BAB II
TEORI PENUNJANG

2.1 Jaringan Komputer

Dengan semakin berkembangnya kebutuhan pengolahan data dan informasi, dalam sebuah organisasi dibutuhkan lebih dari satu komputer yang digunakan oleh banyak orang yang bekerja dalam sebuah tim. Untuk saling bertukar data dan informasi, maka komputer-komputer yang digunakan akan dihubungkan satu dengan yang lainnya. Kumpulan-komputer yang saling terhubung disebut sebagai jaringan komputer [5].

Dengan penggunaan jaringan komputer beberapa keuntungan yang didapat diantaranya adalah :

1. Membagi sumber daya

   Dengan jaringan sumber daya yang ada dapat digunakan secara bersama-sama. Misalkan *user* menggunakan sebuah komputer dan menginginkan mengakses data di komputer lain seolah-olah data tersebut berada di komputer yang user gunakan.
2. Reliabilitas tinggi

   Dengan menggunakan jaringan komputer, *user* mendapatkan reliabilitas yang tinggi dengan memiliki sumber data alternatif. Misalkan data yang digunakan dapat disimpan atau dicopy ke beberapa komputer yang terhubung dengan jaringan. Sehingga bila salah satu komputer rusak, maka salinan di komputer yang lain masih dapat digunakan.
3. *Hardware sharing*

   Dengan menggunakan jaringan komputer *user* dapat menggunakan *hardware* secara bersama-sama. Misalkan sebuah *printer* dapat digunakan bersama-sama dengan user lain. Begitu pula dengan perangkat lainnya.
4. Keamanan data terjamin

   Dengan menggunakan jaringan komputer memungkinkan seorang *administrator* untuk dapat mengontrol data-data penting agar dapat diakses oleh *user* yang mendapatkan ijin dari *administrator*. Sehingga keamanan data akan lebih terjamin.

5. Menghemat biaya

   Dengan menggunakan jaringan komputer penghematan perangkat dapat dilakukan, karena sebuah perangkat dapat digunakan secara bersama-sama. Misalkan penggunaan *printer, scanner* dan CD/DVD ROM.

2.1.1 Jenis Jaringan Komputer

Jaringan komputer dapat dibedakan berdasarkan cakupan geografisnya. Ada tiga kategori utama jaringan komputer yaitu :

1. LAN *(Local Area Network)*

   LAN (*Local Area Network*) adalah suatu kumpulan komputer, dimana terdapat beberapa unit komputer (*client*) dan satu unit komputer bank data (*server*). Antara masing-masing *client* maupun antara *client* dan *server* dapat saling bertukar *file* maupun saling menggunakan *printer* yang terhubung pada unit-unit komputer yang terhubung pada jaringan LAN.

2. MAN *(Metropolitan Area network*)

   Merupakan jaringan sejati (*true networks*), yaitu versi dari LAN yang berukuraan lebih besar dan bisanya memakai teknologi yang sama dengan LAN. MAN mampu menunjang data dan suara. MAN memiliki satu atau dua buah kabel dan tidak mempunyai elemen *switching.*

3. WAN *(Wide Area Network*)

   WAN *(Wide Area Network*) adalah kumpulan dari LAN atau *workgroup* yang dihubungkan dengan menggunakan alat komunikasi, umumnya menggunakan *modem* untuk membentuk hubungan dari atau kekantor pusat dan kantor cabang, maupun antarkantor cabang. Dengan sistem jaringan ini, pertukaran data antarkantor dapat dilakukan dengan cepat serta dengan biaya yang relatif murah.

Berdasarkan media penghantar, jaringan komputer dapat dibagi menjadi dua jenis, yaitu :

1. *Wire Network*

   Wire Network adalah jaringan komputer yang menggunakan kabel sebagai media penghantar, jadi data ditransmisikan pada kabel. Kabel yang umum

digunakan pada jaringan komputer biasanya menggunakan bahan dasar tembaga. Ada juga jenis kabel lain yang menggunakan bahan fiber optik atau serat optik. Bahan tembaga banyak digunakan pada LAN. Sedangkan untuk MAN dan WAN menggunakan gabungan kabel tembaga dan serat optik.

2. *Wireless Network*

   Wireless Network adalah jaringan tanpa kabel yang menggunakan media penghantar gelombang radio. Frekuensi yang digunakan pada radio untuk jaringan komputer biasanya menggunakan frekuensi tinggi, yaitu 2.4 GHz dan 5.8 GHz.

Berdasarkan fungsinya, jaringan komputer dapat dibagi menjadi dua jenis, yaitu:

1. *Client Server*

   Cient server adalah jaringan komputer yang salah satu komputernya difungsikan sebagai *server* atau komputer yang memberikan pelayanan kepada komputer lain. *Server* melayani komputer lain yang disebut *client*. Layanan yang diberikan bisa berupa akses *web*, *e-mail*, *file* atau yang lain. *Client server* banyak dipakai oleh *internet* dan intranet.

2. *Peer to Peer*

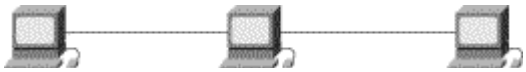
   Peer to peer adalah jaringan komputer dimana setiap komputer bisa menjadi *server* sekaligus *client*. Setiap komputer dapat menerima dan memberikan akses dari atau ke komputer lain.

2.1.2 Topologi Jaringan Komputer

Topologi adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Berikut jenis-jenis topologi dasar jaringan diantaranya :

1. Bus

   Topologi jaringan jenis ini menggunakan sebuah kabel pusat yang merupakan media utama dari jaringan. Terminal-terminal yang akan membangun jaringan dihubungkan dengan kabel utama yang merupakan inti dari jaringan. Gambar 2.1 menunjukan bentuk topologi bus.

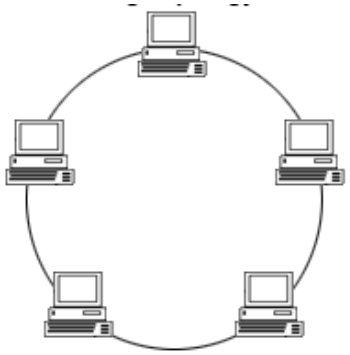


Gambar 2.1. Topologi Bus

Data yang dikirimkan akan langsung menuju terminal yang dituju tanpa harus melewati terminal-terminal dalam jaringan, atau akan di routingkan ke *head end controller*. Tidak bekerjanya sebuah komputer tidak akan menghentikan kerja dari jaringan, jaringan akan tak bekerja apabila kabel utamanya dipotong atau putus.

2. Ring

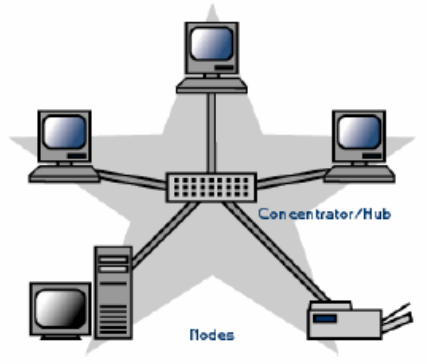
Topologi jenis ring menghubungkan satu komputer didalam suatu *loop* tertutup. Pada topologi ini data atau *message* berjalan mengelilingi jaringan dengan satu arah pengiriman ke komputer selanjutnya terus hingga mencapai komputer yang dituju. Waktu yang di butuhkan untuk mencapai terminal tujuan disebut waktu transmisi.



