

Tugas Akhir Mata Kuliah

EL-695 KEAMANAN SISTEM INFORMASI

Dosen : Ir. Budi Rahardjo PhD.

Nama : ERWIN SYAIFUDDIN

NIM : 23202003

Program Studi : Teknik Sistem Komputer



Program Magister

Departemen Teknik Elektro

Institut Teknologi Bandung

2003

ABSTRAKSI

Seseorang yang membuat program dalam bahasa C di Linux suatu saat mungkin menginginkan supaya program yang dibuatnya tersebut hanya bisa jalan di satu komputer saja. Untuk itu pada makalah ini saya akan mencoba membuat suatu proteksi sederhana program berdasarkan alamat fisik ethernet card dan alamat IP komputer yang diinginkan.

Kata Kunci : Ethernet, Ethernet Physical Address, IP Address

DAFTAR ISI

Pendahuluan	3
Parameter Otentifikasi	3
Algoritma Program	5
Implementasi Program	7
Pengujian Program	16
Kesimpulan	18
Referensi	19

MEMBUAT ‘COPY PROTECTION’ SEDERHANA UNTUK PROGRAM BERBAHASA C DI LINUX

PENDAHULUAN

Ada saat nya dimana seseorang tidak menginginkan program yang telah dibuat dan di-compile-nya dalam bahasa C di Linux di-copy oleh orang lain untuk dijalankan pada komputer lain. Untuk itu perlu ditambahkan ‘*copy protection*’ pada program yang dibuatnya dalam bahasa C. Copy protection di sini bukan berarti program tersebut tidak dapat di-copy ke komputer lain, tetapi jika program tersebut dijalankan dikomputer lain maka akan memberikan output yang lain.

Copy protection berhubungan dengan *License Management* (manajemen perizinan) dari suatu software. Saat ini sudah ada teknologi baru untuk License Management yang bernama FlexLM (Flexible License Management) dari Globetrotter (<http://www.globetrotter.com/flexlm/flexlm.shtml>). FlexLM bekerja berdasarkan prinsip networking client-server. Ada banyak model license yang diberikan oleh FlexLM, contohnya :

1. Evaluation License (demo) yaitu license yang diberikan sampai batas waktu tertentu saja untuk keperluan evaluasi suatu program.
2. Capacity License: komputer dengan performansi tinggi memiliki license lebih banyak dibandingkan komputer dengan performansi rendah.
3. dan sebagainya. (dapat dilihat di <http://www.globetrotter.com/flexlm/lmmodels.shtml>)

PARAMETER OTENTIFIKASI

Dalam makalah ini saya akan mencoba menerapkan ide saya membuat suatu License Management sederhana dengan menggunakan dua parameter otentifikasi yang berhubungan erat dengan network yaitu:

- 1) Alamat Fisik Ethernet
- 2) Alamat Internet (IP Address).

Ide ini saya aplikasikan pada program berbahasa C di sistem operasi Linux. Tujuannya adalah memberikan suatu 'copy protection' pada program yang dibuat dalam bahasa C.

Alamat Fisik Ethernet (Ethernet Physical Address).

Alamat Fisik Ethernet atau disebut juga dengan *Ethernet Hardware Address* adalah suatu nomor pengenal khusus yang dimiliki Ethernet Card dan disimpan secara permanen dalam *Read Only Memory* (ROM) pada card ethernet tersebut. Alamat ini unik dan tidak ada yang sama pada setiap ethernet card, karena alamat ini nantinya berasosiasi dengan alamat internet (IP Address). Semua vendor pembuat ethernet card harus menentukan secara unik alamat fisik ethernet card nya berdasarkan aturan badan internet dunia. Alamat ini terdiri dari 6 byte (6 octet). Pada Linux Ethernet card terinstall sebagai eth0, eth1 dan seterusnya. Dari *shell* Linux alamat fisik ethernet dapat dilihat menggunakan perintah '**ifconfig**' seperti contoh berikut:

```
[Linux_Shell]# ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:80:48:E3:44:20
      inet addr:10.255.57.101  Bcast:10.255.57.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:6 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:0 (0.0 b)  TX bytes:720 (720.0 b)
      Interrupt:10 Base address:0x240
```

Pada contoh di atas alamat fisik ethernet (eth0) terdapat pada bagian *HWaddr* yaitu 00:80:48:E3:44:20. Alamat fisik ethernet ini ditampilkan dalam format Hexadecimal 6 byte dipisahkan oleh tanda titik dua (:).

