Pengantar IPv6 dan Implementasinya di Linux

Wahidi Somad
Ste99120@stttelkom.ac.id
http://wah.id.gg

Dewasa ini perkembangan user dan tuntutan perkembangan aplikasi internet semakin pesat.Ipv4 yang sudah terbukti tangguh menopang internet sekarang mulai bermasalah dengan semakin berkurangnya alokasi ip address yang tersedia.Walaupun IPv4 cukup suskses dalam efisiensi address dengan penggunaan NAT *(Network Address Translation)*,tetapi tuntutan aplikasi internet yang bersifat realtime dan aman tidak dapat terpenuhi.Karena NAT menghambat aplikasi yang bersifat end to end user ,seperti Videoconference dll.Penggunaan IPv6 adalah solusi yang tepat untuk menopang internet sekarang.Banyak keuntungan yang diambil dari penggunaan IPv6 yaitu : Alokasi address yang lebih banyak,Auto configuration address,Adanya traffic class dan flow label untuk mendukung aplikasi realtime dan Ipv6 mendukung mobile ip,IPsec dll.Bagaimana mengkofigurasi IPv6 pada operting system kita?pada tutorial ini akan dijelaskan implementasi IPv6 pada operating system linux distro RedHat.

1. Sejarah Implementasi IPv6 pada Sistem Operasi Linux

Kode network yang berhubungan dengan IPv6 dimasukkan ke kernel linux untuk pertama kalinya pada kernel versi 2.1.8 pada bulan November tahun 1996 oleh Pedro Roque. Penambahan ini berdasarkan pada BSD API seperti pada BSD system. Karena kekurangan sumber daya manusia, maka pengembangan implementasi IPv6 pada kernel linux tidak dapat dilanjutkan dengan penyusunan drafts atau RFC. Namun kemudian pada bulan Oktober 2000, sebuah proyek dijepang (yang sering disebut USAGI), mengembangkan IPv6 yang didukung oleh Linux, dengan cara mengikuti implementasi IPv6 pada sistem operasi FreeBSD yang dikembangkan oleh KAME Project. Namun sangat disayangkan bahwa USAGI patch (tambahan) ini terlalu besar, sehingga

pemelihara jaringan linux yang ada sekarang tidak mampu untuk menambahkannya pada source kernel linux seri 2.4.x. Sehingga kernel 2.4.x kehilangan banyak perkembangan dan tidak dapat memenuhi permintaan dari drafts dan RFC yang ada pada saat itu. Hal ini dapat menyebabkan beberapa masalah dalam interkoneksi dengan sistem operasi yang lain. Saat ini USAGI menggunakan kernel linux versi pengembangan 2.5.x untuk menempatkan seluruh hasil pengembangannya pada kernel versi ini. Dengan demikian diharapkan pada linux kernel versi 2.6.x akan berisi implementasi IPv6 yang sangat lengkap.

2. Pengecekan Dukungan IPv6 di Linux

Untuk menjalankan seluruh aplikasi dan perangkat lunak IPv6 maka kita harus mengimplementasikan IPv6 pada sistem operasi kita. Pada sistem operasi Linux untuk menjalankan IPv6 kita harus menggunakan kernel linux yang mendukung IPv6, yaitu distribusi kernel diatas kernel 2.1.x. Pada distribusi linux dengan versi kernel stabil 2.2.x (seperti RedHat 6.2), modul IPv6 belum terkompilasi. Sehingga apabila kita ingin mengaktifkan modul IPv6 tersebut kita perlu melakukan kompilasi kernel, dengan memilih option pada ***networking*** => ***IPv6 enabled*** . Sedangkan untuk distribusi dengan kernel 2.4.x (seperti RedHat 7.2) modul IPv6 sudah terkompilasi namun belum terinstall. Sehingga untuk mengaktifkannya kita hanya perlu menginstall modul IPv6 tersebut.

