# BAB II

# DASAR TEORI

## 2.1 Microsoft TCP/IP

Protokol adalah himpunan aturan yang memungkinkan komputer untuk saling berhubungan antara yang satu dengan yang lain, biasanya berupa bentuk / waktu / barisan / pemeriksaan *error* saat transmisi data (Craig Hunt, 1998).

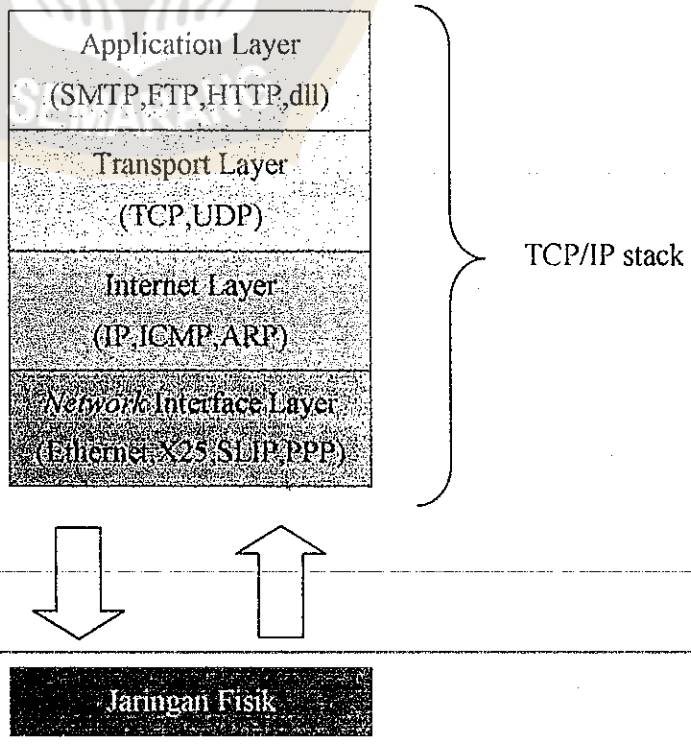
Dalam dunia komunikasi data, protokol mengatur bagaimana sebuah komputer berkomunikasi dengan komputer lain. Dalam sebuah jaringan komputer dapat menggunakan banyak macam protokol, tetapi agar dua buah komputer dapat berkomunikasi, keduanya perlu menggunakan dua protokol yang sama. Protokol ini berfungsi mirip dengan bahasa. Sama halnya dengan manusia, agar dapat berkomunikasi orang-orang perlu menggunakan bahasa yang sama.

TCP/IP *(Transmission Control Protocol / Internet Protocol)* adalah sekelompok protokol yang mengatur komunikasi data antar komputer baik dalam sebuah jaringan komputer lokal maupun internet. Komputer-komputer yang terhubung dalam sebuah jaringan komputer saling berkomunikasi dengan menggunakan protokol ini. Salah satu keistimewaan protokol ini adalah bisa digunakan pada semua jenis komputer maupun sistem operasi. (Onno W. Purbo, 2001)

### 2.1.1 Arsitektur TCP/IP

TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data dalam jaringan komputer. TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data, sehingga tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain selama ia masih bisa mengirim dan menerima data. Arsitektur rangkaian protokol TCP/IP mendifinisikan berbagai cara agar TCP/IP dapat saling menyesuaikan. Protokol ini dapat diterapkan dengan mudah di semua jenis komputer dan interface jaringan.

Sekumpulan protokol TCP/IP ini dimodelkan dengan empat layer TCP/IP sebagaimana terlihat pada gambar 2.1. (Onno W. Purbo, 2001)

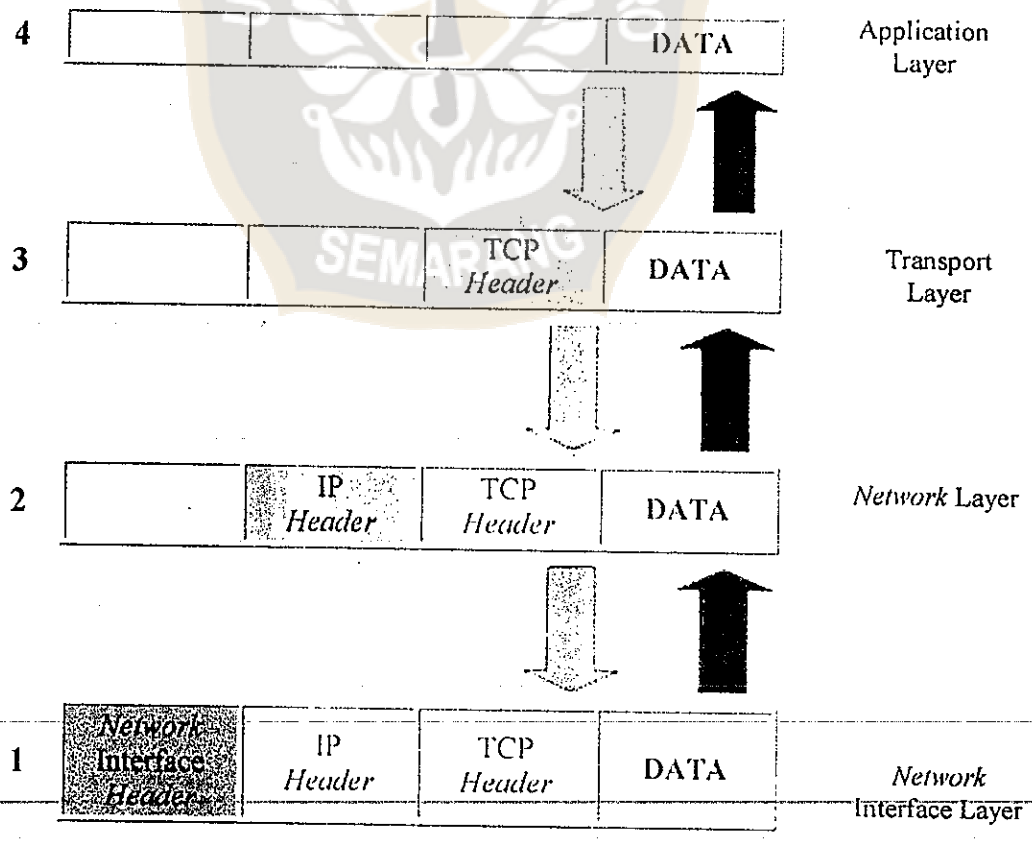



Gambar 2.1 Layer TCP/IP

TCP/IP terdiri atas empat layer/lapis kumpulan protokol yang bertingkat. Keempat layer tersebut adalah :

a. *Network* Interface Layer
b. Internet Layer
c. Transport Layer
d. Application Layer

Dalam TCP/IP terjadi penyampaian data dari protokol yang berada di satu layer ke protokol yang berada pada layer yang lain. Setiap protokol memperlakukan semua informasi yang diterimanya dari protokol lain sebagai data, sebagaimana yang terlihat pada gambar 2.2.




