

# Penerapan Spanning Tree Protocol (STP) pada Jaringan dengan Bridge Loop

Aries Tri Sutrisno K.A  
13515116

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
<sup>1</sup>author@itb.ac.id

Jaringan komputer, dalam komunikasi modern tidak dapat dipungkiri menjadi basis yang sangat krusial dalam menunjang keberjalanan pertukaran informasi dalam skala besar. Kemajuan di bidang ilmu komputer dan teknologi perangkat keras kian mendukung untuk pembuatan jaringan yang kompleks dan komperhensif. Namun, pembangunan jaringan yang kompleks justru menimbulkan masalah baru, salah satu diantaranya adalah bridge loop. Bridge loop menjadi masalah yang krusial apabila tidak ditangani karena akan mengakibatkan paket tidak pernah sampai serta menimbulkan beban pada jaringan dengan memakan bandwidth sehingga dalam skala yang lebih luas akan berujung pada kegagalan infrastruktur jaringan. Untuk mengatasi hal tersebut, diperlukan sebuah mekanisme terstruktur bagi perangkat jaringan untuk menghindari bridge loop yang disebut sebagai Spanning Tree Protocol.

**Keywords:** jaringan komputer, bridge loop, Spanning Tree Protocol,

## I. PENDAHULUAN

Jaringan, dalam konteks ilmu komputer adalah susunan dua atau lebih komputer yang terhubung satu sama lain melalui suatu media sehingga mampu bertukar informasi atau *resources*. Saat ini, jaringan komputer adalah basis komunikasi manusia di seluruh dunia tanpa terkecuali. Komunikasi modern tidak mungkin dapat dilakukan tanpa dilibatkannya jaringan komputer karena kebutuhan akan perpindahan informasi dalam waktu yang singkat hanya dapat dilakukan dalam jaringan komputer dengan teknologi yang kian mutakhir dari zaman ke zaman. Lebih lanjut, jaringan komputer secara umum dapat diklasifikasikan berdasarkan berbagai macam aspek. Berdasarkan jangkauan geografisnya, jaringan komputer dibagi menjadi *Local Area Network (LAN)*, *Metropolitan Area Network (MAN)*, dan *Wide Area Network (WAN)*. Berdasarkan distribusi datanya, jaringan komputer dibagi menjadi *centralized system* dan *distributed system*. Berdasarkan media yang menghubungkan jaringan tersebut, jaringan komputer secara umum dapat dibagi menjadi *wired network* dan *wireless network*. Dalam aspek lain, terdapat klasifikasi yang membagi jaringan menjadi berbagai macam, seperti berdasarkan peranan komputer dalam jaringan ataupun berdasarkan topologi jaringan itu sendiri. Klasifikasi jaringan komputer yang secara umum dikenal juga adalah pembagian menjadi jaringan

*intranet*, *internet*, dan *extranet*, dimana *internet* adalah istilah yang sudah tidak asing lagi bagi penduduk dunia karena akses jaringannya yang global dan memanfaatkan sistem interkoneksi jaringan yang disebut dengan *Web*.



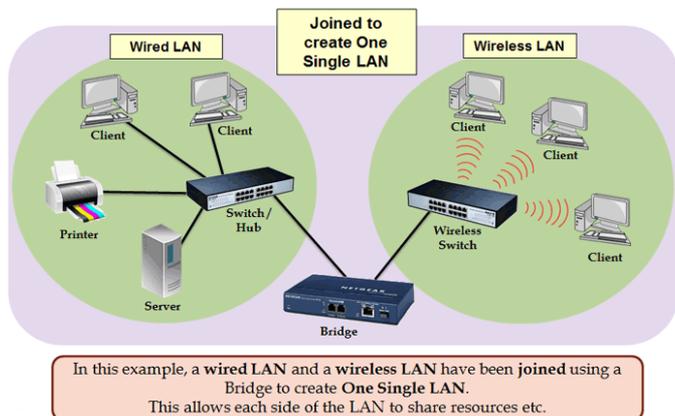
Gambar 1. Visualisasi jaringan sebagai basis komunikasi modern

Berbicara tentang jaringan, selain klasifikasinya yang berbagai macam, sebagai metode pembelajaran terkait fungsionalitasnya jaringan juga dapat ditinjau berdasarkan komponen yang menyusunnya. Jika pada paragraf sebelumnya secara mendasar jaringan didefinisikan sebagai susunan dari komputer-komputer, maka dalam definisi lanjutan komputer-komputer itu akan memiliki peranan yang khusus dalam sebuah jaringan. Secara umum, sebuah komputer dalam jaringan disebut dengan istilah *node*. Lalu, *node-node* pada jaringan inilah yang disebut memiliki peranan yang berbeda-beda.

