

# Interkoneksi IPv6 dan IPv4 dengan Mekanisme Automatic Tunneling

**Wahidi Somad**

ste99120@stttelkom.ac.id

<http://wah.id.gg>

## ***Lisensi Dokumen:***

*Copyright © 2003 IlmuKomputer.Com*

*Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

## **Pendahuluan**

Dengan perkembangan teknologi internet yang semakin pesat menyebabkan user internet semakin bertambah banyak. Tetapi penambahan user internet tidak diimbangi dengan jumlah alamat IPv4 yang ada. Sehingga IETF mengeluarkan standart protocol IP baru yang disebut IPng ( Internet Protokol Next Generations ) atau disebut juga IPv6.

IPv6 mempunyai format alamat dan header yang berbeda dengan IPv4. Sehingga secara langsung IPv4 tidak bisa interkoneksi dengan IPv6. Hal ini tentunya akan menimbulkan masalah pada implementasi IPv6 pada jaringan internet IPv4 yang telah ada. Sebagai solusi masalah implementasi IPv6 ini diperlukan suatu mekanisme Transisi IPv6. Tujuan pembuatan mekanisme transisi ini adalah supaya paket IPv6 dapat dilewatkan pada jaringan IPv4 yang telah ada ataupun sebaliknya. Pada tutorial ini akan dibahas cara implementasi mekanisme Automatic Tunneling untuk interkoneksi IPv6 dan IPv4 , terutama pada operating system Linux.

## **Automatic Tunneling**

**2.4.1.**

**2.4.2.**

**2.4.3.**

**2.4.4.**

**2.4.5.**

**2.4.6.**

**2.4.7.**

**2.4.8.**

**2.4.9.**

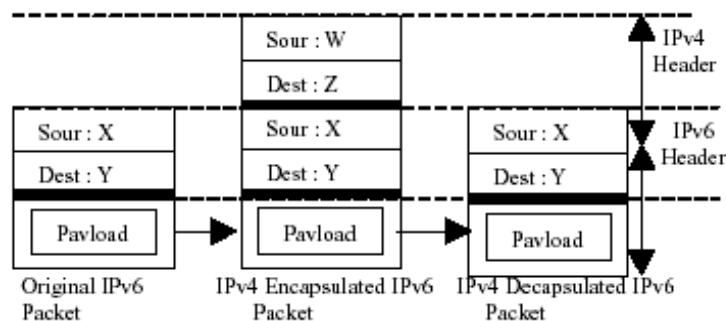
**2.4.10.**

**2.4.11.**

2.4.12.  
2.4.13.  
2.4.14.  
2.4.15.  
2.4.16.  
2.4.17.  
2.4.18.  
2.4.19.  
2.4.20.  
2.4.21.

Mekanisme Automatic Tunneling berfungsi melewati paket IPv6 melalui jaringan IPv4 yang telah ada, tanpa merubah infrastruktur jaringan IPv4.

Mekanisme Automatic Tunneling mempunyai prinsip kerja mengenkapsulasi paket IPv6 dengan header IPv4, kemudian paket tersebut langsung dikirimkan ke jaringan IPv4



**Gambar1. Enkapsulasi Paket pada proses Tunneling**

( Sumber *Application Performance Analysis in Transition Mechanism from IPv4 to IPv6*, Ettikan Kandasamy, hal. 3 )

Fungsi dari enkapsulasi paket IPv6 dengan header IPv4 adalah agar paket dapat diroutingkan oleh router IPv4. Namun dengan penambahan header IPv4 ini tentunya paket akan bertambah besar sesuai panjang header IPv4 yaitu 20 byte. Pertambahan panjang paket ini akan berakibat bertambah pula waktu delay pengiriman paket. Masalah utama dari implementasi mekanisme Transisi IPv6 adalah pertambahan waktu delay proses yang diakibatkan pertambahan panjang paket, adanya proses enkapsulasi dan adanya proses dekapsulasi.

