkan alamat ip penyerang sehingga memudahkan administrator jaringan untuk melakukan analisis serangan.

3.2 Rancangan Sistem

Sesuai dengan diagram alur sistem ids pada Gambar 3.1 terdapat 3 komponen penting dalam pembuatan aplikasi IDS dengan menggunakan metode fuzzy, yaitu paket *sniffing*, pengklasifikasian serangan dan web *log*. Dibawah ini akan dijelaskan bagaimana sistem IDS dibuat.

3.2.1 Perancangan Modul *Sniffing* Paket

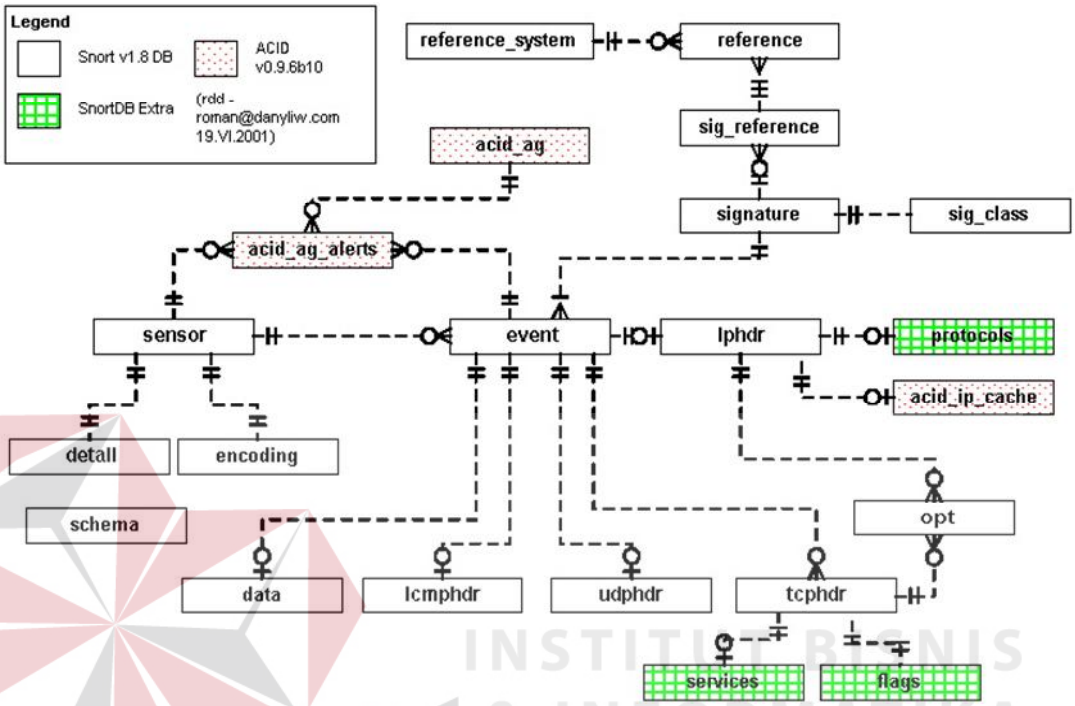
Modul paket *sniffing* merupakan tahapan awal dalam perancangan sistem IDS. Modul ini berperan untuk menangkap data serangan dan menyimpan data tersebut ke dalam *database*. Aplikasi *network sniffing* yang digunakan adalah *snort-mysql*. Dibawah ini merupakan blok diagram alur bagaimana snort bekerja.



Gambar 3.3. Diagram Alur Snort (Nalneesh Gaur, 2001).

Gambar 3.3 merupakan diagram alur Snort, dari Gambar 3.3 dapat dilihat beberapa tahapan snort untuk mendapatkan *log* paket data. Tahap awal snort menangkap paket data mentah, paket data ini masih berupa susunan bilangan biner. Paket data ditangkap oleh *decoder* modul melalui berbagai *interface* jaringan (*contohnya Ethernet, PPP)*, Setelah data diolah oleh *decoder* selanjutnya *detection engine* mengaplikasikan aturan-aturan protokol dalam snort ke dalam paket tersebut. Data yang telah diolah oleh *detection engine* ditampilkan sebagai *log* atau *alert*.

Data *log* hasil *sniffing* snort akan ditampilkan dalam bentuk *database*. Dari data inilah nantinya akan diolah untuk memunculkan hasil keputusan serangan. Berikut ER diagram *database* snort:



Gambar 3.4. ER Diagram Snort (Roman Danyliw, 2002)

Dalam Gambar 3.4 ditunjukkan ER Diagram snort, dalam ER Diagram dapat dilihat bahwa snort mempunyai 22 tabel. Tabel *event* terhubung dengan 8 tabel yaitu tabel *iphdr*, *sensor*, *data*, *icmphdr*, *udphdr*, *tcphdr*, *signature*, *sensor*, dan *acid_ag_alerts*. Tabel *event* menyimpan waktu *capture packet* terjadi, sehingga 8 tabel yang membutuhkan data waktu *capture packet* akan memanggil data dari tabel *event*. Tabel *iphdr* terhubung dengan 3 tabel yaitu *protocols*, *acid_ip_cache* dan *opt*. Dalam tabel *iphdr* menyimpan data ip *host* pengirim, ip *host* penerima, ip protokol dan *ip length.* Dalam sistem IDS pengolahan data diambil dari tabel *event* dan *iphdr*.

3.2.2 *Sampling* Data Serangan

Sampling data serangan berguna untuk membantu menentukan besaran domain kategori variabel fuzzy. Melalui *sampling* data yang tersimpan dalam database snort terdapat 4 jenis data paket yang berbeda yaitu paket normal, paket udp *flooding,* paket *tcp flooding* dan paket icmp *flooding.* Berikut ini *sampling* masing-masing paket data yang ditunjukkan dalam Tabel 3.1.

Tabel 3.1. Sampling Data Serangan

Paket	Frekuensi (/s)	Jumlah Length (B/s)
ICMP Normal	3	180
UDP Flooding	1	1500
TCP Flooding	1	1500
ICMP Flooding	3	65078
Syn-Ack	3	120
HTTP	1	1440

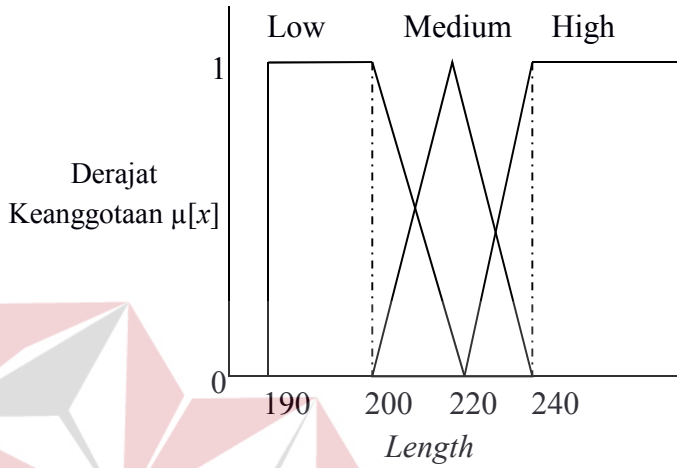
Tabel 3.1 berisi contoh nilai frekuensi dan *length* dari paket serangan yang ditangkap oleh Snort, data tersebut diambil dari data yang tersimpan dalam tabel *acid_event* dan *iphdr* pada database snort.

3.2.3 Perancangan Fuzzy

Aplikasi IDS yang dirancang ini menggunakan algoritma fuzzy yang menghasilkan identifikasi serangan ke dalam 4 bagian yaitu bukan serangan atau normal, *low, middle,* dan *high*. Fuzzy yang digunakan menggunakan metode Sugeno. Dalam metode fuzzy Sugeno terdapat 3 tahapan utama dalam menentukan hasil keputusan yaitu pembentukan fungsi keanggotaan, penyusunan aturan fuzzy dan defuzzyfikasi . Dibawah ini dijelaskan tahapan penyusunan metode fuzzy Sugeno.

Tahap pertama adalah pembentukan fungsi keanggotaan dimana terdapat 2 fungsi keanggotaan fuzzy yaitu fungsi keanggotaan variabel frekuensi dan *length*.

Fungsi keanggotaan variable *length*:



Gambar 3.5. Fungsi Keanggotaan Variabel *Length*

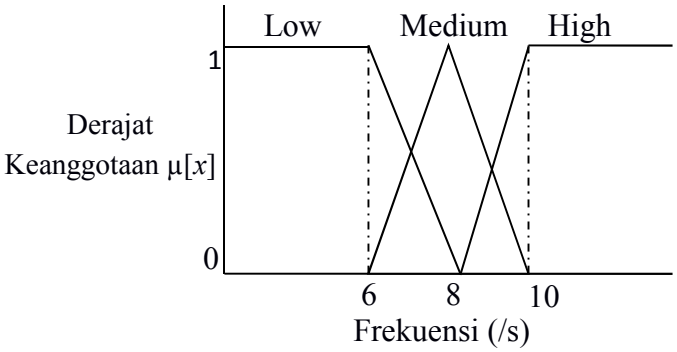
Nilai *length* yang ditunjukkan Gambar 3.5 berdasarkan acuan yang didapat dari hasil *sampling* pada Tabel 3.1. Dari *sampling* tersebut diketahui nilai *length* ICMP normal adalah 180 Byte per detik sehingga ditentukan nilai *length* kategori *low* adalah 190 - 200 Byte (derajat keanggotaan 1), selanjutnya dipilih kelipatan 20 Byte untuk masing masing titik tertinggi derajat keanggotaan (=1) tiap kategori.

Pada variabel *length* (B), data yang dimiliki adalah 200 B, 220 B dan 240 B, dengan demikian pada variabel ini bisa dibagi menjadi 3 himpunan fuzzy yaitu LOW, MEDIUM dan HIGH. Himpunan fuzzy LOW memiliki domain [190,220], dengan derajat keanggotaan tertinggi (=1) terletak pada 190-200 B yang artinya *length* paket adalah LOW, tetapi jika frekuensi lebih dari 200 B maka kondisi *length* paket mendekati MEDIUM. Himpunan fuzzy LOW direpresentasikan dengan fungsi keanggotaan trapesium dengan derajat keanggotaan semakin rendah jika nilai *length* melebihi 200 B.

Himpunan fuzzy MEDIUM memiliki domain [200,240], dengan derajat keanggotaan MEDIUM tertinggi (=1) terletak pada nilai 220. Jika nilai *length* kurang dari 220 B maka kondisi nilai *length* mendekati kondisi LOW, sehingga derajat keanggotaannya pada himpunan MEDIUM akan berkurang. Tetapi jika nilai *length* melebihi nilai 240 B maka kondisi nilai *length* mendekati kondisi HIGH. Himpunan fuzzy MEDIUM direpresentasikan dengan fungsi keanggotaan segitiga dengan derajat keanggotaan tertinggi (=1) jika *length* mendekati nilai 240 B.