*Node* sebagai komponen jaringan secara fungsional dapat dibagi menjadi beberapa peranan seperti *client*, *server*, *switch*, *bridge*, *hub*, *router*, *repeater*, dan lain sebagainya, namun yang akan dibahas pada makalah kali ini adalah *bridge*. Sebelum

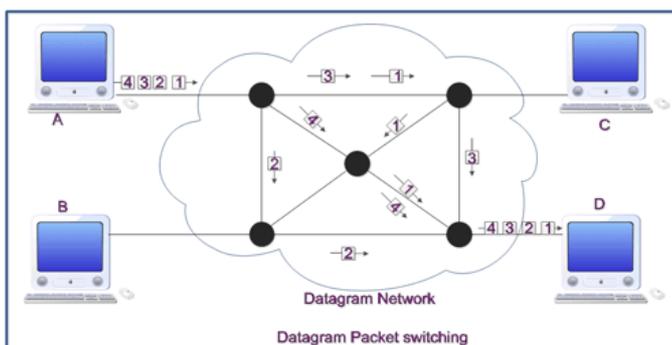
dapat mendefinisikan *bridge*, kita harus terlebih dahulu mengerti tentang *switch* karena pengertian kedua komponen ini tidak dapat dipisahkan satu sama lainnya.

*Switch* adalah sebuah *device* yang dapat menerima banyak *input* dan dapat meneruskannya ke satu atau lebih *output*. Istilah *switch* digunakan sebagai perangkat yang menyambungkan *node-node* lain (biasanya berupa *end node*<sup>1</sup>) tanpa mengurangi performansi *network* tersebut. Istilah *switch* juga lebih umum digunakan sebagai perangkat pada suatu jaringan lokal dengan lingkup kecil.

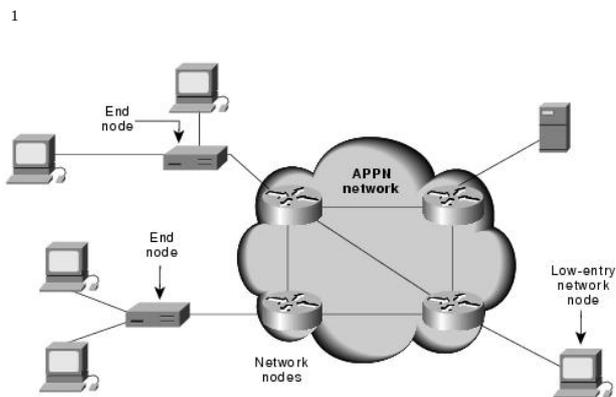


Gambar 2. Visualisasi susunan *switch* dan *bridge* pada LAN

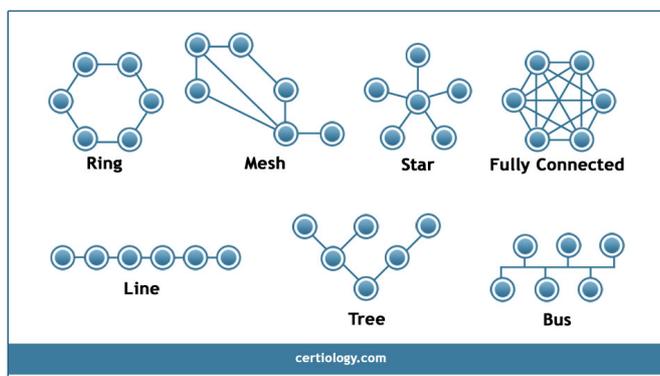
Sedangkan *bridge*, adalah *switch* yang digunakan untuk menghubungkan lebih dari satu *Local Area Network (LAN)*. Kedua perangkat ini dalam menjalankan fungsinya meneruskan paket dari *source* menuju *destination* memiliki konsep yang disebut dengan *forwarding mechanism* yang mencakup antara lain *datagram*, *virtual circuit*, dan *source routing*. Untuk itu, terkait dengan *forwarding mechanism* yang dimiliki oleh *switch* dan *bridge*, sebuah *switch* ataupun *bridge* memiliki sebuah *datagram* yang berfungsi untuk menyimpan informasi jalur yang harus ditempuh oleh sebuah paket untuk mencapai *destination*-nya. Apabila sebuah paket transit pada sebuah *switch* atau *bridge*, maka *switch* atau *bridge* akan melihat informasi kemana paket tersebut akan di-*forward*. Jika pesan tersebut singgah pada *bridge*, maka *bridge* akan melakukan *packet forwarding* kepada *switch* yang bertanggungjawab untuk meneruskan paket tersebut. Begitu pula dengan *switch*, akan melakukan *packet forwarding* sesuai dengan informasi yang dimiliki pada *datagram* apakah menuju *end node* atau menuju *bridge*.



