

SISTEM IDENTIFIKASI PERSONAL KOMPUTER BERBASIS *NETWORK ADMISSION CONTROL* (NAC) MENGUNAKAN JAVA

Dwi Anggara Aditya ¹, Idris Winarno ²

Mahasiswa Jurusan Teknologi Informasi ¹, Dosen Pembimbing ²

Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember

Kampus PENS-ITS Keputih Sukolilo Surabaya 60111

Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114

Email : gandes_pa22@yahoo.com

Makalah Proyek Akhir

Abstrak

Makalah ini berisi pembahasan mengenai permasalahan akses internet dewasa ini yang sangat tinggi, hal ini mengakibatkan peningkatan permintaan akses ke jaringan yang aman semakin tinggi. Keadaan ini menuntut admin jaringan agar lebih selektif dalam memperbolehkan user melakukan akses ke jaringan. Dari permasalahan tersebut maka timbulah suatu metode untuk menanggulangnya yaitu dengan menggunakan metode *Network Admission Control* (NAC). Aplikasi ini bertujuan untuk membuat sistem keamanan yang lebih handal dan mudah, dikarenakan sistem dapat melakukan deteksi identitas sebuah personal komputer yang mendukung adanya *dynamic vlan*. Secara garis besar aplikasi ini dapat melakukan perubahan vlan pada switch *cisco* dilakukan oleh aplikasi java dengan memanfaatkan *library* dari perl dan membuat user yang berkepentingan dapat mengakses dan memperoleh hak akses sesuai aturan yang ada pada jaringan tersebut sehingga dapat menekan resiko terjadinya ilegal user yang dapat membahayakan dan merugikan orang lain.

Kata kunci: dynamic vlan, Network Admission Control, cisco, library

Abstract

This paper contains a discussion of internet access problem today is very high, this resulted in increased demand for access to a secure network is getting higher. This situation requires the network admin to be more selective in allowing users to access to the network. To these problems is a method to overcome that is by using the method of Network Admission Control (NAC). The application aims to make security systems more reliable and easier, because the system can detect the identity of a personal computer that supports the dynamic vlan. Generally these applications can make a change vlan on cisco Switch by java application using the perl libraries and make the interested users can access and obtain access rights according to existing rules on the network so as to reduce the risk of the illegal user can be dangerous and expense of others.

Keyword: dynamic vlan, Network Admission Control, cisco, library

1. PENDAHULUAN

Kebutuhan akan akses internet dewasa ini sangat tinggi, hal ini mengakibatkan peningkatan permintaan akses ke jaringan yang aman semakin tinggi. Keadaan ini menuntut admin jaringan agar lebih selektif dalam memperbolehkan user melakukan akses ke jaringan. Setelah proses seleksi awal pada user admin jaringan juga bertugas untuk memproteksi user dari gangguan yang dilakukan user lain atau dari akses luar jaringan. [1]

Masalah-masalah yang timbul dalam menjaga suatu jaringan diantaranya adalah : Pertama, metode dan teknologi yang digunakan oleh virus, worm, dan program penyerang terus berkembang. Kedua adalah kelemahan yang dimiliki pada suatu software dan kompleksitas dari internet membuat admin sulit melakukan perlindungan terhadap sistem. [2] Ketiga adalah kesalahan yang dilakukan oleh user seperti tidak melakukan update antivirus, kurangnya pengetahuan user sehingga membuka attachment yang berbahaya, dan Keempat

adalah tidak semua user mempunyai hak mengakses jaringan sehingga perlu dipastikan bahwa user yang mengakses jaringan adalah orang-orang yang memang mendapatkan hak akses. Dari permasalahan-permasalahan yang ada tersebut maka timbulah suatu metode untuk menanggulanginya yaitu dengan menggunakan metode *Network Admission Control* (NAC).

Network Admission Control (NAC) yaitu membatasi akses ke jaringan berdasarkan identitas atau keamanan. Ketika sebuah perangkat jaringan (switch, router, jalur akses, server DHCP, dll) dikonfigurasi untuk NAC, sehingga dapat memaksa otentikasi pengguna atau mesin sebelum pemberian akses ke jaringan. Selain itu, akses tamu dapat diberikan kepada daerah karantina untuk perbaikan masalah apapun yang mungkin mengakibatkan kegagalan otentikasi. Hal ini dilaksanakan melalui kebiasaan inline perangkat jaringan, perubahan pada router, atau terbatas kelas DHCP. Tipikal (non-free) koneksi WiFi adalah suatu bentuk NAC. Pengguna harus ada semacam surat-surat (atau kartu kredit) sebelum diberikan akses ke jaringan. [3].

Rumusan Permasalahan

Permasalahan pada proyek akhir ini adalah bagaimana membangun sistem keamanan yang dapat melakukan identifikasi personal komputer sehingga orang yang berkepentingan dapat mengakses dan memperoleh hak akses sesuai aturan yang ada pada jaringan tersebut Untuk itu dari permasalahan tersebut akan dibahas hal-hal penting berikut :

1. Bagaimana melakukan sinkronisasi antara Server dengan Client ketika kita melakukan pendeteksian mac address client.
2. Bagaimana melakukan pendeteksian port client berdasarkan mac-address client yang ada pada switch cisco.
3. Bagaimana melakukan perubahan switch vlan berdasarkan hak akses client pada switch cisco.

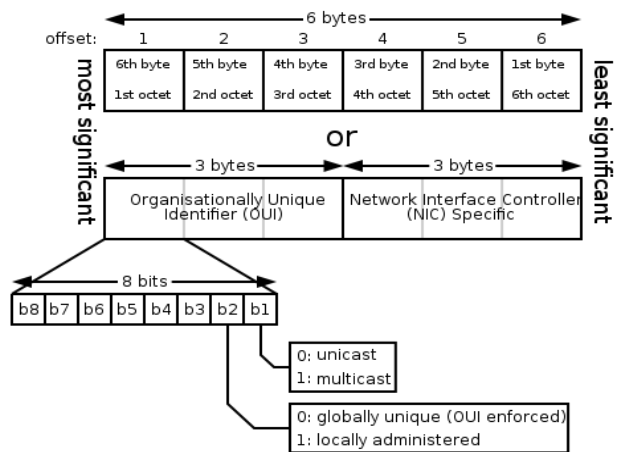
Tujuan

Tujuan dari pembuatan proyek akhir ini adalah Membuat sistem keamanan yang lebih handal dan mudah, dikarenakan sistem dapat melakukan deteksi identitas sebuah personal komputer yang mendukung adanya dinamik Vlan

2. TINJAUAN PUSTAKA

2.1 MAC ADDRESS

MAC Address (Media Access Control Address) adalah sebuah alamat jaringan yang diimplementasikan pada lapisan data-link dalam tujuh lapisan model OSI, yang merepresentasikan sebuah node tertentu dalam jaringan. Dalam sebuah jaringan berbasis Ethernet, MAC address merupakan alamat yang unik yang memiliki panjang 48-bit (6 byte) yang mengidentifikasi sebuah komputer, interface dalam sebuah router, atau node lainnya dalam jaringan. MAC Address juga sering disebut sebagai Ethernet address, physical address, atau hardware address.