Himpunan fuzzy HIGH memiliki domain [240, +] dengan derajat keanggotaan HIGH tertinggi (=1) terletak pada nilai lebih dari sama dengan 240. Jika nilai *length* kurang dari 240 B maka variabrel *length* mendekati kondisi MEDIUM, sehingga derajat keanggotaannya pada himpunan HIGH akan berkurang sedangkan derajat keanggotaan pada himpunan MEDIUM akan bertambah. Tetapi jika frekuensi paket melebihi 240 B maka kondisi *length* paket adalah HIGH. Himpunan fuzzy HIGH direpresentasikan dengan fungsi keanggotaan trapesium dengan derajat keanggotaan tertinggi (=1) jika nilai *length* lebih dari sama dengan 240 B.

Berikut ini adalah fungsi keanggotaan frekuensi:



Gambar 3.6. Fungsi Keanggotaan Variabel Frekuensi

Nilai frekuensi yang ditunjukkan Gambar 3.6 berdasarkan acuan yang didapat dari hasil *sampling* pada Tabel 3.1. Dari *sampling* tersebut diketahui nilai *length* ICMP normal adalah 3 per detik sehingga ditentukan nilai frekuensi low adalah 1 – 6 (derajat keanggotaan 1) dengan nilai terkecil adalah 1, untuk selanjutnya dipilih kelipatan 2 untuk masing masing titik tertinggi derajat keanggotaan (=1) tiap kategori.

Pada variabel frekuensi, data yang dimiliki 6(/s), 8(/s) dan 1(/s), dengan demikian variabel ini bisa dibagi menjadi 3 himpunan fuzzy, yaitu LOW, MEDIUM dan HIGH. Himpunan fuzzy LOW memiliki domain [0,8] dengan derajat keanggotaan LOW tertinggi (=1) terletak pada nilai lebih dari sama dengan 0(/s) dan kurang dari 6(/s). Jika tingkat frekuensi terletak pada *range* 6(/s) sampai 8(/s) maka kondisi frekuensi paket mendekati kondisi MEDIUM. Himpunan fuzzy LOW direpresentasikan dengan fungsi keaanggotaan trapesium dengan derajat keanggotaan tertinggi mencapai (=1) apabila nilai length mencapai *range* 0-40(/s).

Himpunan fuzzy MEDIUM memiliki domain [6,10] dengan derajat keanggotaan tertinggi (=1) terletak pada nilai 8(/s). Jika tingkat nilai frekuensi kurang dari 8(/s), maka kondisi length sudah mendekati LOW. Tetapi jika tingkat frekuensi melebihi 8(/s) maka kondisi frekuensi mendekati HIGH. Himpunan fuzzy MEDIUM direpresentasikan dengan fungsi keanggotaan segitiga dengan derajat keanggotaan tertinggi (=1) jika tingkat nilai length berada pada nilai 8(/s).

Himpunan fuzzy HIGH memiliki domain [10,+] dengan derajat keanggotaan HIGH tertinggi (=1) terletak pada nilai lebih lebih dari sama dengan 10(/s). Jika tingkat nilai frekuensi kurang dari 10(/s) maka kondisi frekuensi

mendekati MEDIUM. Tetapi jika tingkat frekuensi melebihi 10(/s) maka tingkat frekuensi adalah HIGH. Himpunan fuzzy HIGH direpresentasikan dengan fungsi keanggotaan trapesium dengan derajat keanggotaan tertinggi (=1) apabila tingkat frekuensi lebih dari sama dengan 10(/s).

Tahap kedua adalah penyusunan aturan fuzzy, berikut adalah tabel aturan fuzzy:

Tabel 3.2. Aturan Fuzzy

Frekuensi / Length	High	Medium	Low
High	Attack High	Attack High	Attack Med
Medium	Attack High	Attack Med	Attack Med
Low	Attack Med	Attack Med	Attack Low

Dari Tabel 3.2 diketahui terdapat 9 aturan fuzzy yaitu:

1. [R1] *IF* Frekuensi HIGH and Length HIGH *THEN* ATTACK HIGH
2. [R2] *IF* Frekuensi HIGH and Length MEDIUM *THEN* ATTACK HIGH
3. [R3] *IF* Frekuensi HIGH and Length LOW *THEN* ATTACK HIGH
4. [R4] *IF* Frekuensi MEDIUM and Length HIGH *THEN* ATTACK HIGH
5. [R5] *IF* Frekuensi MEDIUM and Length MEDIUM *THEN* ATTACK MEDIUM
6. [R6] *IF* Frekuensi MEDIUM and Length LOW *THEN* ATTACK MEDIUM
7. [R7] *IF* Frekuensi LOW and Length HIGH *THEN* ATTACK MEDIUM
8. [R8] *IF* Frekuensi LOW and Length MEDIUM *THEN* ATTACK MEDIUM
9. [R9] *IF* Frekuensi LOW and Length LOW *THEN* ATTACK LOW

Tahap yang ketiga adalah pembentukan defuzyfikasi. Berikut ini adalah contoh jika paket data yang didapatkan memiliki frekuensi 3(/s) dan length 243B.

1. [R1] *IF* Frekuensi HIGH and Length HIGH *THEN* ATTACK HIGH

α-predikat1 = μFrHIGH ∩ μLthHIGH

= min(μFrHIGH(3), μLthHIGH(243))

=min(0;1) = 0

Nilai z1 = 2400

2. [R2] *IF* Frekuensi HIGH and Length MEDIUM *THEN* ATTACK HIGH

α-predikat2 = μFrHIGH ∩ μLthMEDIUM

= min(μFrHIGH(3), μLthMEDIUM(243))

=min(0;0) = 0

Nilai z1 = 2400

3. [R3] *IF* Frekuensi HIGH and Length LOW *THEN* ATTACK HIGH

α-predikat3 = μFrHIGH ∩ μLthLOW

= min(μFrHIGH(3), μLthLOW(243))

=min(0;0) = 0

Nilai z1 = 2400

4. [R4] *IF* Frekuensi MEDIUM and Length HIGH *THEN* ATTACK HIGH

α-predikat4 = μFrMEDIUM ∩ μLthHIGH

= min(μFrHIGH(3), μLthHIGH(243))

=min(0;1) = 0

Nilai z1 = 2400

5. [R5] *IF* Frekuensi MEDIUM and Length MEDIUM *THEN* ATTACK MEDIUM

α-predikat5 = μFrMEDIUM ∩ μLthMEDIUM

= min(μFrMEDIUM(3), μLthMEDIUM(243))

=min(0;0) = 0

Nilai z2 = 1760

6. [R6] *IF* Frekuensi MEDIUM and Length LOW *THEN* ATTACK MEDIUM

α-predikat6 = μFrMEDIUM ∩ μLthLOW

= min(μFrMEDIUM(3), μLthLOW(243))

=min(0;0) = 0

Nilai z2 = 1760

7. [R7] *IF* Frekuensi LOW and Length HIGH *THEN* ATTACK MEDIUM

α-predikat7 = μFrLOW ∩ μLthHIGH

= min(μFrLOW(3), μLthHIGH(243))

=min(1;1) = 1

Nilai z2 = 1760

8. [R8] *IF* Frekuensi LOW and Length MEDIUM *THEN* ATTACK MEDIUM

α-predikat8 = μFrLOW ∩ μLthMEDIUM

= min(μFrLOW(3), μLthMEDIUM(243))

=min(1;0) = 0

Nilai z2 = 1760

9. [R9] *IF* Frekuensi LOW and Length LOW *THEN* ATTACK LOW

α-predikat9 = μFrLOW ∩ μLthLOW

= min(μFrLOW(3), μLthLOW(243))

=min(1;0) = 0

Nilai z3 = 1200

Nilai Zx adalah nilai perkalian batas masing-masing derajat keanggotaan tertinggi (=1) suatu kategori. Setelah didapatkan masing-masing α-predikat dan nilai z_x untuk setiap aturan maka diperoleh nilai z, yaitu:

Z= αpred1*z1 + αpred2*z1 + αpred3*z1 + αpred4*z1 + αpred5*z2 + αpred6*z2 + αpred7*z2 + αpred8*z2 + αpred9*z3 / αpred1 + αpred2 + αpred3 + αpred4

=0 * 2400 + 0 * 2400 + 0 * 2400 + 0 * 2400 + 0 * 1760 + 0 * 1760+ 1 * 1760 + 0 * 1760 + 0 * 1200 / 0+1+0+0+0+0+0+0

=1760/1 = 1760

Kategori defuzzyfikasi = *Attack Middle*

Dimana kategori untuk masing masing z adalah:

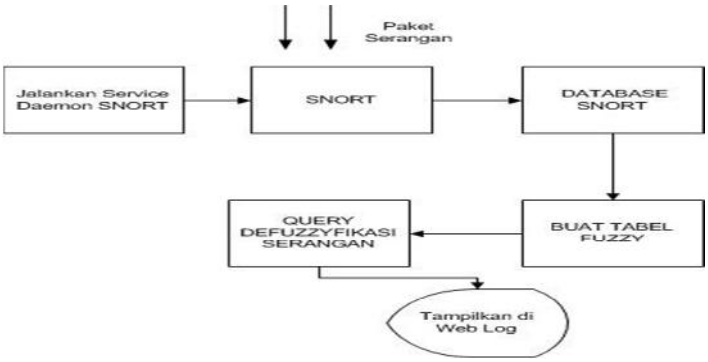
1200 = *Attack Low*

1760 = *Attack Middle*

2400 = *Attack High*

3.3 Desain Sistem

Desain sistem IDS meliputi pengolahan *database*, proses pengklasifikasian serangan dan proses penampilan web *log*. Untuk lebih jelasnya dapat dilihat Gambar 3.7.

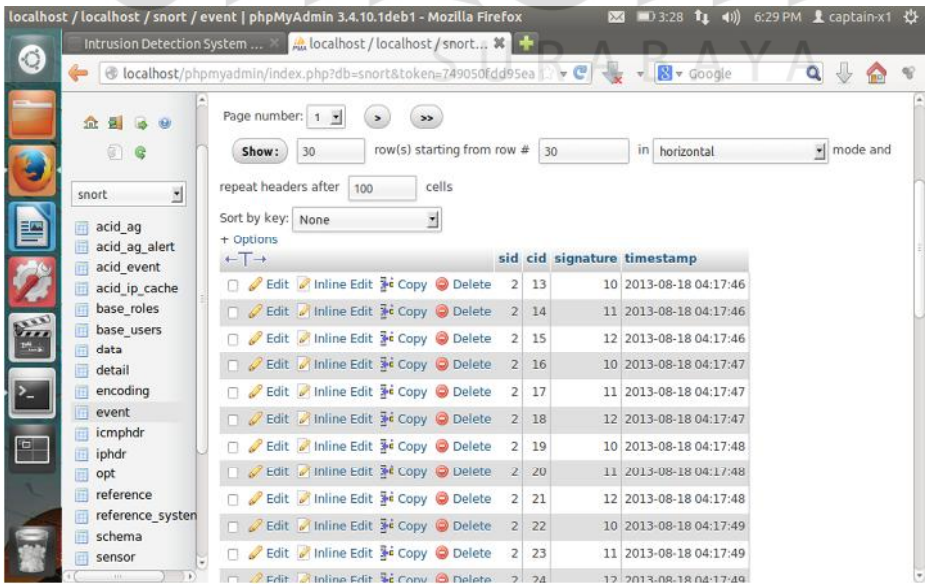


Gambar 3.7. Rancangan Desain Sistem

Pada hasil yang ditampilkan dalam aplikasi PHP, web *log* merupakan hasil *query* defuzzyfikasi dari tabel *database* snort. Proses defuzzyfikasi menggunakan *query* sql tidak terlalu membebani kinerja router, sehingga router dapat memproses defuzzyfikasi secara cepat. Sistem pendeteksian bersifat *realtime* jadi data serangan yang diolah bersifat *up to date* artinya data tersebut setiap saat dapat bertambah. Aplikasi yang dibuat hanya dijalankan sekali dan akan menjalankan proses terus menerus tanpa henti, kecuali jika aplikasi ditutup oleh *user* yang berwenang sehingga secara otomatis sistem tidak akan melakukan proses pendeteksian serangan.