Gambar 2.2. Topologi Ring

3. Star

Topologi model ini dirancang, yang mana setiap *nodes* (*file server*, *workstation*, dan perangkat lainya) terkoneksi ke jaringan melewati sebuah *concentrator*. Data yang dikirim ke jaringan *local* akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Concentrator* akan mengatur dan mengendalikan keseluruhan fungsi jaringan, dan juga bertindak sebagai *repeater*.



Gambar 2.3. Topologi Star

2.2 IP address

IP *address* merupakan identifikasi setiap *host* pada jaringan Internet. Masing-masing *host* mempunyai alamat IP untuk mengidentifikasi suatu *host* dalam melakukan koneksi pada jaringan TCP/IP. Didalam setiap alamat IP terdapat *network* ID dan *host* ID. *Network* ID menunjukan letak atau tempat sistem berada yang terhadap dalam *router* IP pada jaringan fisik yang sama, sedangkan *host* ID lebih dikenal dengan stasiun kerja (*workstasion*), *server*, *roter*, dan *host* TCP/IP yang lainnya dalam suatu jaringan [4].

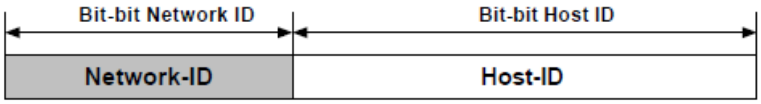
2.2.1 Format IP Address

Pengalamatan IP *address* mempunyai format dalam bilangan biner yang terdiri dari 32-bit dan dibagi atas 4 kelompok 8-bit bilangan biner (atau sering disebut dengan istilah *oktal*).

Table 2.1 Format IP Address

Binary	**Decimal**
00000000.00000000.00000000.00000000	= 0.0.0.0
Sampai dengan	
11111111.11111111.11111111.11111111	= 255.255.255.255

Format IP *address* diatas dapat diubah menjadi 0.0.0.0 sampai dengan 255.255.255.255. Nilai desimal merupakan nilai IP *address* yang dikenal dalam pemakaian sehari-hari. IP *address* dapat dipisahkan menjadi dua bagian yaitu :



Gambar 2.4. Pembagian IP Address

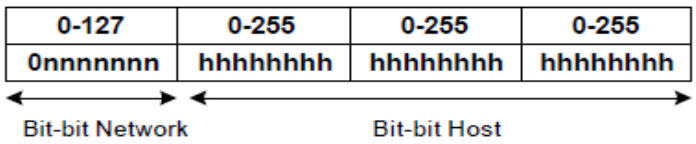
Bit *network* ID berperan dalam identifikasi *network address* sedangkan bit *host* ID berperan dalam identifikasi *host* dalam suatu jaringan. Seluruh *host* yang tersambung dalam jaringan yang sama memiliki bit *network* ID yang sama. Garis pemisah antara bit *network* ID dan bit *host* ID tidak tetap, bergantung kepada *network class* [4].

2.2.2 Pembagian Kelas IP Address

IP *address versi* 4 dapat dibagi menjadi 5 bagian, yaitu kelas A, kelas B, kelas C, kelas D, dan kelas E, tetapi kelas yang paling banyak digunakan adalah kelas A, B, dan C, karena kelas E digunakan untuk alamat *multicast* yang tidak memiliki *network* ID dan host ID, sedangkan kelas E digunakan untuk penggunaan khusus. Berikut adalah pembagian kelas IP *address* yaitu :

1. IP *address* kelas A

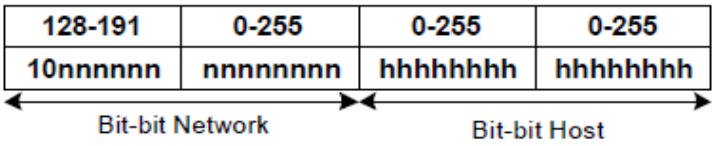
   Pada bit pertama bernilai 0. Bit ini dan 7 bit berikutnya merupakan *network* ID, sedangkan sisanya yaitu 24 bit terakhir merupakan bit untuk host ID, sehingga IP *address* kelas A digunakan untuk jaringan dengan jumlah host yang sangat besar (*n* menyatakan *network*, sedangkan *h* menyatakan host).



Gambar 2.5. IP *Address* Kelas A

2. IP *address* kelas B

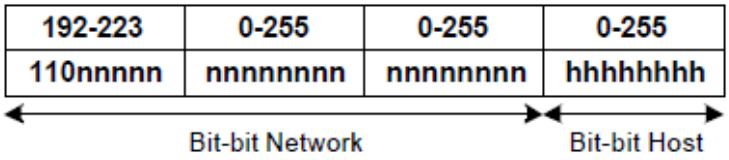
   Pada dua bit pertama bernilai 10. Dua bit ini dan 14 bit berikutnya (16 bit pertama) merupakan *network* ID, sedangkan sisanya yaitu 16 bit terakhir merupakan host ID, sehingga IP *address* kelas B digunakan untuk jaringan dengan jumlah host yang tidak terlalu besar (*n* menyatakan *network*, sedangkan *h* menyatakan host).



Gambar 2.6. IP *Address* Kelas B

3. IP *address* kelas C

   Tiga bit pertama bernilai 110. Tiga bit ini dan 21 bit berikutnya (24 bit pertama) merupakan network ID, sedangkan sisanya yaitu 8 bit terakhir yaitu host ID, sehingga pada IP *address* kelas C biasanya digunakan untuk jaringan Local Area Network atau LAN (*n* menyatakan *network*, sedangkan *h* menyatakan host).



Gambar 2.7. IP *Address* Kelas C

4. IP *address* kelas D

   Jika 4 bit pertama adalah 1110, IP *address* merupakan kelas D yang digunakan untuk *multicast address*, yakni sejumlah komputer yang memakai bersama suatu aplikasi. Pada IP address kelas D tidak dikenal bit-bit *network* dan host.

5. IP *address* kelas E

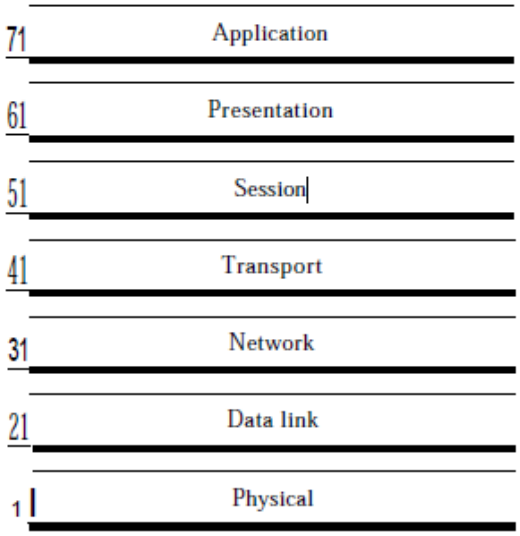
   Empat bit pertama adalah 1111. IP *address* kelas E dicadangkan untuk kegiatan *riset* atau *experimental*. Pada IP *address* kelas E juga tidak dikenal bit-bit *network* dan *host.*

2.3 Protokol Jaringan Komputer

Protokol jaringan adalah perangkat aturan yang digunakan dalam jaringan. Protokol adalah aturan main yang mengatur komunikasi diantara beberapa komputer didalam sebuah jaringan sehingga komputer-komputer anggota jaringan dan komputer berbeda *platform* dapat saling berkomunikasi. semua jenis-jenis jaringan komputer menggunakan protokol. Aturan-aturan protokol adalah termasuk didalamnya petunjuk yang berlaku bagi cara-cara atau metode mengakses sebuah jaringan, topologi fisik, tipe-tipe kabel dan kecepatan transfer data.