Untuk mengecek apakah kernel yang sedang berjalan mendukung IPv6 atau tidak, kita bisa melihat pada file system /proc, yang harus memiliki direktori : /**`proc/net/if_inet6`**
Untuk melihatnya kita dapat menggunakan perintah seperti dibawah ini :

```
# test -f /proc/net/if_net6 && echo “Kernel Mendukung IPv6”
Kernel Mendukung IPv6
```

Jika direktori tersebut belum ada maka IPv6 belum termuat dalam kernel, sehingga perlu kita muat terlebih dahulu.

3. Memuat Modul IPv6 Kernel

Memuat modul IPv6 bertujuan untuk mengaktifkan modul yang akan digunakan untuk menangani IPv6 baik konfigurasi maupun interkoneksi. Kita bisa mencoba memuat modul IPv6

tersebut dengan menggunakan perintah :

```
# insmod ipv6
Using /lib/modules/2.4.7-10/kernel/net/ipv6/ipv6.o
```

Untuk melihat hasil dari aktivasi modul tersebut kita bisa menggunakan perintah :

```
# lsmod | grep ipv6
ipv6                  141952 -1
```

Jika muncul baris diatas maka proses memuat modul IPv6 telah berhasil. Sebagai catatan bahwa proses unload modul belum didukung dan dapat mengakibatkan kernel crash.

Dengan menggunakan perintah insmod tadi maka semua aplikasi dan perangkat lunak yang mendukung IPv6 akan diaktifkan.

Untuk memuat modul IPv6 secara otomatis sesuai dengan permintaan, kita hanya perlu menambahkan baris dibawah ini pada file konfigurasi *kernel module loader* (biasanya file **/etc/modules.conf** atau **/etc/conf.modules**).

alias net-pf-10 ipv6

Selain itu juga dimungkinkan untuk menon-aktifkan proses load modul IPv6 secara otomatis, dengan cara menambahkan baris berikut pada file yang sama :

alias net-pf-10 off

Kompilasi Kernel dengan Dukungan IPv6

Jika dari kedua langkah diatas diperoleh hasil yang negatif maka kernel anda tidak memiliki dukungan IPv6. Ada beberapa cara yang dapat anda lakukan :

- Update distribusi anda dengan salah satu distribusi yang telah mendukung IPv6
- Kompilasi kernel *vanilla* baru (mudah, jika kita tahu pilihan yang dibutuhkan), tetapi sudah jarang digunakan
- Kompilasi ulang source kernel anda yang diberikan oleh distribusi
- Kompilasi kernel anda dengan *USAGI extension*

Pilih salah satu cara diatas dan lakukan kompilasi Kernel, dan pastikan tidak ada error yang terjadi saat kompilasi.

4. Konfigurasi Kernel IPv6 Pada Linux

Untuk mengetahui konfigurasi kernel dari IPv6, dan apa perbedaannya dengan konfigurasi kernel pada IPv4 maka kita dapat melakukannya dengan cara mengakses filesystem **/proc,** dimana semua nilai setting sistem yang sedang berjalan dicatat pada filesistem ini. Sebelum kita melihat filesystem tersebut ada dua persyaratan yang harus dipenuhi yaitu :

- Filesystem **proc** telah terkandung/terkompilasi dalam kernel linux
- Filesystem **proc** telah dimounting terlebih dahulu

Untuk mengeceknya kita dapat menggunakan perintah sebagai berikut :

```
# mount | grep "type proc"
none on /proc type proc (rw)
```

4.1. Perintah “sysctl”

Selain perintah diatas kita, untuk melihat dan mengubah nilai setting kernel IPv6 pada filesystem **proc** kita bisa menggunakan perintah “sysctl”. Misalnya apabila kita ingin melihat nilai setting forwarding digunakan perintah sebagai berikut :

```
# sysctl net.ipv6.conf.all.forwarding
net.ipv6.conf.all.forwarding = 0
```

```
# sysctl –w net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.all.forwarding = 1
```

Ada beberapa format nilai yang ada filesystem **proc,** yaitu :

