**BAB II**

**ANALISIS dan PERANCANGAN**

1. **1 LANDASAN TEORI**

**2.1 .1 Jaringan komputer**

Jaringan komputer merupakan sekumpulan komputer berjumlah banyak yang terpisah-pisah akan tetapi dapat saling dihubungkan dalam melaksanakan tugas. Dua buah komputer misalnya dikatakan terkoneksi bila keduanya dapat saling bertukar informasi. Bentuk koneksi dapat melalui: kawat tembaga, serat optik, gelombang mikro, satelit komunikasi.

Disini secara terbatas dan sederhana dijelaskan secara singkat LAN, MAN, WAN dan Internet.

**1. LAN :**

Menghubungkan komputer-komputer pribadi dalam kantor perpusahaan, pabrik atau kampus: LAN dapat dibedakan dari jenis jaringan lainnya berdasarkan 3 karakteristik: ukuran, teknologi transmisi dan *topologi* jaringan.

**2 . MAN :**

Merupakan versi LAN ukuran lebih besar dan biasanya memakai teknologi yang sama dengan LAN. MAN mampu menunjang data dan suara, dan bahkan dapat berhubungan dengan jaringan televisi kabel. MAN hanya memiliki. sebuah atau dua buah kabel dan tidak mempunyai elemen *switching,* yang berfungsi untuk mengatur paket melalui beberapa kabel *output.*

**3. WAN :**

Mencakup daerah geografis yang luas, seringkali mencakup negara atau benua. WAN terdiri dari kumpulan mesin yang bertujuan untuk menjalankan program-program (aplikasi) pemakai. Mesin ini disebut *HOST*. *HOST* dihubungkan oleh sebuah subnet komunikasi atau cukup disebut *SUBNET*. Tugas subnet adalah membawa pesan dari satu *host* ke *host* lainnya. Pada sebagian besar WAN subnet terdiri dari 2 komponen: kabel transmisi dan elemen *switching*.

1. **Internet :**

Terdapat banyak jaringan di dunia ini, seringkali dengan perangkat keras dan perangkat lunak yang berbeda-beda. Orang yang terhubung ke jaringan sering berharap untuk dapat komunikasi dengan orang lain yang terhubung ke jaringan lainnya. Keinginan seperti ini memerlukan hubungan antar jaringan yang seringkali tidak kompatibel dan berbeda. Kadang menggunakan mesin yang disebut *GATEWAY* sebagai penerjemah antar jaringan yang tidak kompatibel. Kumpulan jaringan yang terkoneksi disebut *INTERNETWORK* atau *INTERNET*. Bentuk *INTERNET* yang umum adalah kumpulan dari *LAN* yang dihubungkan oleh *WAN*.

**2.1.2 *OSI Layer***

Gambar *OSI Layer* juga menggambarkan header - header yang diberikan pada setiap lapisan kepada data yang kirimkan dari lapisan ke lapisan.



Ganbar 1. OSI Model: Gambaran Tiap Layer

Setiap lapisan OSI memiliki tugas yang berbeda satu sama lain. Berikut masing-masing tugas dari tiap lapisan:

1. **Application Layer :**

Bertanggung jawab menyediakan layanan untuk aplikasi misalnya email, akses suatu komputer atau layanan.

1. **Presentation Layer :**

Bertanggung jawab untuk menyandikan informasi. Lapisan ini membuat dua *host* dapat berkomunikasi.

1. **Session Layer :**

Membuat sesi untuk proses dan mengakhiri sesi tersebut. Contohnya jika ada login secara interaktif maka sesi dimulai dan kemudian jika ada permintaan log off maka sesi berakhir. Lapisan ini juga menghubungkan lagi jika sesi login terganggu sehingga terputus.

1. **Transport Layer :**

Lapisan ini mengatur pengiriman pesan dari *host* - *host* di jaringan. Pertama data dibagi-bagi menjadi paket-paket sebelum pengiriman dan kemudian penerima akan menggabungkan paket-paket tersebut menjadi data utuh kembali. Lapisan ini juga memastikan bahwa pengiriman data bebas kesalahan dan kehilangan paket data.

1. **Network Layer :**

Lapisan bertanggung jawab untuk menerjemahkan alamat logis jaringan ke alamat fisik jaringan. Lapisan ini juga memberi identitas alamat, jalur perjalanan pengiriman data, dan mengatur masalah jaringan misalnya pengiriman paket-paket data.