Mekanisme Transisi Automatic Tunneling mempunyai keuntungan sebagai berikut :

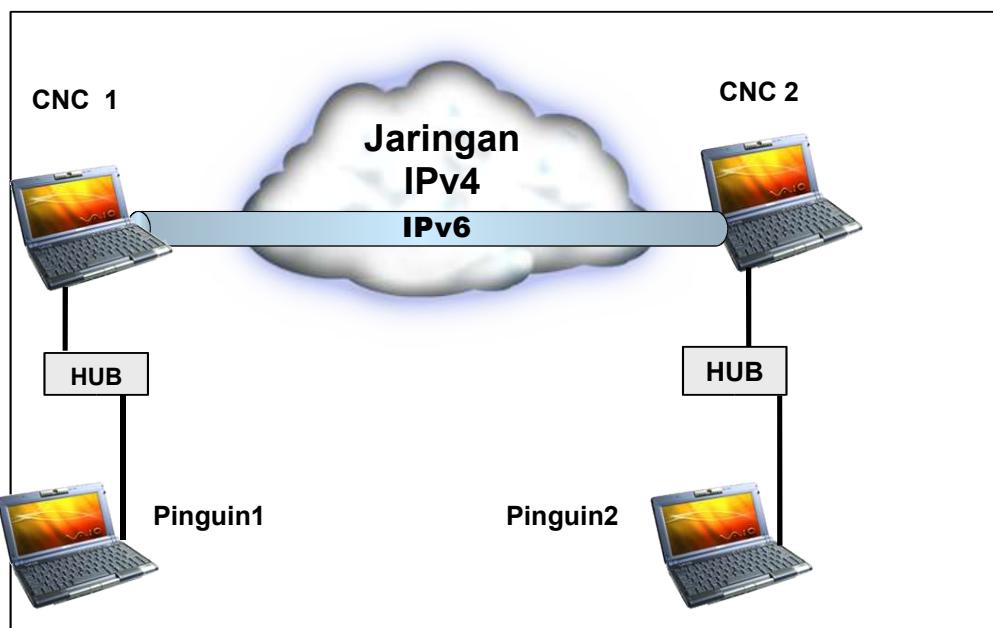
1. Lebih mudah dalam implementasi  
Dalam implementasi tidak memerlukan banyak computer ,cukup menggunakan computer yang sudah ada .Khusus untuk gateway tunnel,operating system perlu diupgrade menjadi operating system yang dualstack mendukung IPv6 dan IPv4.
2. Lebih mudah dalam hal konfigurasi pada system operasi  
Dalam konfigurasi tidak diperlukan script – script konfigurasi yang rumit , cukup dengan konfigurasi interface tunnel dan konfigurasi table routingnya saja.
3. Tidak memerlukan server yang melayani transisi  
Dalam implementasi tidak memerlukan server khusus yang melayani mekanisme transisinya , enkapsulasi dan dekapulasi di lakukan antar gateway tunnel secara point– to– point.

Kelemahan Mekanisme Transisi Automatic Tunneling :

1. Tidak bisa diimplementasikan jika client transisi berada di dalam router NAT
2. Akan menyebabkan penambahan delay ( waktu proses )
3. Rentan terhadap serangan DDOS.

## Implementasi Mekanisme Automatic Tunneling

Untuk mempermudah pemahaman tentang konfigurasi dan implementasi mekanisme automatic tunneling maka akan dibuat contoh kasus konfigurasi sebagai berikut .



Misalkan dengan konfigurasi seperti gambar diatas dengan kondisi sebagai berikut :

- CNC 1 adalah PC dualstack dengan alokasi IPv4 adalah 202.120.120.1  
Alokasi prefix IPv6 adalah : 2002:ca78:7801::/48  
Alokasi alamat IPv6 adalah : 2002:ca78:7801::1/48
- Pinguin1 adalah host IPv6 dengan alokasi IPv6 2002:ca78:7801::2/48
- CNC 2 adalah PC dualstack dengan alokasi IPv4 adalah 202.81.81.2  
Alokasi prefix IPv6 adalah : 2002:ca51:5102::/48  
Alokasi alamat IPv6 adalah : 2002:ca51:5102::1/48
- Pinguin2 adalah host IPv6 dengan alokasi alamat IPv6 2002:ca51:5102::2/48

Untuk membangun sebuah sistem mekanisme transisi Automatic Tunneling diperlukan beberapa langkah-langkah yang harus dilakukan diantaranya :

1. Desain arsitektur sistem Automatic Tunneling
2. Implementasi 2 Automatic Tunelling gateway (Dual Stack )( dalam kasus diatas adalah PC CNC 1 dan CNC 2 )
3. Pengaturan Routing IPv6 pada gateway dan client Tunnel
4. Pengujian mekanisme Automatic Tunnell.