- BOOLEAN : bernilai “1” atau “0”
- INTEGER : bernilai bilangan integer

4.2. Beberapa entry pada direktori /proc/sys/net/ipv6/

1. conf/default/*

   Untuk mengubah antarmuka tertentu ke setting asal

2. conf/all/*

   Untuk mengubah seluruh antarmuka ke setting tertentu

   Misalnya : **conf/all/forwarding** untuk mengaktifkan paket forwarding IPv6 bagi semua antarmuka

3. conf/interface/*

   Untuk mengubah setting antarmuka tertentu ke setting tertentu. Fungsi dari masing-masing setting berbeda-beda, tergantung dari forwarding diijinkan atau tidak. Misalnya : **accept_ra**

untuk menerima *router advertisement* berguna untuk proses konfigurasi alamat secara otomatis**, autoconf** untuk mengaktifkan alamat *link-local* berdasarkan alamat MAC-nya, dan sebagainya.

4. neigh/default/*

   Untuk mengubah setting asal untuk pendeteksian tetangga dan beberapa interval atau treshold yang khusus.

   Misalnya : **gc_tresh1** (bernilai 128) dan **gc_tresh2** (bernilai 512) untuk menentukan besar ukuran table *neighbour*, dan sebagainya.

5. neigh/interface/*

   Untuk mengubah setting tertentu untuk setiap antarmuka yang berguna untuk deteksi tetangga.

   Misalnya : **anycast_delay, proxy_qlen, app_solit**, dan sebagainya.

6. route/*

   Untuk mengubah setting umum untuk keperluan ruting.

   Misalnya : **flush, gc_interval, mtu_expires**

4.3. Entry IPv6 yang berhubungan dengan /proc/sys/net/ipv4

Sampai saat ini ada beberapa setting pada IPv4 yang digunakan IPv6, hal ini akan terus berlanjut hingga dihasilkan kernel modul IPv6 yang benar-benar terpisah dari modul IPv4. Misalnya entry-entry seperti dibawah ini :

1. **ip_***, misalnya **ip_default_ttl, ip_autoconfig**

   Entry ini berguna untuk merubah nilai setting dari IP yang bersifat dasar, misalnya **ip_default_ttl** untuk merubah nilai default dari TTL (time to live). Karena TTL juga masih ada dalam IPv6 maka entry ini juga dipakai oleh IPv6.

2. **tcp_***, misalnya **tcp_sack, tcp_rmem, tcp_syn_retries**

   Entry ini berguna untuk merubah nilai setting dari TCP yang bersifat dasar, misalnya **tcp_rmem** untuk merubah nilai receive memori dari TCP. Karena IPv6 masih menggunakan TCP yang sama dengan IPv4 maka entry **tcp_rmem** masih dipakai.

3. **icmp_*,** misalnya **icmp_echo_ignore_broadcast, icmp_timeexeced**

   Entry ini berguna untuk merubah nilai setting dari ICMP yang bersifat dasar, misalnya **icmp_echo_ignore_broadcast** untuk merubah setting sistem untuk

menerima ICMP broadcast atau tidak. Setting ini tidak dipakai pada IPv6 karena IPv6 sudah memiliki ICMPv6 dengan berbagai macam kelebihan.

4.4 Entry IPv6 pada /proc/sys

Pada direktori /proc/sys ada beberapa setting khusus untuk IPv6. Karena biasanya pada direktori ini berisi entry yang bersifat read-only maka kita hanya bisa melihat dengan menggunakan perintah “cat”, tidak bisa dengan “echo” maupun “sysctl”. Setting tersebut diantaranya :

if_inet6

Entry ini berisi seluruh setting alamat IPv6 setiap antarmuka yang ditunjukkan dalam format yang khusus, sebagai contoh alamat *loopback* akan memiliki nilai setting seperti berikut :

```
# cat /proc/net/if_inet6
00000000000000000000000000000001 01 80 10 80 lo
+--------------------------------+ ++ ++ ++ ++ ++
|                                  |  |  |  |  |
1                                  2  3  4  5  6
```