Saya menggunakan alamat fisik ethernet ini sebagai otentifikasi program karena pada dasarnya alamat atau nomor ini tidak dapat diubah karena tersimpan dalam ROM ethernet card yang bersangkutan, tetapi dengan menggunakan beberapa program tertentu alamat ini dapat diubah, misalnya untuk tujuan *spoofing* (kamufase).

Alamat Internet (IP Address).

Berbeda dengan Alamat Fisik Ethernet, maka Alamat Internet (IP Address) harus dikonfigurasi terlebih dahulu pada ethernet card yang bersangkutan, dan dapat kita ubah/ganti dengan mudah.

IP Address ini terdiri dari 4 byte (4 octet) dan dikelompokkan dalam kelas-kelas tertentu yang menunjukkan banyaknya host yang tercakup dalam kelas tersebut.

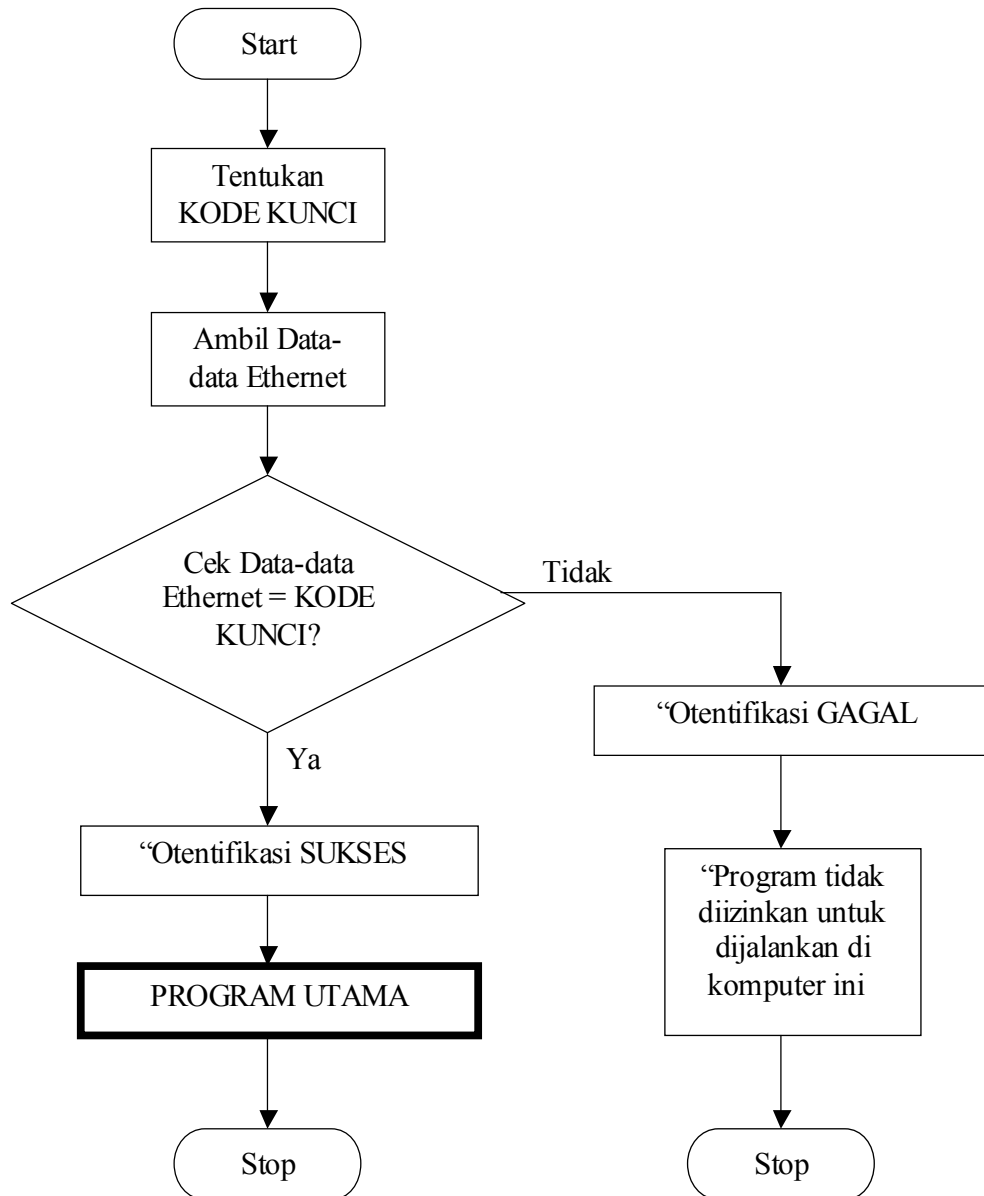
Alamat internet ini pada dasarnya juga unik untuk tiap host yang terhubung ke internet (menggunakan IP ter-register, Fully Qualified Domain Name), kecuali jika host tersebut berada pada *private network* dengan menggunakan *IP private* (10.x.x.x atau 192.168.x.x). IP private ini ditentukan oleh Internet Assign Number Association (IANA) dan hanya boleh digunakan di lingkungan pribadi bukan untuk di-*broadcast* ke jaringan internet.