Gambar 2.1 MacAddress

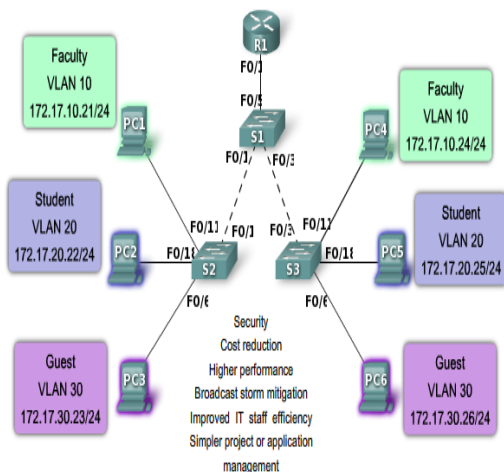
MAC Address mengizinkan perangkat-perangkat dalam jaringan agar dapat berkomunikasi antara satu dengan yang lainnya. Sebagai contoh, dalam sebuah jaringan berbasis teknologi Ethernet, setiap header dalam frame Ethernet mengandung informasi mengenai MAC address dari komputer sumber (*source*) dan MAC address dari komputer tujuan (*destination*). Beberapa perangkat, seperti halnya bridge dan switch Layer-2 akan melihat pada informasi MAC address dari komputer sumber dari setiap frame yang ia terima dan menggunakan informasi MAC address ini untuk membuat "tabel routing" internal secara dinamis. Perangkat-perangkat tersebut pun kemudian menggunakan tabel yang baru dibuat itu untuk meneruskan frame yang ia terima ke sebuah port atau segmen jaringan tertentu di mana komputer atau node yang memiliki MAC address tujuan berada.

2.2 VLAN

Kinerja sebuah jaringan sangat dibutuhkan oleh organisasi terutama dalam hal kecepatan dalam pengiriman data. Salah satu kontribusi teknologi untuk meningkatkan kinerja jaringan adalah dengan kemampuan untuk membagi sebuah broadcast domain yang besar menjadi beberapa broadcast domain yang lebih kecil dengan menggunakan vlan. Broadcast domain yang lebih kecil akan membatasi device yang terlibat dalam aktivitas broadcast dan membagi device ke dalam beberapa grup berdasar fungsinya, seperti layanan database untuk unit akuntansi, dan data transfer yang cepat untuk unit teknik.

Teknologi vlan (Virtual Local Area Network) bekerja dengan cara melakukan pembagian network secara logika ke dalam beberapa subnet. Vlan adalah kelompok device dalam sebuah LAN yang dikonfigurasi (menggunakan software manajemen) sehingga mereka dapat saling berkomunikasi asalkan dihubungkan dengan jaringan yang sama walaupun secara fisik mereka berada pada segmen LAN yang berbeda. Jadi vlan dibuat bukan berdasarkan koneksi fisik namun lebih pada koneksi logikal, yang tentunya lebih fleksibel. Secara logika, vlan membagi jaringan ke dalam beberapa subnetwork. vlan memungkinkan banyak subnet dalam jaringan yang menggunakan switch yang sama.

Dengan menggunakan vlan, kita dapat melakukan segmentasi jaringan switch berbasis pada fungsi, departemen atau pun tim proyek. Kita dapat juga mengelola jaringan kita sejalan dengan kebutuhan pertumbuhan perusahaan sehingga para pekerja dapat mengakses segmen jaringan yang sama walaupun berada dalam lokasi yang berbeda.

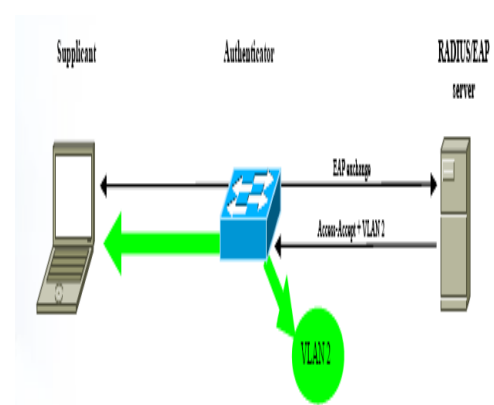


Gambar 2.2. Contoh penerapan teknologi VLAN

2.3 DYNAMIC VLAN

Di samping sebagai autentikasi, fitur dari IEEE 802.1X adalah vlan dinamis. Dengan mencoba masuk ke server Radius, dimana server radius

menerima akses dari user yang melakukan autentikasi. Paket juga meliputi atribut indikasi dimana vlan mengarahkan ke port yang sesuai dengan user autentikasi. Sebagai contoh, seorang mahasiswa login ke jaringan menggunakan autentikasi sebagai mahasiswa maka portnya diarahkan ke 'vlan Mahasiswa'; seorang staf login ke jaringan yang sama sehingga portnya diarahkan ke 'vlan staf'. Jika tak seorangpun login ke terminal, maka port diarahkan ke 'vlan default' seorang admin jaringan memiliki akses ke port 'vlan maintenance' untuk memperbaiki perangkat lunak dan melakukan remote jaringan



Gambar 2.3 Dynamic Vlan

dari gambar 2.3 dapat di jelaskan sebagai berikut :

Supplicant atau disebut user yang mengakses suatu jaringan dimana authenticator berupa switch vlan mengirimkan parameter dari supplicant tersebut ke sebuah server kemudian dari server di identifikasi sehingga memperoleh Access-Accept untuk diarahkan ke Vlan 2 sehingga supplicant tersebut secara otomatis memperoleh hak akses ke Vlan 2.

2.4 JAVA SOCKET

Sebuah socket merupakan IP Port pada sebuah host mesin yang esifik. Socket adalah sebuah konsep yang telah berkembang pada paradigma bahasa pemrograman untuk beberapa waktu. Socket pertama kali digunakan pada sistem Unix pada tahun 1970-an dan sekarang socket merupakan standar low-level komunikasi primitif. Sebenarnya ada dua jenis socket, yaitu connection-oriented socket, yang termasuk dalam TCP (Transport Control Protocol) dan connectionless

socket, yang berdasar pada UDP (User Datagram Protocol). Jika dianalogikan komunikasi socket merupakan komunikasi dua arah antar komputer dengan jalur pipa sebagai penghubungnya dan masing-masing pipa tersebut tertancap pada sebuah gerbang tertentu yang dinamakan port.

Lifetime sebuah socket terdiri atas 3 fase :

- . **Creation** : inisialisasi untuk membuat sebuah socket
- . **Reading and Writing** : menerima dan mengirim ke sebuah socket
- . **Destruction** : menutup atau mengakhiri socket

Java menyediakan utilitas yang lengkap untuk pemrograman network yang dibundel dalam package `java.net.*` dan `javax.net.*`. Java menyediakan dua buah tipe socket yang berbeda dan sebuah socket spesial untuk berkomunikasi antar node yaitu:

- TCP sockets (connection -oriented socket) yang diimplementasikan pada kelas `java.net.Socket`.
- UDP sockets (connection -less socket) yang diimplentasikan oleh kelas `java.net.DatagramSocket`.

Dalam koneksi TCP, socket dan server socket adalah dua class yang biasa digunakan. Server Socket merepresentasikan socket yang berada di server yang menunggu dan mendengarkan request dari client untuk segera dilayani. Socket merepresentasikan end point communication antara client dan server. Ketika server menerima request dari client, server akan membuat socket baru yang digunakan sebagai komunikasi antara server dan client sehingga server bisa mendengarkan request dari client yang lain lagi. Client juga membuat socket untuk komunikasi dengan server. Di java TCP socket terdiri 4 tahap dalam membangun sebuah koneksi yaitu :

1. . Membuka socket
2. . Membuat input data stream
3. . Membuat output data stream
4. . Menutup atau mengakhiri socket

2.5 JDBC

JDBC adalah Application Programming Interface (API) yang dirancang untuk mengakses database universal berdasarkan SQL. JDBC terdiri atas JDBC 1.0 API yang memberika fingsi-fungsi dasar untuk akses data. JDBC 2.0 API memberikan tambahan ke fungsi-fungsi dasar dengan kelebihan-kelebihan lain yang lebih mutakhir.