ipv6_route

Entry ini berisi seluruh setting ruting IPv6 yang ditunjukan dengan format yang khusus, misalnya untuk antarmuka loopback akan mempunyai format seperti berikut :

```
# cat /proc/net/ipv6_route
00000000000000000000000000000000 00 00000000000000000000000000000000 00
+--------------------------------+ ++ +--------------------------------+ ++
|                                  |  |                                  |
1                                  2  3                                  4
¬ 00000000000000000000000000000000 ffffffff 00000001 00000001 00200200 lo
¬ +--------------------------------+ +------+ +------+ +------+ +------+ ++
¬ |                                  |        |        |        |        |
¬ 5                                  6        7        8        9        10



# cat /proc/net/sockstat6
TCP6: inuse 7
UDP6: inuse 2
RAW6: inuse 1
FRAG6: inuse 0 memory 0
```

berisi stratistik tentang statistik IPv6, sebagai contoh :

```
[root@pinguin root]# cat /proc/net/tcp6
  sl   local_address                                         remote_address
st tx_queue rx_queue tr tm->when retrnsmt   uid  timeout inode
   4:                                 00000000000000000000000000000000:0016
00000000000000000000000000000000:0000   0A   00000000:00000000   00:00000000
00000000     0        0 1153 1 c7cd7a40 300 0 0 2 -1
   6:                                 0000000000000000FFFF000067C80E0A:0016
0000000000000000FFFF00005FC80E0A:07B2   01   00000030:00000000   01:00000026
00000000     0        0 1880 4 c209a5a0 39 4 7 2 -1
```

snmp6

Entry ini berisi informasi data MIB IPv6, sebagai contoh :

```
[root@pinguin root]# cat /proc/net/snmp6
Ip6InReceives                           72
Ip6InHdrErrors                          0
Ip6InTooBigErrors                       0
Ip6InNoRoutes                           0
Ip6InAddrErrors                         0
Ip6InUnknownProtos                      0
Ip6InTruncatedPkts                      0
Ip6InDiscards                           0
Ip6InDelivers                           0
Ip6OutForwDatagrams                     0
Ip6OutRequests                          0
Ip6OutDiscards                          0
Ip6OutNoRoutes                          0
Ip6ReasmTimeout                         0
. . . . . .
```

rt6_stats

```
[root@pinguin root]# cat /proc/net/rt6_stats
0000 000f 0000 0013 0000 0013
```

Selain itu masih ada beberapa entry diantaranya **udp6, igmp6, raw6, ip6_flowlable,** dan **ip6_tables_names**

5. Antarmuka Jaringan Pendukung IPv6

Tidak semua perangkat jaringan yang ada sekarang ini mendukung transportasi paket IPv6. Hal ini disebabkan karena struktur lapisan jaringan pada implementasi paket IPv6 pada kernel menghasilkan IP header yang belum jelas (IPv6 dan IPv4). IP header ini akan dicocokan dengan nomor protokol pada lapisan kedua TCP/IP yaitu protokol tranport. Jika protokol transport tidak

menggunakan nomor protokol seperti pada IP header maka protokol tansport tidak mempunyai kemampuan untuk mengirimkan paket IPv6.

5.1 Pengecekan perangkat konfigurasi IPv6

Setelah mengecek dukungan kernel, kemudian kita melakukan pengecekan perangkat lunak untuk konfigurasi IPv6 pada Linux. Ada beberapa perangkat yang digunakan yaitu :

Paket Net-tools

Paket ini berisi perangkat seperti perintah "ifconfig" dan "route", yang berguna untuk melakukan konfigurasi IPv6 pada antarmuka anda. Untuk mengecek perintah "ifconfig" gunakan perintah sebagai berikut :

```
# /sbin/ifconfig | grep -qw 'inet6' && echo “Utilitas
ifconfig sudah mendukung IPv6”
Utilitas ifconfig sudah mendukung IPv6
```

Sama juga untuk perintah "route", yaitu :

```
# /sbin/route | grep -qw 'inet6' && echo “Utilitas route
sudah mendukung IPv6”
Utilitas route sudah mendukung IPv6
```

Jika terdapat option yang menunjukkan tentang pengguanaan protokol IPv6 maka perangkat kita ini sudah mendukung IPv6.