Pada contoh di atas IP Address eth0 ditunjukkan oleh *inet addr*, yaitu : 10.255.57.101. Alamat Internet ini ditampilkan dalam format Decimal 4 byte dipisahkan oleh tanda titik (.)..

ALGORITMA PROGRAM

Pada dasarnya algoritma program ini sederhana yaitu hanya membandingkan data-data ethernet (alamat fisik dan alamat internet) pada saat program dijalankan dengan data KODE KUNCI yang telah kita tentukan sebelumnya. Jika data ethernet sama dengan kode kunci maka program dapat melanjutkan eksekusinya ke bagian program utama.

Flow chart program ini dapat digambarkan sebagai berikut:



IMPLEMENTASI PROGRAM

Untuk melakukan otentifikasi berdasarkan alamat fisik ethernet tentunya hal utama yang harus dilakukan oleh program adalah mendapatkan alamat fisik dari ethernet yang diinginkan. Informasi yang mendefinisikan struktur dari Ethernet Card seperti alamat fisik dan IP address terdapat pada library `/usr/include/net/if.h`. Untuk membuka dan membaca informasi pada ethernet digunakan *file descriptor* dengan *socket* (`/usr/include/sys/socket.h` dan `/usr/include/linux/if_ether.h`) dan fungsi *ioctl* (`/usr/include/sys/ioctl.h`). Untuk menyalin data ethernet digunakan fungsi *string copy* (*strcpy*). Untuk menyimpan hasil pembacaan data ethernet digunakan variabel pointer bertipe karakter (`unsigned char`). Data-data ethernet yang terdapat dalam alamat pointer diasumsikan dalam tipe integer dari 0 sampai 255 (1 byte / 8 bit).

Potongan program pengambilan data alamat fisik ethernet `eth0` sebagai berikut:

```
unsigned char *d0;
int fd;
struct ifreq ifr;
fd = socket(PF_INET, SOCK_PACKET, htons(ETH_P_ALL));
strcpy(ifr.ifr_name, "eth0");
ioctl(fd, SIOCGIFHWADDR, &ifr);
d0 = ifr.ifr_hwaddr.sa_data;
```

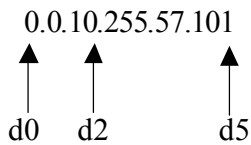
Isi pointer `d0` adalah data alamat fisik pada byte ke-5 (most significant byte) yaitu data pada posisi paling kiri. Data paling kanan (least significant byte) byte ke-0 adalah isi dari pointer `d5` (`d0 + 5`).

Ilustrasi pointer dengan data alamat fisik sebagai berikut:

```
00:80:48:E3:44:20
  ↑           ↑
  d0         d5=d0+5
```

Baik alamat fisik maupun alamat internet dari ethernet diambil per-byte. Pada library `if.h` data alamat fisik hardware didefinisikan dalam 6 byte, dimana format ini lengkap untuk alamat fisik ethernet tapi terlalu panjang untuk alamat internet (IP address), sehingga untuk alamat internet dua byte paling kiri dianggap nol. Format data yang didapat untuk alamat fisik adalah hexadecimal sedangkan untuk alamat internet dalam decimal.