2.5.1 JDBC API

JDBC adalah suatu nama trade mark, bukan sebuah singkatan. Tapi JDBC sering dikira singkatan dari Java Database Connectivity. JDBC API terdiri dari sejumlah class dan interface yang ditulis dalam bahasa Java yang menyediakan API standar sebagai alat bantu bagi pembuat program dan memberikan kemungkinan untuk menulis aplikasi database dengan menggunakan semua Java API. JDBC API memudahkan untuk mengirim statement SQL ke sistem database relasional dan mendukung bermacam-macam bahasa SQL. Keunggulan JDBC API adalah sebuah aplikasi dapat mengakses sembarang sumber data dan dapat berjalan pada sembarang platform yang mempunyai Java Virtual Machine (JVM). Sehingga kita tidak perlu menulis satu program untuk mengakses database Sybase, Oracle atau Access dan lain-lain. Kita cukup menulis satu program yang menggunakan JDBC API, dan program dapat mengirimkan statement SQL atau statement lain ke sumber data tertentu. Dengan aplikasi yang ditulis dalam bahasa Java seseorang tidak perlu khawatir untuk menulis aplikasi yang berbeda-beda agar dapat berjalan pada platform yang berbeda-beda.

2.5.2 Model 2-tier dan 3-tier

Untuk akses data base, JDBC API mendukung baik model 2-tier maupun 3-tier. Untuk model 2-tier, sebuah applet atau aplikasi java berbicara langsung ke database. Dalam hal ini diperlukan JDBC driver yang dapat berkomunikasi ke sumber data tersebut. Sebuah perintah atau statement dari user dikirim ke database dan hasil dari statement dikirim balik ke user. Database dapat terletak pada mesin yang sama atau berbeda dengan klien, yang disambungkan dengan jaringan. Jika letak database berbeda dengan mesin klien maka disebut dengan client/server. Mesin user disebut dengan client dan mesin dimana database berada disebut dengan server. Jaringan ini bisa berupa LAN atau internet.

Dalam model 3-tier, user mengirimkan perintah ke sebuah middle tier. Selanjutnya middle tier mengirimkan perintah tersebut ke database. Database memproses perintah tersebut dan mengirim balik hasilnya ke middle tier. Kemudian middle tier mengirimkannya ke user. Keuntungan model 3-tier ini adalah mempermudah aplikasi untuk dideploy dan meningkatkan performansi.

2.6 Network Admission Control (NAC)

Network Admission Control (NAC) merujuk pada versi Cisco Network Access Control, yang membatasi akses ke jaringan

berdasarkan identitas atau postur keamanan. Ketika sebuah perangkat jaringan (switch, router, jalur akses, server DHCP, dll) dikonfigurasi untuk NAC, itu dapat memaksa otentikasi pengguna atau mesin sebelum pemberian akses ke jaringan. Selain itu, akses tamu dapat diberikan kepada daerah karantina untuk perbaikan masalah apapun yang mungkin mengakibatkan kegagalan otentikasi. Hal ini dilaksanakan melalui kebiasaan inline perangkat jaringan, perubahan pada saklar yang ada atau router, atau terbatas kelas DHCP. Tipikal (non-free) koneksi WiFi adalah suatu bentuk NAC. Pengguna harus ada semacam surat-surat (atau kartu kredit) sebelum diberikan akses ke jaringan.

Dalam tahap awal, Cisco Network Admission Control (NAC) memungkinkan fungsi router Cisco untuk menegakkan hak istimewa akses ketika endpoint mencoba untuk menyambung ke jaringan. Akses ini dapat keputusan berdasarkan informasi tentang perangkat titik akhir, seperti antivirus saat ini negara. Negara antivirus termasuk informasi seperti versi perangkat lunak antivirus, definisi virus, dan versi mesin scan. Jaringan sistem kontrol penerimaan noncompliant memungkinkan perangkat untuk dapat mengakses, ditempatkan di daerah karantina, atau Pembatasan diberikan akses ke sumber daya komputasi, sehingga tetap tidak aman menjangkiti node dari jaringan.

Komponen kunci dari Cisco Network Program Kontrol Penerimaan Cisco Trust Agent, yang berada pada sistem endpoint dan berkomunikasi dengan router Cisco di dalam jaringan. Cisco Trust Agent negara keamanan mengumpulkan informasi, seperti perangkat lunak antivirus apa yang sedang digunakan, dan mengkomunikasikan informasi ini untuk Cisco router. Informasi ini kemudian diteruskan ke Cisco Secure Access Control Server (ACS) di mana kontrol akses keputusan dibuat. ACS mengarahkan router Cisco untuk melakukan penegakan terhadap endpoint

Konsep NAC sebagai berikut pada saat Client komputer mengakses ke jaringan, server melakukan report kondisi sistem pada client. Sehingga, selain identitas Client komputer seperti MAC address, Windows SMI, dan parameter lainnya Server juga menganalisa kondisi kesehatan Client. Saat ini yang termasuk kategori "kesehatan" adalah Anti-Virus apakah client sudah melakukan scan antivirus?, sudahkah antivirus di client terupdate?, berapa banyak virus yang terdeteksi di client? dan Filtering apakah semua Port dalam kondisi baik?, apakah ada filter atau port yang tidak konsisten dengan IT policy dan sebagainya. NAC akan memastikan setiap user menaati semua aturan keamanan sebelum diperbolehkan masuk ke dalam jaringan

2.7 PEMOGRAMAN PERL

Perl adalah bahasa pemrograman untuk segala keperluan, dikembangkan pertama kali oleh Larry Wall di mesin Unix. Perl dirilis pertama kali pada tanggal 18 Desember 1987 ditandai dengan keluarnya Perl 1.

Pada versi-versi selanjutnya, Perl tersedia pula untuk berbagai sistem operasi varian Unix (SunOS, Linux, BSD, HP-UX), juga tersedia untuk sistem operasi seperti DOS, Windows, PowerPC, BeOS, VMS, EBCDIC, dan PocketPC. Dukungan terhadap pemrograman berbasis obyek (*object oriented programming/OOP*) ditambahkan pada Perl 5, yang pertama kali dirilis pada tanggal 31 Juli 1993. Proyek pengembangan Perl 6 dimulai pada tahun 2000, dan masih berlangsung hingga kini tanpa tanggal yang jelas kapan mau dirilis. Ini dikatakan sendiri oleh Larry Wall [1] dalam satu pidatonya yang dikenal dengan seri The State of the Onion.

Dua di antara karakteristik utama Perl adalah penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum. Tidak heran jika Perl sangat populer digunakan dalam program-program CGI ([Common Gateway Interface](#)) dan berbagai protokol Internet lainnya. Seperti diketahui, TCP/IP sebagai basis bagi semua protokol Internet yang dikenal sekarang ini menggunakan format teks dalam komunikasi data. Seperti juga bahasa populer lainnya, Perl menerima banyak kritikan. Meski banyak di antaranya hanya berupa mitos, atau berlebih-lebihan, tapi terdapat juga sejumlah kritikan yang valid. Salah satunya adalah, sintaksnya susah dibaca, karena banyak menggunakan simbol-simbol yang bukan huruf dan angka. Kode sumber Perl tidak perlu dikompilasi, melainkan bisa langsung dijalankan oleh sebuah program [interpreter](#) bernama perl

3. METODOLOGI

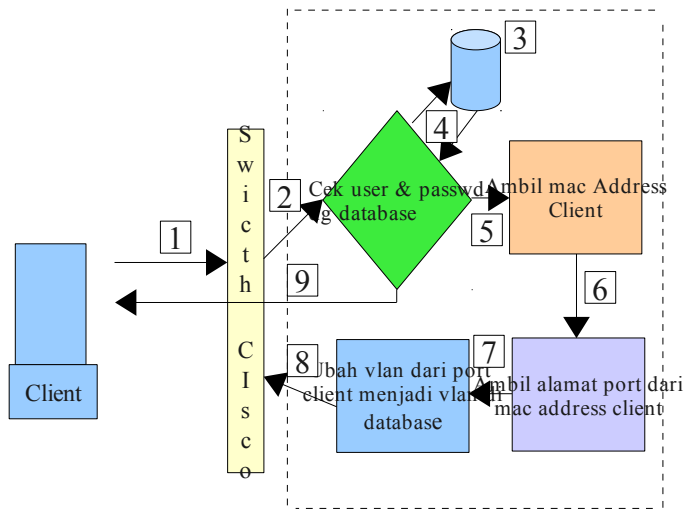
Dalam melakukan penyusunan program ini dilakukan dengan penelitian dan percobaan secara nyata sehingga diharapkan nanti bisa diaplikasikan dengan kondisi lingkungan yang sesungguhnya.