Gambar 3. Contoh skema *forwarding mechanism*



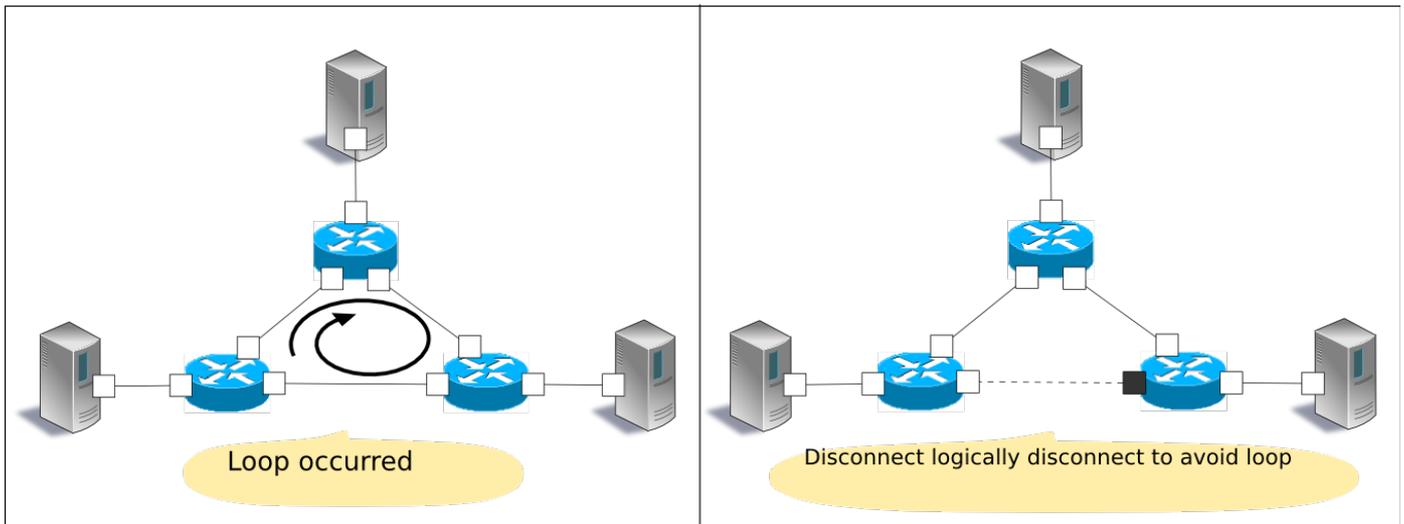
Gambar 4. Visualisasi *node* pada jaringan



Gambar 5. Berbagai macam topologi jaringan

Setelah memahami pengertian tentang jaringan beserta komponennya, dalam hal ini *bridge*, kita juga harus memahami (seminimal mungkin) tentang topologi jaringan. Dalam topologi jaringan, dapat dirangkai sebuah arsitektur jaringan yang memiliki *loop* sehingga dapat mengakibatkan kemungkinan terjadinya perputaran paket yang tiada henti. Pada praktiknya, *bridge loop* bermanfaat untuk menyediakan jalur alternatif apabila status sebuah *switch* atau *bridge* pada jaringan mengalami *down*. Namun, arsitektur jaringan yang memiliki *bridge loop* ini juga menjadi kelemahan dari *forwarding mechanism* yang dimiliki oleh *bridge*. Hal itu terjadi karena sebuah paket yang transit ke dalam sebuah *bridge*, dapat ditransmisikan ke lebih dari satu jalur dalam sebuah jaringan. Apabila sebuah paket terus menerus ditransmisikan tanpa tujuan yang jelas, maka paket tersebut akan terus beredar (mengalami *endless loop*) di jaringan yang justru akan menjadi beban pada jaringan tersebut. Untuk mengatasi permasalahan tersebut maka diperlukan sebuah