2.3.1 OSI (*Open System Interconnection*) Model

Model OSI merupakan *framework* berlapis untuk merancang sistem jaringan yang memungkinkan komunikasi antara semua jenis sistem komputer. Hal itu terdiri dari tujuh lapisan terpisah namun berhubungan, yang masing-masing mendefinisikan bagian dari proses perpindahan informasi melalui jaringan. Sebuah pemahaman dasar-dasar dari model OSI memberikan dasar yang kuat untuk menjelajahi komunikasi data [4].



Gambar 2.8. Model OSI Layer

Ketujuh lapisan tersebut mempunyai peran dan fungsi yang berbeda satu terhadap yang lain. Setiap *layer* bertanggung jawab secara khusus pada proses komunikasi data. Tujuan utama penggunaan model OSI adalah untuk membantu desainer jaringan memahami fungsi dari tiap-tiap *layer* yang berhubungan dengan aliran komunikasi data, termasuk jenis-jenis protokol jaringan dan metode transmisi. Model OSI dibagi menjadi 7 *layer*, dengan karakteristik dan fungsinya masing-masing. Tiap *layer* harus dapat berkomunikasi dengan *layer* diatasnya maupun di bawahnya secara langsung melalai beberapa protokol dan *standard* [1].

1. *Physical Layer*

   Berfungsi untuk mendefinisikan media transmisi jaringan, sinkronisasi bit, arsitektur jaringan. Selain itu, *layer* ini juga mendefinisikan bagaimana *network interface card* berinteraksi dengan media *wire* atau *wireless.* Contoh protokol pada *layer* ini yaitu Hub, Repeater, Fibre Optic, 10base-t, 100base-tx.

2. *Data Link Layer*

   Berfungsi untuk menentukan bagaimana bit-bit dikelompokan menjadi format yang disebut *frame*. Pada *layer* ini juga terjadi pengalamatan perangkat keras dan menentukan bagaimana perangkat-perangkat jaringan beroprasi. Contoh protokol pada *layer* ini, Ethernet, Frame Relay, Token Ring, FDDI.

3. *Network Layer*

   Berfungsi untuk mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket data, dan melakukan *routing*. Contoh protokol pada *layer* ini, IP, IPX, ICMP, ARP, RIP, IGMP, OSPF.

4. *Transport Layer*

   Berfungsi untuk memecah data menjadi paket-paket data serta untuk memberikan nomor urut setiap paket sehingga dapat disusun kembali setelah diterima. Contoh protokol yang digunakan pada *layer* ini TCP dan UDP.

5. *Session Layer*

   Berfungsi untuk mendefinisikan bagaimana koneksi dimulai, dipelihara, dan diakhiri. Contoh protokol pada *layer* ini, Netbios, ADSP, PAP.

6. *Presentation Layer*

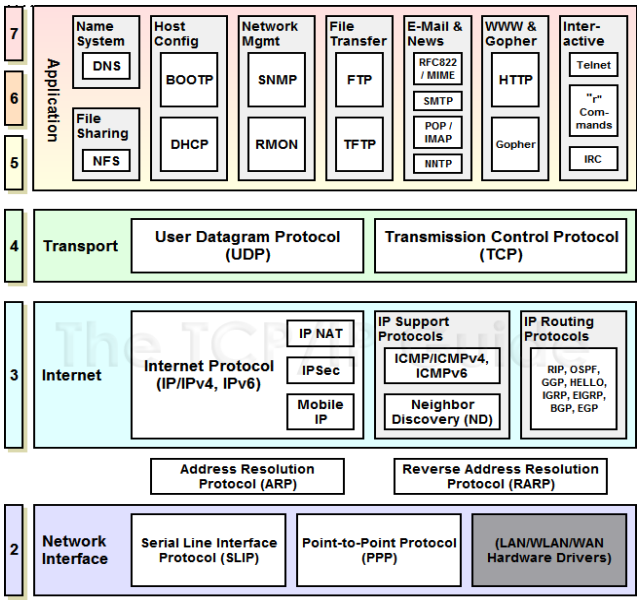
   Berfungsi untuk mentranslasikan data yang hendak ditransmisikan kedalam format yang dimengerti dalam jaringan. Contoh protokol pada *layer* ini TDI, ASCII, MIDI, MPEG.

7. *Application Layer*

   Berfungsi sebagai antarmuka dengan *user*. Mengatur bagaimana aplikasi dapat mengakses jaringan. Contoh protokol yang berada pada *layer* ini, FTP, Telnet, HTTP, SMTP, POP3, NFS, SMNP.

2.3.2 Protokol TCP/IP

Protokol TCP/IP terbentuk dari 2 komponen yaitu *Transmission Control Protocol* (TCP) dan *Internet Protocol* (IP). Protokol ini dikembangkan oleh *Advence Research Project Agency* (ARPA) untuk departemen pertahanan Amerika pada tahun 1969. TCP/IP sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). Sebuah alamat TCP/IP adalah nilai biner berukuran 32 bit yang diberikan ke setiap *host* dalam sebuah jaringan [4].



Gambar 2.9. Protokol TCP/IP

TCP/IP protokol dikembangkan sebelum model OSI. Oleh karena itu, lapisan di TCP/IP protokol tidak sama persis dalam model OSI. TCP/IP protokol memiliki empat lapisan yaitu *network*, Internet, *transport*, dan *application*. Namun, ketika TCP/IP dibandingkan dengan OSI dapat dikatakan bahwa lapisan *network* setara dengan gabungan dari *layer data link* dan *physical*. Internet *layer* setara dengan *layer network*, dan *applicaton* mengerjakan pekerjaan yang sama seperti *session*, *presentation*, dan *application* [4]. Berikut penjelasan lebih lanjut tentang *layer network*, Internet, *transport*, dan *application* :

1. *Network Layer*

   Pada model OSI dapat merupakan gabungan dari *layer physical* dan *data link* yang bertanggung jawab mengirim dan menerima data ke atau dari media fisik. Media fisik bisa berupa kabel, serat optik atau gelombang radio, sehingga protokol ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasa dari peralatan lain.

2. *Internet Layer*

   Pada model OSI sama dengan *network layer* yang bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada *layer* ini terdapat tiga macam protokol yaitu IP, ARP, dan ICMP.

3. *Transport Layer*

   Protokol yang bertanggung jawab untuk mengadakan komunikasi antar dua *host*/komputer. Protokol tersebut adalah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*). TCP merupakan Protokol yang berorientasi pada hubungan yang andal (*reliable connection-oriented protocol*) yang mengizinkan sebuah aliran *byte* yang berasal pada suatu mesin untuk dikirimkan tanpa *error* ke sebuah mesin yang ada di Internet. TCP memecah aliran *byte* data menjadi pesan-pesan diskret dan meneruskannya ke Internet *layer*. UDP merupakan protocol yang tidak andal (*unreliable*) dan tanpa sambungan (*connectionless*) bagi aplikasi-aplikasi yang tidak memerlukan pengurutan TCP atau pengendalian aliran dan bagi aplikasi-aplikasi yang ingin melayani dirinya sendiri.

4. *Application layer*

   Pada model OSI identik dengan *application layer, presentation layer, session layer*. Pada *layer* ini terletak semua aplikasi atau protokol-protokol tingkat tinggi yang menggunakan protokol TCP/IP. Contoh protokol tingkat tinggi adalah terminal *virtual* (TELNET), transfer file (FTP), surat elektronik (SMTP), *domain name service* (DNS), untuk manajemen jaringan (SNMP).