PERANCANGAN SISTEM

3.1 UMUM

Dalam bab ini diarahkan pada langkah-langkah perencanaan proyek akhir yang meliputi perencanaan perangkat lunak secara keseluruhan, yang merupakan pokok dari bahasan utama proyek akhir. Dalam proyek akhir ini dibuat sebuah aplikasi yang digunakan untuk melakukan identifikasi personal komputer yang nantinya identifikasi tersebut akan dapat membedakan hak akses setiap user yang terhubung ke jaringan

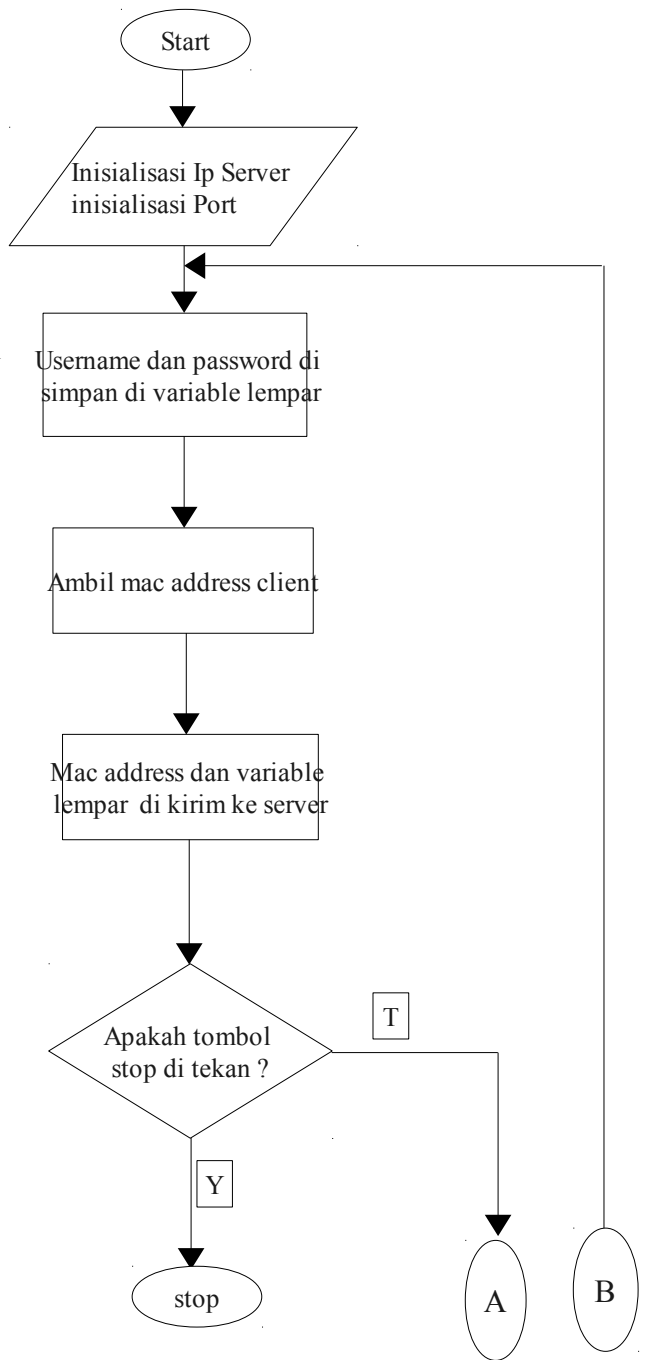
Disini diterangkan bagaimana merencanakan program untuk melakukan identifikasi personal



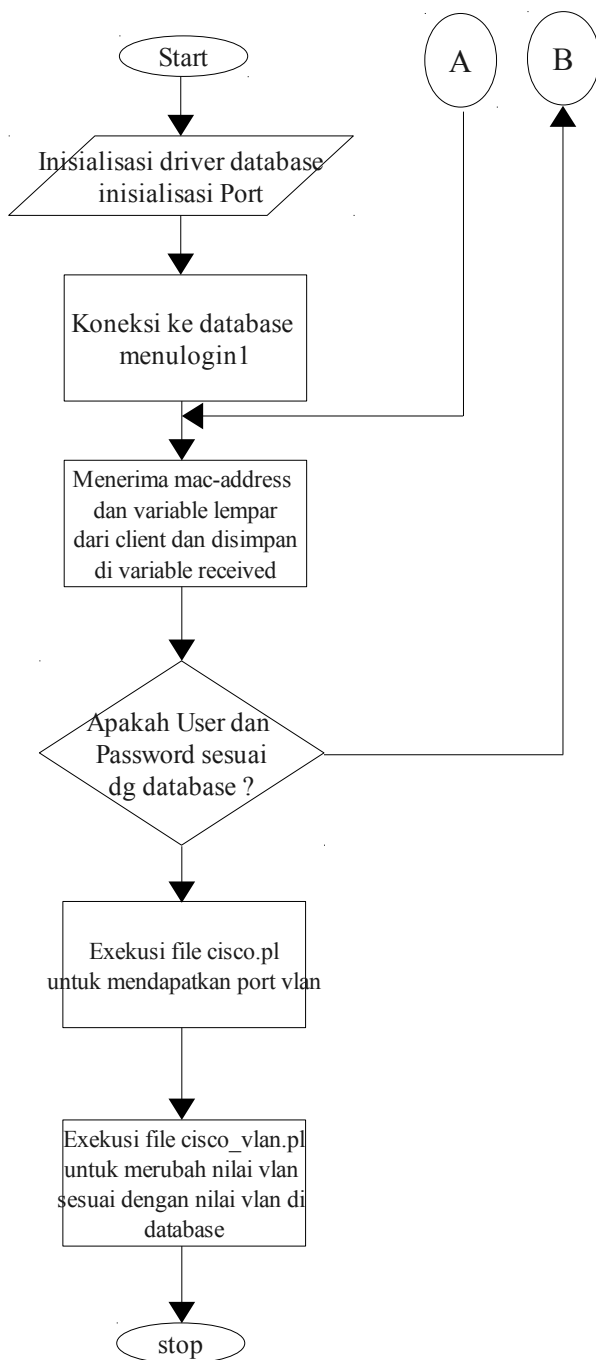
komputer. Langkah pembuatan seperti pada blok diagram berikut :

Dari Sekema 3.1 dapat dijelaskan sebagai berikut:

- Pada no1. Client melakukan koneksi ke server melalui switch cisco.
2. Client mengirimkan username dan password untuk dilakukan pengecekan dengan database.
3. Merupakan database
4. Merupakan tempat yang digunakan untuk melakukan pengecekan antara username dan password dari client dengan username dan password dari database, dan apabila diperoleh kesesuaian maka menuju ke nomor 5 dan apabila tidak di peroleh kesesuaian maka akan menuju ke no 9
5. Proses pengambilan mac address client server
6. Proses untuk mendapatkan alamat port di cisco berdasarkan mac address client
7. Proses merubah nilai vlan pada port cisco yang berdasarkan mac address client sesuai dengan nilai vlan yang ada di database
8. Vlan port cisco telah berubah sesuai dengan nilai vlan di database
9. User dan password tidak ditemukan



Gambar 3.2 Flowchart Sistem di sisi Client



Gambar 3.3 Flowchart Sistem disisi Server

3.3.2. PERANCANGAN SISTEM DISISI CLIENT

3.3.2.1 KONEKSI CLIENT KE SERVER

```

.....
12 public class MultiClient {
13     private static final int PORT = 1234;
14     private static Socket link ;
15     private static BufferedReader in ;
16     private static PrintWriter out ;
17     private static BufferedReader keyboard ;
18     private static String oke;
19     public static void koneksi(String tangkap){
20     try{
21         InetAddress host = InetAddress.getByNome("10.252.10.219");
22         link = new Socket(host,PORT);
23         in = new BufferedReader(new InputStreamReader(link.getInputStream()));
24         out = new PrintWriter(link.getOutputStream(),true);
25         keyboard = new BufferedReader(new InputStreamReader(System.in));
26         String message, response,hasil="";
.....
  
```

Potongan Program 3.1 Koneksi client ke server

Pada potongan program 3.1 dapat dijelaskan sebagai berikut :

- pada baris 13. Merupakan deklarasi variabel PORT dengan tipe data berupa integer dengan nilai 1234 yang digunakan untuk koneksi dengan server
- pada baris 14. merupakan deklarasi variabel link dengan tipe data berupa Socket
- pada baris 21. Merupakan coding yang digunakan untuk mengenali ip server sebelum dilakukan sinkronisasi ke server
- pada baris 22. merupakan terjadinya koneksi antara client dengan server dengan ip server 10.252.10.219 dan port 1234

3.3.2.2 Input dan Output Stream Client

Pada potongan program 3.1 input output stream dapat dijelaskan sebagai berikut :

- pada baris 15 deklarasi in dengan tipe data BufferedReader
- pada baris 16 deklarasi out dengan tipe data PrintWriter
- pada baris 23 merupakan listing program yang nantinya digunakan untuk menangkap respon server sedangkan,
- pada baris 24 digunakan untuk melakukan pengiriman ke server

3.3.2.3 Proses mendapatkan Mac Address client

```
28 InetAddress address = InetAddress.getLocalHost();
29 NetworkInterface ni = NetworkInterface.getByInetAddress(address);
32 if (ni != null) {
33     byte[] mac = ni.getHardwareAddress();
34     if (mac != null) {
36         for (int i=0; i < mac.length; i++) {
38             out.format("%02x%s", mac[i], i < mac.length-1 ? "" : "");
40         }
42     } else {
43         System.out.println("Address doesn't exist or is not accessible.");
44     }
45     } else {
46         System.out.println("Network Interface for the specified address is not found.");
47     }
```

Potongan program 3.2 Proses mendapatkan Mac Address Client

Pada potongan program 3.2 dapat dijelaskan sebagai berikut :

pada baris 28 merupakan proses yang digunakan untuk mendapatkan ip address dari client sedangkan **pada baris 29** merupakan proses yang digunakan untuk mencari network interface berdasarkan ip address client **pada baris 32** merupakan suatu kondisi jika nilai ni sama dengan null maka menuju ke baris 46 dan apabila tidak maka menuju baris 33 **pada baris 33** merupakan proses pengambilan mac address client yang nilainya di simpan di variabel array "mac" dengan tipe data byte **pada baris 34** merupakan suatu kondisi dimana apabila nilai mac sama dengan null maka menuju baris 43 dan apabila tidak sama dengan null maka menuju baris 36 **pada baris 36** merupakan looping dimana i=0 dan i < mac.length serta i melakukan increment sampai ditemu i < mac.length **pada baris 38** merupakan proses pengiriman mac address ke server dengan format xxxxxxxxxxxx

3.3.3 PERANCANGAN SITEM DI SISI SERVER

3.3.3.1 KONEKSI SERVER KE CLIENT

```
18 private static ServerSocket servSock;
19 private static final int PORT = 1234;
20 public static void main (String args[]) throws IOException {
21     System.out.println ("Opening Port ..... \n");
22     try {
23         servSock = new ServerSocket (PORT);
24     }
25     } catch (Exception e) {
26         System.out.println ("unable to attach to port");
27         System.exit(1);
28     }
29     do {
30         Socket client = servSock.accept();
```

Potongan program 3.3 Proses Koneksi Server ke Client

Pada potongan program 3.3 dapat dijelaskan sebagai berikut :

Pada baris 18 merupakan inialisasi variabel servSock dengan tipe data ServerSocket pada baris 19 merupakan inialisasi variabel PORT dengan tipe data berupa integer yang memiliki nilai 1234 pada baris 21 cetak string berikut "Opening Port \n" pada server pada baris 23 merupakan proses dimana server menunggu koneksi dari client dengan menggunakan PORT 1234 pada baris 30 merupakan proses menunggu client (listen) yang pada dasarnya menunggu permintaan koneksi ke jaringan

3.3.3.2 FUNGSI THREAD DI SERVER

```
.....
31     ClientHandler handler = new ClientHandler(client);
32     handler.start();
33     }while (true);
34 }
35
36 }
37
38 class ClientHandler extends Thread {
39     private Socket client ;
40     private BufferedReader in ;
41     private PrintWriter out;
42     static int portNumber;
43
44     .....
118     public ClientHandler(Socket socket) throws IOException {
119         client = socket;
120     }
121     try {
122     }
123         in = new BufferedReader(new InputStreamReader(client.getInputStream()));
124         out = new PrintWriter(client.getOutputStream(), true);
125     } catch (Exception e) {
126     }
127     }
128 }
```

Potongan Program 3.4 Fungsi Thread

Fungsi Thread ini memiliki kegunaan agar setiap client yang koneksi dengan server bisa lebih dari satu client dan biasa disebut multi client. Potongan program 3.4 dapat dijelaskan sebagai berikut :

Pada baris 31 membuat object baru yaitu ClientHandler. Konstruktord ClientHandler memerlukan "client" sebagai argumen. Dan merupakan sebagai proses menunggu permintaan apabila ada client lain yang akan melakukan koneksi ke server. **Pada baris 32** proses melakukan pengaktifan kustrator ClientHandler pada baris 118

3.3.3.3 Input dan Output Stream Server

Pada potongan program 3.4 proses input dan output stream server terletak pada baris berikut :

pada baris 40 inialisasi variabel "in" dengan tipe data BufferedReader

pada baris 41 inialisasi variabel "out" dengan tipe data PrintWriter
 pada baris 123 merupakan listing program yang nantinya digunakan untuk menangkap respon dari client sedangkan,
 pada baris 124 digunakan untuk melakukan pengiriman ke client

3.3.3.4 Koneksi ke Database Mysql

```

.....
43 private static Connection conn = null;
45 public Connection getConnection(){
46     conn=null;
47     try {
48
49         String userName = "root";
50         String password = "root";
51         String url = "jdbc:mysql://localhost/menulogin1";
52         Class.forName("com.mysql.jdbc.Driver").newInstance();
53         conn = DriverManager.getConnection(url, userName, password);
.....
  