Untuk alamat internet format data sebagai berikut:



Sebelum dilakukan pengambilan data-data ethernet, harus ditentukan dulu KODE KUNCI sebagai kode otentifikasi. Tentunya kode kunci ini berisi alamat fisik dan alamat internet dari host komputer yang kita izinkan untuk menjalankan program kita. Kode kunci ini bisa kita dapatkan menggunakan perintah 'ifconfig' dari shell linux. Karena C bekerja pada format decimal, maka untuk alamat fisik ethernet harus kita konversi-kan dalam bentuk decimal pada kode kuncinya.

Otentifikasi dilakukan dengan membandingkan data-data ethernet (alamat fisik dan alamat internet) pada komputer saat program dijalankan dengan kode kunci yang telah kita tentukan sebelumnya. Jika data-data ethernet sama dengan kode kunci maka program akan terus menjalankan PROGRAM UTAMA.

Penentuan kode kunci, pengambilan data-data ethernet, dan pengecekan kesamaan antara kode kunci dengan data-data internet diimplementasikan sebagai sebuah FUNGSI CEK yang menghasilkan variabel-variabel global bernilai 0 jika otentifikasi benar dan bernilai 1 jika otentifikasi salah. Sebelum menjalankan program yang lain, Program Utama akan menjalankan terlebih dahulu FUNGSI CEK dan memeriksa apakah variabel global bernilai 0 atau 1.

Pada program ini saya gunakan dua parameter otentifikasi yaitu otentifikasi alamat fisik ethernet dan otentifikasi alamat internet dari ethernet, sehingga dalam implementasi program terjadi dua kali otentifikasi pada Fungsi Cek dengan dua variabel global. Variabel global tersebut adalah 'y' untuk otentifikasi alamat fisik ethernet, dan 'z' untuk otentifikasi alamat internet ethernet.

Potongan program berikut menunjukkan proses penentuan kode kunci dan penentuan nilai variabel global 'y' untuk eth0 dengan alamat fisik sebagai kode kunci adalah 00:80:48:E3:44:20.

```
-----
/* variabel global 'y' untuk otentifikasi alamat fisik */
int y=0;

/* Fungsi Otentifikasi CEK untuk alamat fisik*/
int cek(void)
{
unsigned char *d0, *d1, *d2, *d3, *d4, *d5;
int kd0, kd1, kd2, kd3, kd4, kd5;

int fd;
struct ifreq ifr;

/* KODE KUNCI ALAMAT FISIK ETHERNET dari komputer sumber sebanyak 6 byte
   yaitu Hexa=00:80:48:E3:44:20 atau dalam Desimal=00:128:72:227:68:32 */
kd0 = 0;
kd1 = 128;
kd2 = 72;
kd3 = 227;
kd4 = 68;
kd5 = 32;

/* Mengambil data Alamat Fisik Ethernet */
fd = socket(PF_INET, SOCK_PACKET, htons(ETH_P_ALL));
strcpy(ifr.ifr_name, "eth0");
ioctl(fd, SIOCGIFHWADDR, &ifr);
d0 = ifr.ifr_hwaddr.sa_data;
d1 = d0+1;
d2 = d0+2;
d3 = d0+3;
d4 = d0+4;
d5 = d0+5;

/* Membandingkan Data Alamat Fisik Ethernet dengan Kode kunci*/
if ((*d0==kd0) && (*d1==kd1) && (*d2==kd2) && (*d3==kd3) && (*d4==kd4) &&
(*d5==kd5))
    y = 0;
else y = 1;

close(fd);
}
-----
```

Program di atas jika di-compile akan menghasilkan file *binary executable* (file hasil yang dijalankan) . Jika file binary executable tersebut di-edit menggunakan binary editor (atau hexa editor) maka akan ditemukan dengan mudah isi kode kunci 'kd0' sampai 'kd5'. Tentunya hal ini tidak kita harapkan karena seseorang yang mencoba menjalankan program tersebut dikomputer lain akan dapat mengubah kode kunci sesuai dengan alamat fisik ethernet card pada komputernya, yang

Erwin Syaifuddin 2003

akhirnya otentifikasi menjadi sukses. Untuk sedikit mengelabui kemungkinan di'bongkar' otentifikasi tersebut melalui binary editor oleh orang yang tidak berwenang, maka saya melakukan sedikit modifikasi pada kode kunci dan data-data yang diambil dari ethernet yaitu dengan cara menambahkan 1 pada tiap data, kecuali untuk data bernilai 255 akan menjadi 0 supaya angka-angka tersebut tidak keluar dari range nilai 1 byte.