Gambar 2.2 Pergerakan data dalam layer TCP/IP

Jika suatu protokol menerima data dari protokol lain di layer atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut. Informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu data diteruskan lagi ke protokol pada layer di bawahnya. Demikian pula sebaliknya, jika suatu protokol menerima data dari protokol lain yang berada pada layer di bawahnya.

Layer terbawah, yaitu *Network Interface Layer* bertanggung jawab mengirim dan menerima data ke dan dari media fisik. Media fisik disini dapat berupa kabel, serat optik atau gelombang radio. Karena tugasnya ini, protokol pada layer ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer yang berasl dari peralatan lain yang sejenis.

Lapisan/layer protokol berikutnya adalah *Internet Layer*. Protokol yang berada pada layer ini bertanggung jawab dalam proses pengiriman paket data ke alamat yang tepat. Pada layer ini terdapat tiga macam protokol, yaitu IP, ARP dan ICMP.

a. IP *(Internet Protocol)* berfungsi untuk menyampaikan paket data ke alamat yang tepat.
b. ARP *(Address Resolution Protocol)* ialah protokol yang digunakan untuk menemukan alamat hardware dari *host*/komputer yang terletak dalam jaringan yang sama.
c. ICMP *(Internet Control Message Control Protocol)* adalah protokol yang berfungsi untuk megirimkan pesan dan melaporkan kegagalan pengiriman data.

Layer berikutnya, *Transport Layer* berisi protokol yang bertanggung jawab untuk mengadakan komunikasi antara dua buah *host*/komputer. Kedua protokol pada lapisan ini adalah TCP *(Transmission Control Protocol)* dan UDP *(User Datagram Protocol)*.

Layer teratas, ialah *Application Layer*. Pada layer inilah terletak semua aplikasi yang menggunakan protokol TCP/IP ini.

Karena TCP/IP merupakan salah satu lapisan protokol OSI *(Open System Interconnections)*, berarti bahwa hierarki TCP/IP merujuk kepada 7 lapisan OSI tersebut. Berikut adalah model referensi OSI 7 lapisan, yang mana setiap lapisan menyediakan tipe khusus pelayanan jaringan : (Heywood, 2001)

1. *Application layer*
2. *Presentation layer*
3. *Session layer*
4. *Transport layer*
5. *Network layer*
6. *Data link layer*
7. *Physical layer*

Tiga lapisan teratas biasa dikenal sebagai "*upper level protocol*" sedangkan empat lapisan terbawah dikenal sebagai *"lower level protocol"*. Tiap lapisan berdiri sendiri, tetapi fungsi dari masing-masing lapisan bergantung dari keberhasilan operasi layer sebelumnya. Sebuah lapisan pengirim hanya perlu

berhubungan dengan lapisan yang sama di penerima (jadi misalnya lapisan data link penerima hanya berhubungan dengan *data link* pengirim) selain dengan satu layer di atas atau dibawahnya (misalnya lapisan *network* berhubungan dengan lapisan transport diatasnya atau dengan lapisan data link dibawahnya).

Model dengan menggunakan lapisan ini merupakan sebuah konsep yang penting karena suatu fungsi yang rumit yang berkaitan dengan komunikasi dapat dipecahkan menjadi sejumlah unit yang lebih kecil. Tiap lapisan bertugas memberikan layanan tertentu pada lapisan diatasnya dan juga melindungi lapisan diatasnya dari rincian cara pemberian layanan tersebut. Tiap lapisan harus transparan sehingga modifikasi yang dilakukan atasnya tidak akan menyebabkan perubahan pada lapisan yang lain. Lapisan menjalankan perannya dalam pengalihan data dengan mengikuti peraturan yang berlaku untuknya dan hanya berkomunikasi dengan lapisan yang setingkat.

Akibatnya sebuah layer pada satu sistem tertentu hanya akan berhubungan dengan lapisan yang sama dari sistem yang lain. Proses ini dikenal sebagai *"peer process"*.

### 2.1.2 Protokol TCP/IP

Berikut ini akan dibahas mengenai cara kerja masing-masing layer dalam protokol TCP/IP :

1. *Network Interface Layer*

   Layer ini bertanggung jawab mengirim data dan menerima data dari media fisik. Lapisan ini hanya menggambarkan bagaimana data dikodekan menjadi sinyal-sinyal dan karakteristik antarmuka tambahan media.

2. *Internet layer / network layer*

   a. *IP (Internet Protocol)*

      Protokol IP merupakan inti dari protokol TCP/IP. Seluruh data yang akan dikirimkan, akan diolah terlebih dahulu oleh protokol IP, dan dipancarkan sebagai paket IP, agar sampai ke tujuan. Untuk mengirimkan pesan pada suatu *internetwork* (suatu jaringan yang mengandung beberapa segmen jaringan), tiap jaringan harus secara unik diidentifikasi oleh alamat jaringan. Dalam melakukan pengiriman data, IP memiliki sifat yang dikenal sebagai *unreliable connectionless* dan *datagram selivery sevice.*

      *Unreliable*/ketidakandalan berarti bahwa protokol IP tidak menjamin datagram yang dikirim pasti sampai ke tempat tujuan, tetapi akan melakukan usaha sebaik-baiknya *(best effort delivery service)*, agar paket yang dikirim tersebut sampai ke tujuan.

      Jika dalam perjalanan pengiriman paket tersebut terjadi hal-hal yang tidak diinginkan (salah satu jalur putus, router mengalami kongesti/macet, atau

*host/network* tujuan sedang *down*), protokol IP hanya memberitahukan ke pengirim paket (melalui protokol ICMP), bahwa terjadi masalah dalam pengiriman paket IP ke tempat tujuan.

*Connectionless* berarti dalam mengirim paket data dari tempat asal ke komputer tujuan, pihak pengirim dan penerima paket IP sama sekali tidak mengadakan perjanjian *(handsake)* terlebih dahulu.