```

Potongan program 3.5 Koneksi Server ke Database

Pada potongan program 3.5 digunakan untuk koneksi dari server ke database Mysql berikut penjelasannya:
 pada baris 43 merupakan inialisasi variabel "conn" dengan tipe data berupa Connection dan memiliki nilai null
 pada baris 45 merupakan konstruktor getConnection()
 pada baris 46 variabel "conn" di signmen dengan null
 pada baris 49 inialisasi userName untuk mencocokkan userName database pada waktu login ke database disini disignmen dengan String root
 pada baris 50 inialisasi password yang digunakan untuk mencocokkan password didatabase pada waktu login nantinya, yang disignmen dengan String "root"
 pada baris 51 inialisasi variabel "url" dengan tipe data String yang digunakan untuk menunjukkan alamat dimana database menulogin1 berada
 pada baris 52 melakukan pemanggilan driver ojdbc yang nantinya digunakan untuk melakukan koneksi ke database mysql
 pada baris 53 merupakan perintah untuk melakukan koneksi database dengan memasukan tiga parameter dari baris 49,50,dan 51

3.3.3.5. Proses Validasi User

Proses validasi user digunakan untuk mengecek apakah user yang terkoneksi ke server merupakan user yang valid dan untuk menghindari dari ilegal user berikut listing validasi user yang di sesuaikan dengan data di database :

```

93 public boolean getExistUser(String userN, String pass) {
94     boolean hasil = false;
95     try {
96         getConnection();
97         Statement s = conn.createStatement();
98         s.executeQuery ("SELECT * from T_USER where username='"+userN+"'
          'and password='"+pass+"'");
99         ResultSet rs = s.getResultSet ();
100         int count = 0;
101         if(rs.next()){
102             hasil = true;
103         }
104         else {
105             hasil = false;
106         }
107
108         rs.close ();
109         s.close ();
110
111     } catch (SQLException ex) {
112         Logger.getLogger(MultiServer.class.getName()).log(Level.SEVERE, null, ex);
113     }
114
115     return hasil;
116 }
117 }
  
```

Potongan program 3.6 Proses Query Tabel T_USER

Pada potongan program 3.6 dapat dijelaskan sebagai berikut :

pada baris 93 merupakan fungsi getExistUser dengan dua parameter userN dengan tipe data String dan pass dengan tipe data String
 pada baris 94 inialisasi variabel "hasil" dengan tipe data boolean dengan value false
 pada baris 96 membuka koneksi database menulogin1
 pada baris 97 membuat object baru s untuk menampung perintah sql pada baris 98 object s melakukan query tabel T_USER terhadap field username dan field password
 pada baris 99 membuat object baru rs untuk menangkap satu record di database
 pada baris 100 inialisasi variabel count dengan tipe data interger yang di sigdmen dengan 0
 pada baris 101 merupakan suatu kondisi dimana rs.next bernilai dan sesuai maka hasil disignmen true bila tidak maka menuju baris 105

```

.....
164         if (getExistUser(fin[1],fin[2])){
165             System.out.println(akhir);
166             out.println("Terkoneksi dengan server");
.....
  
```

Potongan program 3.7 Proses Validasi User

pada potongan program 3.7 dapat dijelaskan sebagai berikut :

pada baris 164 merupakan pemanggilan fungsi getExistUser dengan parameter fin[1] dan fin[2] dimana fin[1] dan fin[2] merupakan username dan password yang dikirim oleh client yang ditangkap oleh server
 pada baris 165 apabila dari pengecekan user dan password sesuai dengan data yang ada didatabase maka cetak variabel "akhir" menuju baris 166 dan mengirimkan pesan ke client berupa string "Terkoneksi dengan server"

berikut potongan program untuk mendapatkan mac address sesuai dengan format yang dikenali di cisco seperti xxxx.xxxx.xxxx :

```

136 String received ;
137 String vlan = "";
138
139 do {
140     received = in.readLine();
141     //System.out.println("received "+received);
142     String[] input = received.split("|");
143     String tmphasil="";
144
145     int size = input.length;
146     for (int x=0;x<size;x++){
147         tmphasil += input[x];
148     }
149     String[] data = tmphasil.split("|");
150     String akhir = "";
151     for (int x=0;x<data.length;x++){
152         if (x==5 || x==9) akhir += ".";
153         akhir +=data[x];
154     }

```

Potongan program 3.8 Proses mendapatkan mac address sesuai format cisco

Pada potongan program 3.8 dapat dijelaskan sebagai berikut :

pada baris 136 inialisasi variabel “received” dengan tipe data berupa String

pada baris 137 inialisasi variabel “vlan” dengan tipe data String

pada baris 140 mensignmen nilai in.readLine() yang dikirim dari client ke variabel “received”

pada baris 142 melakukan split dengan indikator “|” pada variabel “received” yang kemudian ditampung di array input yang betipe data String

pada baris 143 inialisasi variabel “tmphasil ” yang bertipe data String

pada baris 145 melakukan inialisasi variabel “size” yang betipe data interger untuk menampung nilai dari input.lenght

pada baris146 melakukan looping dimana x=0 sampai x<size dan x diincrement

pada baris 147 hasil dari looping pada baris 146 ditampung di variabel “tmphasil”

pada baris 149 melakukan split pada variabel “ tmphasil” dengan indikator “|” yang kemudian ditampung di array data bertipe data String

pada baris 150 inialisasi variabel “akhir” dengan tipe data String

pada baris 151 melakukan looping dimana x=0 sampai x<data.lenght dan dilakukan increment pada x

pada baris 152 suatu kondisi dimana jika x ke 5 atau x ke 9 memenuhi nilai 5 atau 9 maka tambahkan “.” dan hasil ditampung di variabel “akhir”

Berikut Potong program untuk mendapatkan fin[1] dan fin[2] :

```

155 String fin[]= new String[3];int param = 0;
156 StringTokenizer st = new StringTokenizer(akhir,"|");
157 while(st.hasMoreTokens()){
158     fin[param]=st.nextToken();
159     param++;
160     if(param >= 3)
161         param=0;

```

Potongan program 3.9 Proses mendapatkan fin[1] dan fin[2]

pada potongan program 3.9 dapat dijelaskan sebagai berikut :

pada baris 155 inialisasi array fin bertipe data String yang berjumlah 3 array dan inialisasi variabel “param” bertipe data interger

pada baris 156 membuat object baru st dan melakukan Tokenizer pada variabel “akhir” dengan indikator “|”

pada baris 157 melakukan perulangan sampai variabel “st” masih memiliki token

pada baris 158 mensignmen nilai st.nextToken() ke array fin dengan indek param

pada baris 159 indek param di increment

3.3.3.6 Proses Mendapatkan Alamat Port Mac Address Client di Switch Cisco

Proses selanjutnya adalah memperoleh alamat port yang berada di switch cisco dari mac address client yang nantinya digunakan dasar untuk merubah nilai vlannya berikut listing programnya :

```

177 Process p = Runtime.getRuntime().exec("./cisco.pl "+fin[0]);
178 BufferedReader stdInput = new BufferedReader
(new InputStreamReader(p.getInputStream()));
179 BufferedReader stdError = new BufferedReader
(new InputStreamReader(p.getErrorStream()));
180 System.out.println("Here is the standard output of the command:\n");
181 while ((s = stdInput.readLine()) != null) {
182     String fan,int param = 0;
183     StringTokenizer stt = new StringTokenizer(s);
184     while(stt.hasMoreTokens()){
185         if(param !=3){
186             param++;stt.nextToken();
187             continue;
188         }
189         else{
190             param=0;
191             fan=stt.nextToken();
192             System.out.println(fan + vlan);
193         }
194         //System.out.println("./cisco_vlan.pl "+fan+" "+vlan);
195         Process q = Runtime.getRuntime().exec("./cisco_vlan.pl "+fan+" "+vlan);
196     }
197     //System.out.println(s);
198 }
199 // read any errors from the attempted command
200 System.out.println("Here is the standard error of the command (if any):\n");