3.3.1 Desain *Input*

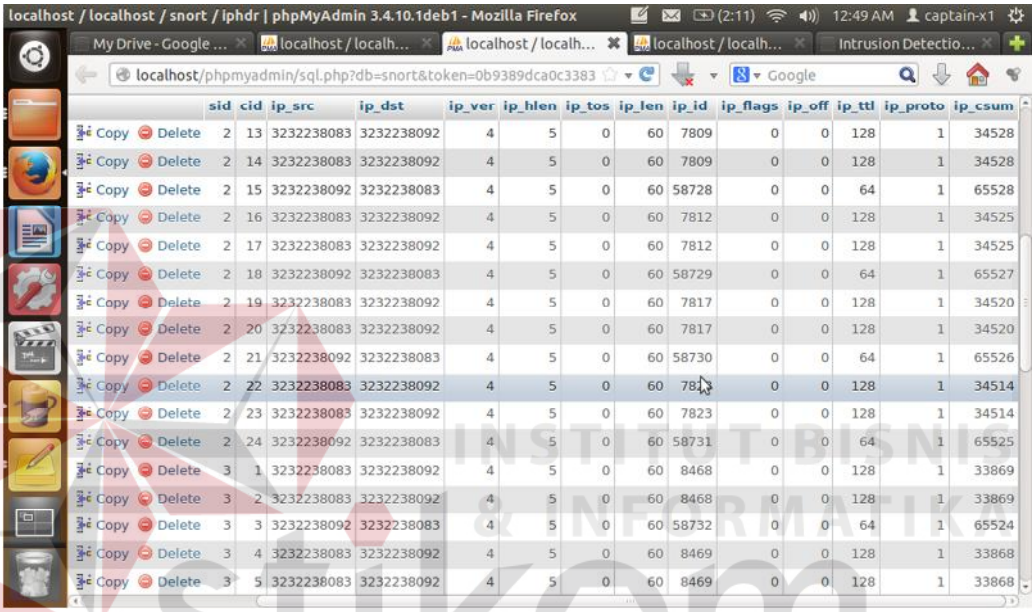
Input dari sistem pendeteksian serangan adalah berupa data serangan yang setiap saat terus bertambah dari *database log* milik snort selanjutnya akan diproses oleh program sesuai *query* fuzzy yang ada. Data input terdiri dari 2 tabel snort yaitu tabel *event* dan *iphdr*. Gambar 3.8 merupakan contoh data tabel *event*.



Gambar 3.8. Data Tabel Event

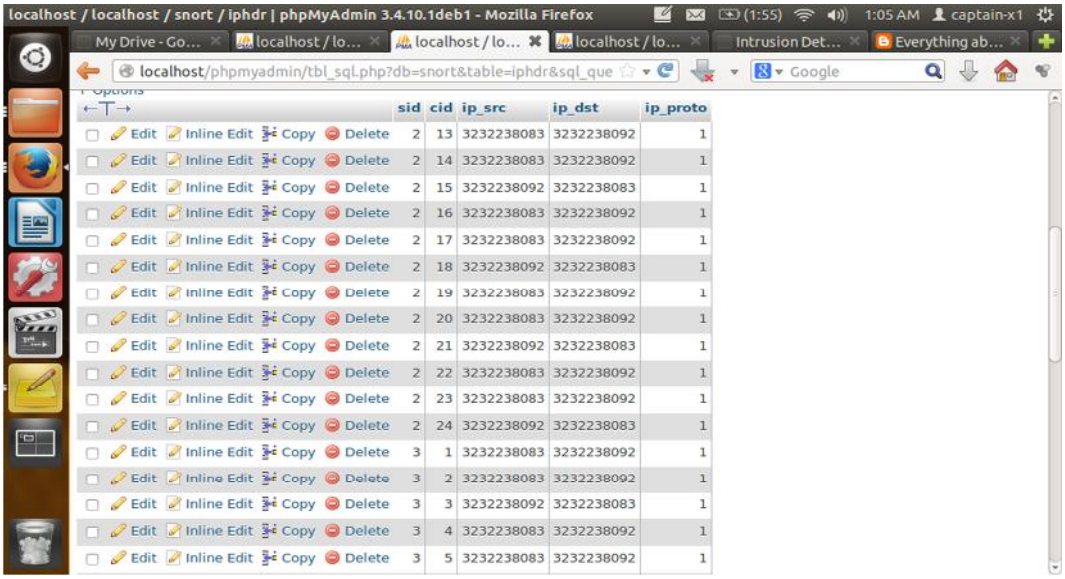
Gambar 3.8 menunjukkan Tabel event yang mempunyai beberapa *field* yaitu *sid, cid, signature* dan *timestamp. Field* data yang diambil untuk diolah dalam *query* defuzzyfikasi hanya 3 *field* yaitu *sid, cid* dan *timestamp*. *Field sid* dan *cid* menghubungkan ke tabel paket data seperti IP/TCP/UDP/ICMP *header*, sedangkan *timestamp* berisi waktu data ditangkap oleh *packet sniffer* snort.

Berikut ini adalah data tabel *iphdr*:



Gambar 3.9. Data Tabel iphdr

Pada Gambar 3.9 diatas merupakan *database* snort yang terdiri dari berbagai field hasil *log* paket data. Namun tidak semua data yang ada di *database log* digunakan. Untuk membangun sistem IDS ini hanya 5 parameter yang digunakan yaitu sid, cid, ip_src, ip_dst dan ip_proto. Data *log* yang sudah dilakukan seleksi dan siap diolah ditunjukkan pada Gambar 3.10

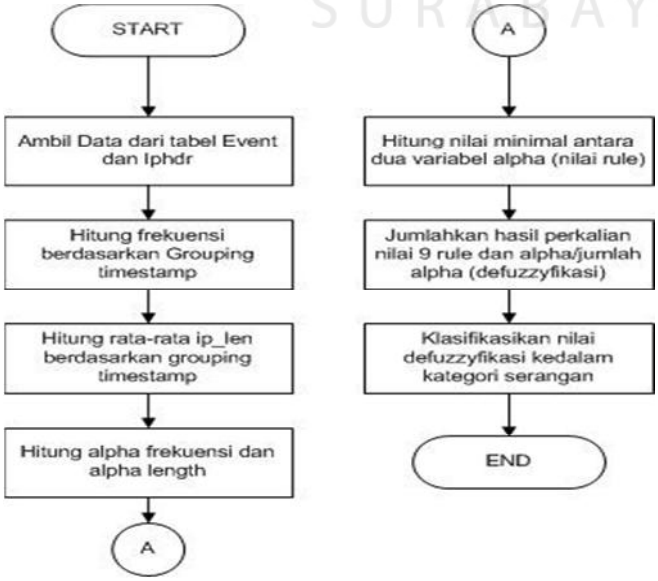


Gambar 3.10. Data iphdr yang sudah diseleksi

3.3.2 Desain Proses

Perintah *query* yang dipakai untuk melihat dan menyeleksi data dari berbagai field adalah *select. Syntax* dasar penggunaan *select* adalah *select [nama field] from [nama tabel] where [expr...] group by [expr..]*. Perintah *query* untuk menampilkan data yang terdapat pada Gambar 3.10 adalah *select sid, cid, ip_src, ip_dst, ip_proto from iphdr.*

Berikut blok diagram *query* defuzzyfikasi yang ditunjukkan oleh Gambar 3.11.



Gambar 3.11. Diagram Blok *Query* Fuzzy

Pada Gambar 3.11 *query* mulai bekerja dengan memilih data yang ada pada tabel event dan iphdr. Adapun field yang diseleksi adalah *timestamp, ip_src, ip_dst, ip_len* dan *ip_proto*. Selanjutnya *query* mencari nilai frekuensi dan rata-rata *ip_len* berdasarkan data field *timestamp* yang telah digroup per satuan detik. Untuk nilai alpha frekuensi dan alpha length didapatkan dengan memasukkan nilai frekuensi dan rata-rata ip_len hasil group timestamp kedalam metode fuzzy sugeno. Tahapan akhir yaitu defuzzyfikasi didapatkan dengan menjumlahkan hasil perkalian alpha dan nilai zx yang sudah ditentukan kemudian dibagi dengan penjumlahan 9 nilai rule. Pencocokan data dilakukan berdasarkan sid dan cid.

3.3.3 Desain Output

Output yang dihasilkan oleh sistem pendeteksian serangan ini berupa data hasil pengolahan *query* yang menghasilkan nilai defuzzyfikasi. Nilai defuzzyfikasi inilah yang digunakan sebagai acuan untuk menentukan hasil serangan. Berikut ini adalah kategori nilai serangan, dapat dilihat pada Tabel 3.3.

Tabel 3.3. Pengklasifikasian Hasil Serangan

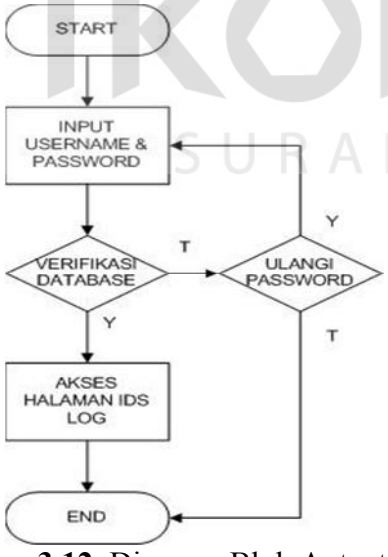
Nilai Defuzzyfikasi	Kategori Attack
0	Bukan Serangan
1 – 1200	Low
1201 – 2399	Medium
>= 2400	High

Nilai defuzzyfikasi yang ditunjukkan oleh Tabel 3.3 didapatkan dari hasil perkalian masing-masing batas derajat keanggotaan 1 dari tiap kategori variabel *length* dan frekuensi. Batas derajat keanggotaan 1 kategori *low* terletak pada nilai 200 untuk variabel *length* dan 6 untuk variabel frekuensi maka hasil perkalian dari 2 variabel adalah 200 * 6 = 1200. Batas derajat keanggotaan 1 kategori *middle*

terletak pada nilai 220 untuk variabel *length* dan 8 untuk variabel frekuensi maka hasil perkalian dari 2 variabel tersebut adalah 220 * 8 = 1760. Batas derajat keanggotaan 1 kategori *high* terletak pada nilai 240 untuk variabel *length* dan 10 untuk variabel frekuensi maka hasil perkalian dari 2 variabel tersebut adalah 240 * 10 = 2400. Dari perkalian tersebut didapatkan jangkauan nilai *low* adalah 1-1200, jangkauan kategori *high* lebih dari sama dengan 2400 sehingga jangkauan nilai *middle* adalah 1201-2399. Nilai defuzzyfikasi 0 adalah nilai untuk kategori bukan serangan sebab nilai 0 menyatakan bahwa suatu paket memiliki derajat keanggotaan 0.