*Datagram delivery service* berarti setiap paket data yang dikirim adalah independen terhadap paket data yang lain. Akibatnya jalur yang ditempuh oleh masing-masing paket data IP ke tujuannya bisa jadi berbeda satu dengan yang lainnya. Karena jalur yang ditempuh berbeda, kedatangan peketpun bisa jadi tidak berurutan.

b. *ICMP (Internet Control Message Protocol)*

ICMP adalah protokol yang bertugas mengirimkan pesan-pesan kesalahan dan kondisi lain yang memerlukan perhatian khusus. Pesan/paket ICMP dikirim jika terjadi masalah pada layer IP dan layer atasnya (TCP/UDP).

Pada kondisi normal, protokol IP berjalan baik dan menghasilkan proses penggunaan memori serta sumber daya transmisi yang efisien. Namun ada beberapa kondisi dimana koneksi IP terganggu, misalnya karena router yang crash, putusnya kabel, atau matinya *host* tujuan. Pada saat inilah ICMP berperan membantu menstabilkan kondisi jaringan. Hal ini dilakukan dengan cara mamberikan pesan-pesan tertentu sebagai respon atas kondisi tertentu yang terjadi pada jaringan tersebut.

c. *ARP (Address Resolution Protocol)*

Dalam jaringan lokal, paket IP biasanya dikirim melalui *ethernet card*. Untuk dapat mengenali dan dapat berkomunikasi dengan *ethernet card* lainnya, digunakan *ethernet address*. *Ethernet address* ini besarnya 48 bit. Setiap *ethernet card* memiliki *ethernet address* yang berbeda-beda.

Pada saat hendak melakukan pengiriman data ke komputer tertentu dengan IP tertentu, suatu *host* pada jaringan ethernet perlu mengetahui, di atas *ethernet address* manakah tempat IP tersebut terletak. Untuk keperluan pemetaan *IP address* dengan *ethernet address* ini, digunakan protokol ARP.

4. *Transport layer /host to host*

*Transport layer* merupakan layer komunikasi data yang mengatur aliran data antara dua *host* untuk keperluan aplikasi di atasnya, yaitu dengan membagi pesan-pesan menjadi fragmen-fragmen yang cocok dengan pembatasan ukuran yang dibentuk oleh jaringan. Ada dua buah protokol pada layer ini, yaitu TCP dan UDP.

a. *TCP (Transmission Control Protocol)*

TCP merupakan protokol yang terletak pada layer transport. Protokol ini menyediakan service yang dikenal sebagai *connection oriented, reliable, byte stream service.*

*Connection oriented* berarti sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukan hubungan

(*handsake*) terlebih dulu. *Reliable* berarti TCP menerapkan proses deteksi kesalahan paket dan retransmisi. *Byte Stream Service* berarti paket dikirimkan dan sampai ke tujuan secara berurutan.

Protokol ini berusaha secara seksama untuk mengirimkan data ke tujuan, memeriksa kesalahan, mengirimkan data ulang bila diperlukan dan mengirimkan *error* ke lapisan atas hanya bila TCP tidak berhasil mengadakan komunikasi. Tetapi kehandalan TCP tercapai dengan mengorbankan *bandwidth* jaringan yang besar.

Untuk menjaga agar reliabilitas pengiriman data terjamin, TCP melakukan hal-hal berikut :

1. Data yang diterima oleh aplikasi dipecah menjadi segmen-segmen yang besarnya menurut TCP paling sesuai untuk mengirimkan data.
2. Ketika TCP menerima data dari mitranya, TCP mengirimkan *acknowledgement* (pemberitahuan bahwa ia telah menerima data).
3. Ketika TCP mengirimkan sebuah data, TCP mengaktifkan pewaktu (*software timer*) yang akan menunggu *acknowledgement* dari penerima segmen data tersebut. Jika sampai pada waktu yang ditentukan tidak diterima *acknowledgement*, data tersebut dikirimkan kembali oleh TCP.
4. Sebelum segmen data terebut dikirim, TCP melakukan perhitungan *checksum* pada *header* dan datanya. Hal ini berbeda dengan protokol IP yang hanya melakukan perhitungan *checksum* pada *header*nya saja. Jika segmen yang diterima memiliki *checksum* yang tidak valid, TCP

akan membuang segmen ini dan berharap sisi pengirim akan melakukan retransmisi.

5. Karena segmen TCP dikirim menggunakan IP, dan datagram IP dapat sampai ke tujuan dalam keadaan tidak berurutan, segmen TCP yang dikirimnyapun dapat mengalami hal yang sama. Karenanya sisi penerima paket TCP harus mampu melakukan pengurutan kembali segmen TCP yang ia terima *(resequencing)* dan memberikan data dengan urutan yang benar ke aplikasi penggunanya.
6. Karena paket IP dapat terduplikasi di perjalanan, penerima TCP harus membuang data tersebut.
7. Untuk mencegah agar server yang cepat tidak membanjiri server yang lambat, TCP melakukan proses *flow control*. Setiap koneksi TCP memiliki *buffer* dengan ukuran yang terbatas. Sisi penerima TCP hanya memperbolehkan sisi pengirim mengirimkan data sebesar *buffer* yang ia miliki.

b. *UDP (User Datagram Protocol)*

UDP merupakan protokol transport yang sederhana. Berbeda dengan TCP yang *connection oriented*, UDP bersifat *connectionless*. Dalam UDP tidak ada *sequencing* (pengurutan kembali) paket yang datang, *acknowledgement* terhadap paket yang datang, atau retransmisi, jika paket mengalami masalah di tengah jalan.

Kemiripan UDP dengan TCP ada pada penggunaan *port number*. Sebagaimana digunakan pada TCP, UDP menggunakan port number ini membedakan pengiriman datagram ke beberapa aplikasi berbeda yang terletak pada komputer yang sama.

Karena sifatnya yang *connectionless* dan *unreliable*, UDP digunakan oleh aplikasi-aplikasi yang secara periodik melakukan aktivitas tertentu (misalnya *query routing table* pada jaringan lokal), serta hilangnya satu data akan dapat diatasi pada *query* periode berikutnya dan melakukan pengiriman data ke jaringan lokal. Pendeknya jarak tempuh datagram akan mengurangi resiko kerusakan data.

Bersifat *broadcasting* atau *multicasting*. Pengiriman datagram ke banyak klien sekaligus akan efisien jika prosesnya menggunakan metode *connectionless*.

UDP *(User Datagram Protocol)* disisi lain adalah protokol yang tidak handal. Protokol ini hanya "semampunya" saja mengirimkan data. UDP tidak akan berusaha untuk mengembalikan datagram yang hilang dan proses pada lapisan atas harus bertanggung jawab untuk mendeteksi data yang hilang atau rusak dan mengirimkan ulang data tersebut bila dibutuhkan.