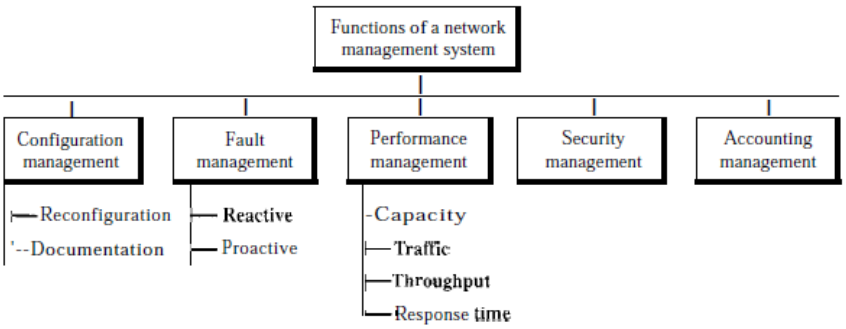
2.4 *Network Monitoring System* (NMS)

NMS merupakan *tool* untuk melakukan monitoring atau pengawasan pada elemen-elemen dalam jaringan komputer. Fungsi dari NMS adalah melakukan pemantauan terhadap kualitas SLA *(Service Level Agreement)* dari *banwidth* yang digunakan. SLA *(Service Level Agreement)* adalah bagian dari perjanjian layanan secara keseluruhan antara 2 dua entitas untuk peningkatan kinerja atau waktu pengiriman harus diperbaiki selama perjanjian. Hasil dari pantauan tersebut biasanya dijadikan bahan dalam pengambilan keputusan oleh pihak manajemen, disisi lain digunakan oleh *administrator* jaringan *(technical person)* untuk menganalisa apakah terdapat kejanggalan dalam operasional jaringan. Terdapat 10 alasan utama menggunakan aplikasi monitoring jaringan komputer, yaitu:

1. Mengetahui apa yang sedang terjadi dalam jaringan, dimana solusi NMS selalu memberikan informasi tentang operasional dan konektifitas dari peralatan dan sumber daya yang ada dalam jaringan.
2. Untuk perencanaan peningkatan *(upgrade)* dan perubahan peralatan jaringan.
3. Dapat digunakan untuk mendiagnosa masalah-masalah dalam jaringan.
4. Sebagai bahan untuk keperluan SLA *(service level agreement).*
5. Mengetahui kapan saat yang tepat untuk mengimplementasikan solusi *disaster recovery system* (pemulihan bencana atau masalah) dapat dilaksanakan.
6. Memastikan keamanan sistem beroperasi dengan baik.
7. Memastikan pengguna *(client)* layanan dalam jaringan terkoneksi dengan *server* yang mereka butuhkan.
8. Mendapatkan infomasi status jaringan secara *remote.*
9. Memastikan *uptime* untuk keperluan pengguna yang tergantung dengan ketersediaan jaringan komputer.
10. Menghemat pengeluaran dengan menekan jumlah waktu jaringan *down* dan memangkas waktu untuk menganalisa masalah.

Untuk melakukan monitoring pada komponen atau elemen-elemen jaringan serta mengumpulkan informasi yang sangat banyak dari aktifitas jaringan, melihat, menganalisa secara tepat dan cepat memerlukan sebuah solusi dalam menampilkan informasi-informasi tersebut (dimana didalamnya termasuk peta jaringan, pelaporan, sistem peringatan, informasi historis, pengelompokan masalah dan informasi yang berguna lainnya) dalam sebuah *dashboard* NMS. Selain mempermudah *troubleshooting*, sistem ini akan membantu dalam mengumpulkan data historis jaringan untuk melihat kecendrungan yang timbul pada penggunaan sumber daya dan kapasitas jaringan sehingga dapat didesain dan direncanakan sebuah jaringan yang akurat dan efektif.

Kita dapat mengatakan fungsi yang dilakukan oleh sistem manajemen jaringan dapat dibagi menjadi lima kategori besar yaitu: manajemen konfigurasi, manajemen kesalahan, manajemen kinerja, manajemen keamanan, dan manajemen akuntansi, seperti yang ditunjukkan pada Gambar 2.10 [1].



Gambar 2.10. Fungsi Sistem Manajemen Jaringan

Berdasarkan gambar diatas dapat di jelaskan sebagai berikut :

1. Manajemen Konfigurasi (*configuration management*)

   Sebuah jaringan yang besar biasanya terdiri dari ratusan entitas yang secara fisik atau logis terhubung satu sama lain. Entitas ini memiliki konfigurasi awal ketika jaringan sudah diatur, tetapi dapat berubah seiring dengan waktu. Sistem manajemen konfigurasi harus mengtahui setiap saat status masing-masing entitas dan hubungannya dengan entitas lain. Manajemen konfigurasi dapat dibagi menjadi dua subsistem yaitu rekonfigurasi dan dokumentasi.

2. Manajemen Kesalahan *(fault management)*

   Jaringan yang kompleks saat ini terdiri dari ratusan dan kadang-kadang terdapat ribuan komponen. Operasi yang tepat dari jaringan tergantung pada operasi yang tepat dari masing-masing komponen secara individu maupun dalam hubungan satu sama lain. Kesalahan manajemen adalah bidang manajemen jaringan yang menangani masalah ini. Sebuah sistem manajemen kesalahan yang efektif memiliki dua subsistem yaitu kesalahan manajemen reaktif *(reactive fault management)* dan proaktif kesalahan manajemen *(proactive fault management).*

3. Kinerja Manajemen (*performance management*)

   Kinerja manajemen terkait erat dengan manajemen kesalahan mencoba untuk memantau dan mengontrol jaringan untuk memastikan bahwa hal itu berjalan seefisien mungkin. Kinerja manajemen mencoba untuk mengukur kinerja dengan menggunakan beberapa terukur kuantitas seperti kapasitas *(capacity), lalu lintas (traffic), throughput,* atau *response time.*

4. Manajemen Keamanan (*security management*)

   Manajemen keamanan bertanggung jawab untuk mengendalikan akses ke jaringan berdasarkan kebijakan yang telah ditetapkan.

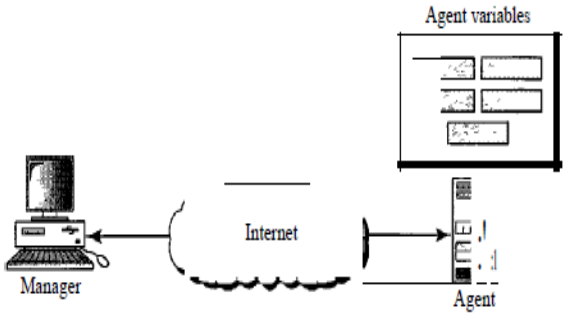
5. Manajemen Akuntansi (*accounting management*)

   Accounting management adalah kontrol akses pengguna ke sumber daya jaringan melalui biaya. Di bawah manajemen akuntansi, pengguna individu, departemen, divisi, atau bahkan proyek akan dikenakan biaya untuk layanan yang mereka terima dari jaringan. Pengisian tidak berarti transfer tunai, mungkin berarti mendebet departemen atau divisi untuk tujuan penganggaran.

2.5 *Simple Network Management Protocol* (SNMP)

Simple Network Management Protocol (SNMP) adalah standar manajemen jaringan pada TCP/IP. Gagasan dibalik SNMP adalah bagaimana supaya informasi yang dibutuhkan untuk manajemen jaringan bisa dikirim menggunakan TCP/IP. Protokol tersebut memungkinkan *administrator* jaringan untuk menggunakan perangkat jaringan khusus yang berhubungan dengan perangkat jaringan yang lain untuk mengumpulkan informasi dari mereka, dan mengatur bagaimana mereka beroperasi. Model umum yang digunakan SNMP adalah adanya *network management station* (NMS) yang mengirim *request* kepada SNMP *agent.* Terdapat dua jenis NMS, yaitu *hardware* dan *software.*

SNMP menggunakan konsep *manager* dan *agent*. Artinya *manager* biasanya sebuah *host*, kontrol dan memantau satu set *agent*.