<sup>1</sup>Cara yang umum untuk mendefinisikan *end node* erat berkaitan dengan keamanan jaringan yang kompleks dan komperhensif serta interface dari perangkat keras yang digunakan. Pakar IT cenderung mengartikan *end node* sebagai *personal computer* yang umum digunakan oleh *end user* yang pada arsitektur jaringan biasanya menempati ujung-ujung topologi.



Gambar 6. Contoh arsitektur jaringan dengan *bridge loop* dan mekanisme lojik sederhana untuk mengatasinya

protokol agar sebuah paket mendapatkan jalur yang deterministik walaupun beredar pada arsitektur jaringan yang memiliki *bridge loop*.

Apakah itu protokol? Protokol, dalam konteks jaringan komputer, adalah serangkaian standar formal dan kebijakan yang disepakati bersama, berupa aturan-aturan dan prosedur yang terdefinisi untuk dua atau lebih perangkat yang melakukan komunikasi melalui jaringan komputer. Protokol pada jaringan komputer sendiri diadakan untuk menjaga komunikasi yang efektif, aman, dan teratur dalam jaringan. Lalu, protokol seperti apa yang dapat menjawab permasalahan *bridge loop* pada jaringan komputer? Protokol tersebut disebut dengan *Spanning Tree Protocol*.

## II. LANDASAN TEORI

Topologi jaringan kian berkembang dari waktu ke waktu terutama pada tingkat kompleksitas yang dapat dibangun pada suatu arsitektur jaringan. Tidak dapat dipungkiri bahwa aksesibilitas yang tinggi dapat dicapai dengan mengintegrasikan interkoneksi dari lebih dari satu segmen ke dalam suatu jaringan yang lebih besar. Hal tersebut dilakukan salah satunya untuk menjamin ketersediaan mekanisme agar pertukaran data dan informasi pada jaringan dapat diandalkan meskipun terjadi kegagalan pada suatu perangkat di dalam jaringan.

Namun, arsitektur yang kompleks tentu memiliki harga yang harus dilunasi untuk mencapai kondisi yang ideal. Membangun arsitektur jaringan dengan *multiple links* akan sangat membangkitkan *redundancy* yang menjadi masalah baru bagi jaringan tersebut. *Redundancy* tersebut lalu akan menyebabkan kemungkinan paket yang melalui jaringan tersebut mengalami *loop* dan beredar selamanya pada jaringan. Tentu saja sebuah paket yang terus-menerus beredar dalam jaringan tidak akan

merusak keberadaan jaringan. Namun, lama kelamaan paket tersebut akan menjadi beban pada jaringan. Bayangkan apabila paket tersebut tidak hanya satu melainkan berasal dari *broadcast message* yang pada praktiknya sering dikirimkan pada jaringan yang berbasis *switch*. Sekumpulan paket yang beredar tersebut lalu akan memenuhi *bandwidth* dari jaringan sehingga lambat laun akan mengakibatkan kegagalan infrastruktur jaringan.

Beruntunglah berkat kemajuan pengetahuan di bidang jaringan, hal tersebut dapat dihindari apabila dalam jaringan terdapat mekanisme untuk mendeteksi adanya *loop* tersebut. *Institute of Electrical and Electronics Engineer (IEEE)* telah menerbitkan standarisasi untuk mencegah *bridge loop* pada *data link layer* dalam jaringan pada *IEEE 802.1D*. Standarisasi yang merupakan solusi dari *bridge loop* tersebut lalu dikenal dengan sebutan *Spanning Tree Protocol (STP)*.

*Spanning Tree Protocol* adalah sebuah protokol pada *data link layer* (karena *bridge* bekerja pada *data link layer* pada jaringan) yang memungkinkan sebuah jaringan dengan arsitektur yang memiliki *bridge loop* untuk memiliki jalur-jalur yang deterministik untuk setiap nodenya dengan menonaktifkan hubungan pada jalur tertentu sehingga membangkitkan sebuah upagraf dari arsitektur jaringan tersebut. *Spanning Tree Protocol* berjalan saat jaringan akan membangun *datagram* untuk seluruh *switch* dan *bridge* yang aktif pada jaringan tersebut. Hal ini dimungkinkan karena konsep dasarnya untuk membangun *datagram* pada *switch* dan *bridge* memerlukan pertukaran data serta informasi yang terstruktur dari suatu *node* dengan *node* yang lain, sehingga *Spanning Tree Protocol* menyediakan aturan-aturan dan prosedur yang disisipkan saat proses pembangunan *datagram* tersebut.