Potongan program berikut menunjukkan proses pergeseran satu nilai ke atas (kecuali 255 menjadi 0) untuk kode kunci dan data alamat fisik ethernet.

```
-----  
unsigned char *d0, *d1, *d2, *d3, *d4, *d5;  
int kd0, kd1, kd2, kd3, kd4, kd5;  
int keyd0, keyd1, keyd2, keyd3, keyd4, keyd5;  
int data0, data1, data2, data3, data4, data5;  
  
/* KODE KUNCI ALAMAT FISIK ETHERNET dari komputer sumber sebanyak 6  
byte  
   yaitu Hexa=00:80:48:E3:44:20 atau dalam Desimal=00:128:72:227:68:32  
*/  
kd0 = 0;  
kd1 = 128;  
kd2 = 72;  
kd3 = 227;  
kd4 = 68;  
kd5 = 32;  
/* menggeser satu nilai ke atas kode alamat fisik, kecuali 255 menjadi  
0 */  
keyd0 = 1;  
keyd1 = 129;  
keyd2 = 73;  
keyd3 = 228;  
keyd4 = 69;  
keyd5 = 33;  
  
/* Mengambil data Alamat Fisik Ethernet */  
fd = socket(PF_INET, SOCK_PACKET, htons(ETH_P_ALL));  
strcpy(ifr.ifr_name, "eth0");  
ioctl(fd, SIOCGIFHWADDR, &ifr);  
d0 = ifr.ifr_hwaddr.sa_data;  
d1 = d0+1;  
d2 = d0+2;  
d3 = d0+3;  
d4 = d0+4;  
d5 = d0+5;  
  
/* Menggeser satu nilai ke atas data Alamat Fisik Ethernet
```

```
    kecuali jika data bernilai 255 maka data menjadi 0 */
if (*d0==255)
    data0 = 0;
else
    data0 = *d0 + 1;

if (*d1==255)
    data1 = 0;
else
    data1 = *d1 + 1;

if (*d2==255)
    data2 = 0;
else
    data2 = *d2 + 1;

if (*d3==255)
    data3 = 0;
else
    data3 = *d3 + 1;

if (*d4==255)
    data4 = 0;
else
    data4 = *d4 + 1;

if (*d5==255)
    data5 = 0;
else
    data5 = *d5 + 1;

/* Membandingkan Data Alamat Fisik Ethernet */
if ((data0==keyd0) && (data1==keyd1) && (data2==keyd2) &&
    (data3==keyd3) && (data4==keyd4) && (data5==keyd5))
    y = 0;
else y = 1;
```

Trik ini tidak sepenuhnya aman tapi paling tidak kode alamat fisik ethernet komputer yang terotentifikasi tidak bisa langsung kelihatan dari file *binary executable* nya.

Implementasi lengkap program otentifikasi ini saya tulis dalam file bernama *erwins.c*. Listing lengkap dari program otentifikasi ini dengan dua parameter otentifikasi alamat fisik dan alamat internet dari ethernet card sebagai berikut:

File: erwins.c

```
-----
#include <stdio.h>
#include <net/if.h>
#include <sys/ioctl.h>
#include <sys/socket.h>
#include <linux/if_ether.h>

/* deklarasi variabel global sebagai identifier
   y untuk otentifikasi alamat fisik ethernet
   z untuk otentifikasi alamat internet ethernet */
int y=0, z=0;

/* Permulaan Fungsi Otentifikasi CEK */
int cek(void)
{
unsigned char *d0, *d1, *d2, *d3, *d4, *d5;
unsigned char *a0, *a1, *a2, *a3, *a4, *a5;
int kd0, kd1, kd2, kd3, kd4, kd5;
int ka0, ka1, ka2, ka3, ka4, ka5;
int keyd0, keyd1, keyd2, keyd3, keyd4, keyd5;
int keya0, keya1, keya2, keya3, keya4, keya5;
int data0, data1, data2, data3, data4, data5;
int alamat0, alamat1, alamat2, alamat3, alamat4, alamat5;

int fd;
struct ifreq ifr;

/* KODE KUNCI ALAMAT FISIK ETHERNET dari komputer sumber sebanyak 6 byte
   yaitu Hexa=00:80:48:E3:44:20 atau dalam Desimal=00:128:72:227:68:32 */
kd0 = 0;
kd1 = 128;
kd2 = 72;
kd3 = 227;
kd4 = 68;
kd5 = 32;
/* menggeser satu nilai ke atas kode alamat fisik, kecuali 255 menjadi 0 */
keyd0 = 1;
keyd1 = 129;
keyd2 = 73;
keyd3 = 228;
keyd4 = 69;
keyd5 = 33;

/* KODE KUNCI ALAMAT INTERNET (IP Address) dari komputer sumber sebanyak 4
byte
   yaitu Desimal=10.255.57.101 atau dalam 6 byte Desimal=0.0.10.255.57.101
*/
ka0 = 0;
ka1 = 0;
ka2 = 10;
ka3 = 255;
```

```
ka4 = 57;
ka5 = 101;
/* Menggeser satu nilai ke atas alamat internet, kecuali 255 menjadi 0 */
keya0 = 1;
keya1 = 1;
keya2 = 11;
keya3 = 0;
keya4 = 58;
keya5 = 102;

/* Dalam prakteknya variabel 'ka' dan 'kd' harus dihilangkan untuk
mengelabui Binary Editor :) */

/* Mengambil data Alamat Fisik Ethernet */
fd = socket(PF_INET, SOCK_PACKET, htons(ETH_P_ALL));
strcpy(ifr.ifr_name, "eth0");
ioctl(fd, SIOCGIFHWADDR, &ifr);
d0 = ifr.ifr_hwaddr.sa_data;
d1 = d0+1;
d2 = d0+2;
d3 = d0+3;
d4 = d0+4;
d5 = d0+5;

/* Menggeser satu nilai ke atas data Alamat Fisik Ethernet
kecuali jika data bernilai 255 maka data menjadi 0 */
if (*d0==255)
    data0 = 0;
else
    data0 = *d0 + 1;

if (*d1==255)
    data1 = 0;
else
    data1 = *d1 + 1;

if (*d2==255)
    data2 = 0;
else
    data2 = *d2 + 1;

if (*d3==255)
    data3 = 0;
else
    data3 = *d3 + 1;

if (*d4==255)
    data4 = 0;
else
    data4 = *d4 + 1;

if (*d5==255)
```

```
data5 = 0;
else
    data5 = *d5 + 1;

/* Membandingkan Data Alamat Fisik Ethernet */
if ((data0==keyd0) && (data1==keyd1) && (data2==keyd2) && (data3==keyd3) &&
    (data4==keyd4) && (data5==keyd5))
    y = 0;
else y = 1;

printf("KODE KUNCI ALAMAT FISIK ETHERNET=
%d:%d:%d:%d:%d:%d\n", keyd0, keyd1, keyd2, keyd3, keyd4, keyd5);
printf("KODE DATA ALAMAT FISIK ETHERNET=
%d:%d:%d:%d:%d:%d\n", data0, data1, data2, data3, data4, data5);
printf("\n");
printf("KUNCI ALAMAT FISIK ETHERNET=
%02hX:%02hX:%02hX:%02hX:%02hX:%02hX\n", kd0, kd1, kd2, kd3, kd4, kd5);
printf("ALAMAT FISIK ETHERNET=
%02hX:%02hX:%02hX:%02hX:%02hX:%02hX\n", *d0, *d1, *d2, *d3, *d4, *d5);
printf("-----\n");

/* Mengambil data Alamat Internet */
ioctl(fd, SIOCGIFADDR, &ifr);
a0 = ifr.ifr_addr.sa_data;
a1 = a0+1;
a2 = a0+2;
a3 = a0+3;
a4 = a0+4;
a5 = a0+5;

/* Menggeser data Alamat Internet satu nilai ke atas, kecuali 255 menjadi 0
*/
if (*a0==255)
    alamat0 = 0;
else
    alamat0 = *a0 +1;

if (*a1==255)
    alamat1 = 0;
else
    alamat1 = *a1 +1;

if (*a2==255)
    alamat2 = 0;
else
    alamat2 = *a2 +1;

if (*a3==255)
    alamat3 = 0;
else
    alamat3 = *a3 +1;
if (*a4==255)
```

```
alamat4 = 0;
else
    alamat4 = *a4 +1;

if (*a5==255)
    alamat5 = 0;
else
    alamat5 = *a5 +1;

/* Membandingkan Data Alamat Internet */
if ((alamat0==keya0) && (alamat1==keya1) && (alamat2==keya2) &&
(alamat3==keya3) && (alamat4==keya4) && (alamat5==keya5))
    z = 0;
else z = 1;

printf("KODE KUNCI ALAMAT INTERNET=
%d.%d.%d.%d.%d.%d\n", keya0, keya1, keya2, keya3, keya4, keya5);
printf("DATA ALAMAT INTERNET=
%d.%d.%d.%d.%d.%d\n", alamat0, alamat1, alamat2, alamat3, alamat4, alamat5);
printf("\n");
printf("KUNCI ALAMAT INTERNET= %d.%d.%d.%d\n", ka2, ka3, ka4, ka5);
printf("ALAMAT INTERNET= %d.%d.%d.%d\n", *a2, *a3, *a4, *a5);
printf("-----\n");

close(fd);
}
/* Akhir dari Fungsi CEK */

/* PROGRAM UTAMA */
int main()
{
printf("\n");
printf("Identifler awal y = %d, z = %d\n", y, z);
printf("-----\n");
cek();
printf("Identifler cek y = %d, z = %d\n", y, z);
printf("===== \n");
if ((y==1) || (z==1))
{
printf("STOP! INI ADALAH SEJENIS 'COPY PROTECTION PROGRAM'\n");
printf("ANDA TIDAK DIIZINKAN MENJALANKANNYA DI KOMPUTER INI !!\n");
}
else
{
printf("Komputer Anda ter-otentifikasi untuk menggunakan program ini.\n");
printf("Silahkan diteruskan....\n");
printf("\n");
/* Anda dapat meletakkan program utama anda di sini */
}
}
}
```