Paket iproute

Pada sistem operasi linux ada perangkat yang berguna untuk mengkonfigurasi jaringan melalui *netlink device*. Perangkat ini memiliki kemampuan lebih dibandingkan dengan net-tools. Perangkat tersebut disebut dengan paket iproute. Untuk melakukan pengecekan apakah paket ini telah mendukung IPv6, yaitu dengan perintah :

```
# /sbin/ip |grep -qw 'inet6' && echo "Perintah ip sudah
mendukung IPv6"
```

ini tidak ada (/sbin/ip), maka kita harus melakukan installasi terlebih dahulu, baik dari CD distribusi linux, maupun download dari internet.

5.2 Pengecekan Dukungan program test/debug

Setelah sistem kita mendukung IPv6, maka kita harus memastikan bahwa sistem kita dapat menggunakan IPv6 untuk berkomunikasi melalui jaringan. Untuk melakukan pengamatan sebaiknya kita gunakan program seperti sniffer untuk mengamati paket IPv6.

Program Ping

Program ini biasanya termasuk dalam paket ***iputils***, yang didesain untuk melakukan pengetesan koneksi dengan cara mengirimkan paket *ICMPv6 echo-request* dan menunggu balasan paket *ICMPv6 echo-reply*. Hal ini dapat dilakukan dengan menggunakan perintah :

```
# ping6 <ipv6address>
# ping6 [-I <device>] <link-local-ipv6address>
```

```
# ping6 -c 3 ::1
PING ::1(::1) from ::1 : 56 data bytes
64 bytes from ::1: icmp_seq=0 hops=64 time=292 usec
64 bytes from ::1: icmp_seq=0 hops=64 time=292 usec
64 bytes from ::1: icmp_seq=0 hops=64 time=292 usec
--- ::1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.292/0.292/0.292/0.000 ms

# ping6 –c 1 –I eth0 fe80::250:daff:fe7d:fe11
PING      fe80::250:daff:fe7d:fe11(fe80::250:daff:fe7d:fe11)
from ::1 eth0: 56 data bytes
Warning: time of day goes back, taking countermeasures.
64 bytes from fe80::250:daff:fe7d:fe11: icmp_seq=0 hops=64
round-trip min/avg/max/mdev = 0.292/0.292/0.292/0.000 ms
time=292 usec
---  fe80::250:daff:fe7d:fe11 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
```

Program ini juga termasuk dalam paket ***iputils***. Program ini mirip dengan program "traceroute" pada IPv4. Contoh penggunaan tersebut adalah :

```
# traceroute6 ::1
traceroute to ::1 (::1) from ::1, 1 hops max, 16 byte
packets
```

tracepath6

Program ini juga masuk dalam paket ***iputils***. Program ini mirip dengan program "tracepath" pada IPv4 yang digunakan untuk menjelajahi jalur ke tujuan untuk menemukan MTU sepanjang jalur ini. Program dapat dijalankan dengan menggunakan perintah :

```
(root@pinguin /root) # tracepath6 ::1
1?:    [LOCALHOST]                pmtu 16436
1:     ::1                        0.210 ms reached
Resume: pmtu 16436 hops 1 back 1
```

tcpdump IPv6

Pada sistem operasi linux tcpdump merupakan perangkat utama untuk menangkap paket. Paket tcpdump versi 3.6 telah mendukung IPv6, sehingga dapat mengamati paket IPv6 yang melewati jaringan, yang meliputi icmp6 (trafik ICMPv6), ip6 (native trafik IPv6), proto ipv6 (trafik tunneling IPv6 melalui IPv4), dan sebagainya. Untuk melakukan penangkapan paket tersebut dapat digunakan perintah sebagai berikut :

```
# tcpdump -t -n -i eth0 -s 512 -vv ip6 or proto ipv6
tcpdump: listening on eth0
3ffe:ffff:100:f101:2e0:18ff:fe90:9205 >
3ffe:ffff:100:f101::1: icmp6: echo request (len 64, hlim 64)
3ffe:ffff:100:f101::1 >
3ffe:ffff:100:f101:2e0:18ff:fe90:9205: icmp6: echo reply
(len 64, hlim 64)
```

Jika baik kernel maupun perangkat pendukung jaringannya sudah mendukung IPv6 maka kita dapat memulai konfigurasi alamat jaringan IPv6.