```

Potongan program 3.10 Proses untuk mendapatkan alamat port di cisco

```

1#!/usr/bin/perl -w
2 use Net::Telnet::Cisco;
3 my $session = Net::Telnet::Cisco->new(Host => "10.252.1.50");
4 $session->login('cisco','cisco');
5 if($session->enable("cisco")) {
6     $cmd="show mac-address-table | i $ARGV[0]";
7     @output = $session->cmd($cmd);
8     print "@output";
9 } else {
10     warn "Can't enable: ". $session->errmsg;
11 }
12
13 $session->close;

```

Potongan Program 3.11 Listing Perl dari Cisco.pl

Pada potongan program 3.10 dapat dijelaskan sebagai berikut :

pada baris 177 melakukan proses *execute* command line di console linux berupa program perl yaitu menjalankan perintah `cisco.pl fin[0]` dimana `fin[0]` merupakan mac address client yang sudah sesuai dengan format cisco

pada potongan program 3.11 dapat dijelaskan sebagai berikut :

- pada baris 2** inialisasi untuk melakukan telnet
- pada baris 3** telnet ke 10.252.1.50
- pada baris 4** login sebagai user cisco dengan password cisco
- pada baris 5** mengaktifkan user root dengan password cisco
- pada baris 6** melakukan perintah untuk menampilkan tabel mac address yang berdasarkan mac address client yang ditampung di variabel `"$ARGV[0]"` yang diambil dari variabel `"fin[0]"`
- pada baris 7** inialisasi variabel `"output"` untuk menampung hasil dari cmd
- pada baris 8** cetak hasil dari cmd untuk ditampilkan

kembali pada potongan program 3.10

pada baris 181 melakukan perulangan dan pengecekan dimana variabel `"s"` yang digunakan untuk menampung nilai dari `stdInput.readLine()` bila nilainya null maka menuju ke baris 200 dan apabila variabel `"s"` tidak bernilai null menuju baris berikutnya

pada baris 182 inialisasi variabel `"fan"` bertipe data String dan inialisasi variabel `"peram"` yang bertipe data integer

pada baris 183 membuat object baru `stt` dan melakukan tokenizer pada variabel `"s"` dengan indikator `" "`

pada baris 184 melakukan perulangan dan pengecekan pada variabel `"stt"` sampai masih memiliki token

pada baris 185 merupakan suatu kondisi dimana jika variabel `"peram"` tidak sama dengan 3 maka increment `peram` dan `next token` dan apabila variabel `"peram"` sama dengan 3 maka menuju ke baris 190

pada baris 190 inialisasi variabel `"peram"` dengan value 0

pada baris 191 variabel `"fan"` digunakan untuk menampung nilai dari `stt.nextToken()`
pada baris 192 cetak variabel `"fan"` yang merupakan alamat port dari mac address client

3.4 PERENCANAAN GUI

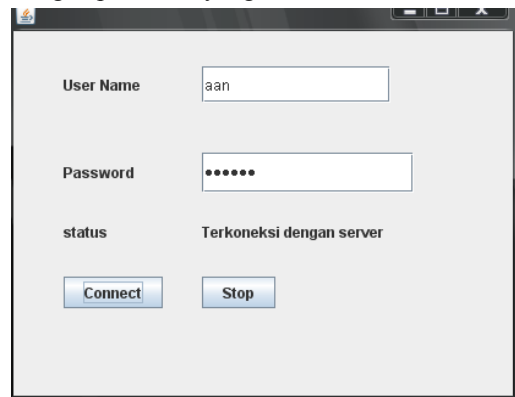
Pada tampilan program akan dibuat sedemikian rupa sehingga pemakai atau pengguna program dapat menjalankan dan memahami jalan kerja program dengan mudah.

3.4.1 GUI Form Konfigurasi User

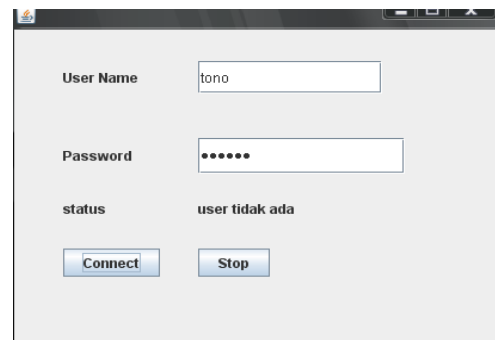
Form ini akan tampil saat pertama kali program di jalankan disisi client, form ini digunakan pada saat kita akan melakukan koneksi ke server, semua field harus diisi karena field – field ini merupakan parameter dalam melakukan koneksi.

Field *Username* adalah suatu inputan yang digunakan untuk memasukan username dari client yang nantinya akan dicross check dengan username yang ada di database.

Field *Password* adalah suatu inputan yang digunakan untuk memasukan password dari client yang nantinya akan di cross check dengan password yang ada di database.



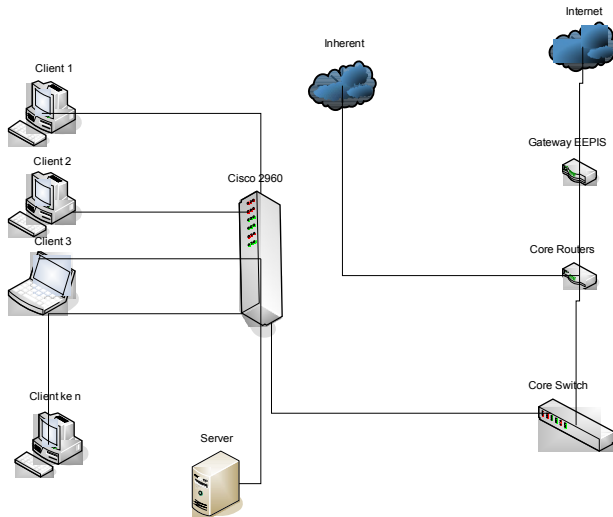
Gambar 3.4 Form Login di Client



Gambar 3.5 Form Login di Client user dan password tidak sesuai dengan database

4. ANALISA HASIL SISTEM

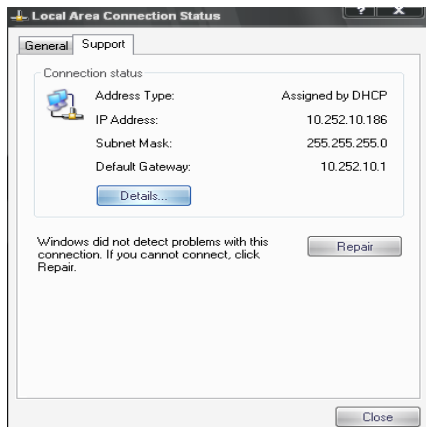
Pada bab ini terdapat beberapa hal yang dapat dianalisa berdasarkan hasil dari sistem identifikasi personal komputer



Gambar 4.1 Topologi Jaringan yang digunakan saat percobaan

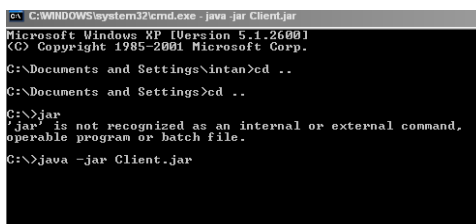
Pada gambar 4.1 dapat di ambil keterangan sebagai berikut :

Client 1 berada pada port 11 di cisco 2960, secara default mendapatkan vlan 10 dengan gateway default 10.252.10.1 ip 10.252.10.186