## Perancangan arsitektur Sistem Automatic Tunelling

Komponen peyusun sistem Automatic Tunneling terdiri dari: Tunnel Client IPv6 , dan 2 Router dualstack untuk gateway tunnel .Arsitektur sistem Automatic Tunneling adalah sebagai berikut :

### Implementasi Gateway Tunneling pada CNC 1 (Dual Stack)

Tunneling Gateway pada buku ini akan diambil contoh dengan menggunakan sistem operasi Linux Redhat 9.0.

Untuk host CNC 1 harus dialokasikan dengan menggunakan alamat IPv6 global supaya dapat di routekan pada jaringan IPv6 dan Internet.Format alamat IPv6 untuk gateway tunnel dalam sebagai berikut :

Prefix Global :	Alamat IPv4 dalam hexadesimal
<b>2002:ca78:7801::/48</b>	

2002 = prefix global  
ca78:7801= Alamat IPv4 dalam hexa ( 202.120.120.1 = ca78:7801 )

Supaya operating system kita mendukung IPv6 kita perlu instalasi modul IPv6 adapun perintah untuk aktivasi modulnya adalah:

```
# insmod ipv6
```

Misalkan yang terhubung ke jaringan IPv4 adalah interface eth0 dan yang terhubung ke clinet tunnel adalah eth1 .Setelah itu kita perlu konfigurasi alamat IPv4 yaitu dengan perintah :

```
# ifconfig eth0 202.120.120.1 netmask 255.255.255.0 up
```

untuk pengecekan interface kita gunakan perintah :

```
# ifconfig eth0
```

Langkah selanjutnya adalah mengaktifkan interface untuk tunneling yang digunakan untuk membangun jembatan menembus jaringan IPv4.Aktivasinya dengan menggunakan perintah :

```
# ifconfig sit0 up
```

Dan konfigurasi address tunnelingnya adalah dengan format sebagai berikut :

*:: address IPv4 host /prefix*

Dalam implementasi gateway tunnel address IPv4 pada eth0 yang digunakan adalah 202.10.120.1 maka format address yang dimunculkan pada interface sit0 pada CNC 1 adalah :: 202.120.120.1 dengan prefix 96 .Biasanya alamat ini akan terkonfigurasi secara otomatis saat kita aktivasi sit0 .Untuk mengkonfigurasinya dapat dilakukan dengan perintah :

```
# ifconfig sit0 add ::202.120.120.1/96 up atau untuk menghapusnya digunakan  
#ifconfig sit0 del ::202.120.120.1/96
```

untuk pengecekan interface sit0 kita gunakan perintah :

```
# ifconfig sit0

sit0  Link encap:IPv6-in-IPv4
      inet6 addr: ::127.0.0.1/96 Scope:Unknown
      inet6 addr: ::202.120.120.1/96 Scope:Compat
      UP RUNNING NOARP MTU:1480 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:0 (0.0 b) TX bytes:0 (0.0)
```

Sehingga akan muncul tampilan diatas .Alamat ::202.120.120.1 adalah merupakan alamat IPv6 kompatibel IPv4.Yang biasanya alamat IPv6 kompatibel IPv4 ini digunakan untuk interface – interface tunnel 6over4.