Gambar 2.11. Konsep SNMP

Sebuah stasiun manajemen disebut *manager* adalah sebuah *host* yang menjalankan program *client* SNMP. Sebuah stasiun yang dikelola disebut *agent* yaitu *host* yang menjalankan program *server* SNMP. Manajemen dicapai melalui interaksi sederhana antara seorang *manager* dan *agent*. *Agent* menyimpan informasi kinerja dalam *database*. *Manager* memiliki akses dengan nilai-nilai dalam *database*. Sebagai contoh, sebuah *host* dapat menyimpan dalam *variabel* yang sesuai dengan jumlah paket yang diterima dan diteruskan. *Manager* dapat mengambil dan membandingkan nilai-nilai dari dua *variabel* untuk melihat apakah *router* tersebut padat atau tidak. *Manager* juga dapat membuat *host* melakukan tindakan tertentu. Sebagai contoh, *host* secara berkala memeriksa nilai *counter reboot* untuk melihat ketika harus *reboot* sendiri. *Reboot* sendiri misalnya, jika nilai *counter* adalah 0. *Manager* dapat menggunakan fitur ini untuk *reboot agent* jarak jauh setiap saat. Ini hanya mengirimkan sebuah paket nilai 0 di *counter*. *Agent* juga dapat berkontribusi untuk proses manajemen. Program *server* yang berjalan pada *agent* dapat memeriksa kondisi dan jika pemberitahuan sesuatu yang tidak biasa dapat mengirimkan pesan peringatan kepada *manajer* [1]. Dengan kata lain, manajemen dengan SNMP dibagi menjadi tiga ide dasar yaitu:

1. *Manager* memeriksa *agent* dengan meminta informasi yang mencerminkan perilaku *agent.*
2. *Manager* memaksa *agent* untuk melakukan tugas dengan mengatur ulang nilai-nilai dalam *database agent.*
3. *Agent* memberikan kontribusi terhadap proses manajemen dengan memperingatkan *manager* dari situasi yang tidak biasa.

2.6 *Internet Control Message Protocol* (ICMP)

ICMP adalah merupakan sebuah protokol yang didesain untuk menangani pesan kesalahan yang tidak mampu dilaporkan oleh Internet *protocol* (IP) jika terjadi masalah pada sebuah jaringan [1]. Pesan atau paket ICMP dikirim jika terjadi masalah pada layer IP dan layer diatasnya (TCP dan UDP). Beberapa karakteristik dari ICMP adalah:

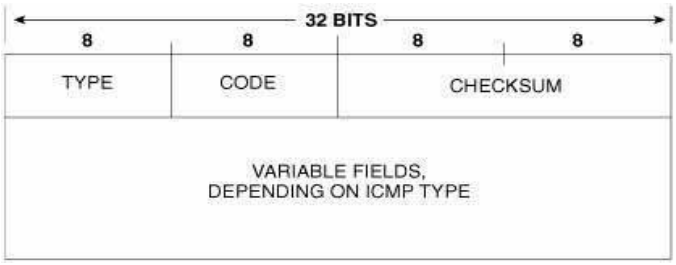
1. ICMP merupakan bagian internal dari IP dan diimplementasikan di setiap modul IP.
2. ICMP digunakan untuk menyediakan *feedback* tentang beberapa kesalahan pada sebuah proses datagram.
3. Tidak mendukung kehandalan pengiriman paket IP. Datagram atau paket bisa tidak terkirim dan tidak ada laporan pemberitahuan tentang kehilangan datagram. Jika diperlukan adanya keandalan maka harus diimplementasikan pada lapisan *transport* (pada model TCP/IP).

2.6.1 Format Pesan ICMP

Sebuah pesan ICMP memiliki 3 *field*, yaitu:

1. 8 bit : *field TYPE* yang mengidentfikasikan pesan.
2. 8 bit : *field CODE* yang menyediakan informasi tentang jenis pesan.
3. 16 bit : *field CHECKSUM* untuk pemerikasaan pesan ICMP.

Format umum dari pesan ICMP adalah sebagai berikut :

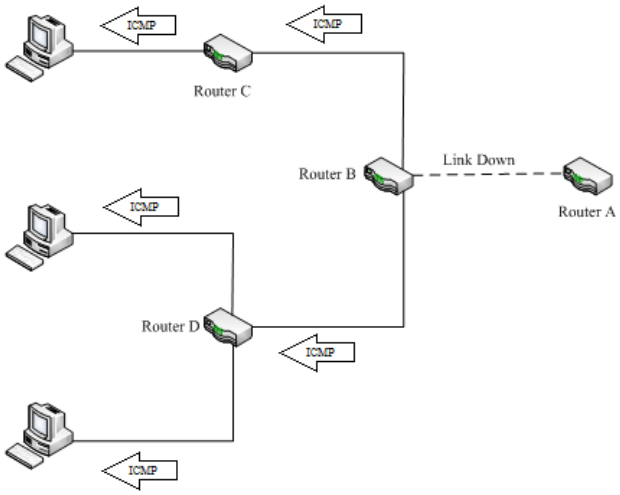


Gambar 2.12 Format Umum Pesan ICMP

2.6.2 ICMP *Error Messages*

Setiap ICMP *error messages* terdiri dari IP *header* dan sekurangnya 8-byte pada datagram. Pada *field typ*e menspesifikasikan tipe dari pesan, sementara kode kesalahan pada datagram dilaporkan oleh pesan ICMP pada *field code*. Pada kondisi normal IP berjalan baik dan menghasilkan proses penggunaan memori serta sumber daya transmisi yang efisien. Namun ada beberapa kondisi di mana koneksi IP terganggu, misalnya karena *router* yang *crash*, putusnya kabel atau matinya *host* tujuan. Pada saat ini ICMP berperan membantu menstabilkan

kondisi jaringan. Hal ini dilakukan dengan cara memberikan pesan-pesan tertentu, sebagai respon atas kondisi tertentu yang terjadi pada jaringan tersebut. Di bawah ini merupakan cara kerja ICMP saat terjadi masalah dalam suatu jaringan.



Gambar 2.13 Masalah yang Terjadi pada Suatu Jaringan

Pada gambar 2.13 diatas hubungan antar *router* A dan B mengalami masalah, sehingga *router* B akan secara otomatis mengirimkan paket ICMP *Destination Unreachable* ke *host* pengirim paket yang berusaha melewati *host* A menuju tujuannya. Dengan adanya pemberitahuan ini maka *host* tujuan tidak akan terus menerus berusaha mengirimkan paketnya melewati *router* A.

Terdapat dua jenis pesan yang dapat dihasilkan oleh ICMP, yaitu ICMP *Error Message* dan ICMP *Query Message*. ICMP *Error Message* dihasilkan jika terjadi kesalahan pada jaringan. Sedangkan ICMP *Query Message* ialah jenis pesan yang dihasilkan oleh ICMP jika pengirim paket menginginkan informasi tertentu yang berkaitan dengan kondisi jaringan. ICMP *error message* dibagi menjadi beberapa jenis diantaranya:

1. *Destination Unreachable*. Pesan ini dihasilkan oleh *router* jika pengiriman paket mengalami kegagalan akibat masalah putusnya jalur, baik secara fisik maupun secara logik. *Destination Unreachable* ini dibagi menjadi beberapa tipe. Beberapa tipe yang penting adalah :
    1. *Network Unreachable*, jika jaringan tujuan tak dapat dihubungi.