6. Konfigurasi Alamat Ipv6

6.1. Menampilkan Alamat IPv6

Sebelum kita mencoba untuk melakukan konfigurasi alamat IPv6 pada antarmuka jaringan kita, maka sebaiknya kita cek terlebih dahulu, apakah alamat IPv6 sudah terkonfigurasi atau belum (karena bisa saja terkonfigurasi dengan mekanisme *stateless auto-configuration*).

Untuk melihat alamat IPv6 tersebut dapat menggunakan dua buah perintah yaitu :

1. Perintah “ip”

   Dengan syntax :

```
# /sbin/ip –6 addr show dev <interface>
```

   antarmuka ini menunjukkan antarmuka jaringan yang kita miliki. Sebagai contoh :

```
# /sbin/ip –6 addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc
pfifo_fast qlen 100
inet6 fe80::210:a4ff:fee3:9566/10 scope link
```

2. Perintah “ifconfig”

   Dengan syntax :

```
# /sbin/ifconfig <interface>
```

   Sebagai contoh :

```
# /sbin/ifconfig eth0 |grep "inet6 addr:"
inet6 addr: fe80::210:a4ff:fee3:9566/10 Scope:Link
```

6.2. Konfigurasi Alamat IPv6

Setelah IPv6 berhasil diimplementasikan pada sistem operasi Linux , selanjutnya akan dicoba untuk memberikan alamat IPv6 untuk antarmuka sistem kita. Ada dua metode untuk mengkonfigurasi alamat IPv6 yaitu konfigurasi manual dan konfigurasi otomatis. Pada bagian ini alamat IPv6 akan diberikan antarmuka jaringan secara manual dengan

cara memasukkan alamat yang kita kehendaki. Seperti perintah untuk melihat alamat IPv6, perintah untuk menambahkan alamat IPv6 juga dapat dilakukan dengan menggunakan perintah “ip” dan “ifconfig”.

Untuk perintah “ip” menggunakan syntax sebagai berikut :

```
# /sbin/ip -6 addr add <ipv6address>/<prefixlength> dev
<interface>
```

misalnya :

```
# /sbin/ip -6 addr add 3ffe:ffff:0:f100::1/64 dev eth0
```

Sedangkan untuk perintah “ifconfig” menggunakan syntax sebagai beikut :

```
# /sbin/ifconfig <interface> add <ipv6address>/
<prefixlength>
```

Kemudian itu untuk mengecek antarmuka jaringan kita gunakan program ping6.

```
# ping6 3ffe:ffff:0:f100::1
```

6.3 Menghapus Alamat IPv6

Apabila alamat IPv6 tidak digunakan lagi, kita bisa menghapus alamat IPv6 tersebut dari antarmuka jaringan kita. Untuk melakukan penghapusan tersebut kita bisa menggunakan perintah “ip” dan “ifconfig”.

Untuk perintah “ifconfig” memiliki syntax sebagai berikut :

```
# /sbin/ip -6 addr del <ipv6address>/<prefixlength> dev
<interface>
```

```
# /sbin/ip -6 addr del 3ffe:ffff:0:f101::1/64 dev eth0
```

Sedangkan untuk perintah “ifconfig” memiliki syntax seprti berikut :

```
# /sbin/ifconfig del <ipv6address>/<prefixlength>
```

misalnya :

```
# /sbin/ifconfig eth0 inet6 del 3ffe:ffff:0:f101::1/64
```

Untuk melihat berhasil atau tidaknya kita melakukan penghapusan alamat tersebut kita bisa lihat kembali alamat pada antarmuka kita dengan perintah “ifconfig”.