3.3.4 Desain Autentifikasi

Autentifikasi digunakan untuk mengatur *user* yang diperbolehkan mengakses log IDS. Blog diagram proses autentifikasi *user* adalah sebagai berikut.



Gambar 3.12. Diagram Blok Autentifikasi

Pada Gambar 3.12 proses autentifikasi dimulai dengan memasukkan *username* dan *password* oleh *user.* Selanjutnya *username* dan *password user* diverifikasi

sesuai dengan data yang terdapat pada *database.* Jika *username* dan *password* benar maka oleh sistem akan diijinkan untuk masuk pada halaman web *log* IDS, sebaliknya jika salah terdapat dua kemungkinan yaitu ingin mengulangi memasukkan *username* dan *password* atau ingin mengakiri proses autentifikasi.

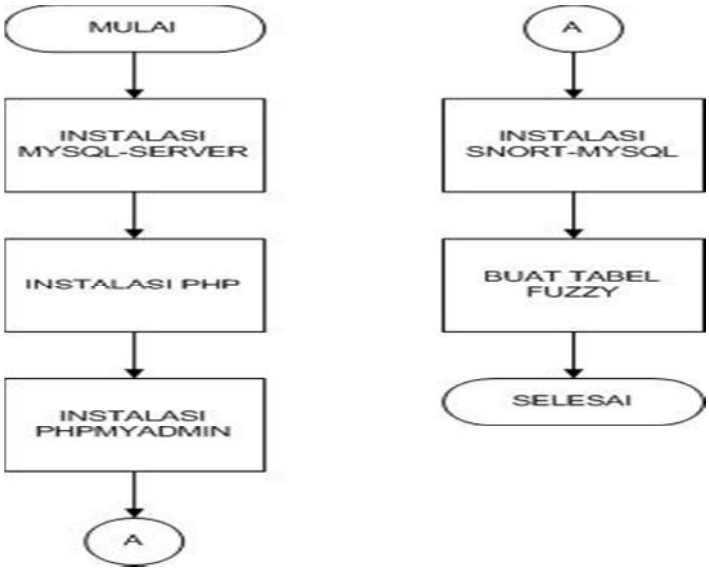
3.4 Implementasi Program

Berdasarkan uraian dari desain sistem, maka tahapan utama untuk membangun aplikasi pendeteksian serangan dengan menggunakan metode fuzzy dengan algoritma fuzzy Sugeno yaitu tahap perolehan data oleh snort sebagai data *log input*. Sebelum dilakukan proses pengolahan data menggunakan fuzzy Sugeno maka data dari *log* snort harus diseleksi sesuai kebutuhan variabel fuzzy. Setelah didapatkan data variabel fuzzy selanjutnya diolah menurut tahapan-tahapan desain sistem fuzzy.

3.4.1 Pengumpulan Data (*Database Log* Snort)

Snort mempunyai kemampuan mendeteksi serangan dan menuliskan data-data serangannya di file *log*, namun snort tidak hanya bisa menulis *log* pada sebuah file saja, namun snort juga bisa menuliskan *log* nya kedalam sebuah *database* baik itu *database* mysql, postgre ataupun oracle. Agar snort dapat menuliskan data *log* kedalam *database* maka perlu dilakukan konfigurasi yang berbeda dari konfigurasi standar.

Diagram blok pengumpulan data dapat dilihat pada Gambar 3.13.



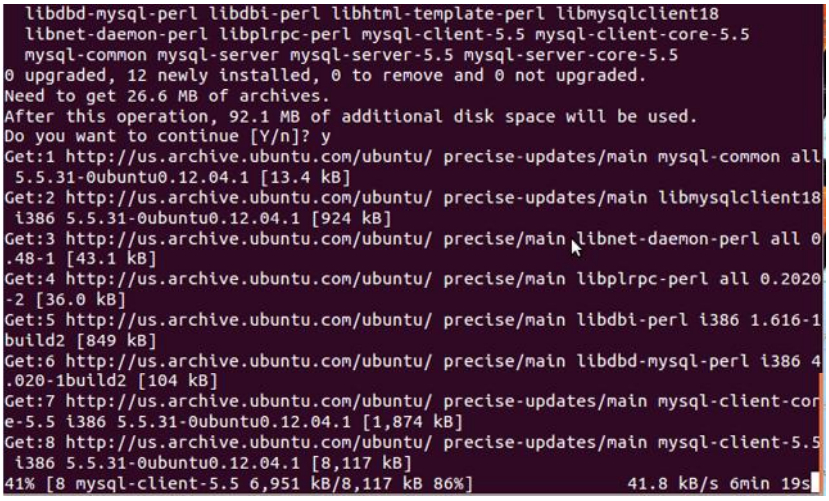
Gambar 3.13. Diagram Blok Pengumpulan Data

Pada Gambar 3.13 langkah yang dilakukan pertama kali adalah melakukan instalasi mysql-server. Kemudian langkah kedua adalah melakukan instalasi php, selanjutnya melakukan instalasi phpmyadmin. *Service* terakhir yang perlu diinstal adalah snort-mysql. Langkah terakhir adalah membuat *database* fuzzy yang berisi variabel-variabel fuzzy.

Dibawah ini merupakan penjelasan langkah-langkah instalasi *service* pada Gambar 3.13, sebagai berikut:

1. Instalasi mysql-server

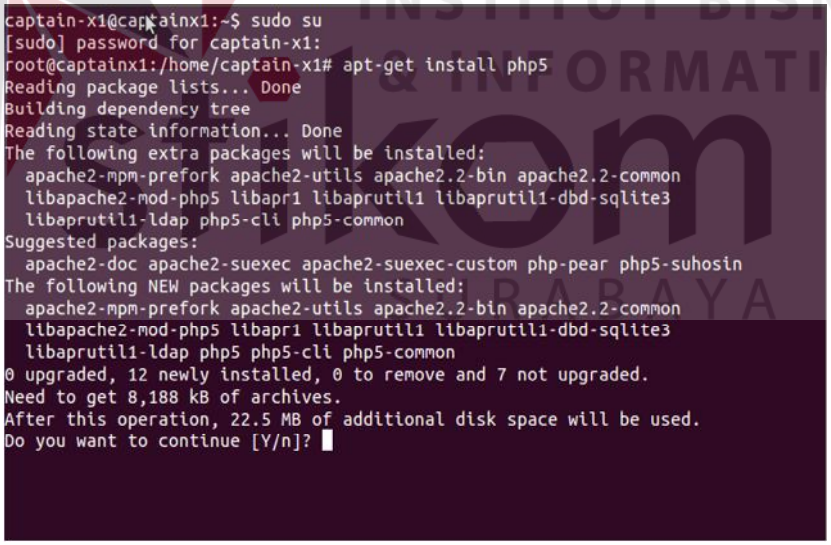
   Sebelum instalasi snort maka perlu dilakukan instalasi *database* server yaitu menggunakan mysql-server. Untuk menginstall *package* mysql-server dilakukan pada *shell* linux dengan mengetikkan perintah *#apt-get install mysql-server*. Secara otomatis *package* dan *library dependencies* akan terinstall pada sistem linux. Proses instalasi ditunjukkan seperti Gambar 3.14.



Gambar 3.14. Proses Instalasi MySQL-Server

2. Instalasi php5

   Langkah instalasi php5 dilakukan dengan mengetikkan perintah *#apt-get install php5.* Kemudian ketik y untuk konfirmasi instalasi seperti ditunjukkan pada Gambar 3.15.

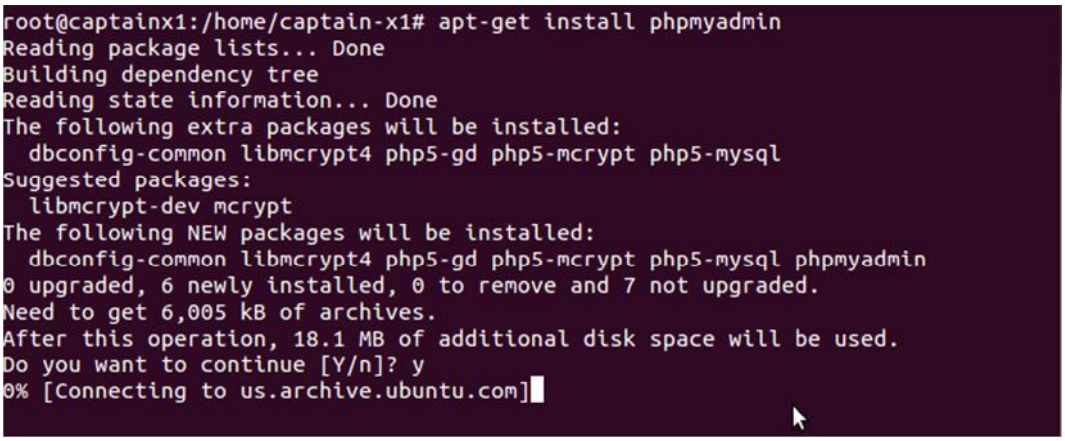


Gambar 3.15. Proses Instalasi PHP5

3. Instalasi phpmyadmin

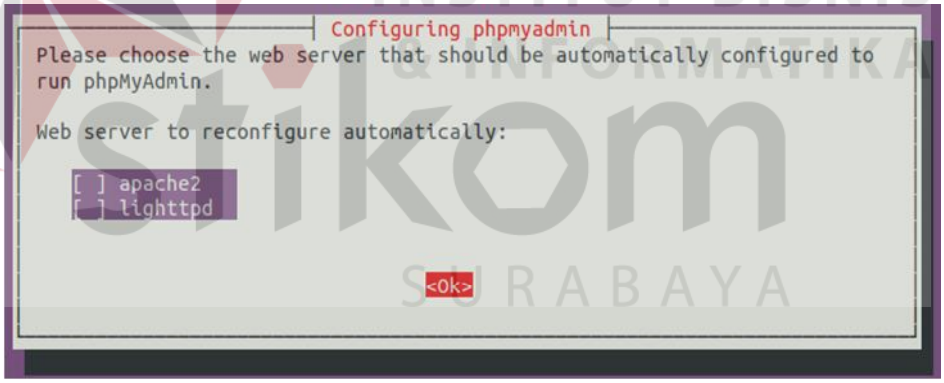
   Phpmyadmin digunakan untuk memudahkan memonitor dan memanajemen *database* snort, Langkah instalasi dengan mengetikkan perintah *#apt-get*

install phpmyadmin. Kemudian ketik *y* untuk konfirmasi instalasi seperti ditunjukkan pada Gambar 3.16.