1. **Data Link Layer :**

Lapisan data link mengendalikan kesalahan antara dua komputer yang berkomunikasi lewat lapisan *physical*. Data link biasanya digunakan oleh *hub* dan *switch*.

1. **Physical Layer :**

Lapisan *physical* mengatur pengiriman data berupa bit lewat kabel. Lapisan ini berkaitan langsung dengan perangkat keras seperti kabel, dan kartu jaringan (*LAN CARD*).

* + 1. **PENGALAMATAN**

**2.1.3.1 IP *ADDRESS* dan *SUBNETTING***

**2.1.3.1.1 IP *ADDRESS***

Ip *Address* dirancang untuk memungkinkan sebuah *host* dalam suatu *network* dapat berkomunikasi dengan *host* lain dalam *network* berbeda. Pada dasarnya sebuah Ip *address* terdiri atas 32 bit informasi. Bit-bit ini dibagi lagi menjadi empat seksi,atau yang dikenal dengan *octet,*dimana masing-masing memuat 1 byte (8 bit).

Alamat yang unik terdiri dari 32 bit yang dibagi dalam 4 oktet (8 bit)

00000000 . 00000000 . 00000000 . 00000000

 o 1 o 2 o 3 o 4

Komponen Ip *address* dibagi menjadi 2 bagian yaitu *Network* ID dan *Host* ID, *Network* ID yang akan menentukan alamat dalam jaringan (*network address*) sedangkan *Host* ID menentukan alamat dari peralatan jaringan yang sifatnya unik untuk membedakan antara satu mesin dengan mesin lainnya Ibaratkan *Network* ID Nomor jalan dan alamat jalan sedangkan *Host* ID adalah nomor rumahnya.

**Menentukan alamat *Network***

Untuk menentukan alamat *network* dari sebuah alamat IP dengan menggunakan sebuah *subnet mask* tertentu, dapat dilakukan dengan menggunakan sebuah operasi matematika, yaitu dengan menggunakan operasi logika *AND*. Di dalam sebuah *AND*, nilai dari dua hal yang dioperasikan akan bernilai true hanya ketika dua item tersebut bernilai true; dan menjadi false jika salah satunya false. Dengan mengaplikasikan prinsip ini ke dalam *bit*-*bit*, nilai 1 akan didapat jika kedua *bit* yang diperbandingkan bernilai 1, dan nilai 0 jika ada salah satu di antara nilai yang diperbandingkan bernilai 0.

Cara ini akan melakukan sebuah operasi logika *AND* dengan menggunakan 32-bit alamat IP dan dengan 32-bit *subnet mask*, yang dikenal dengan operasi *bitwise logical AND*. Hasil dari operasi bitwise alamat IP dengan *subnet mask* itulah yang disebut dengan *network identifier*.

Contoh:

Alamat IP 10000011 01101011 10100100 00011010 (131.107.164.026)

Subnet Mask 11111111 11111111 11110000 00000000 (255.255.240.000)

----------------------------------------------------- AND

Network ID 10000011 01101011 10100000 00000000 (131.107.160.000)

IP address dibagi menjadi 5 kelas yaitu ;



 Table class IP *Address*

* Kelas A digunakan untuk jaringan yang sangat besar dengan memiliki banyak *host*. 1 *bit* pertama IP *Address*-nya “0”
* Kelas B digunakan pada jaringan yang menengah sampai ke jaringan yang besar. 2 *bit* pertama IP *Address*-nya “10”.
* Kelas C digunakan pada jaringan yang kecil. Kelas ini banyak dipakai pada perusahaan yang berinvestasi pada jaringan. 3 *bit* pertama IP *Address*-nya “110”
* Kelas D hanya digunakan untuk *Multicating*. Yaitu apabila kita menginginkan untuk mengirim data kepada beberapa penerima sekaligus atau sebuah grup. 4 *bit* pertama IP *Address*-nya “1110”
* Kelas E digunakan hanya untuk kepentingan penelitian . Tidak digunakan oleh umum. 4 *bit* pertama IP *Address*-nya “1111”

**2.1.3.1.2 *IP PRIVATE DAN IP PUBLIC***

Berdasarkan jenisnya IP *address* dibedakan menjadi 2 macam yaitu *IP Private dan IP Public.*

*IP Private* adalah suatu IP address yang digunakan oleh suatu organisasi yang diperuntukkan untuk jaringan lokal. Sehingga organisasi lain dari luar organisasi tersebut tidak dapat melakukan komunikasi dengan jaringan lokal tersebut. Contoh pemakaiannya adalah pada jaringan intranet.