Selanjutnya kita konfigurasi alamat IPv6 pada interface eth1.Alatmat IPv6 untuk eth1 harus menggunakan prefik yang tadi telah kita hitung yaitu : 2002:ca78:7801::1/48 .Perintah yang digunakan :

```
#ifconfig eth1 add 2002:ca78:7801::1/48 up
```

Untuk mengecek konfigurasinya kita gunakan :

```
# ifconfig eth1
```

Langkah yang terakhir untuk konfigurai gateway tunell CNC 1 ini adalah dengan mengkonfigurasi entri tabel routingnya ,inti table routingnya yaitu semua alamat yang bertujuan ke CNC 2 atau dengan prefix 2002:ca51:5102::/48 di lewatkan ke interface tunnel sit0 , yaitu dengan perintah:

```
#route -A inet6 add 2002:ca51:5102::/48 gw ::202.120.120.1 dev sit0
```

supaya gateway dapat memforward paket IPv6 maupun IPv4 maka perlu kita lakukan aktivasi IP forward pada system linux kita dengan perintah :

```
# echo "1" > /proc/sys/net/ipv4/conf/all/forwarding
# echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
```

Untuk menghindari supaya tabel routingnya tidak berubah saat di reboot atau di matikan maka kita bisa meletakkan beberapa script pada file /etc/rc.local.

```
/sbin/insmod ipv6
/sbin/ifconfig eth0 add 2002:ca78:7801::1/48 up
/sbin/ifconfig sit0 up
/sbin/route -A inet6 add 2002:ca51:5102::/48 gw ::202.120.120.1 dev sit0
echo "1" > /proc/sys/net/ipv4/conf/all/forwarding
echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
```

kemudian anda save .

## Implementasi Gateway Tunneling CNC 2

Sama halnya dengan implementasi gateway tunnel CNC 1 .Gateway Tunnel CNC 2 mempunyai prinsip konfigurasi yang sama. Untuk alamat IPv6 yang dipakai

Prefix Global :	Alamat IPv4 dalam hexadesimal
2002:ca51:5102::/48	

2002 : prefix global  
ca51:5102 : Alamat IPv4 CNC 2 dalam hexa (202.81.81.2 = ca51:5102 )

Supaya operating system kita mendukung IPv6 kita perlu instalasi modul IPv6 adapun perintah untuk aktivasi modulnya adalah:

```
# insmod ipv6
```

Misalkan yang terhubung ke jaringan IPv4 adalah interface eth0 dan yang terhubung ke client tunnel adalah eth1 .Setelah itu kita perlu konfigurasi alamat IPv4 yaitu dengan perintah :

```
# ifconfig eth0 202.81.81.2 netmask 255.255.255.0 up
```

Seperti halnya konfigurasi pada gateway tunnel CNC 1 maka kita perlu aktivasi interface tunnel sit0 .

```
#ifconfig sit0 up
```

Dan kemudian kita lihat konfigurasi sit0

```
#ifconfig sit0
```

```
sit0 Link encap:IPv6-in-IPv4
      inet6 addr: ::127.0.0.1/96 Scope:Unknown
      inet6 addr: ::202.81.81.2/96 Scope:Compat
      UP RUNNING NOARP MTU:1480 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:0 (0.0 b) TX bytes:0 (0.0)
```

Apabila inet6 address sudah terkonfigurasi alamat IPv6 kompatibel IPv4 maka kita tidak perlu melakukan penambahan alamat pada interface sit0.

Setelah itu kita perlu konfigurasi untuk interface yang terhubung ke jaringan IPv6 client tunnel yaitu interface eth1.Sesuai dengan perhitungan format address tunneling maka kita akan memasukan / mengkonfigurasi eth1 dengan alamat IPv6 2002:ca51.5102::1/48 .Adapun perintah yang digunakan :

```
#ifconfig eth1 add 2002:ca51.5102::1/48 up
```

Jangan lupa untuk konfigurasi gateway tunnel ini kita mengatur dan menambah entry table routing IPv6 . Agar paket yang bertujuan ke CNC 1 yaitu dengan prefix 2002:ca78:7801::/48 di lewatkan ke interface tunnel.

```
#route -A inet6 add 2002:ca78:7801::/48 gw ::202.81.81.2 dev sit0
# echo "1" > /proc/sys/net/ipv4/conf/all/forwarding
# echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
```

Untuk menghindari supaya tabel routingnya tidak berubah maka kita perlu meletakkan beberapa script pada file /etc/rc.local.

```
/sbin/insmod ipv6
```

```
#ifconfig eth1 add 2002:ca51:5102::1/48 up
#route -A inet6 add 2002:ca78:7801::/48 gw ::202.81.81.2 dev sit0
# echo "1" > /proc/sys/net/ipv4/conf/all/forwarding
# echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
```

Jangan lupa anda save .File yang ada pada /etc/rc.local akan di eksekusi setelah system linux selesai melakukan booting.