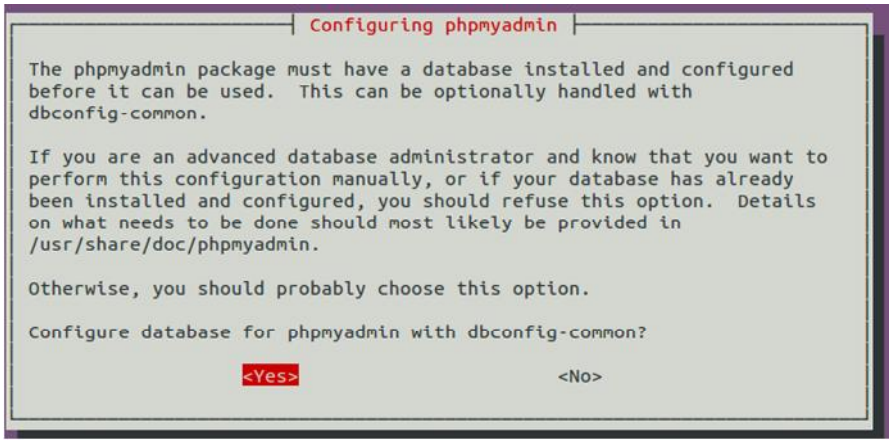
Gambar 3.16. Konfirmasi Instalasi Phpmyadmin

Setelah proses instalasi selesai akan muncul tampilan antarmuka konfigurasi web server. Pilih Apache2 sebagai web server. Proses ditunjukkan pada Gambar 3.17.



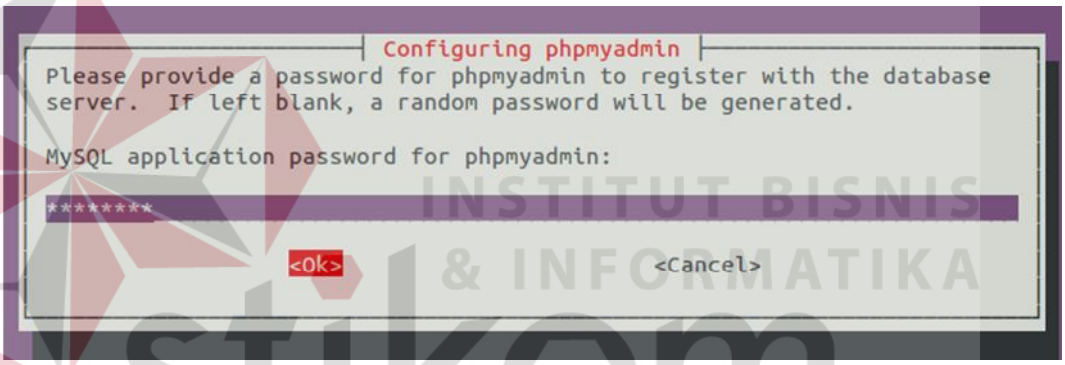
Gambar 3.17 Pemilihan Web Server

Proses selanjutnya adalah pemilihan *auto configure* atau *manual configure database* mysql. Untuk memudahkan set up *database* pilih yes. Proses ditunjukkan seperti pada Gambar 3.18.



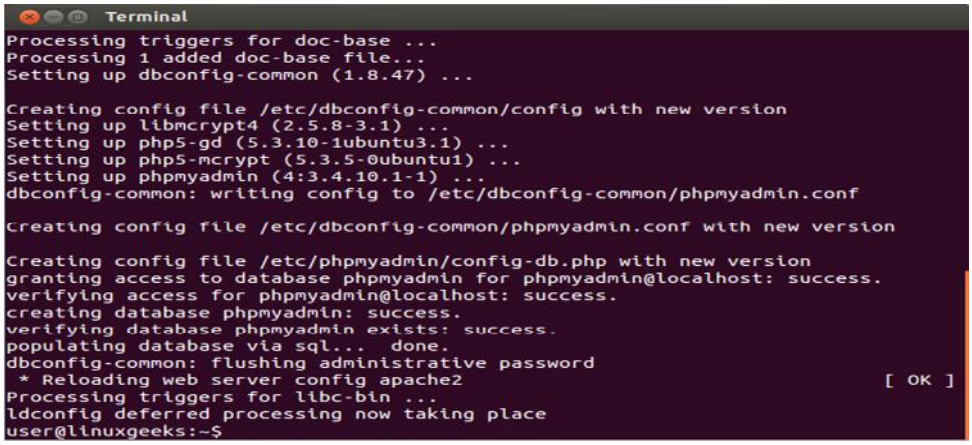
Gambar 3.18 Pemilihan auto konfigurasi *database*

Proses berikutnya adalah memasukkan password administrasi *database* pengguna. Proses ditunjukkan seperti pada Gambar 3.19



Gambar 3.19 Password MySQL untuk phpmyadmin

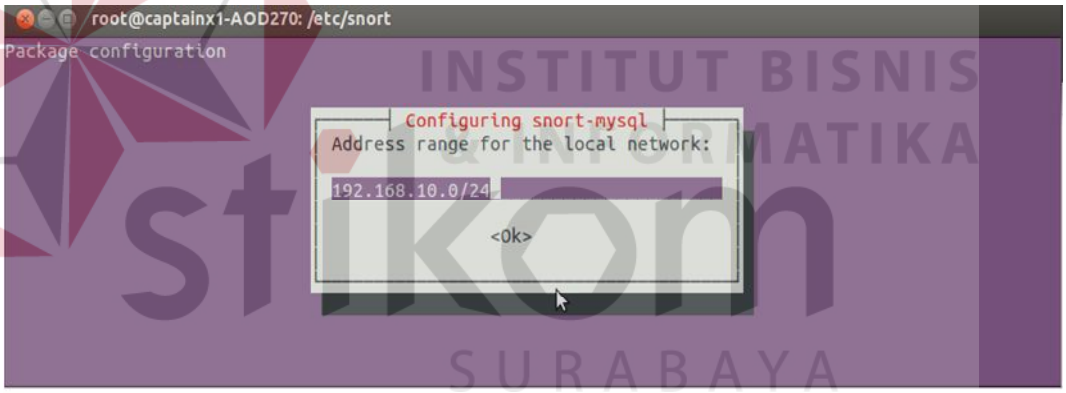
Proses pada Gambar 3.19 merupakan tahapan akhir konfigurasi phpmyadmin. Gambar 3.20 menunjukkan akhir proses instalasi dan konfigurasi phpmyadmin.



Gambar 3.20. Proses Akhir Instalasi dan Konfigurasi Phpmyadmin

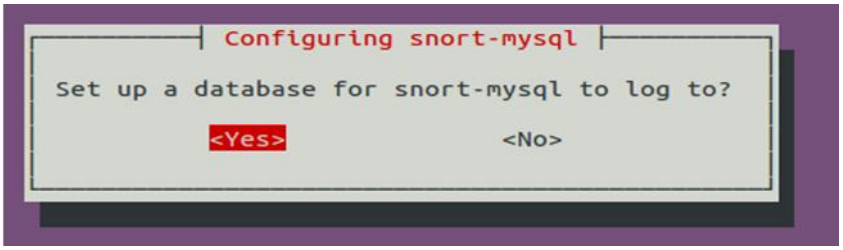
4. Instalasi snort-mysql

Untuk instalasi snort-mysql dapat menggunakan *console* dengan perintah *#apt-get-install snort-mysql.* Selanjutnya akan muncul jendela untuk mengisi *address range* jaringan lokal.



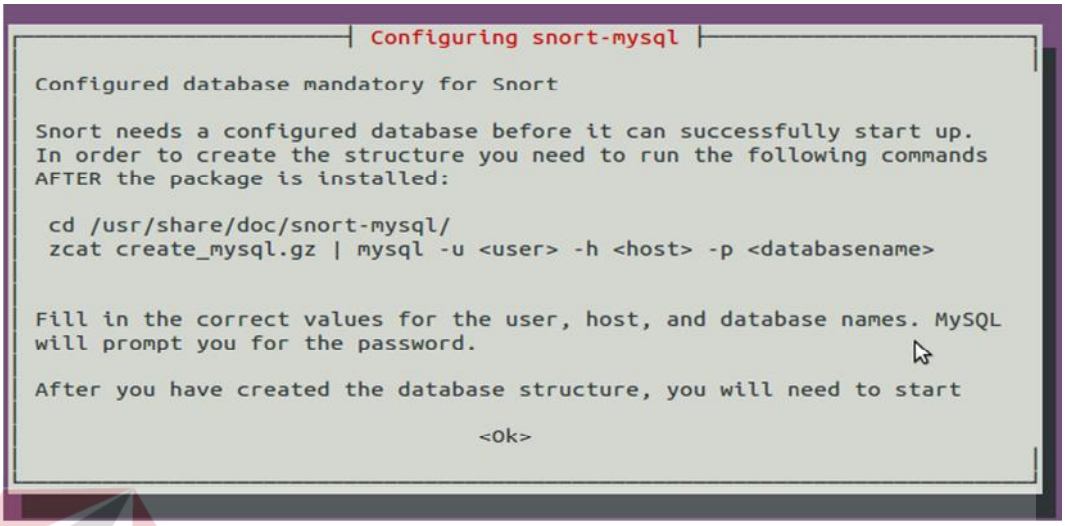
Gambar 3.21. Konfigurasi IP Address Local Network

Gambar 3.21 merupakan konfigurasi IP Address jaringan lokal, pada sistem IDS ini alamat jaringan lokal adalah 192.168.10.0/24. Artinya snort akan memantau jaringan lokal 192.168.10.0/24.



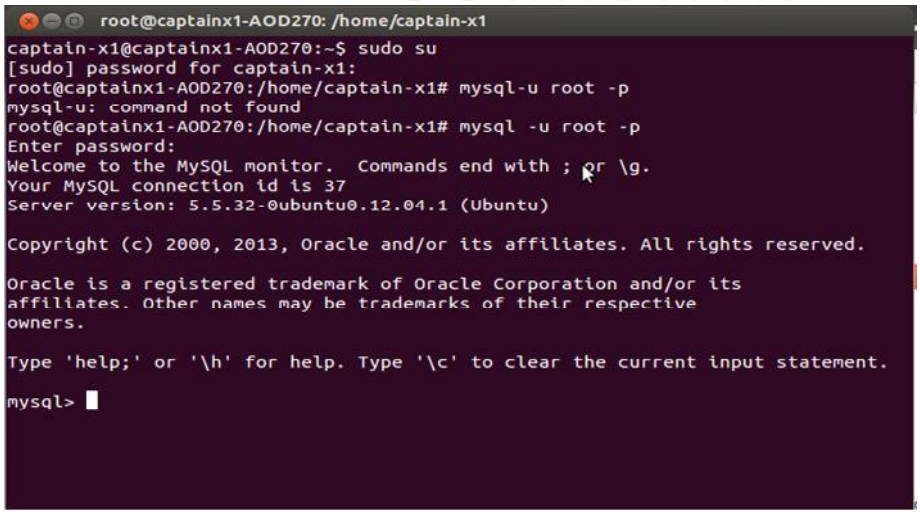
Gambar 3.22. Konfirmasi Set Up *database* snort-mysql

Gambar 3.22 merupakan konfirmasi untuk set up *database* snort-mysql. Pilih <*Yes*> untuk melanjutkan konfigurasi *database*.