### III. PEMBAHASAN

#### A. Root Bridge Election

Spanning Tree Protocol memungkinkan switch dan bridge untuk berkomunikasi dengan mekanisme yang khusus agar memiliki struktur yang sama dalam menjalankan protokol ini. Untuk mencapai kesepakatan tersebut, seluruh bridge memiliki suatu standar pesan data yang ditransmisikan selama sebelum pembangunan *datagram* yang disebut dengan *Bridge Protocol Data Unit (BPDU)*.

BPDU ditransmisikan secara *multicast* pada jaringan di *data link layer*. Dalam menentukan *bridge* yang akan dijadikan acuan (*root bridge*), setiap *bridge* melakukan pertukaran BPDU satu sama lain dan menentukan *root bridge* berdasarkan *Bridge ID* yang terkecil yang dimiliki oleh salah satu *bridge*. *Bridge ID* sendiri ditentukan berdasarkan *header* dari *MAC address* dan *bridge priority* milik *bridge* tersebut.

#### B. Root Port Election

Setelah seluruh *bridge* menentukan *bridge root* sebagai acuan dari jaringan, maka setiap *non-root bridge* harus menentukan *root port* sebagai akses menuju *root bridge* tersebut. *Root port* didefinisikan sebagai sebuah *port* yang terhubung ke jalur terbaik menuju *root bridge*. Pemilihan *root port* tersebut ditentukan dengan menggunakan *root path cost* yang terdapat pada BPDU. Lalu, perhitungan tersebut secara singkat mencakup beberapa aspek berikut.

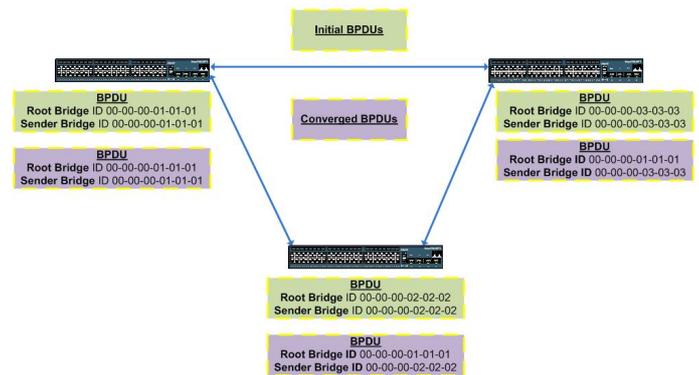
- Setiap *switch port* memiliki *path cost*-nya masing-masing berdasarkan *bandwidth* dari *port* tersebut.
- Semakin tinggi *bandwidth* yang dimilikinya, maka semakin kecil *cost* yang dibutuhkan apabila melalui *port* tersebut.
- Untuk setiap BPDU yang transit pada perangkat, maka *path cost* tersebut ditambahkan ke dalam *path cost field* pada BPDU sehingga akan diakumulasikan jumlah *path cost* yang dilalui suatu jalur tertentu oleh BPDU tersebut.
- Langkah terakhir adalah menghitung setiap *port* pada perangkat dan menentukan *port* dengan *path cost* terkecil, lalu menjadikan *port* tersebut sebagai *root port*.

#### C. Designated Port Election

Langkah terakhir yang dilakukan pada *Spanning Tree Protocol* adalah menentukan *designated port* untuk setiap segmen pada jaringan. Pemilihan *designated port* ini sendiri berdasarkan pada *root port* yang telah ditentukan pada langkah sebelumnya. Apabila dalam menentukan *designated port* menuju *root bridge* terjadi kesamaan *path cost* antara dua atau lebih *switch*, maka *switch* dengan *sender ID* yang lebih kecil yang akan dipilih dan *port* yang terhubung dengan *switch* tersebut yang akan menjadi *designated port* pada segmen jaringan tersebut.