Range IP *Private* menurut *Internet Assigned Numbers Authority* (IANA) RFC 1918 telah mencadangkan tiga blok dari alamat IP ruang untuk Internets pribadi sebagai berikut :

Kelas A : 10.0.0.0 - 10.255.255.255 (10 / 8 *prefix*)

Kelas B : 172.16.0.0 - 172.31.255.255 (172.16/12 *prefix*)

Kelas C : 192.168.0.0 - 192.168.255.255 (192.168/16 *prefix*)

*IP Public* *address* oleh semua pihak dapat menyebabkan terjadinya kesamaan nomor IP yang digunakan. Hal ini menyebabkan dijelaskannya aturan mengenai *IP Public*. IP jenis ini digunakan oleh setiap orang yang ini memasuki kominitas jaringan yang besar seperti internet. *IP Public* adalah semua nomor IP pada semua kelas yang ada kecuali *IP Private*.

**2.1.3.1.3 Konversi IP *ADDRESS***

**1. *Binary* ke *Desimal***

IP *address* dapat direpresentasikan kedalam 2 macam cara:

* *Decimal* (misal 131.107.2.200)
* *Binary* (misal 1000 0011. 01101011. 00000010. 11001000)

Menggunakan IP *address* dengan menggunakan format notasi bertitik seperti 131.107.2.200, sementara computer secara internal menggunakan system *binary* untuk berkomunikasi antar *host*. Jangan meremehkan kemampuan untuk melakukan konversi dari *decimal* ke binary atau sebaliknya. Kemampuan ini berguna untuk membuat *address* pada jaringan.

Table berikut adalah patokan untuk mengkonversikan *decimal* ke *binary*. Baris pertama adalah posisi *bit* yang dari kanan menuju ke kiri dimulai dari nilai 0 sampai 7. Posisi *bit* pertama dengan nilai 0 dan sampai posisi *bit* terakhir (posisi ke 8 ) dengan nilai 7. Sementara nilai *bit* hanya 1 atau 0 sebagai bilangan *binary*.

Misalkan pada posisi *bit* ke 4 dengan nilai *bit* 1 mempunyai harga *decimal* (2 pangkat 3) = 8, dengan rumusan:

*2 pangkat (n – 1) dimana n adalah posisi bit*

Pada posisi *bit* ke 8 dengan nilai bit 1 mempunyai harga decimal (2 pangkat 7) = 128 dst.



Table 1. Konversi *Biner* ke *Desimal*

Sebagai pedoman yang perlu dihafalkan adalah angka 128 di sebelah kiri bawah table, posisi *bit* ke 8 dengan *bit* value 1 adalah 128 (2 pangkat 7). Untuk posisi *bit* ke 7 bagi saja dengan dua hasil tadi jadi (128 / 2) = 64, posisi bit ke 6 (64 / 2) = 32 dan seterusnya.

Dengan table ini akan memudahkan untuk mengkonversikan nilai *decimal* ke *binary*. Misalkan saja untuk IP *address* dalam bentuk binary berikut: 11000000. 10101000. 11001000.11111110

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Posisi bit | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Nilai bit | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Decimal | 128 | 64 | 0 | 0 | 0 | 0 | 0 | 0 |

Nilai decimal dibaris terakhir jumlahkan, maka binary 11000000 nilai decimalnya adalah (128+64) = 192

**2. *Decimal* ke *Binary***

Konversi dari *binary* ke *decimal* dapat dijelaskan dengan menggunakan table berikut ini, dengan masih mengacu pada table konversi . Dimisalkan adalah konversi IP *address* 218.132.10.55 kedalam format dapat dijelaskan sebagai berikut.