Tanggung jawab lapisan transport yang paling berat dalam hal pengiriman pesan adalah mendeteksi kesalahan dalam pengiriman data tersebut. Ada

dua kategori umum deteksi kesalahan dapat dilakukan oleh lapisan transport :

a. *Reliable delivery*, berarti kesalahan tidak dapat terjadi, tetapi kesalahan akan dideteksi jika terjadi. Pemulihan kesalahan dilakukan dengan jalan memberitahukan lapisan atas bahwa kesalahan telah terjadi dan meminta pengiriman kembali paket yang kesalahannya terdeteksi.

b. *Unreliable delivery*, bukan berarti kesalahan mungkin terjadi, tetapi menunjukkan bahwa lapisan transport tidak memeriksa kesalahan tersebut. Karena pemeriksaan kesalahan memerlukan waktu dan mengurangi penampilan jaringan. Biasanya kategori ini digunakan jika setiap paket mengandung pesan yang lengkap, sedangkan *reliable delivery*, jika mengandung banyak paket. *Unreliable delivery*, sering disebut *"datagram delivery"* dan paket-paket bebas yang dikerimkan dengan cara ini sering disebut "datagram".

4. *Application layer*

Lapisan inilah biasa disebut lapisan akhir *(front end)* atau bisa disebut *user program*. Lapisan inilah yang menjadi alasan keberadaan lapisan sebelumnya. Lapisan sebelumnya hanya bertugas mengirimkan pesan yang ditujukan untuk lapisan ini. Di lapisan ini dapat ditemukan program yang menyediakan pelayanan jaringan, seperti *mail server* (email program), file transfer server (FTP program), *remote terminal*.

## *2.2 IP Address*

*IP address* adalah alamat unik yang harus dimiliki oleh setiap komputer yang terhubung ke dalam sebuah jaringan komputer. Tidak boleh ada dua komputer yang memiliki *IP address* yang sama. *IP address* ini nantinya dibutuhkan dalam pengiriman paket-paket data, karena dalam setiap paket data tersebut terdapat *header* yang berisi *IP address* dari tujuan paket-paket data tersebut. *IP address* ini diberikan ke jaringan dan peralatan jaringan yang menggunakan protokol TCP/IP. *IP address* terdiri atas 32-bit angka biner yang dapat dituliskan sebagai empat angka desimal yang dipisahkan oleh tanda titik seperti: 192.168.0.19 (Onno W. Purbo, 2001) .

### 2.2.1 Format *IP address*

#### 2.2.1.1 Bentuk Biner

*IP address* merupakan bilangan biner 32 bit yang dipisahkan oleh tanda pemisah berupa tanda titik setiap 8 bitnya. Bentuk *IP address* adalah sebagai berikut :

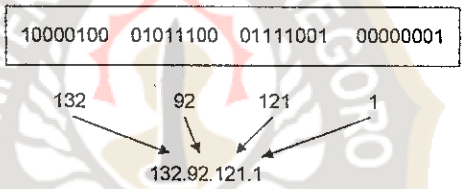
xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

Setiap simbol "x " dapat digantikan oleh angka 0 dan 1, misalnya sebagai berikut :

10000100.1011100.1111001.00000001

### 2.2.1.2 Bentuk *dotted decimal*

Notasi *IP address* dengan bilangan biner seperti di atas tidaklah mudah dibaca. Untuk membuatnya lebih mudah dibaca dan ditulis, *IP address* sering ditulis sebagai 4 bilangan desimal yang masing-masing dipisahkan oleh sebuah titik. Format penulisan seperti ini disebut *"dotted-decimal notation"* (notasi desimal bertitik). Setiap bilangan desimal tersebut merupakan nilai dari satu oktet (delapan bit) *IP address*. Gambar 2.3 memperlihatkan bagaimana sebuah *IP address* yang ditulis dengan notasi *dotted-decimal*.




*Gambar 2.3 Notasi Dotted Desimal*

### 2.2.2 Struktur *IP address*

*IP address* terdiri atas bilangan biner sepanjang 32 bit yang dibagi atas 4 segmen. Tiap segmen terdiri atas 8 bit yang memiliki nilai desimal dari 0 – 255 (255 adalah nilai terbesar dari bilangan biner 8 bit yang diperoleh dari = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 1 ). Karena *IP address* terdiri atas 4 buah bilangan 8 bit, maka jumlah *IP address* yang tersedia ialah 255 x 255 x 255 x 255. *Range address* yang bisa digunakan adalah dari 00000000. 00000000. 00000000. 00000000 sampai dengan 11111111. 11111111. 11111111. 11111111. Jadi ada sebanyak 232 kombinasi *IP address* yang bisa dipakai di seluruh dunia. Jaringan

TCP/IP dengan 32 *bit address* mampu menampung sebanyak 232 atau lebih dari 4 milyar *host*. Untuk mempermudah pembacaan dan penulisan, *IP address* biasanya direpresentasikan dalam bentuk desimal. *Range address* di atas dapat diubah menjadi address 0.0.0.0 sampai address 255.255.255.255. Nilai desimal dari *IP address* inilah yang dikenal dalam pemakaian sehari-hari.

*IP address* sebanyak ini harus dibagi-bagikan ke seluruh pengguna jaringan (internet) di seluruh dunia. Untuk mempermudah proses pembagiannya, *IP address* dukelompokkan ke dalam kelas-kelas. Dasar pertimbangan pembagian *IP address* ke dalam kelas-kelas adalah untuk memudahkan pendistribusian *IP address*.

*IP address* ini dikelompokkan ke dalam lima kelas, yaitu kelas A, kelas B, kelas C, kelas D dan kelas E (Onno W. Puro, 2001). Perbedaan pada tiap kelas tersebut adalah pada ukuran dan jumlahnya. IP kelas A dan B dipakai oleh sedikit jaringan, namun jaringan ini memiliki anggota yang besar. Kelas C dipakai oleh banyak jaringan, namun anggota masing-masing jaringan sedikit. Kelas D dan E juga didefinisikan, tetapi tidak digunakan dalam penggunaan normal. Kelas D diperuntukkan bagi jaringan *multicast* dan kelas E untuk keperluan eksperimental.