7. Konfigurasi ruting IPv6

Selain dapat menampilkan alamat IPv6 kita juga bisa melihat tabel ruting IPv6 untuk mengetahui jalur yang akan ditempuh oleh paket untuk sampai ke tujuan. Untuk melakukannya kita bisa menggunakan dua perintah yaitu “ip” dan “route”. Untuk perintah “ip” memiliki syntaks sebagai berikut :

```
# /sbin/ip -6 route show [dev <device>]
```

misalnya :

```
# /sbin/ip -6 route show dev eth0
3ffe:ffff:0:f100::/64 proto kernel metric 256 mtu 1500 advmss 1440
fe80::/10             proto kernel metric 256 mtu 1500 advmss 1440
ff00::/8              proto kernel metric 256 mtu 1500 advmss 1440
default               proto kernel metric 256 mtu 1500 advmss 1440
```

kemudian untuk menambahkan tabel ruting melalui sebuah gateway dapat dilakukan perintah :

```
# /sbin/ip -6 route add <ipv6network/prefixlenght> via
  <gateway>
# /sbin/ip -6 route add <ipv6network/prefixlenght> dev
  <device> metric 1
```

```
# /sbin/ip -6 route add 2000::/3 via 3ffe:ffff:0:f101::1
# /sbin/ip -6 route add 2000::/3 dev eth0 metric 1
```

kemudian untuk meghapus tabel ruting tersebut dapat digunakan perintah :

```
# /sbin/ip -6 route del 2000::/3 dev eth0
```

Kesimpulan

Linux merupakan salah satu operating system yang telah mendukung IPv6.Untuk mengetahui linux mendukung IPv6 apa belum,kita perlu chek kernel kita dan perlu install modul IPv6 untuk aktivasi modul IPv6 pada kernel linux kita.Pada makalah ini telah dijelaskan cara implementasi IPv6 pada operating system Linux RedHat 8.0 dan hanya dijelaskan cara konfigurasi dasar IPv6 saja.Mungkin pada makalah selanjutnya akan dijelaskan untuk aplikasi – aplikasi Linux IPv6 server dan implementasi mekanisme transisi IPv6 pada linux.Maturnuwun!

Referensi

- Implementing IPV6. Supporting the Next Generation Internet Protocols by P. E. Miller, Mark A.Miller; Publisher: John Wiley & Sons; Maret 2000.
- Big Book of Ipv6 Addressing Rfcs by Peter H. Salus (Compiler), Morgan Kaufmann Publishers, April 2000.
- Understanding IPV6 by Davies, Joseph; ISBN 0735612455; Date Published 05/01/2001; Number of Pages: 350. Understanding IPV6 by Davies, Joseph;Date Published 13/11/2002.
- Linux IPv6 How To
- Aris Cahyadi , “Analisa dan Implementasi IPv6 Tunnel Broker untuk interkoneksi IPv4 dan IPv6 ”,Stt telkom,2002

Biografi Penulis

Wahidi Somad. Lahir di Klaten, 2 Mei 1981. Menamatkan SMU di SMU Muhammadiyah 1 Klaten pada tahun 1999. Sekarang sedang menyelesaikan program S1 pada jurusan Teknik Elektro Telekomunikasi di STTTelkom bandung.Saat ini aktif sebagai asissten pada Computer and Communication Laboratory (C&C Lab) STTTelkom Bandung dan Asisten Dsn Jaringan Komunikasi Data untuk D3 Teknik Elektro STTTelkom bandung. Kompetensi inti adalah pada bidang Komunikasi data,Networking,Security Network dan Web Engineering.Sekarang sedang aktif mengerjakan riset tentang mekanisme transisi IPv6 (NAT-PT,Teredo,DSTM,Tunnell Broker) dan Security network (LAN,WLAN,internet) bersama senior member Lab Computer And Communication STT Telkom bandung.Pernah menjadi system administrator di salah satu ISP WaveLAN di bandung,dan sekarang menjadi system administrator server pinguin.stttelkom.ac.id.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:
URL : *http://wah.id.gg*
Email : *ste99120@stttelkom.ac.id*