2. *Host Unreachable*, jika host tujuan tak bisa dihubungi.
3. *Protocol at Destination is Unreachable*, jika ditujuan tak tersedia protokol tersebut.
4. *Port is Unreachable*, jika tidak ada *port* yang dimaksud pada tujuan.
5. *Destination Network is Unknown*, jika jaringan tujuan tak diketahui.
6. *Destination Host is Unknown*, jika *host* tujuan tidak diketahui.

2. *Time Exceeded*. Paket ICMP jenis ini dikirimkan jika isi *field* TTL dalam paket IP sudah habis dan paket belum juga sampai ke tujuannya. Setiap kali sebuah paket IP melewati satu *router*, nilai TTL dalam paket tersebut dikurangi satu. TTL ini diterapkan untuk mencegah timbulnya paket IP yang terus menerus berputar di jaringan karena suatu kesalahan tertentu, sehingga menghabiskan sumber daya jaringan yang ada.
3. *Parameter Problem*. Paket ini dikirimkan jika terdapat kesalahan parameter pada *header* paket IP.
4. *Source Quench*. paket ICMP ini dikirimkan jika *router* atau tujuan mengalami kongesti. Sebagai respon pada paket ini, pihak pengirim paket harus memperlambat pengiriman paketnya.
5. *Redirect*. Paket ini dikirimkan jika *router* merasa *host* mengirimkan paket IP melalui router yang salah. Paket ini seharusnya dikirimkan melalui *router* lain.

2.6.3 ICMP *Query Messages*

ICMP *Query Messages* terdiri dari:

1. *Echo Request dan Echo Reply*. Bertujuan untuk memeriksa apakah sistem tujuan dalam keadaan aktif. Program ping merupakan program pengiriman paket ini. Sistem tujuan harus mengembalikan data yang sama dengan data yang dikirimkan.
2. *Timestamp dan Timestamp Reply*. Menghasilkan informasi waktu yang diperlukan sistem tujuan untuk memproses suatu paket.
3. *Address Mask*. Untuk mengetahui berapa *netmask* yang harus digunakan oleh suatu *host* dalam suatu jaringan.

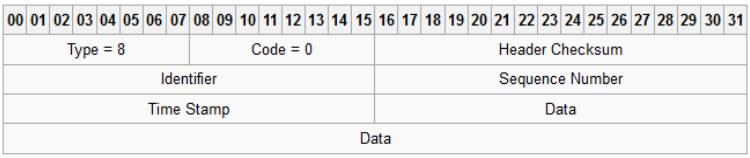
2.6.4 Cara Kerja Ping

Ping bekerja dengan cara mengirimkan ICMP *Echo Request* ke *host* tujuan dan menunggu ICMP *Echo Reply* dari *host* tujuan. ICMP bekerja pada *layer network* (Internet) berdampingan dengan IP. ICMP bergantung pada IP sebagai transportasi untuk melintas dalam jaringan. Program ping dijalankan pada *layer* aplikasi. Program ping memanfaatkan *raw socket* untuk dapat melewati *layer* transport. Socket merupakan fungsi untuk mengalokasikan *resource* sehingga program bisa berinteraksi dengan protokol TCP/IP di bawahnya. Ada tiga jenis socket, yaitu *stream*, *datagram*, dan *raw*. TCP menggunakan *stream* dan UDP menggunakan *datagram*. *Raw socket* digunakan oleh semua aplikasi yang ingin berinteraksi langsung dengan IP, tapi melewati TCP dan UDP, termasuk program ping.

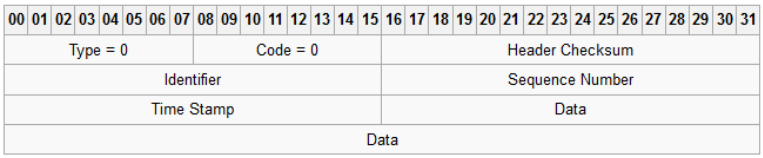
Pada waktu program ping memulai eksekusi, raw socket terbuka hanya untuk ICMP. Pada *output*, pengiriman ICMP *Echo Request*, program harus memformat pesan ICMP. Sistem akan memberikan *IP header* dan *Ethernet header*. *IP address* sumber dan *IP address* tujuan terdapat pada *IP header*. Pada *input*, program harus memeriksa semua pesan ICMP yang datang. Format pesan pada *Echo Request*, *Field Type* bernilai 8 dan *Field Code* bernilai 0 (gambar 2.29). Sedangkan format pesan pada *Echo Reply*, *Field Type* dan *Field Code* bernilai 0 (gambar 2.30). *Checksum* digunakan untuk mendeteksi *error*. *Field Identification* yang berisi ID proses (PID) yang menunjukkan eksekusi program ping. Selanjutnya adalah *Field Sequence*, yang dimulai dari nol dan naik satu setiap *Echo Request* dikirim. Kemudian terdapat field opsional bernama *Test Data*. Ini meliputi waktu yang digunakan dalam kalkulasi waktu pulang-pergi saat menerima *Echo Reply*.

Pada waktu pesan ICMP datang, tugas ping sedikit lebih kompleks. Karena ping menggunakan *raw socket*, program menerima semua pesan ICMP yang datang. Ini berarti ping tidak hanya melihat *Echo Reply* tapi juga yang lain, seperti pesan *Destination Unreachable Source Quench* dan *Time Exceeded*. Jika ada dua program ping yang dijalankan pada waktu bersamaan, maka masing-masing

program harus mengidentifikasi pesan yang ditujukan untuk dirinya. Itu bisa diketahui dengan melihat ID proses pada *Field Identification*. Gambar 2.29 dan gambar 2.30 merupakan gambar format pesan *Echo Request* dan *Echo Reply*.



Gambar 2.14 *Echo Request*



Gambar 2.15 *Echo Reply*

2.7 Server

Server adalah sebuah system komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan. *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau pencetak, dan memberikan akses kepada stasiun kerja anggota jaringan. Umumnya, di dalam sistem operasi *server* terdapat berbagai macam layanan yang menggunakan arsitektur *client/server*. *Server* biasanya terhubung dengan *client* dengan kabel UTP dan sebuah kartu jaringan. Kartu jaringan ini biasanya berupa kartu PCI.

Dilihat dari fungsinya, *server* bisa di kategorikan dalam beberapa jenis, seperti *server* aplikasi, *server* data maupun *server proxy*. *Server* aplikasi adalah *server* yang digunakan untuk menyimpan berbagai macam aplikasi yang dapat diakses oleh *client, server* data sendiri digunakan untuk menyimpan data baik yang digunakan *client* secara langsung maupun data yang diproses oleh *server* aplikasi. *Server proxy* berfungsi untuk mengatur lalu lintas di jaringan melalui

pengaturan *proxy*. Orang awam lebih mengenal *proxy server* untuk mengkoneksikan komputer klien ke Internet.

Layanan *server* yang akan dimonitor oleh Nagios yaitu FTP *server* dimana Nagios memungkinkan untuk memverifikasi koneksi *server* FTP dengan digunakan perintah *check_tcp*. *Plugin* ini identik dengan *check_tcp*, dengan perbedaan bahwa *port* opsional, dan secara *default* diharapkan mendapat pesan selamat datang FTP yang valid.

2.8 Nagios

Nagios adalah NMS *open source* yang dirancang khusus untuk memonitor *host* dan *service* serta untuk menginformasikan kejadian yang terjadi pada jaringan, misalnya pada *host* yang mengalami maslah pada jaringan, Nagios akan melakukan pengiriman notifikasi dengan beberapa cara, yaitu dengan melalui *email*, SMS, dan antarmuka *web* [2].