### Implementasi Client tunnel Penguin 1

Client tunnel Penguin 1 berada dibawah layanan gateway tunnel CNC 1 , maka kita harus membuat alamat prefix dari Penguin 1 ini sama dengan alamat prefix dari gateway tunnel CNC 1 . Yaitu 2002:ca78:7801::/48 , maka kita dapat mengalokasikan alamat untuk client tunnel Penguin1 yaitu : 2002:ca78:7801::2/48 .Jangan lupa untuk mengatur routing IPv6 pada client tunnel ini .Semua paket yang bertujuan ke prefix CNC 2 kita lewatkan ke gateway tunnel.Kita dapat konfigurasi dengan perintah sebagai berikut :

```
#ifconfig eth0 add 2002:ca78:7801::2/48 up
#route -A inet6 add 2002:ca51:5102::/48 gw 2002:ca78:7801::1 dev eth0
```

### Implementasi Client tunnel Penguin 2

Client tunnel Penguin 2 berada dibawah layanan gateway tunnel CNC 2 , maka kita harus membuat alamat prefix dari Penguin 2 ini sama dengan alamat prefix dari gateway tunnel CNC 2 . Yaitu 2002:ca51:5102::/48 , maka kita dapat mengalokasikan alamat untuk client tunnel Penguin2 yaitu : 2002:ca51:5102::2/48 .Jangan lupa untuk mengatur routing IPv6 pada client tunnel ini .Semua paket yang bertujuan ke prefix CNC 1 kita lewatkan ke gateway tunnel CNC 2.Kita dapat konfigurasi dengan perintah sebagai berikut :

```
#ifconfig eth0 add 2002:ca51:5101::2/48 up
#route -A inet6 add 2002:ca78:7801::/48 gw 2002:ca51:5102::1 dev eth0
```

### Tahap pengujian konektivitas mekanisme Automatic Tunnel

Untuk pengujian konektivitas kita bisa lakukan dengan program traceroute6 atau pun ping6.Pengecekan konektifitas sebaiknya dilakukan antar gateway tunnel dulu.

Misal dari Host CNC 1 , lakukan :

```
#ping6 :: 202.81.81.2
#ping6 2002:ca51:5102::1
#ping6 2002:ca51:5102::2
```

Dari host **penguin 1**

```
#ping6 2002:ca51:5102::1
#ping6 2002:ca51:5102::2
```

Apabila semua sudah replay insyaallah konfigurasi tunneling kita sudah berjalan.Apabila kita tertarik menangkap dan menganalisa paket hasil enkapsulasi kita dapat melakukan snifer pada interface terluar dari gateway tunnel dalam hal ini eth0 pada host **CNC 1** dan **CNC 2**.

Apabila anda tertarik hasil enkapsulasi pada mekanisme transisi Automatic Tunneling ini , pada tutorial ini saya lampirkan contoh hasil capturing paket ICMPv6 hasil output program ethereal.

```
.....
..
Frame 1 (138 on wire, 138 captured)
  Arrival Time: Feb  4, 2004 00:42:27.009321000
  Time delta from previous packet: 0.000000000 seconds
  Time relative to first packet: 0.000000000 seconds
  Frame Number: 1
```

Packet Length: 138 bytes  
Capture Length: 138 bytes

#### Ethernet II

Destination: 00:50:da:7d:fe:11 (3COM\_7d:fe:11)  
Source: 00:50:da:7d:fd:eb (3COM\_7d:fd:eb)  
Type: IP (0x0800)