#### 2.2.2.1 *Network* ID dan *host* ID

Pembagian kelas-kelas *IP address* didasarkan pada dua hal, yaitu *network ID* dan *host ID* dari suatu *IP address*. Setiap *IP address* selalu merupakan sebuah pasangan dari *network ID* (identitas jaringan) dan *host id* (identitas *host* dalam jaringan tersebut). *Network ID* ialah bagian dari *IP address* yang digunakan untuk

menunjukkan jaringan tempat sebuah komputer berada. Sedangkan *host ID* ialah bagian dari *IP address* yang digunakan untuk menunjukkan workstation, server, router dan semua *host* TCP/IP lainnya dalam jaringan tersebut. Dalam satu jaringan, *host ID* ini harus unik (tidak boleh ada yang sama).

Untuk pengelompokan kelas-kelas *IP address* diberikan sebagai berikut:

**a. Kelas A**

Karakteristik :

| | |
|---|---|
| Format | : 0nnnnnnn hhhhhhhh hhhhhhhh hhhhhhhh |
| Bit pertama | : 0 |
| Panjang NetID | : 8 bit |
| Panjang *Host*ID | : 24 bit |
| Byte pertama | : 0 – 127 |
| Jumlah | : 126 Kelas A (0 dan 127 dicadangkan) |
| Range IP | : 1.xxx.xxx.xxx sampai 126.xxx.xxx.xxx |
| Jumlah IP | : 16.774.214 *IP address* tiap kelas A |

*IP address* kelas A diberikan untuk jaringan dengan jumlah *host* yang sangat besar. Bit pertama dari *IP address* kelas A selalu diset 0 (nol) sehingga byte terdepan dari *IP address* kelas A selalu bernilai antara angka 0 dan 127. Pada *IP address* kelas A, *network* ID ialah delapan bit pertama, sedangkan 24 bit berikutnya adalah *host* ID.

Contoh :

Diberikan alamat IP 113.46.5.6. Dengan demikian cara pembacaannya ialah sebagai berikut :

| | |
|---|---|
| *Network* ID | : 113 |
| *Host* ID | : 46.5.6 |

Sehingga *IP address* tersebut berarti *host* nomor 46.5.6 pada *network* nomor 113. Dengan panjang *host* ID yang 24 bit, *network* dengan *IP address* kelas A ini dapat menampung sekitar 16 juta *host*.

**b. Kelas B**

Karakteristik :

| | |
|---|---|
| Format | : 10nnnnnn nnnnnnnn hhhhhhhh hhhhhhhh |
| Bit pertama | : 10 |
| Panjang NetID | : 16 bit |
| Panjang *Host*ID | : 16 bit |
| Byte pertama | : 128 – 191 |
| Jumlah | : 16.384 Kelas B |
| Range IP | : 128.0.xxx.xxx sampai 191.155.xxx.xxx |
| Jumlah IP | : 65.532 *IP address* tiap kelas B |

*IP address* kelas B biasanya dialokasikan untuk jarinngan berukuran sedang dan besar. Dua bit pertama dari *IP address* kelas B selalu diset 10 (satu nol), sehingga byte terdepan dari *IP address* kelas B selalu bernilai antara 128 hingga 191. Pada

*IP address* kelas B, *network* ID adalah 16 bit pertama, sedangkan 16 bit berikutnya adalah *host* ID

Contoh :

Diberikan alamat IP 132.92.121.1. Dengan demikian cara pembacaannya ialah sebagai berikut :

| | |
|---|---|
| *Network* ID | : 132.92 |
| *Host* ID | : 121.1 |

Sehingga *IP address* di atas berarti *host* nomor 121.1 pada *network* nomor 132.92. Dengan panjang *host* ID yang 16 bit, *network* dengan *IP address* kelas B ini dapat menampung sekitar 65.000 *host*.

**c. Kelas C**

Karakteristik :

| | |
|---|---|
| Format | : 110nnnnn nnnnnnnn nnnnnnnn hhhhhhhh |
| Bit pertama | : 110 |
| Panjang NetID | : 24 bit |
| Panjang *Host*ID | : 8 bit |
| Byte pertama | : 192 – 223 |
| Jumlah | : 2.097.152 Kelas C |
| Range IP | : 192.0.0.xxx sampai 223.255.255.xxx |
| Jumlah IP | : 254 *IP address* tiap kelas C |

*IP address* kelas C pada awalnya digunakan untuk jaringan berukuran kecil (misalnya LAN). Tiga bit pertama dari *IP address* kelas C selalu berisi 111. Bersama 21 bit berikutnya, angka ini membentuk *network* ID 24 bit. *Host* ID ialah 8 bit terakhir. Dengan konfigurasi ini, bisa dibentuk sekitar dua juta *network* dengan masing-masing *network* memiliki 256 *IP address*.

Contoh :

Diberikan alamat IP 200.100.101.1. Dengan demikian cara pembacaannya ialah sebagai berikut :

*Network* ID : 200.100.101

*Host* ID : 1

**d. Kelas D**

Karakteristik :

Format : 1110mmmm mmmmmmmm mmmmmmmm mmmmmmmm

4 bit pertama : 1110

Bit multicast : 28 bit

Byte inisial : 224-247

Deskripsi : Kelas D adalah ruang alamat multicast (RFC 1112)

*IP address* kelas D digunakan untuk keperluan IP *multicasting*. 4 bit pertama *IP address* kelas D diset 1110. Bit-bit berikutnya diatur sesuai keperluan multicast group yang menggunakan *IP address* ini. Dalam *multicasting* tidak dikenal *network bit* dan *host bit*.

**e. Kelas E**

Karakteristik :

| | |
|---|---|
| Format | : 1111rrrr rrrrrrr rrrrrrr rrrrrrr |
| 4 bit pertama | : 1111 |
| Bit cadangan | : 28 bit |
| Byte inisial | : 248-255 |
| Deskripsi | : Kelas E adalah ruang alamat yang dicadangkan untuk keperluan eksperimental |

*IP address* kelas E tidak digunakan untuk umum. 4 bit pertama *IP address* ini diset 1111. Selain *network* ID, istilah lain yang digunakan untuk menyebut bagian *IP address* yang menunjukkan jaringan ialah *Network Prefix*.

Tabel 2.1 menggambarkan secara rinci pembagian *IP address* ke dalam kelas-kelas seperti yang telah diuraikan di atas.