Nagios juga merupakan sebuah sistem dan aplikasi monitoring jaringan. Nagios mengawasi host-host dan servis yang telah ditetapkan, memberi peringatan jika keadaan memburuk, dan memberi tahu kapan keadaan tersebut membaik. Nagios awalnya didesain untuk berjalan pada sistem operasi Linux, namun dapat juga berjalan dengan baik hampir disemua sistem operasi *Unix like* [3].

Nagios terdapat 2 macam pemeriksaan, yaitu *host check* dan *service check*. Pemeriksaaan terhadap *host* melibatkan sebuah PING *(Packet Internet Groper)* sederhana menggunakan ICMP *(Internet Control Message Protocol),* sedangkan pemeriksaan *service* menggunakan beberapa *Internet Protocol*. *Host dan service check* didefinisikan di Nagios sebagai objek dan menyediakan metode *inheritance* dengan kemampuan untuk melakukan *over-ride* terhadap karakteristik dari *parent object*. Konsep yang digunakan Nagios ini terdapat pada pemrograman berorientasi objek, dan fungsionalitasnya disediakan di Nagios. Dalam mengamati terhadap objek yang dipantau, terdapat empat pernyataan untuk mendeskripsikan status, yaitu *ok, warning, critical,* dan *unknown.* [3].

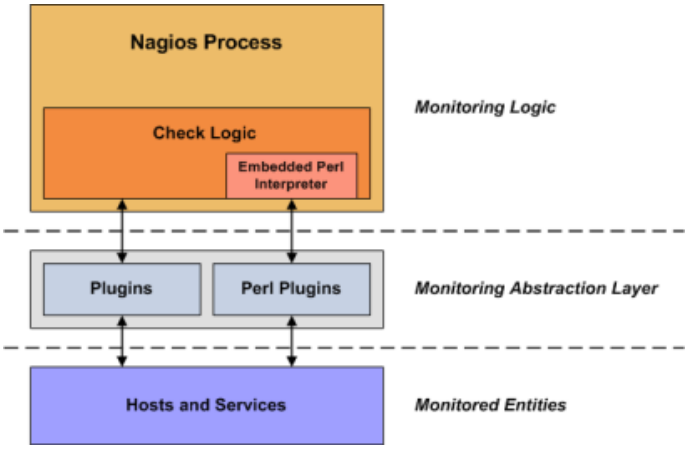
Berikut beberapa fitur-fitur yang tersedia pada Nagios diantaranya adalah:

1. Memonitoring *service* jaringan (HTTP, FTP, SMTP, POP3, dan sebagainya).
2. Memonitoring sumber-sumber *host* (*load prosesor*, penggunaan disk, penggunaan memori, *log file*, dan sebagainya).
3. Desain plugin yang serderhana, yang mengijinkan pengguna untuk lebih mudah menggunakan pemeriksaan terhadap servisnya.
4. Pemberitahuan ketika terjadi masalah pada servis atau host dan mendapatkan pemecahannya (lewat SMS atau *metode user-defined*).
5. Kemampuan untuk mendefinisikan kejadian yang ditangani selama *service/host* berlangsung untuk mempermudah pemecahan masalah.
6. Perputaran *file log* yang otomatis.
7. Mendukung implementasi monitoring dengan *host* yang berlebih.
8. *Web interface* yang fakultatip untuk melihat status *network*, urutan masalah dan pemberitahuan.

Monitoring dengan Nagios juga sangat fleksibel ketika notifikasi datang untuk memberitahu admin tentang *host* atau *service* yang tidak berfungsi dengan benar. Notifikasi dapat diatur untuk mengirim SMS kepada admin yang berbeda tergantung pada layanan yang tidak berfungsi dengan baik.

2.9 *Plugins* Nagios

Plugins merupakan program monitoring yang dipanggil oleh Nagios untuk memonitoring *service* tertentu, dengan kata lain *plugins* yang berhubungan dengan *port-port service* yang dimonitoring.



Gambar 2.16 *Plugins* Nagios

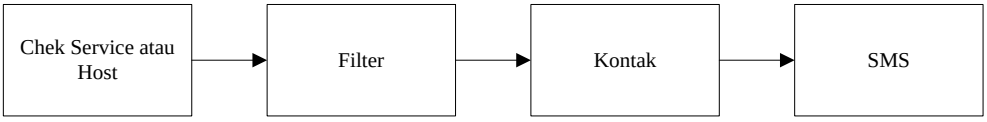
Nilai dari *Plugins* Nagios :

Tabel 2.2 Nilai *Plugins* Nagios

Nama	Keterangan
OK	Semua berjalan dengan baik
WARNING	Melewati batas *warning* tetapi belum *critical*
CRITICAL	Batas critical atau plugins berhenti melakukan test setelah timeout
UNKNOWN	Plugins error

2.10 Notifikasi *Alert* Nagios

Notifikasi merupakan laporan yang akan dikirimkann Nagios pada kontak admin jaringan ketika terjadi gangguan pada *host* atau *service* yang dimonitor.



Gambar 2.17 Notifikasi *Alert* Nagios

Pada gambar 2.17 diatas dapat dijelaskan sebagai berikut :

1. *Check Service* atau *Host*

   Ketika Nagios mendeteksi perubahan status pada suatu *host* atau *service* maka Nagios akan memberitahukan notifikasi melalui *web interface* dan *log file.*
2. *Filter*

   Filter pesan merupakan aturan-aturan yang dibuat dalam konfigurasi agar aktifitas monitoring dan notifikasi yang dikirim ke admin jaringan sesuai dengan yang diinginkan.
3. Kontak

   Kontak yang merupakan isi nomor *handphone* Nagios admin yang telah dikonfigurasi.

4. SMS

   Ketika jaringan mengalami gangguan, Nagios akan mengirimkan notifikasi berupa SMS ke nomor *handphone* admin jaringan.

Tabel 2.3 Status Notifikasi *Host* dan *Service*

***Host* Status**	***Service* status**
Unreachable (u)	*Unknown (u)*
Down (d)	*Critical (c)*
Recovered (r)	*Warning (w)*

2.11 Nagios *Add-ons*

Nagios *add-ons* merupakan paket-paket tambahan yang bertujuan untuk membantu aktifitas Nagios dalam melakukan pemantauan jaringan. Sama seperti Nagios, paket-paket tambahan ini bersifat *open source* dan prosesnya bergantung dari Nagios. Dengan adanya paket-paket ini, efektifitas dan efisiensi Nagios dalam memantau jaringan akan semakin meningkat.

2.11.1 Pnp4nagios

Pnp4nagios adalah sebuah *add-ons* yang menganalisa data kinerja yang diberikan oleh *plugin* dari Nagios dan menyimpannya secara otomatis kedalam *RRD-Database (Round Robin Database)* melalui *RRDTool*. Data tersebut kemudian dipresentasikan dalam bentuk grafik & *chart*, sehingga memudahkan bagi *administrator* jaringan untuk menilai kinerja dari jaringan [8].

Pada sistem monitoring yang dirancang, *pnp4nagios* akan berperan sebagai *graphic presentation* dari data keadaan jaringan yang dikumpulkan oleh Nagios. Data yang dikumpulkan dari Nagios akan dimasukkan terlebih dahulu ke dalam *database* RRD (*Round Robin Database*) menggunakan *RRDTool* kemudian ditampilkan dalam bentuk grafik oleh *pnp4nagios*. Grafik yang ditampilkan oleh *pnp4nagios* bisa diatur berdasarkan waktu *interval*. Waktu *interval* yang disediakan adalah dalam satuan jam, minggu, bulan, dan tahun. Fitur ini dapat

membantu *administrator* jaringan dalam menentukan kinerja jaringan selama waktu *interval* yang ditentukan [8].