Table 2. konversi *binary*

Untuk angka *decimal* pada *octet* pertama 218, kurangi 218 dengan 128. Jika dapat dikurangi, maka pada posisi *bit* ke 8 nilai binary nya dalah 1, dan sisa pengurangan = (218-128) = 90.Begeser pada posisi *bit* ke 7, kurangi sisa tadi (90) dengan 64, karena dapat dikurangi nilai *bit* posisi ke 7 adalah 1, dan sisa pengurangan adalah (90-64) = 26. Bergeser kekanan lagi ke posisi *bit* ke 6, kurangi angka sisa tadi (26) dengan angka 32, karena tidak dapat dikurangi (minus) maka posisi *bit* ke 6 adalah angka *binary* 0. Geser lagi ke kanan ke posisi *bit* ke 5, kurangi angka sisa 26 dengan angka 16. Karena dapat dikurangi maka posisi *bit* ke 5 adalah 1.

Geser kekanan lagi ke posisi *bit* ke 4, kurangi angka sisanya tadi (10) dengan angka 8, karena dapat dikurangi maka posisi ke 4 adalah nilai *bit* 1.Geser lagi kekanan ke posisi *bit* ke 3, kurangi angka sisa (2) dengan angka 4, karena tidak dapat maka posisi *bit* ke tiga ini adalah 0.Geser lagi ke kanan ke posisi bit ke 2, kurangi angka sisa tadi (2) dengan angka 2, karena dapat dikurangi maka posisi *bit* ke dua ini adalah 1.Dan untuk posisi *bit* terakhir ke 1 adalah angka sisa pengurangan posisi *bit* ke 2, yaitu 0, tidak ada sisa. Jadi angka decimal 218 = 11011010

Anda dapat menyelesakan dengan cara yang sama untuk angka decimal 132, 10, dan 55 seperti pada contoh table diatas. Sehingga akhirnya diketemukan angka *decimal* IP *address* 218.132.10.55 adalah 11011010. 10000100. 00001010. 00110111

**2.1.4 *ROUTER***

*Cisco* *router* adalah peratan utama yang banyak digunakan pada Jaringan Area Luas atau *Wide Area Network (WAN)*. Dengan *cisco* *router*, informasi dapat diteruskan ke alamat - alamat yang berjauhan dan berada di jaringan computer yang berlainan. Untuk dapat meneruskan paket data dari suatu LAN ke LAN lainnya, *Cisco* *router* menggunakan tabel dan *protocol* *routing* yang berfungsi untuk mengatur lalu lintas data. Paket data yang tiba di *router* diperiksa dan diteruskan ke alamat yang dituju. Agar paket data yang diterima dapat sampai ke tujuannya dengan cepat, *router* harus memproses data tersebut dengan sangat tepat.

Untuk itu, *Cisco* *Router* menggunakan *Central Processing Unit* (CPU) seperti yang digunakan di dalam komputer untuk memproses lalu lintas data tersebut dengan cepat. Seperti komputer, *cisco* *router* seri 2500 juga mempunyai sejumlah jenis memori yaitu *ROM, RAM, NVRAM* dan *FLASH*, yang berguna untuk membantu kerjanya *CPU*. Selain itu dilengkapi pula dengan sejumlah *interface* untuk berhubungan dengan dunia luar dan keluar masuk data. Sistem operasi yang digunakan oleh *cisco* *router* adalah *Internetwork Operating System (IOS)*.



Gambar 2. Blok Internal *Router*

Memori yang digunakan oleh *cisco* *router* masing-masing mempunyai kegunaan sendiri sendiri sebagai berikut :

• *ROM* berguna untuk menyimpan sistem *bootstrap* yang berfungsi untuk mengatur proses boot dan menjalankan *Power On Self Test (POST)* dan *IOS* image. Untuk *router* *cisco* 2500: 2 MB

• *RAM* berguna untuk menyimpan *running* *conﬁguration* dan dan sistem operasi *IOS* yang aktif. Untuk *router* *cisco* 2500: 16 MB

• *NVRAM* berguna untuk menyimpan konﬁgurasi awal (*start-up conﬁguration*). Untuk *router* *cisco* 2500: 34 KB

• *FLASH* berguna untuk menyimpan *IOS* image. Dengan menggunakan *FLASH*, *IOS* versi baru dapat diperoleh dari *TFTP* server tanpa harus mengganti komponen dalam *router*. Untuk *router* *cisco* 2500: 34 KB

**Menghubungkan *Cisco* *router* ke PC**

Untuk menghubungkan *cisco* *router* ke suatu terminal atau komputer, diperlukan kabel *console* dan *adaptor* RJ-45 ke DB-9 yang biasanya disertakan dengan peralatan *router* tersebut. Cara memeriksa apakah suatu kabel berjenis *console* sangat mudah.