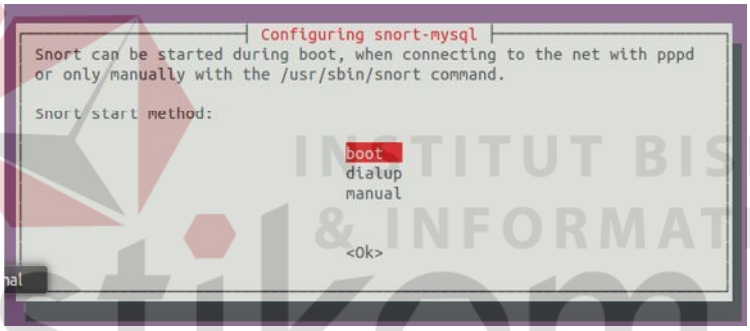
Gambar 3.23. Tahap akhir Package Configuration

Gambar 3.23 merupakan tahap akhir *package configuration*. Pilih <*Ok*> untuk mengakhiri konfigurasi paket. Langkah selanjutnya adalah membuat *database* snort melalui mysql command prompt. *Log*in pada terminal dengan mengetikkan perintah *#mysql –u root –p,* selanjutnya muncul pesan *Enter password:*. Jika password telah benar dimasukkan maka dapat masuk *mysql command promt* seperti Gambar 3.24.



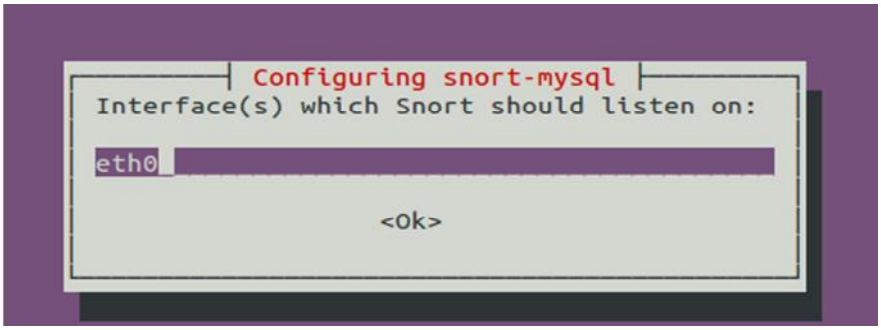
Gambar 3.24. MySQL Command Prompt

Gambar 3.24 menunjukkan tampilan *mysql command* prompt yang berguna untuk menjalankan operasi mysql. Untuk membuat *database* snort dilakukan melalui perintah *#create database snort;*. Langkah selanjutnya adalah kembali keluar dari mysql *prompt* dengan perintah *#quit;* sehingga akan kembali pada *shell* linux. Setelah *database* berhasil dibuat langkah selanjutnya adalah memasukkan skema *database* snort pada *database* snort-mysql dengan perintah *#zcat create /usr/share/doc/snort-mysql/create_mysql.gz | mysql –u root –p snort.* Langkah selanjutnya adalah melakukan konfigurasi ulang *database* snort-mysql dengan cara mengetikkan perintah *#dpkg-reconfigure snort-mysql*.



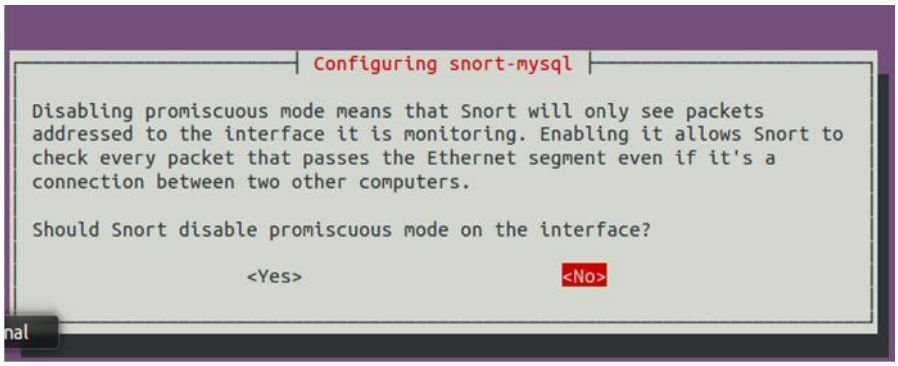
Gambar 3.25. Snort Start Method

Gambar 3.25 adalah pemilihan metode kapan *service* snort dijalankan. Metode yang dipilih adalah *boot*. Artinya pada saat komputer dihidupkan secara otomatis snort akan dijalankan, sehingga user tidak perlu untuk memulai *service* snort secara manual.



Gambar 3.26. Konfigurasi Interface Snort

Gambar 3.26 merupakan konfigurasi *interface* snort yang digunakan untuk mendengarkan paket data pada suatu *host*. Interface yang dipilih adalah eth0.



Gambar 3.27. Mengaktifkan Snort dalam Mode Promiscuous

Gambar 3.27 merupakan proses untuk mengaktifkan snort dalam mode *promiscuous*. Mode *promiscuous* membuat snort mampu mendengarkan paket yang melewati segmen Ethernet dalam koneksi antar 2 host komputer lainnya. Langkah terakkhir adalah menghilangkan file *db-pending-config* pada *database* snort, agar snort dapat berjalan dengan baik. Perintah untuk menghapus file adalah *#rm –rf /etc/snort/db-pending-config*

5. Menjalankan Snort

Untuk menjalankan snort pertama kali dilakukan dengan mengetik perintah *#/etc/init.d/snort start*. Untuk selanjutnya snort akan berjalan otomatis pada saat booting komputer. Untuk melihat apakah snort sudah jalan atau belum adalah dengan menjalankan perintah *#ps -aux | grep snort* seperti yang ditunjukkan pada Gambar 3.28.

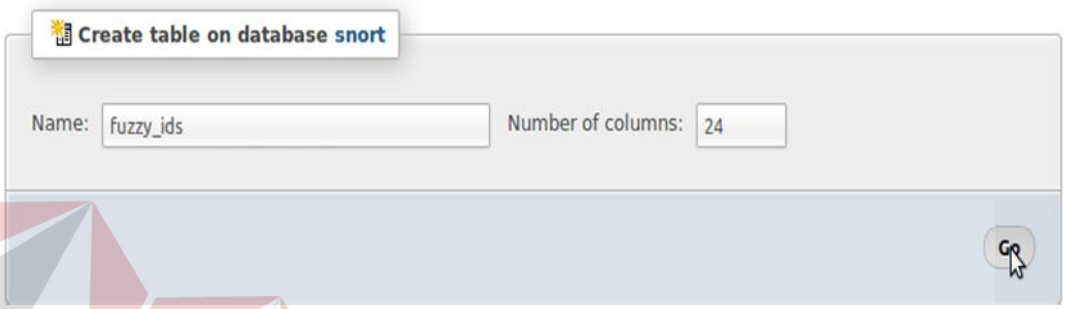
```
root@captainx1:/usr/share/doc/snort-mysql# ps -aux | grep snort
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html
snort     6417  0.1  7.8 352756 80164 ?        Ssl  23:37   0:01 /usr/sbin/snort
 -m 027 -D -d -l /var/log/snort -u snort -g snort -c /etc/snort/snort.conf -S HO
ME_NET=[192.168.10.0/24] -i eth0
root      6769  0.0  0.0   4388   836 pts/1    S+   23:57   0:00 grep --color=au
to snort
```

Gambar 3.28. Pengecekan Service Snort

Seperti yang ditunjukkkan oleh Gambar 3.28 *service* snort telah berjalan dalam mode *daemon*.

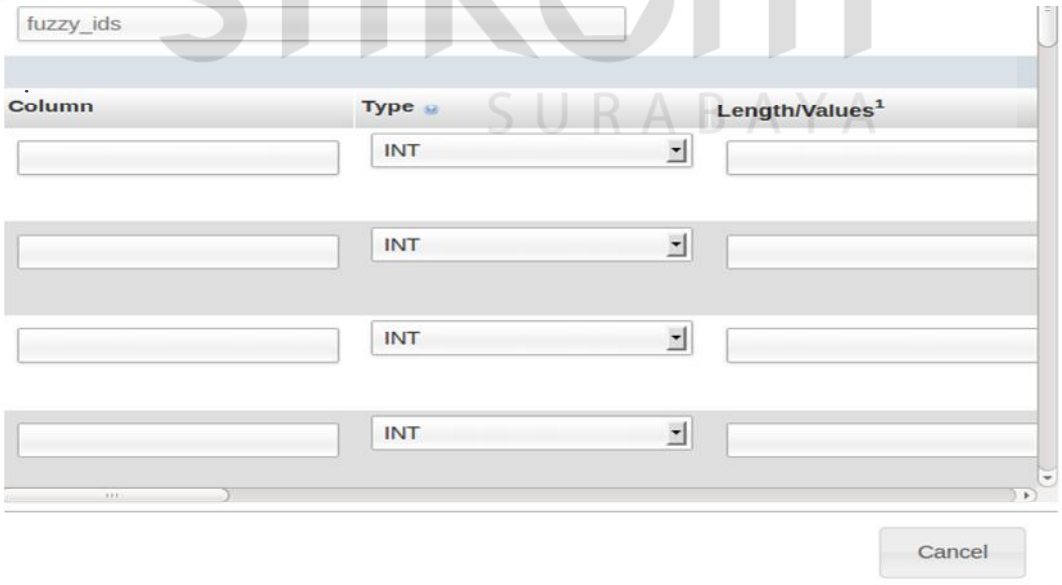
6. Membuat tabel Fuzzy

Secara *default* tabel snort-mysql hanya terdiri dari 22 tabel. Tabel fuzzy berisi 26 kolom yang berisikan data variabel fuzzy dan klasifikasi serangan. Untuk membuat tabel fuzzy dapat dilakukan melalui phpmyadmin.



Gambar 3.29. Membuat Tabel Baru

Tabel fuzzy dibuat dengan cara mengisikan nama tabel dan jumlah kolom, kemudian klik *Go*. Seperti yang ditunjukkan pada Gambar 3.29 nama tabel adalah *fuzzy_ids* dan jumlah kolom adalah 26.



Gambar 3.30. Pengisian Nama Kolom

Gambar 3.30 merupakan halaman pengisian nama kolom tabel beserta dengan atributnya. Setelah melakukan pengisian secara lengkap klik *Save* untuk menyimpan tabel fuzzy. Berikut ini adalah struktur tabel fuzzy yang telah dibuat.