Tabel 2.1 Pembagian kelas IP Address

| **Bit Inisial a** | **Format** | **Range a** | **Jumlah Kelas** | **Kelas** | **Bagian *Network*** | **Bagian *Host*** | **Penggunaan** |
|---|---|---|---|---|---|---|---|
| 0... | 0nnnnnnn.<br>hhhhhhhh.hhhhhhhh.hhhhhhhh | 0 – 127 | 126 | A | a | b,c,d | Untuk Jaringan Besar |
| 10... | 10nnnnnn.nnnnnnnn.<br>hhhhhhhh.hhhhhhhh | 128 – 191 | 16.384 | B | a,b | c,d | Untuk Jaringan Menengah |
| 110... | 110nnnnn.nnnnnnnn.nnnnnnnn.<br>hhhhhhhh | 192 – 233 | 2.097.152 | C | a,b,c | d | Untuk Jaringan Kecil |
| 1110... | 1110mmmm.mmmmmmmm.mmmmmmmm.<br>mmmmmmmm | 224 – 247 | - | D | a,b,c | d | Cadangan : IP Multicasting |
| 1111... | 1111rrrr.rrrrrrrr.rrrrrrrr.rrrrrrrr | 248 – 255 | - | E | a,b,c | d | Cadangan: eksperimen |

#### 2.2.2.2 Aturan dasar pemilihan *network* ID dan *Host* ID

Terdapat beberapa aturan dasar dalam menentukan *network* ID dan *host* ID yang hendak digunakan. Aturan tersebut ialah:

1 *Network* ID tidak boleh sama dengan 127

*Network* ID tidak dapat digunakan karena secara default digunakan untuk keperluan loopback. Loopback ialah *IP address* yang digunakan komputer untuk menunjukkan dirinya sendiri.

2 *Network* ID dan *Host* ID tidak boleh sama dengan 255

Seluruh bit dari *Network* ID dan *Host* ID tidak boleh semuanya diset 1. Jika hal ini dilakukan, *Network* ID dan *Host* ID tersebut akan diartikan sebagai alamat *broadcast*. ID *broadcast* merupakan alamat yang mewakili seluruh anggota jaringan. Pengiriman paket ke alamat *broadcast* akan menyebabkan paket akan dikirimkan ke seluruh anggota *network* tersebut.

3 *Network* ID dan *Host* ID tidak boleh sama dengan 0 (nol)

*IP address* dengan *Host* ID 0 diartikan sebagai alamat *network*. Alamat *network* ialah alamat yang digunakan untuk menunjukkan suatu jaringan, dan tidak menunjukkan suatu *host*.

4. *Host* ID harus unik dalam satu *network*

Dalam satu *network* tidak boleh ada dua *host* yang memiliki *Host* ID yang sama.

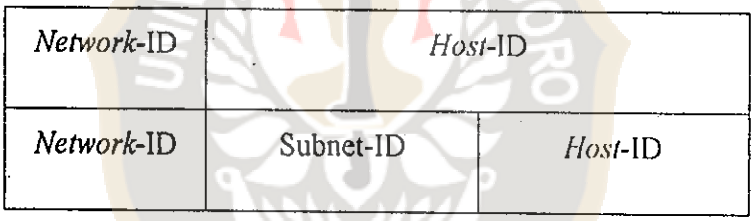
## 2.3 Subnetting

Subnetting adalah pembagian suatu kelompok alamat IP menjadi bagian-bagian yang lebih kecil lagi. Tujuan dalam melakukan subnetting ini adalah :

a. Membagi suatu kelas jaringan menjadi bagian-bagian yang lebih kecil.

b. Menempatkan suatu host, apakah berada dalam satu jaringan atau tidak.

Caranya adalah dengan mengorbankan sebagian *host* ID untuk dipakai dalam membuat *network* ID tambahan. Penjelasan subnetting dapat dilihat pada gambar 2.4 berikut.




*Gambar 2.4 Subnetting*

### 2.3.1 Subnet mask

Subnet mask ialah angka biner 32 bit yang digunakan untuk :

a. Membedakan network ID dan host ID

b. Menunjukkan letak suatu host, apakah ia berada di jaringan lokal atau jaringan luar.

Tabel 2.2 berikut ini menunjukkan subnet mask dari masing-masing kelas *IP address.*

Tabel 2.2 Subnet mask untuk tiap-tiap kelas *IP address*

| Kelas | Bit subnet mask | Subnet Mask Default |
|---|---|---|
| A | 11111111.00000000. 00000000. 00000000 | 255.0.0.0 |
| B | 11111111. 11111111.00000000. 00000000 | 255.255.0.0 |
| C | 11111111. 11111111. 11111111.00000000 | 255.255.255.0 |

Contoh :

Misal suatu jaringan kelas C dengan *IP address* 192.168.10.0 ingin dibagi menjadi 5 jaringan kecil (masing-masing terdiri dari 48 host). Dalam hal ini akan dilakukan proses subnetting.

Penyelesaian :

Langkah pertama yang harus kita lakukan adalah menentukan subnet mask dari *IP address* tesebut. Cara menentukan subnet masknya adalah sebagai berikut :

a. Mengubah jumlah network yang dibutuhkan menjadi bilangan biner. Dalam hal ini jumlah network yang dibutuhkan adalah 5. Untuk merepresentasikan persis sama dengan 5 tidak ada, maka dipilih bilangan yang di atasnya yaitu 8.
b. Menghitung jumlah bit yang dibutuhkan untuk merepresentasikan angka tersebut. Untuk merepresentasikan angka 5 ke dalam biner dibutuhkan 3 bit.

Bit sebanyak inilah yang dibutuhkan oleh subnet ID. Jumlah bit host ID sekarang ialah jumlah bit host ID yang lama dikurangi bit yang diperlukan untuk subnet ID. Jika sebelumnya IP kelas C yang digunakan memakai 8 bit untuk host ID, sekarang berarti tersisa 5 saja.

c. Mengisi subnet ID ini dengan bit 1, sehingga subnet mask baru adalah :

11111111.11111111.11111111.11100000

Seperti telah diketahui bahwa tiap-tiap kelas C mempunyai 255 *IP address*, maka perhitungannya adalah sebagai berikut :

255 / 5 = 51

Masing-masing subnet mempunyai 49 alamat IP (masing-masing diambil 2 untuk IP broadcast dan IP network).

Berikut adalah pengelompokan dari jaringan-jaringan tersebut :

a. 192.168.10.0 - 192.168.10.50 digunakan oleh jaringan 1
b. 192.168.10.51 - 192.168.10.101 digunakan oleh jaringan 2
c. 192.168.10.102 - 192.168.10.152 digunakan oleh jaringan 3
d. 192.168.10.153 - 192.168.10.203 digunakan oleh jaringan 4
e. 192.168.10.204 - 192.168.10.224 digunakan oleh jaringan 5

Subneting diperlukan agar host pada satu jaringan tidak dapat mengakses host pada jaringan lain secara langsung.