Kabel rollover ini dihubungkan dari *console* port *router* ke serial port COM 1 atau COM 2 komputer. Bergantung pada serial port, jika panel belakang komputer menggunakan konektor DB-25 atau DB-9, maka diperluakan adapter RJ-45 ke DB-9 atau DB-25 yang sesuai. Setelah itu, *cisco* *router* dan komputer yang saling terhubung dapat dihidupkan. *Hy*-*perterminal* digunakan sebagai program untuk menghubungkan antara *cisco* *router* dengan *computer.*

* + 1. ***ROUTING***

Konfigurasi *routing* secara umum terdiri dari 3 macam yaitu:

1. Minimal *Routing*

Dari namanya dapat diketahui bahwa ini adalah konfigurasi yang paling sederhana tapi mutlak diperlukan. Biasanya minimal *routing* dipasang pada *network* yang *terisolasi* dari *network* lain atau dengan kata lain hanya pemakaian lokal saja.

2. Static *Routing*

Konfigurasi *routing* jenis ini biasanya dibangun dalam *network* yang hanya mempunyai beberapa *gateway*, umumnya tidak lebih dari 2 atau 3. Static *routing* dibuat secara manual pada masing-masing *gateway*. Jenis ini masih memungkinkan untuk jaringan kecil dan stabil. Stabil dalam arti kata jarang down. Jaringan yang tidak stabil yang dipasang static *routing* dapat membuat kacau seluruh *routing*, karena tabel *routing* yang diberikan oleh gateway tidak benar sehingga paket data yang seharusnya tidak dapat diteruskan masih saja dicoba sehingga menghabiskan *bandwicth*.

Setiap penambahan sebuah *router*, maka *router* yang telah ada sebelumnya harus diberikan tabel *routing* tambahan secara manual. Jadi jelas, static *routing* tidak mungkin dipakai untuk jaringan besar, karena membutuh effort yang besar untuk mengupdatenya.

3. *Dynamic* *Routing*

Dalam sebuah *network* dimana terdapat jalur *routing* lebih dari satu rute untuk mencapai tujuan yang sama biasanya menggunakan *dynamic* *routing*. Dan juga selain itu *network* besar yang terdapat lebih

dari 3 *gateway*. Dengan *dynamic* *routing*, tinggal menjalankan *routing* protokol yang dipilih dan biarkan bekerja. Secara otomatis tabel *routing* yang terbaru akan didapatkan.

Seperti dua sisi uang, *dynamic* *routing* selain menguntungkan juga sedikit merugikan. *Dynamic* *routing* memerlukan *routing* protokol untuk membuat tabel *routing* dan *routing* *protokol* ini dapat memakan *resource* komputer.

* 1. **PERANCANGAN**

*Administrator - administrator* yang mengelola jaringan besar sering sekali merasa perlu membagi-bagi jaringan mereka menjadi bgian-bagian yang lebih kecil lagi (disebut *subnetwork*) sebagai usaha memberikan *fleksibelitas* *addressing*. Dengan *subneting*,sebuah *network* *address* tunggal dapat dipecah menjadi banyak *subnetwork* atau *subnet.*

**2.2.1 *SUBNETTING* PADA IP *ADDRESS* CLASS C**

*Subnetting* seperti apa yang terjadi dengan sebuah *NETWORK ADDRESS* **192.168.1.0/26**

**Analisa**: 192.168.1.0 berarti kelas C dengan Subnet Mask /26 berarti 11111111.11111111.11111111.11000000 (255.255.255.192).