Pada program lengkap di atas saya menambahkan beberapa perintah *printf* untuk menampilkan secara jelas hasil langkah yang dilakukan oleh program, termasuk mendeteksi bekerja normal atau tidaknya program. Perbandingan data ethernet dengan kode kunci dilakukan dengan perintah *if* menggunakan operator logika AND (&&) karena parameter yang dibandingkan terdiri dari enam buah dimana keenamnya harus memberikan kondisi yang benar supaya memenuhi persyaratan *if*. Sementara itu pendeteksian valid tidaknya kedua alamat fisik dan alamat internet dilakukan pada program utama dengan menggunakan *if* dan operator logika OR (||). Di sini dimanfaatkan output dari Fungsi Cek yang mengubah variabel global *y* dan *z* menjadi 1 jika kondisi tidak terpenuhi (data-data ethernet tidak valid). Dengan operator logika OR pada *if* berarti jika satu saja dari variabel global *y* atau *z* bernilai 1 (alamat fisik atau alamat internet, atau kedua nya berbeda dengan kode kunci), berarti komputer yang menjalankan program tidak ter-otentifikasi.

PENGUJIAN PROGRAM

Pengujian program otentifikasi ini cukup saya lakukan pada satu komputer saja, dimana yang saya ubah hanya Kode-kode kunci nya saja. Selanjutnya dengan mengamati output yang dihasilkan program, terutama identifier *y* dan *z* maka saya pastikan program ini berjalan sesuai dengan harapan.

Output dari program ini jika dijalankan pada komputer yang terotentifikasi berdasarkan alamat fisik dan alamat internet ethernet yang sama dengan kode kunci program sebagai berikut:

```
[Linux_Shell]# erwins

Identifier awal y = 0, z = 0
-----
KODE KUNCI ALAMAT FISIK ETHERNET= 1:129:73:228:69:33
KODE DATA ALAMAT FISIK ETHERNET= 1:129:73:228:69:33

KUNCI ALAMAT FISIK ETHERNET= 00:80:48:E3:44:20
ALAMAT FISIK ETHERNET= 00:80:48:E3:44:20
-----
KODE KUNCI ALAMAT INTERNET= 1.1.11.0.58.102
DATA ALAMAT INTERNET= 1.1.11.0.58.102

KUNCI ALAMAT INTERNET= 10.255.57.101
ALAMAT INTERNET= 10.255.57.101
-----
Identifier cek y = 0, z = 0
```

```
=====  
Komputer Anda ter-otentifikasi untuk menggunakan program ini.  
Silahkan diteruskan....
```