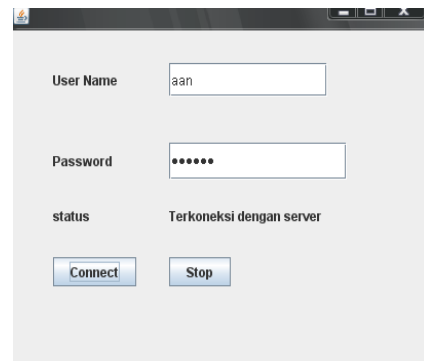
Gambar 4.2 Local Area Connection Status Client default

di client 1 dilakukan eksekusi file Client.jar melalui console command line seperti pada gambar 4.3:



Gambar 4.3 Console command line di client

tampil menu login sebagai berikut :



Gambar 4.4 Form login dari client

Kemudian akan dilakukan perubahan vlan ke vlan 408 sesuai daftar table T_USER yang berada di database menulogin1 sebagai berikut :

	id	username	password	vlan
<input type="checkbox"/>	1	aan	123456	408
<input type="checkbox"/>	2	ahmad	qwerty	407
<input type="checkbox"/>	3	client	linux	405
<input type="checkbox"/>	4	suhendro	qwerty	406

Gambar 4.5 Tabel T_USER

vlan 408 dimiliki oleh username aan. Setelah user dan password sudah sesuai dengan database maka disisi server akan ada output sebagai berikut :

run:

1. Opening Port
2. 0016.d390.e91b|aan|123456
3. Here is the standard output of the command:
4. Fa0/5408
5. Here is the standard error of the command (if any):

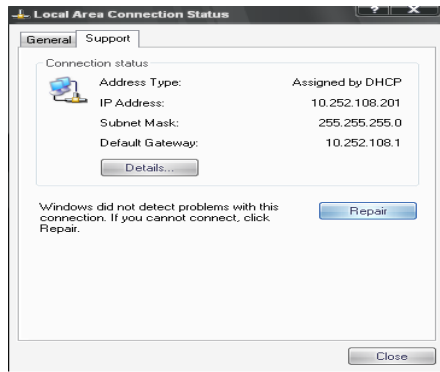
keterangan :

pada baris 1. menunjukkan bahwa server telah siap untuk menerima akses dari client

pada baris 2. menerima mac-address, username serta password dari client

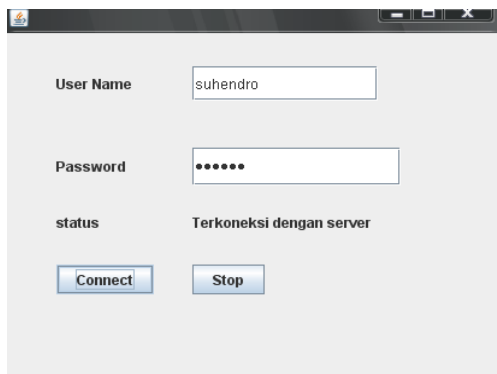
pada baris 4. Fa0/5 menunjukkan letak port yang digunakan oleh mac-address client pada switich cisco sedangkan 408 merupakan nilai vlan yang ada di database yang nantinya digunakan untuk merubah nilai vlan di port Fa0/5 ke vlan 408.

Setelah progam dijalankan maka client akan memperoleh gateway 10.252.108.1 dan ip 10.252.108.201



Gambar 4.6 Local Area Connection Status setelah vlan di ubah menjadi 408

Kemudian dilakukan inputan user yang berbeda saat ini menggunakan username suhendro dengan nilai vlan 406



Gambar 4.7 Form login client dengan user yang memiliki vlan 406

Disisi server akan tampil output :

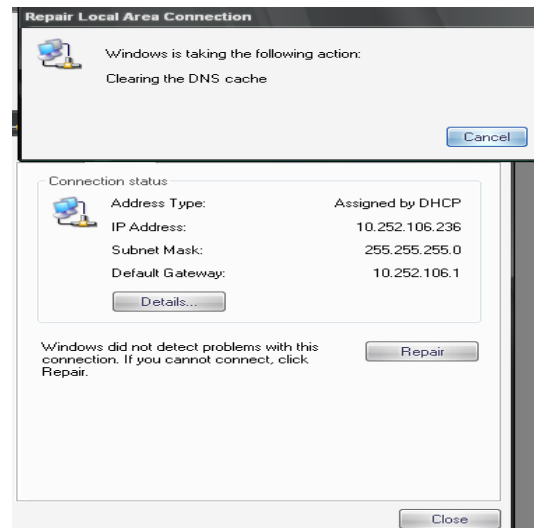
1. Opening Port
2. 0016.d390.e91b|suhendro|qwerty
3. Here is the standard output of the command:
4. Fa0/5406
5. Here is the standard error of the command (if any):

keterangan :

pada baris 1. menunjukkan bahwa server telah siap untuk menerima akses dari client

pada baris 2. menerima mac-address, username serta password dari client

pada baris 4. Fa0/5 menunjukkan letak port yang digunakan oleh mac-address client pada swithc cisco sedangkan 406 merupakan nilai vlan yang ada di database yang nantinya digunakan untuk merubah nilai vlan di port Fa0/5 ke vlan 406.



Gambar 4.8 Local Area Connection Status setelah vlan diubah menjadi 406

Setelah program dijalankan maka client akan memperoleh gateway 10.252.106.1 dan ip 10.252.106.236

5. KESIMPULAN

Pada bagian ini akan diulas tentang kesimpulan dari seluruh percobaan dan pengujian dari software yang penulis buat :

1. Mac adress client dapat di deteksi melalui fungsi java yang ada di client yang dikirim ke server untuk melakukan pendeteksian port mac adress client yang terdaftar di table mac address *cisco* dengan menggunakan perl.
2. Perubahan vlan pada swithc *cisco* dilakukan oleh aplikasi java dengan memanfaatkan *library* dari perl *cisco*.
3. Data dari user yang terkoneksi ke sistem disimpan di database sehingga java bisa melakukan komunikasi ke database tersebut.

Aplikasi ini jauh dari kata sempurna maka jika ada kritik dan saran yang membangun kiranya penulis akan menerima dengan ihlas dan terima kasih.

5.1 SARAN – SARAN

Dari beberapa perancangan disini banyak hal yang seharusnya masih bisa disempurnakan, namun karena keterbatasan waktu, biaya serta pikiran, maka ada beberapa hal yang perlu di kembangkan lagi diantaranya :

1. Diperlukannya timer untuk mengecek koneksitas client ke server.
2. Belum adanya administrasi untuk menghubungkan ke phpmyadmin
3. Diperlukannya enkripsi teks dari client ke server

DAFTAR PUSTAKA

1. <https://www.ee.ui.ac.id/online/main/mhssemta/show/id/5313>.
2. <http://hari.narmadi.net/internet/jaringan-java-socket> . Date April 27th, 2009
Author Hari Category [Internet](#) Tags
[JAVA SOCKET](#)
3. <http://www.agungnugroho.net/?p=62>_
IMPLEMENTASI NETWORK ADMISION
CONTROL
4. http://lecturer.ukdw.ac.id/cnuq/wpcontent/uploads/bridging/modul_bab3VLAN.doc
5. IEEE Computer Society, 802.1X - Port Based Network Access Control, 2001.
6. <http://lecturer.eepisits.edu/~yuliana/Prog%20Lanjut/Networking/>
7. [http://lecturer.eepis-its.edu/~idris/files/oop_lanjut/p9%20%20JDBC\(1\)/odbc.ppt](http://lecturer.eepis-its.edu/~idris/files/oop_lanjut/p9%20%20JDBC(1)/odbc.ppt)
8. http://id.wikipedia.org/wiki/MAC_address
9. <http://id.wikipedia.org/wiki/Perl>