Untuk pembagian 51 host : 51 = 00110011 (biner). Nilai 8 bit tertinggi dari subnetting kelas C adalah : 255 =

11111111

00110011

---------- (negasi)

11001100 (8 bit terakhir dari subnet kelas C) = 204

maka IP subnetmask nya : 255.255.255.204.

## 2.3 Bootstrap Protocol (BOOTP)

Bootstrap Protocol (BOOTP) adalah sebuah protokol yang dirancang agar komputer dapat menginisialisasi dirinya pada suatu jaringan (Berry Kercheval, 2001). Karena protokol DHCP dibangun di atas protokol BOOTP, maka perlu dipelajari struktur pesan dan bagaimana cara kerja protokol ini.

Komputer klien dapat meminta informasi kepada komputer server BOOTP, antara lain informasi *IP address* klien, *IP address* server, dan informasi file-file konfigurasi agar komputer klien dikenali dalam jaringan.

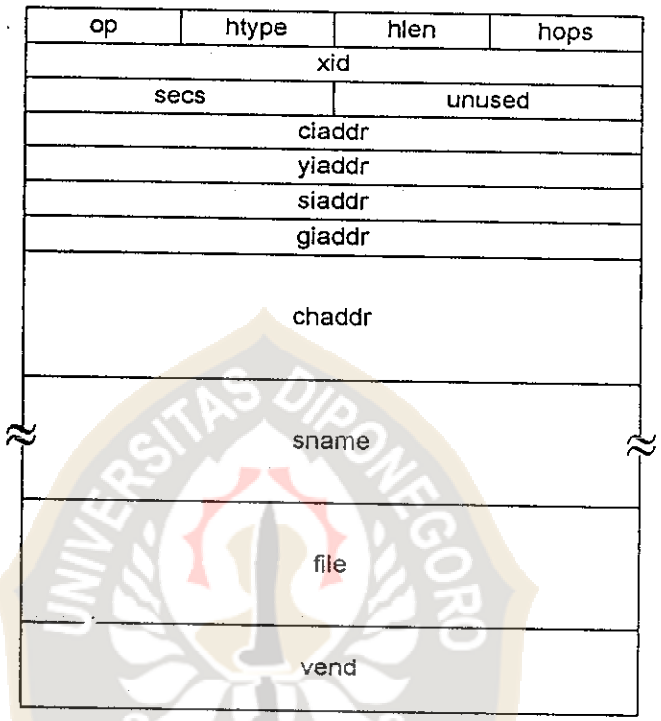
Berikut ini merupakan field-field yang ada pada format paket pesan BOOTP:

Tabel 2.3 Field paket BOOTP

| Field | Panjang field | Keterangan |
|---|---|---|
| Op | 1 bit | Kode op menunjukkan jenis paket. Jika field berisi nilai 1, maka paket itu adalah BOOTREQUEST dan jika nilainya 2 paket berupa BOOTREPLY |
| Htype | 1 bit | Jenis hardware atau alamat MAC yang sedang digunakan oleh klien. Jika nilainya 1 berarti Ethernet, 4 berarti token ring. |
| Hlen | 1 byte | Panjang alamat hardware, dalam byte. Alamat Ethernet dan token ring adalah 6 bit, sedangkan ATM 20 bit. |
| Hops | 1 byte | Diatur ke nol oleh klien; digunakan dalam booting transrouter. |
| Xid | 4 byte | Transaksi ID adalah nilai random yang dipilih klien untuk mencocokkan jawaban yang dikirim server dengan permintaan yang dikirim. |
| Secs | 2 byte | Diatur oleh klien yaitu waktu sisa (dalam detik) terhitung sejak klien memulai proses booting. |
| Flags | 2 byte | Byte yang berisi flag 16 byte. Semua bit yang lain diatur ke nol untuk penggunaan di masa datang. |
| Ciaddr | 4 byte | Alamat IP klien. Jika klien mengetahui alaamat IPnya, maka nilai ini dipenuhi. Jiak tidak, nilainya diatur ke nol. |

| | | |
|---|---|---|
| Yiaddr | 4 byte | Alamat IP "Anda". Server akan memberikan *IP address* untuk klien, jika field ciaddr oleh klien diatur ke nol. |
| Siaddr | 4 byte | Alamat IP server, yaitu alamat IP server yang menghasilkan jawaban. |
| Giaddr | 4 byte | Alamat gateway yang digunakan dalam booting trans-router. |
| Chaddr | 16 byte | Alamat hardware klien, disini klien mengisi alamat Ethernet-nya atau alamat MAC lainnya. |
| Sname | 64 byte | *Host*name server, ditunjukkan sebagai string dengan terminasi nol. |
| File | 128 byte | Nama bootfile yang harus diload dan diaktifkan oleh klien. |
| Vend | 64 byte | Area yang digunakan untuk berbagai ekstensi khusus vendor. |

Format paket BOOTP terlihat seperti gambar 2.5 berikut ini :




*Gambar 2.5 Format Paket BOOTP*

Proses yang terjadi dalam BOOTP adalah :

1. Untuk memulai pertukaran BOOTP, komputer klien harus membuat dan mengirim sebuah paket BOOTREQUEST.
   a. Klien menginisialisasi alamat *ethernet card*-nya.
   b. Klien BOOTP mengirimkan alamat kartu jaringan / MAC-nya (*Media Access Layer)* dalam bentuk datagram UDP. Jika alamat server diketahui, maka alamat tujuan dari paket IP sama dengan alamat tersebut, tetapi sebaliknya, jika klien tidak tahu alamat IPnya, maka alamat tujuan dapat

diatur ke *broadcast* address (255.255.255.255). Port UDP yang digunakan adalah port 67.