Tabel 3.4. Struktur Tabel fuzzy_ids

Column	Type
Sid	int(10)
cid	int(10)
Id	int(10)
Timestamp	Datetime
ip_src	int(10)
ip_dst	int(10)
Frekuensi	int(10)
Iplen	smallint(10)
ip_proto	int(10)
Alphaflow	Float
Alphafmid	Float
Alphallow	Float
Alphalmid	Float
Alphalhigh	Float
rule1	Float
rule2	Float
rule3	Float
rule4	Float
rule5	Float
rule6	Float
rule7	Float
rule8	Float
rule9	Float
Attack	Float
attack_kategori	varchar(40)

Dimana:

Alphaflow : alpha fuzzy frekuensi low

Alphafmid : alpha fuzzy frekuensi middle

Alphafhigh : alpha fuzzy frekuensi high

Alphallow : alpha fuzzy length low

Alphalmid : alpha fuzzy length middle

Alphalhigh : alpha fuzzy length high

Rule 1 -9 : Aturan fuzzy

Tabel 3.4 merupakan struktur tabel *fuzzy_ids,* kolom *sid* dan *cid* merupakan *foreign key* dari tabel *acid_event,* sedangkan id menjadi *primary key* dari tabel fuzzy_ids. Kolom *timestamp, ip_src, ip_dst* dan *ip_proto* berisikan data yang sama dengan tabel *acid_event*. Kolom *frekuensi* berisi jumlah banyaknya paket data persatuan detik, kolom *ip_len* berisi banyaknya jumlah data persatuan detik. Kolom *alphaflow, alphafmid, alphafhigh, alphallow, alphalmid* dan *alphalhigh* berisi perhitungan alpha fuzzy dari variabel frekuensi dan *length*. Kolom *rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8* dan *rule9* berisi perhitungan nilai z_x aturan fuzzy. Kolom *attack* berisi perhitungan defuzzyfikasi dan kolom *attack_kategori* berisi klasifikasi serangan.

3.4.2 Konfigurasi IP *Address*

Dalam pengujian sistem IDS berdasarkan metode fuzzy ini terdapat 2 konfigurasi utama yaitu konfigurasi ip address router dan ip *address host*.Berikut ini konfigurasi masing-masing ip *address*:

1. Konfigurasi ip *address* router

   Konfigurasi ip *address* router dilakukan dengan mengedit file */etc/network/interfaces,* dengan mengetikkan perintah *#gedit /etc/network/interfaces.* Kemudian mengisi konfigurasi file dibawah ini:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.10.11
```

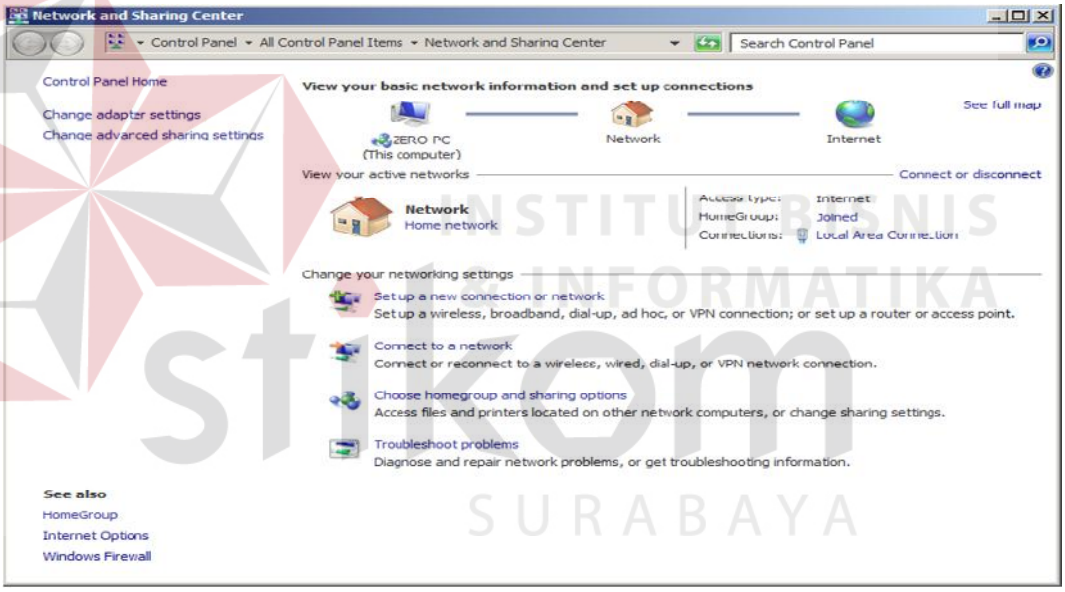
```
netmask 255.255.255.0
network 192.168.10.0
broadcast 192.168.10.0
```

Kemudian klik File – Save pada *window* gedit, selanjutnya jalankan perintah *#/etc/init.d/networking restart*

2. Konfigurasi ip address client

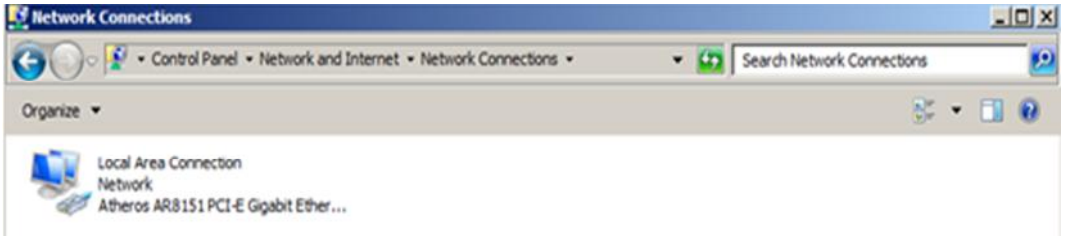
   Adapun sistem operasi *client* adalah Windows 7, oleh karena itu pengaturan ip *address* dilakukan dengan cara:

   Klik kanan pada *icon network* pilih *Network and Sharing Center,* seperti yang ditunjukkan pada Gambar 3.31.



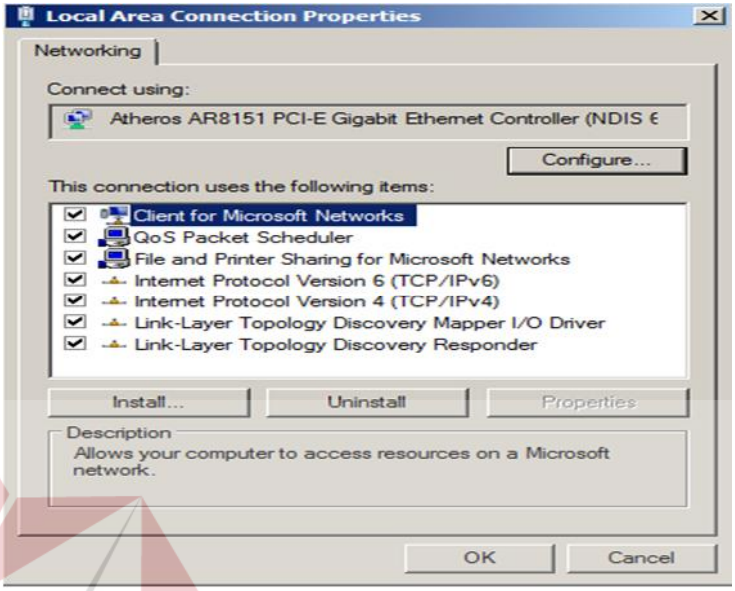
Gambar 3.31. Network and Sharing Center

Klik *Change Adapter Setting* sehingga muncul jendela *Network Connection* seperti yang ditunjukkan pada Gambar 3.32.



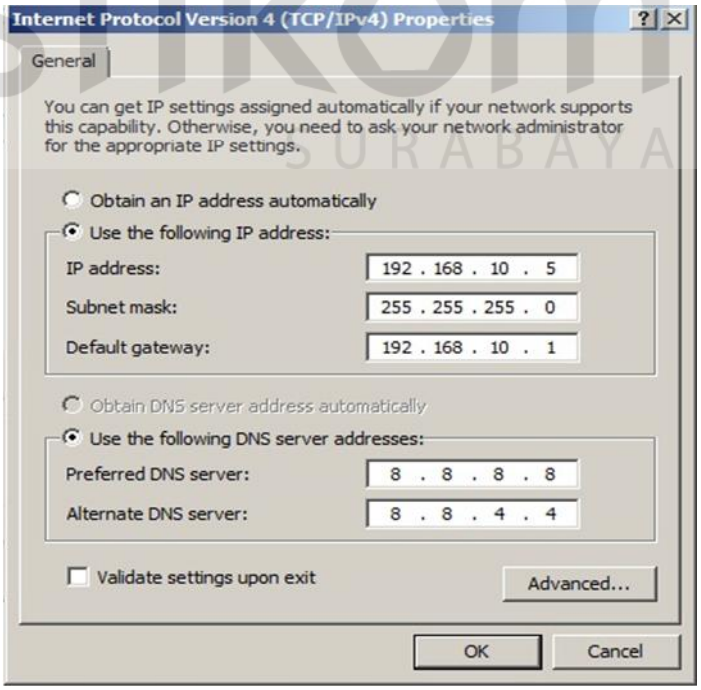
Gambar 3.32 Jendela Network Connection

Klik kanan pada *icon Local Area Connection* pilih Properties, sehingga muncul jendela *Local Area Connection Properties* seperti yang ditunjukkan pada Gambar 3.33.



Gambar 3.33. Jendela Local Area Connection Properties

Kemudian klik *Internet Protocol Version 4 (TCP/IPv4),* pilih *Properties* sehingga muncul jendela baru seperti pada Gambar 3.34.



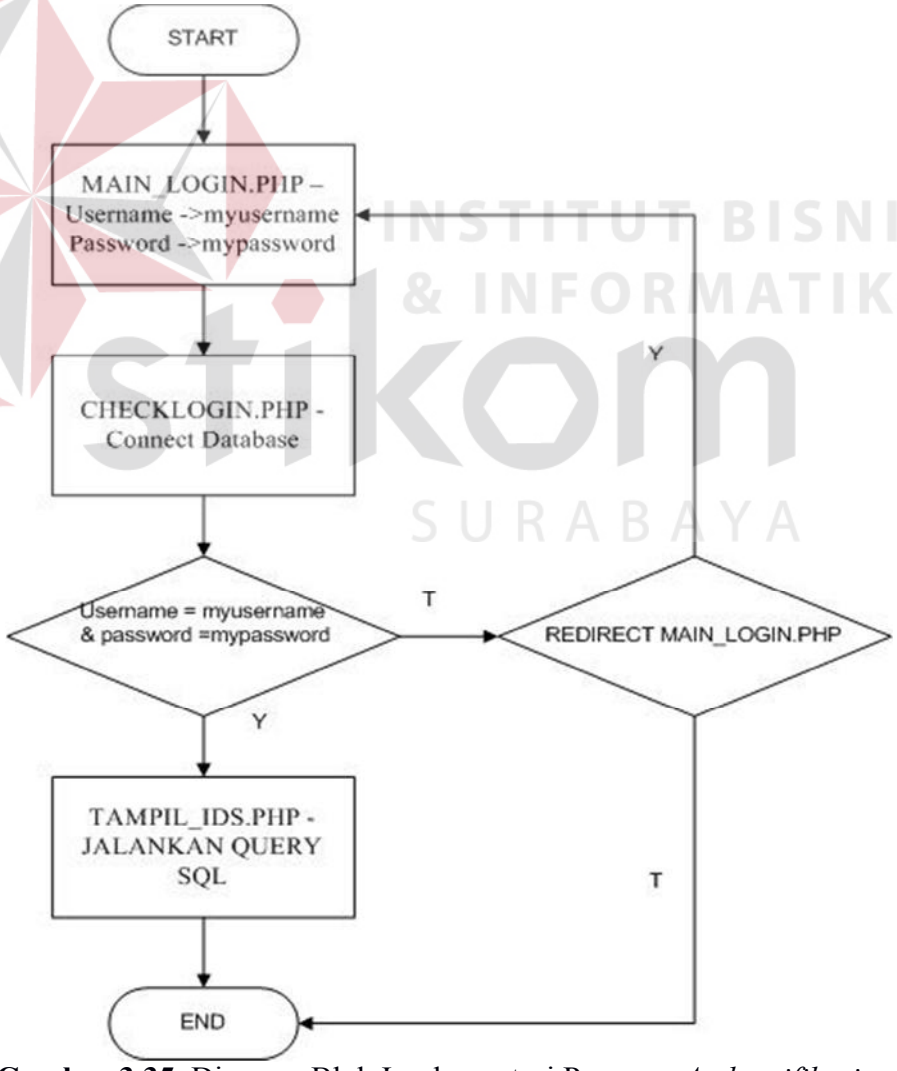
Gambar 3.34. Konfigurasi IP *Address*

Konfigurasi ip yang ditunjukkan oleh Gambar 3.34 terdiri dari beberapa bagian yaitu ip *address* dengan alamat 192.168.10.5. *subnet mask* 255.255.255.0, *default gateway* 192.168.10.12. Konfirmasi ip *address* dapat dilakukan dengan menekan tombol *OK*.

3.4.3 Penulisan Program

A. Program *Authentifikasi*

Berdasarikan alur diagram yang ditunjukkan pada Gambar 3.12 dibuat diagram blok implementasi program yaitu:



Gambar 3.35. Diagram Blok Implementasi Program *Authentifikasi*

Gambar 3.35 menunjukkan *username* dan *password* disimpan dalam variabel *myusername* dan *mypassword* pada halaman web *main_login.php* . *Script* program tersebut adalah sebagai berikut:

```
<input name="myusername" type="text" id="myusername">
<input name="mypassword" type="password" id="mypassword">
```

Selanjutnya program akan memverifikasi *username* dan *password* pada halaman *checklogin.php*. Verifikasi dimulai dengan membuka koneksi dengan *database* terlebih dahulu, yaitu dengan *script:*

```
$host="localhost";
$username="root";
$password="password";
$db_name="login";
$tbl_name="login_ids";
mysql_connect("host","$username","$password")or        die("cannot
connect");
mysql_select_db("&db_name")or die("cannot select DB");
```

Setelah koneksi berhasil selanjutnya langkah yang dilakukan adalah dengan mendefinisikan *username* dan *password* ke dalam metode POST dengan *script:*

```
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];
```

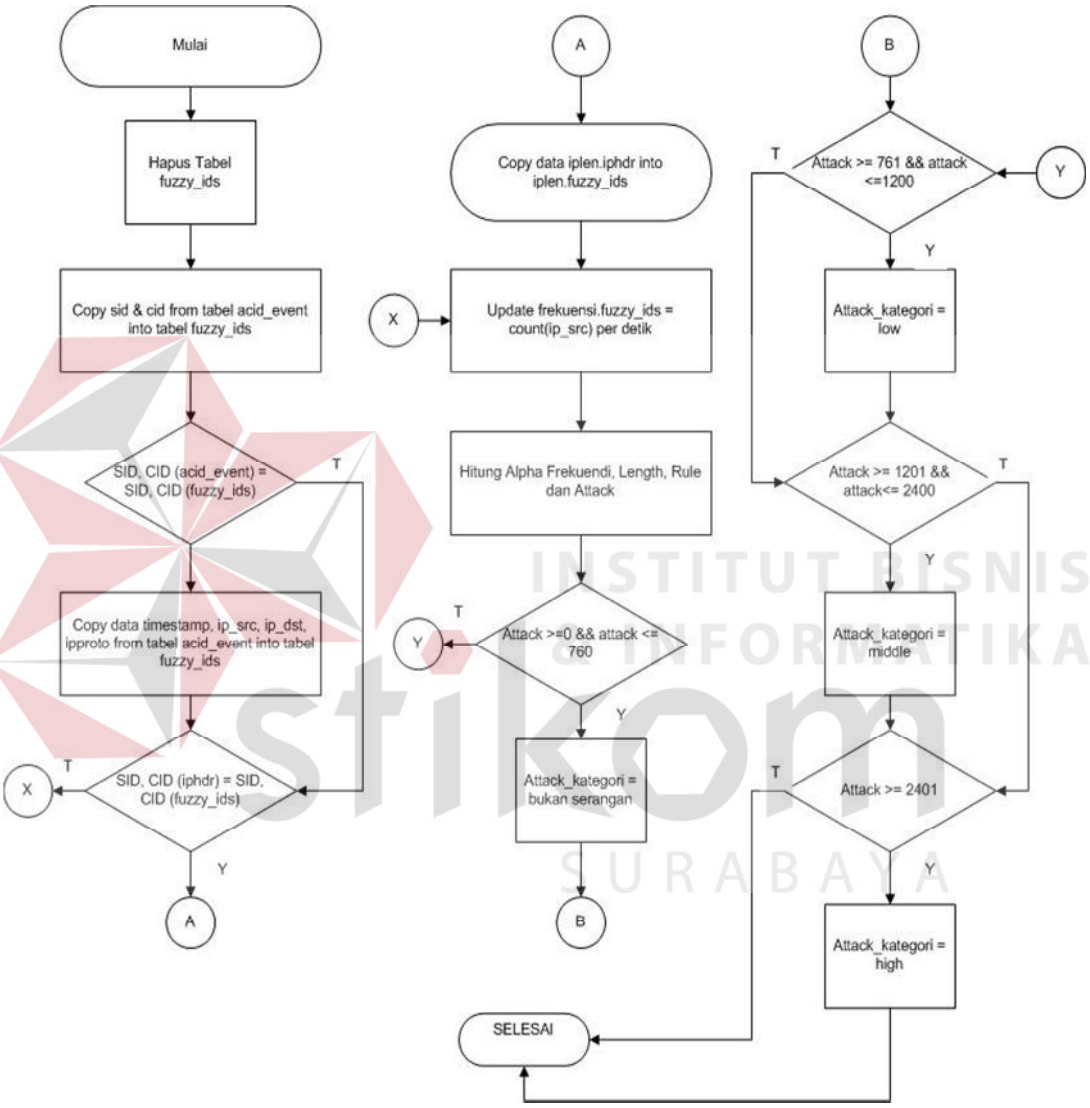
Pengecekan *username* dan *password* dilakukan dengan *script:*

```
$sql="SELECT * FROM $tbl_name WHERE username='$myusername' and
password='$mypassword'";
$count=mysql_num_rows($result);
if($count==1){
session_register("myusername");
session_register("mypassword");
header("location:tampil_ids.php");
}
else{
echo "Wrong Username or Password";
}
```

Jika *myusername* dan *mypassword* sama dengan data *username* dan *password* pada tabel *login_ids* secara otomatis *user* akan diarahkan pada halaman web tampil_ids.php, sebaliknya jika salah akan ditampilkan pesan *Wrong Username or Password.*

B. Program Fuzzy

Berdasarkan alur kerja sistem IDS yang ditunjukkan pada Gambar 3.1 dibuat blok diagram implementasi program seperti yang ditunjukkan pada Gambar 3.36.



Gambar 3.36. Diagram Blok Penulisan Program

Dari Gambar 3.36 terdapat beberapa langkah *sequensial* penulisan program, adapun langkah-langkah tersebut adalah.

1. Langkah pertama yang dilakukan program adalah menghapus isi tabel fuzzy_ids, langkah ini dilakukan agar tidak terjadi *duplicate entries* pada saat

proses update kolom *sid, cid, timestamp, ip_src* dan *ip_dst. Script* program dari hapus tabel adalah.

```
$qry= "truncate fuzzy_ids";
$sql = mysql_query($qry);
```

2. Langkah selanjutnya mengkopikan kolom *sid, cid, timestamp, ip_src* dan *ip_dst* dari tabel acid_event ke dalam tabel fuzzy_ids, *script* program tersebut adalah:

```
$qry2= "insert into fuzzy_ids (sid,cid,timestamp,ip_src,ip_dst)
select sid,cid,timestamp,ip_src,ip_dst from acid_event";
$sql = mysql_query($qry2);
```

3. *Update* kolom frekuensi dengan *script:*

```
$qry3= "update fuzzy_ids as t left join (select
sid,cid,timestamp,count(ip_src) as NumEvents from acid_event
group by timestamp) as m on m.sid = t.sid and m.cid = t.cid and
m.timestamp = t.timestamp set t.frekuensi = m.NumEvents where
t.id is not null";
$sql = mysql_query($qry3);
```

4. *Update* kolom *iplen* dengan *script:*

```
$qry4= "update fuzzy_ids inner join iphdr on
fuzzy_ids.sid=iphdr.sid and fuzzy_ids.cid = iphdr.cid set
fuzzy_ids.iplen = iphdr.ip_len";
$sql = mysql_query($qry4);
```

5. Hitung alpha frekuensi dan alpha length dengan *script:*

```
$qry5= "update fuzzy_ids left join (select frekuensi,
if(frekuensi >=0 and frekuensi <=6, 1, if(frekuensi >=6 and
frekuensi <=8, (8 - frekuensi)/2,9)) as alpha_flow from
fuzzy_ids ) as alpha using (frekuensi) set alphaflow =
alpha_flow";
$sql = mysql_query($qry5);
$qry8= "update fuzzy_ids left join (select iplen, if(sum(iplen)
>=190 and sum(iplen) <=220, 1, if(sum(iplen) >=200 and
sum(iplen) <=220, (220 - sum(iplen))/20,9)) as alpha_llow from
fuzzy_ids where ip_dst = 3232238091 group by timestamp ) as
alpha using (iplen) set alphallow = alpha_llow";
$sql = mysql_query($qry8);
```

6. *Update* rule fuzzy dengan *script:*

```
$qry11= "update fuzzy_ids left join (select alphafhigh,
alphalhigh, least (alphafhigh,alphalhigh) as rule_1 from
fuzzy_ids) as nilairule1 using (alphafhigh,alphalhigh) set
rule1 =rule_1";
$sql = mysql_query($qry11);
```

7. Update nilai defuzzyfikasi attack dengan *script:*

```
$qry20= ”update fuzzy_ids left join (select rule1, rule2,
rule3, rule4, rule5, rule6, rule7, rule8, rule9, (rule1*2400 +
rule2*2400 + rule3*2400 +rule4*2400 + rule5*1760 +rule6*1760
+rule7*1760 + rule8*1760 + rule9*1200) /
(rule1+rule2+rule3+rule4+rule5+rule6+rule7+rule8+rule9) as
tempcount from fuzzy_ids) as totalcount using
(rule1,rule2,rule3,rule4,rule5,rule6,rule7,rule8,rule9 set
attack=tempcount ";
$sql = mysql_query($qry20);
```

8. *Update* kategori attack dengan *script:*

```
$qry21 = “ update fuzzy_ids set attack_kategori=if((attack) =0
, \"bukan serangan\",if ((attack) > 0 and (attack)
<=1200,\"low\", if ((attack) >1200 and (attack)
<2400,\"middle\",\"high\")))";
$sql = mysql_query($qry21);
```