**Penghitungan**: Seperti sudah disebutkan sebelumnya semua pertanyaan tentang subnetting akan berpusat di 4 hal, jumlah subnet, jumlah *host* per *subnet*, *blok* *subnet*, alamat *host* dan *broadcast* yang valid. Jadi diselesaikan dengan urutan seperti itu:

1. **Jumlah *Subnet*** = 2x, dimana x adalah banyaknya binari 1 pada oktet terakhir *subnet* *mask* (2 *oktet* terakhir untuk kelas B, dan 3 *oktet* terakhir untuk kelas A). Jadi Jumlah *Subnet* adalah 22 = 4 *subnet*
2. **Jumlah *Host* per *Subnet*** = 2y – 2, dimana y adalah adalah kebalikan dari x yaitu banyaknya binari 0 pada *oktet* terakhir subnet. Jadi jumlah *host* per subnet adalah 26 – 2 = 62 *host*
3. **Blok *Subnet*** = 256 – 192 (nilai *oktet* terakhir *subnet* *mask*) = 64. *Subnet* berikutnya adalah 64 + 64 = 128, dan 128+64=192. Jadi *subnet* lengkapnya adalah **0, 64, 128, 192**.
4. Bagaimana dengan alamat ***host* dan *broadcast* yang valid** Kita langsung buat tabelnya. Sebagai catatan, *host* pertama adalah 1 angka setelah *subnet*, dan *broadcast* adalah 1 angka sebelum *subnet* berikutnya.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Subnet*** | 192.168.1.**0** | 192.168.1.**64** | 192.168.1.**128** | 192.168.1.**192** |
| ***Host* Pertama** | 192.168.1.**1** | 192.168.1.**65** | 192.168.1.**129** | 192.168.1.**193** |
| ***Host* Terakhir** | 192.168.1.**62** | 192.168.1.**126** | 192.168.1.**190** | 192.168.1.**254** |
| ***Broadcast*** | 192.168.1.**63** | 192.168.1.**127** | 192.168.1.**191** | 192.168.1.**255** |

Selesaikan *subnetting* untuk IP *address* Class C. Dan dapat melanjutkan lagi untuk *subnetmask* yang lain, dengan konsep dan teknik yang sama. *Subnetmask* yang dapat digunakan untuk *subnetting* class C adalah seperti di bawah.

|  |  |
| --- | --- |
| ***Subnet Mask*** | **Nilai *CIDR*** |
| 255.255.255.128 | /25 |
| 255.255.255.192 | /26 |
| 255.255.255.224 | /27 |
| 255.255.255.240 | /28 |
| 255.255.255.248 | /29 |
| 255.255.255.252 | /30 |

**2.2.2 *SUBNETTING* PADA IP *ADDRESS* CLASS B**

Berikut ini akan dicoba melakukan subnetting untuk IP *address* class B. Pertama, subnet mask yang dapat digunakan untuk *subnetting* class B. Sengaja dipisahkan jadi dua, blok sebelah kiri dan kanan karena masing-masing berbeda teknik terutama untuk oktet yang “dimainkan” berdasarkan blok *subnetnya*. CIDR /17 sampai /24 caranya sama persis dengan *subnetting* Class C, hanya blok *subnetnya* masukkan langsung ke oktet ketiga, bukan seperti Class C yang di *oktet* keempat. Sedangkan *CIDR* /25 sampai /30 (kelipatan) blok *subnet* di *oktet* keempat, tapi setelah selesai oktet ketiga berjalan maju dari 0, 1, 2, 3, dst.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |
| --- | --- |
| ***Subnet Mask*** | **Nilai *CIDR*** |
| 255.255.128.0 | /17 |
| 255.255.192.0 | /18 |
| 255.255.224.0 | /19 |
| 255.255.240.0 | /20 |
| 255.255.248.0 | /21 |
| 255.255.252.0 | /22 |
| 255.255.254.0 | /23 |
| 255.255.255.0 | /24 |

 |

|  |  |
| --- | --- |
| ***Subnet* *Mask*** | **Nilai *CIDR*** |
| 255.255.255.128 | /25 |
| 255.255.255.192 | /26 |
| 255.255.255.224 | /27 |
| 255.255.255.240 | /28 |
| 255.255.255.248 | /29 |
| 255.255.255.252 | /30 |

 |

Kedua teknik *subnetting* untuk Class B. Mulai dari yang menggunakan *subnetmask* dengan *CIDR* /17 sampai /24. Contoh *network* *address* **172.16.0.0/18**.

**Analisa**: 172.16.0.0 berarti kelas B, dengan *SubnetMask* /18 berarti 11111111.11111111.11000000.00000000 (255.255.192.0).