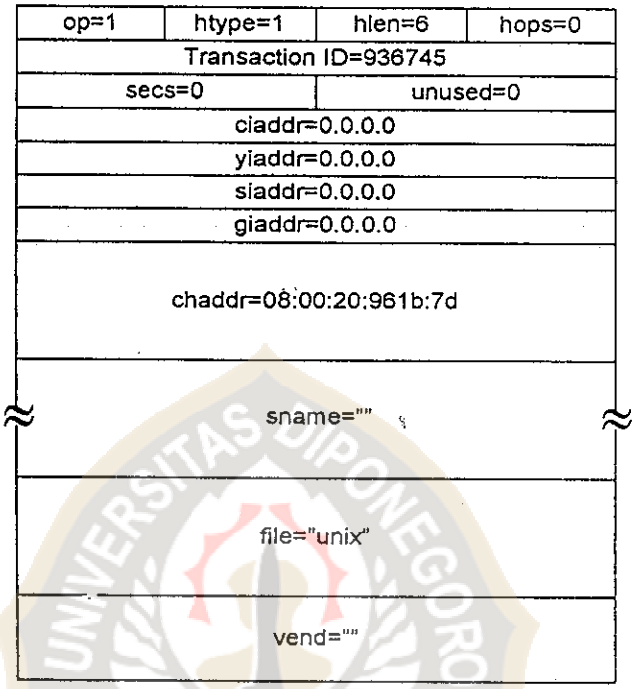
c. Server menerima datagram, kemudian memproses paket tersebut dan mencari file konfigurasi yang sesuai dengan informasi hardware yang dikirimkan klien. Kemudian apabila klien:

   a. Tahu alamat IPnya (yang termuat dalam paket BOOTREQUEST), server akan mengirimkan datagram ke alamat tersebut.

   b. Tidak tahu alamat IPnya (klien mengisi 0.0.0.0 dalam paket BOOTREQUEST), maka server hanya mempunyai dua pilihan. Jika server mampu, ia akan menyusun entry tabel ARP-nya sendiri untuk klien, karena server mengetahui alamat hardware klien (Ethernet) dari paket BOOTREQUEST yang diterimanya. Jika server tidak mampu melakukannya, server akan segera mem-*broadcast* jawaban.

d. Setelah klien menerima paket BOOTREPLY dari server, ia akan segera memprosesnya.

Berikut ini adalah contoh paket BOOTREQUEST yang telah diisi klien




*Gambar 2.6 Format BOOT REQUEST yang telah diisikan*

Keterangan :

a. op = 1

   berarti jenis paket tersebut adalah paket BOOTREQUEST

b. htype = 1

   berarti bahwa jenis hardware yang digunakan adalah Etehrnet

c. hlen = 6

   menunjukkan panjang alamat hardware klien yang menggunakan jenis Ethernet

d. hops = 0

   Diatur ke nol oleh klien, digunakan dalam booting transrouter.

e. Transaction ID = 936745

Nilai random yang dipilih klien untuk mencocokkan jawaban yang dikirim server dengan permintaan yang dikirim.

f. Secs = 0

Waktu sisa terhitung sejak klien memulai proses booting (dalam detik).

g. ciaddr = 0.0.0.0

Alamat IP klien, berisi 0.0.0.0 karena klien belum mengetahui alamat IP-nya.

h. yiaddr = 0.0.0.0

Jika ciaddr bernilai 0.0.0.0, server DHCP akan memberikan *IP address* pada field ini.

i. siaddr = 0.0.0.0

Alamat IP server, berisi 0.0.0.0 karena klien belum mengatahui alamat server.

j. giaddr = 0.0.0.0

Alamat gateway atau router, berisi 0.0.0.0 karena tidak ada gateway dalam jaringan tersebut.

k. chaddr = 08:00:20:961b:7b

Alamat hardware klien ( identitas kartu jaringan ).

l. snames = “”

Hostname server, nama komputer server.

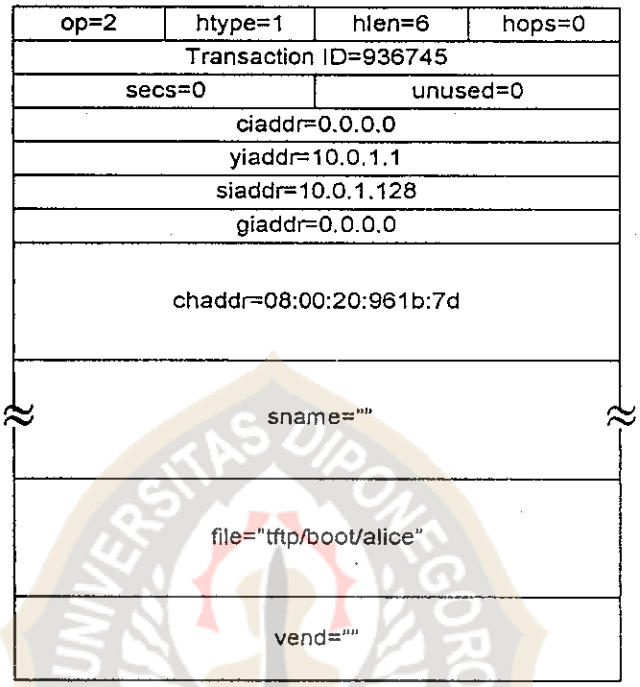
m. file=”unix”

Nama file yang diload server dan diaktifkan oleh klien.

n. vend = “”

Nama vendor yang digunakan.

Berikut ini adalah contoh paket BOOTREPLY yang telah diisi server.




*Gambar 2.7 Format BOOTREPLY yang telah diisikan.*

Keterangan :

a. op = 2

   berarti jenis paket tersebut adalah paket BOOTREPLY

b. htype = 1

   berarti bahwa jenis hardware yang digunakan adalah Etehrnet

c. hlen = 6

   menunjukkan panjang alamat hardware klien yang menggunakan jenis Ethernet

d. hops = 0

   Diatur ke nol oleh klien, digunakan dalam booting transrouter.

e. Transaction ID = 936745

Nilai random yang dipilih klien untuk mencocokkan jawaban yang dikirim server dengan permintaan yang dikirim.

f. Secs = 0

Waktu sisa terhitung sejak klien memulai proses booting (dalam detik).

g. ciaddr = 0.0.0.0

Alamat IP klien, berisi 0.0.0.0 karena klien belum mengetahui alamat IP-nya.

h. yiaddr = 10.0.1.1

Alamat IP untuk klien yang diberikan server, karena pada paket BOOTREQUEST field ciaddr bernilai 0.0.0.0.

i. siaddr = 10.0.1.128

Alamat IP server, berisi 0.0.0.0 karena klien belum mengatahui alamat server.

j. giaddr = 0.0.0.0

Alamat gateway atau router, berisi 0.0.0.0 karena tidak ada gateway dalam jaringan tersebut.

l. chaddr = 08:00:20:961b:7b

Alamat hardware klien ( identitas kartu jaringan ).

m. snames = “”

Hostname server, nama komputer server.

n. file=”tftp/boot/alice”

Nama file yang diload server dan diaktifkan oleh klien.

o. vend = “”

Nama vendor yang digunakan.