Setelah mendapatkan data, admin jaringan kemudian dapat membuat perencanaan dan pengembangan. Jika data kinerja jaringan yang didapat ternyata kurang memuaskan atau dibawah standar kinerja jaringan, maka dapat direncanakan untuk melakukan *maintenance* perangkat jaringan atau peremajaan perangkat jaringan. Fitur yang lain dari *pnp4nagios* ini adalah dapat membuat laporan data dalam bentuk *file pdf*. Fitur ini berguna bagi admin jaringan untuk membuat laporan keadaan jaringan dalam bentuk tercetak. Data yang ada dalam *graphic presentation* dari *pnp4nagios*, akan sesuai dengan objek yang sedang diperiksa oleh Nagios. Contohnya, jika objek yang diperiksa adalah PING, data yang akan tampilan adalah RTA (*Round Time Around*) dan berapa banyak *packet lost*-nya [8].

2.11.2 NSClient++

NSClient++ adalah aplikasi sederhana yang *powerfull* dan sebuah *daemon* yang aman untuk monitoring sistem operasi Windows. NSClient++ merupakan jembatan penghubung antara server monitoring dengan *host* yang akan dimonitoring. Untuk melakukan monitoring *private service* pada OS Windows, seperti *Memory usage*, CPU *load*, *disk usage*, *service states*, dan sebagainya dilakukan penambahan agent NSClient++ pada server dengan OS Windows tersebut. Tanpa penambahan agent tersebut, hanya dapat dimonitor *service public* seperti HTTP, FTP, POP3, dan sebagainya [2].

Untuk memonitoring OS Windows, seorang admin akan menginstal *add-ons* NSClient++ pada mesin Windows dan menggunakan *check_nt plugin* untuk berkomunikasi dengan NSClient ++ add-on. *Check_nt plugin* harus sudah diinstal pada Nagios [2].

2.12 SMS *Gateway*

SMS *Gateway* adalah sebuah perangkat lunak yang menggunakan bantuan komputer dan memanfaatkan teknologi seluler yang diintegrasikan agar mampu

mendistribusikan pesan-pesan yang dihasilkan lewat sistem informasi melalui media SMS yang ditangani oleh jaringan seluler. SMS ditangani oleh jaringan melalui suatu layanan data *text* atau SMS *Center* (SMSC) yang berfungsi menyimpan dan meneruskan pesan dari sisi pengirim ke sisi penerima.

SMS merupakan sebuah layanan yang bersifat *non real-time* dengan sebuah pesan singkat dapat dikirim ke suatu tujuan, tidak mempedulikan aktif tidaknya nomor tujuan. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin pengiriman dari suatu pesan hingga sampai ke tujuan.

2.13 Gammu SMS *Gateway*

Gammu merupakan salah satu pustaka atau *library opensource* yang dibuat sebagai *gateway* antara *handphone* dengan perangkat komputer. Pengembangan Gammu awalnya dari pendahulunya yaitu Gnooki yang dari segi konsep masih sangat sederhana dan rumit. Namun pada Gammu proses instalasi dan penerapan pembangunan aplikasi SMS semakin mudah. Gammu dibuat menggunakan gabungan bahasa pemrograman Phyton dan C. Ada juga versi lain yang telah dikemas menjadi aplikasi jadi bernama Wammu yang dibangun menggunakan bahasa C++. Gammu dapat digunakan disemua *operating system*. Baik itu Linux, Unix, Windows. Namun setiap *operating system* memiliki proses instalasi yang berbeda-beda [7].

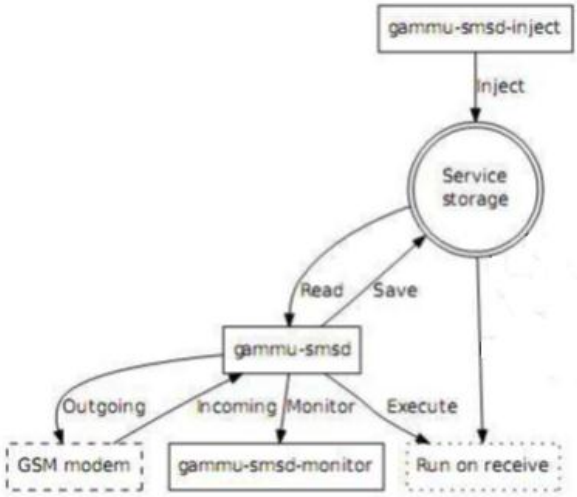
Cara kerja Gammu SMS *gateway* untuk cara pertama, yaitu dengan menggunakan *command inject* yang disediakan oleh Gammu, caranya adalah masuk ke folder atau direktori Gammu melalui *command promp* dimana terdapat *file* bernama *gammu-smsd-inject*, lalu ketikan perintah berikut ini kemudian tekan Enter [7].

gammu-smsd-inject - c pathtoconfigfile TEXT notujuan – “pesan”

Dengan file *pathtoconfigfile* adalah *path* direktori *file* SMSDRC Gammu. Cara yang kedua, yaitu dengan cara menyisipkan sebuah *record* ke dalam tabel *outbox*. Yang perlu diperhatikan dalam tabel *outbox* hanyalah *field Destination*

Number dan *Text Decoded. Field Destination Number* itu nanti akan diisi dengan nomor HP tujuan dan *Text Decoded* diisi dengan pesan sms yang akan dikirimkan.

Secara garis besar alur kerjanya dapat dilihat melalui gambar berikut ini :



Gambar 2.18 Alur Kerja Gammu

Pada gambar 2.18 diatas dapat dijelaskan sebagai berikut :

1. *Gammu-Smsd*

   Gammu-smsd adalah program yang secara periodik memeriksa modem untuk menerima pesan, menyimpannya pada tempat penyimpanan, dan kemudian mengirimkan antrian pesan (*message enqueued*) pada tempat penyimpanan.

2. *Gammu-Smsd-Inject*

   Gammu-smsd-inject adalah program yang memasukkan pesan ke dalam *gammu* SMS *daemon*, yang kemudian akan dikirimkan dengan menggunakan modem.

3. Modem

   Modem adalah perangkat keras yang sering digunakan untuk komunikasi pada komputer. Data dari komputer yang berbentuk sinyal digital diberikan kepada modem untuk diubah menjadi sinyal analog. Ketika modem menerima data dari luar berupa sinyal analog, modem mengubahnya kembali ke sinyal digital supaya dapat diproses lebih lanjut oleh komputer.

4. *Gammu-Smsd-Monitor*

   Gammu-smsd-monitor adalah program yang memonitor keadaan gammu SMS *daemon*. Program ini menampilkan informasi secara periodik tentang nomor telepon dan pesan yang telah diproses.

5. *Run-On-Receive*

   Mengeksekusi program setelah menerima pesan. Parameter dengan pengidentifikasi pesan yang diterima dilampirkan pada baris perintah. Pengidentifikasi tergantung pada layanan *backend* yang digunakan. Biasanya itu adalah ID dari baris yang dimasukkan ke dalam *database backend* atau nama *file* untuk *backend* berbasis *file*.

2.14 Apache *Web Server*

Apache adalah *web server* yang dapat dijalankan pada banyak sistem operasi (Unix, BSD, Linux, dan Microsoft Windows) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas ini adalah HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan *server* menjadi mudah. Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.