**Penghitungan**:

1. **Jumlah *Subnet*** = 2x, dimana x adalah banyaknya binari 1 pada 2 oktet terakhir. Jadi Jumlah *Subnet* adalah 22 = 4 subnet
2. **Jumlah *Host* per *Subnet*** = 2y – 2, dimana y adalah adalah kebalikan dari x yaitu banyaknya binari 0 pada 2 *oktet* terakhir. Jadi jumlah *host* per *subnet* adalah 214 – 2 = 16.382 *host*
3. **Blok *Subnet*** = 256 – 192 = 64. *Subnet* berikutnya adalah 64 + 64 = 128, dan 128+64=192. Jadi *subnet* lengkapnya adalah **0, 64, 128, 192**.
4. **Alamat *host* dan broadcast yang valid**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Subnet*** | 172.16.**0.0** | 172.16.**64.0** | 172.16.**128.0** | 172.16.**192.0** |
| ***Host* Pertama** | 172.16.**0.1** | 172.16.**64.1** | 172.16.**128.1** | 172.16.**192.1** |
| ***Host* Terakhir** | 172.16.**63.254** | 172.16.**127.254** | 172.16.**191.254** | 172.16.**255.254** |
| ***Broadcast*** | 172.16.**63.255** | 172.16.**127.255** | 172.16.**191.255** | 172.16.**.255.255** |

Satu lagi untuk Class B khususnya untuk yang menggunakan subnetmask CIDR /25 sampai /30. Contoh *network* address **172.16.0.0/25**.

**Analisa**: 172.16.0.0 berarti kelas B, dengan *SubnetMask* /25 berarti 11111111.11111111.11111111.10000000 (255.255.255.128).

**Penghitungan**:

1. **Jumlah *Subnet*** = 29 = 512 *subnet*
2. **Jumlah *Host* per *Subnet*** = 27 – 2 = 126 *host*
3. **Blok *Subnet*** = 256 – 128 = 128. Jadi lengkapnya adalah (**0, 128**)
4. **Alamat *host* dan *broadcast* yang valid**?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Subnet*** | 172.16.**0.0** | 172.16.**0.128** | 172.16.**1.0** | …  | 172.16.**255.128** |
| ***Host* Pertama** | 172.16.**0.1** | 172.16.**0.129** | 172.16.**1.1** | …  | 172.16.**255.129** |
| ***Host* Terakhir** | 172.16.**0.126** | 172.16.**0.254** | 172.16.**1.126** | … | 172.16.**255.254** |
| ***Broadcast*** | 172.16.**0.127** | 172.16.**0.255** | 172.16.**1.127** | … | 172.16.**255.255** |

**2.2.3 SUBNETTING PADA IP ADDRESS CLASS A**

Dilanjut ke *Class* A. Konsepnya semua sama saja. Perbedaannya adalah *dioktet* mana anda mainkan blok *subnet*. Class C *dioktet* ke 4 (terakhir), kelas B *dioktet* 3 dan 4 (2 *oktet* terakhir), Class A *dioktet* 2, 3 dan 4 (3 *oktet* terakhir). Kemudian *subnetmask* yang dapat digunakan untuk *subnetting* class A adalah semua *subnetmask* dari *CIDR* /8 sampai /30.

Coba latihan untuk *network* *address* **10.0.0.0/16**.

**Analisa**: 10.0.0.0 berarti kelas A, dengan *SubnetMask* /16 berarti 11111111.11111111.00000000.00000000 (255.255.0.0).

**Penghitungan**:

1. **Jumlah *Subnet*** = 28 = 256 *subnet*
2. **Jumlah *Host* per *Subnet*** = 216 – 2 = 65534 *host*
3. **Blok *Subnet*** = 256 – 255 = 1. Jadi *subnet* lengkapnya: 0,1,2,3,4, etc.
4. **Alamat *host* dan *broadcast* yang valid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Subnet*** | 10.**0.0.0** | 10.**1.0.0** | …  | 10.**254.0.0** | 10.**255.0.0** |
| ***Host* Pertama** | 10.**0.0.1** | 10.**1.0.1** | …  | 10.**254.0.1** | 10.**255.0.1** |
| ***Host* Terakhir** | 10.**0.255.254** | 10.**1.255.254** | … | 10.**254.255.254** | 10.**255.255.254** |
| ***Broadcast*** | 10.**0.255.255** | 10.**1.255.255** | … | 10.**254.255.255** | 10.**255.255.255** |



**Table 3. *subnetmask***