#### Internet Protocol, Src Addr: 172.168.1.2 (172.168.1.2), Dst Addr: 192.168.1.2 (192.168.1.2)

Version: 4  
Header length: 20 bytes  
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)  
    0000 00.. = Differentiated Services Codepoint: Default (0x00)  
    .... ..0. = ECN-Capable Transport (ECT): 0  
    .... ...0 = ECN-CE: 0  
Total Length: 124  
Identification: 0x0000  
Flags: 0x04  
    ..1.. = Don't fragment: Set  
    ..0. = More fragments: Not set  
Fragment offset: 0  
Time to live: 64  
Protocol: IPv6 (0x29)  
Header checksum: 0xcb04 (correct)  
Source: 172.168.1.2 (172.168.1.2)  
Destination: 192.168.1.2 (192.168.1.2)

#### Internet Protocol Version 6

Version: 6  
Traffic class: 0x00  
Flowlabel: 0x00000  
Payload length: 64  
Next header: ICMPv6 (0x3a)  
Hop limit: 64  
Source address: 2002:aca8:102::1 (2002:aca8:102::1)  
Destination address: 2002:c0a8:102::1 (2002:c0a8:102::1)

#### Internet Control Message Protocol v6

Type: 128 (Echo request)  
Code: 0  
Checksum: 0x923f (correct)  
ID: 0xf330  
Sequence: 0x1b00  
Data (56 bytes)

```
0000  c3 bc 2b 40 4e b9 07 00 08 09 0a 0b 0c 0d 0e 0f  ..+@N.....
0010  10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f  .....
0020  20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f  !"#$%&'()*+,-./
0030  30 31 32 33 34 35 36 37                          01234567
```

.....  
.

## Kesimpulan

Untuk implementasi IPv6 pada jaringan internet IPv4 dibutuhkan suatu mekanisme transisi , mekanisme transisi IPv6 yang akan kita gunakan harus sesuai dengan kondisi jaringan kita.Misalkan kita ingin melewati paket Ipv4 melalui Ipv6 anda bisa menggunakan mekanisme DSTM.Untuk melewati paket Ipv6 melalui NAT Ipv4 anda bisa menggunakan mekanisme Teredo.Hampir semua mekanisme transisi menggunakan sistem enkapsulasi dan dekapsulasi . Hal ini akan berakibat bertambahnya panjang paket dan bertambahnya waktu delay akibat proses enkapsulasi dan dekapsulasi.



## Referensi

1. Linux IPv6 How To
2. Marc Blanchet, Florent parent, "IPv6 Transition Mechanisms", Viagenie, 2000
3. Ettikan Kandasamy, Application Performance Analysis in Transition Mechanism from IPv4 to IPv6, Cyberjaya, Malaysia.
4. <http://penguin.stttelkom.ac.id>

## BIOGRAFI PENULIS



**Wahidi Somad.** Biasa dipanggil **Wahid**, Lahir di Klaten, 2 Mei 1981. Sekarang masih terdaftar sebagai mahasiswa tingkat akhir STT Telkom Bandung. Aktif dalam melakukan riset bersama SM dan Asisten Laboratorium Computer & Communications STT Telkom Bandung. Menjadi system administrator server [penguin.stttelkom.ac.id](http://penguin.stttelkom.ac.id). Bidang kesukaan adalah Komunikasi data, networking, Linux, FreeBSD, IPv6, WLAN, Security Network dan VOIP (baru belajar). Kegiatan saat ini adalah menjadi asisten Lab.C&C Stttelkom, Asisten Dosen matakuliah Komunikasi Data, Perangkat Keras & LAN, Jaringan Komunikasi

Data dosen Gunawan Adi Sulistyono, ST (Boz Goen) dan menjadi tenaga Freelance pada CV. Technia Karya Abadi sebagai IT Engineer.

### Ucapan Terimakasih :

Semua crew Lab. Computer And Communications atas kebersamaan dalam melakukan riset dan proses belajar bersama. Pak asep, Bos goen, STT Telkom atas segala fasilitas yang diberikan. Salam SPTJS buat crew penguin (Bu6z, Thunder, Mbah Arj0, DragoN, P-Biru, Mandrix, Paghist)

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:

URL: <http://wah.id.gg>

URL: <http://penguin.stttelkom.ac.id>

Email: [ste99120@stttelkom.ac.id](mailto:ste99120@stttelkom.ac.id)