Output program jika dijalankan pada komputer dengan alamat fisik ethernet yang berbeda dengan kode kunci meskipun alamat internet sama:

```
[Linux_Shell]# erwins  
  
Identifier awal y = 0, z = 0  
-----  
KODE KUNCI ALAMAT FISIK ETHERNET= 1:14:101:228:69:33  
KODE DATA ALAMAT FISIK ETHERNET= 1:129:73:228:69:33  
  
KUNCI ALAMAT FISIK ETHERNET= 00:0D:64:E3:44:20  
ALAMAT FISIK ETHERNET= 00:80:48:E3:44:20  
-----  
KODE KUNCI ALAMAT INTERNET= 1.1.11.0.58.102  
DATA ALAMAT INTERNET= 1.1.11.0.58.102  
  
KUNCI ALAMAT INTERNET= 10.255.57.101  
ALAMAT INTERNET= 10.255.57.101  
-----  
Identifier cek y = 1, z = 0  
=====
```

```
STOP! INI ADALAH SEJENIS 'COPY PROTECTION PROGRAM'  
ANDA TIDAK DIIZINKAN MENJALANKANNYA DI KOMPUTER INI !!
```

Output program jika dijalankan pada komputer dengan alamat fisik ethernet yang sama dengan kode kunci tetapi alamat internetnya berbeda, sebagai berikut:

```
[Linux_Shell]# erwins  
  
Identifier awal y = 0, z = 0  
-----  
KODE KUNCI ALAMAT FISIK ETHERNET= 1:129:73:228:69:33  
KODE DATA ALAMAT FISIK ETHERNET= 1:129:73:228:69:33  
  
KUNCI ALAMAT FISIK ETHERNET= 00:80:48:E3:44:20  
ALAMAT FISIK ETHERNET= 00:80:48:E3:44:20  
-----  
KODE KUNCI ALAMAT INTERNET= 1.1.168.206.50.51  
DATA ALAMAT INTERNET= 1.1.11.0.58.102
```

```
KUNCI ALAMAT INTERNET= 167.205.49.50
ALAMAT INTERNET= 10.255.57.101
-----
Identifier cek y = 0, z = 1
=====
STOP! INI ADALAH SEJENIS 'COPY PROTECTION PROGRAM'
ANDA TIDAK DIIZINKAN MENJALANKANNYA DI KOMPUTER INI !!
```

Output program jika dijalankan pada komputer dengan alamat fisik dan alamat internet ethernet yang berbeda dengan kode kunci:

```
[Linux_Shell]# erwins

Identifier awal y = 0, z = 0
-----
KODE KUNCI ALAMAT FISIK ETHERNET= 1:129:204:228:69:16
KODE DATA ALAMAT FISIK ETHERNET= 1:129:73:228:69:33

KUNCI ALAMAT FISIK ETHERNET= 00:80:CB:E3:44:0F
ALAMAT FISIK ETHERNET= 00:80:48:E3:44:20
-----
KODE KUNCI ALAMAT INTERNET= 1.1.168.206.50.51
DATA ALAMAT INTERNET= 1.1.11.0.58.102

KUNCI ALAMAT INTERNET= 167.205.49.50
ALAMAT INTERNET= 10.255.57.101
-----
Identifier cek y = 1, z = 1
=====
STOP! INI ADALAH SEJENIS 'COPY PROTECTION PROGRAM'
ANDA TIDAK DIIZINKAN MENJALANKANNYA DI KOMPUTER INI !!
```

KESIMPULAN

Dari pembahasan di atas jelas bahwa pembuatan program dan otentifikasinya sangat tergantung dari data Ethernet Card. Untuk program-program pribadi yang tidak kita inginkan di-copy oleh orang lain, cara ini bisa dipertimbangkan. Tapi untuk program yang akan kita jual dalam jumlah banyak maka cara ini akan menyulitkan karena kita diharuskan meng-compile ulang program untuk tiap komputer dengan alamat fisik dan alamat internet yang berbeda, kecuali jika program

yang kita jual menyertakan ethernet card, dan kita yang menentukan alamat internet untuk komputer yang akan diinstall – suatu hal yang kemungkinan besar jarang terjadi.

REFERENSI

1. Kris Jamsa, **1001 Tip C/C++**, Penerbit ANDI Yogyakarta, 1996.
2. Neil Matthew & Richard Stones, **Beginning Linux Programming**, Wrox Press, 1996.
3. http://www.experts-exchange.com/Programming/Programming_Platforms/Linux_Programming/
4. Online Manual RedHat Linux 7.3.
5. <http://www.globetrotter.com>