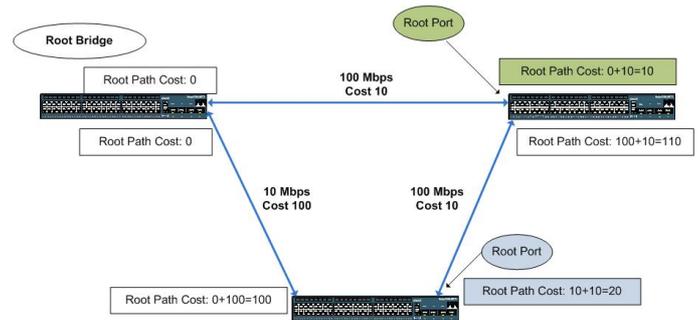
Lalu, *port* yang bukan merupakan *root port* atau *designated port* akan berpindah *state* menjadi *blocking state* dan tidak akan dapat menerima ataupun mentransmisikan paket sehingga memastikan bahwa jaringan tersebut bebas dari *bridge loop*.

#### Root Bridge Election



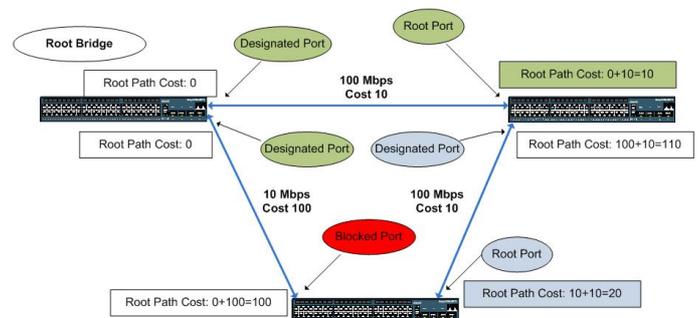
Gambar 7. Skema pemilihan root bridge pada STP

#### Root Port Election



Gambar 8. Skema pemilihan root port pada STP

#### Designated Port Election



Gambar 9. Skema pemilihan designated port pada STP

## IV. KESIMPULAN

Jaringan komputer, dalam komunikasi modern tidak dapat dipungkiri menjadi basis yang sangat krusial dalam menunjang keberjalanan pertukaran informasi dalam skala besar. Kemajuan di bidang ilmu komputer dan teknologi perangkat keras kian mendukung untuk pembuatan jaringan yang kompleks dan komperhensif. Namun, pembangunan jaringan yang kompleks justru menimbulkan masalah baru, salah satu diantaranya adalah *bridge loop*. *Bridge loop* menjadi masalah yang krusial apabila tidak ditangani karena akan mengakibatkan paket tidak pernah sampai serta menimbulkan beban pada jaringan dengan memakan *bandwidth* sehingga dalam skala yang lebih luas akan berujung pada kegagalan infrastruktur jaringan.

Menanggapi hal tersebut, *IEEE* memberikan solusi yang terstandardisasi untuk menangani permasalahan *bridge loop* ini yang tertuang pada *IEEE 802.1D*. Solusi yang terstandardisasi tersebut lalu disebut dengan *Spanning Tree Protocol* karena protokol ini merentang upagraf dari arsitektur jaringan yang memiliki *bridge loop*.

*Spanning Tree Protocol* bekerja dengan mekanisme dasar menonaktifkan jalur-jalur yang menyebabkan *loop* pada jaringan. Secara teknis, seluruh perangkat akan berkomunikasi menggunakan pesan terstruktur dari STP yang disebut dengan *Bridge Protocol Data Unit (BPDU)*. Sebelum proses pembangunan *datagram* oleh perangkat-perangkat jaringan, seluruh *switch* dan *bridge* akan saling berkomunikasi untuk menentukan *root bridge*, *root port*, dan *designated port*, serta menonaktifkan *port-port* yang tidak termasuk di dalamnya. Setelah upagraf dari jaringan dibentuk, barulah seluruh perangkat jaringan diperbolehkan untuk saling bertukar informasi dalam membangun *datagram* untuk kepentingan *forwarding mechanism*.

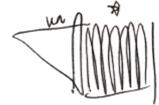
## REFERENCES

- [1] <https://www.computerhope.com/jargon/n/network.htm> Diakses pada 3 Desember 19:30
- [2] <http://cisco2960.over-blog.com/2014/02/internet-intranet-and-extranet.html> Diakses pada 3 Desember 20:03
- [3] <http://www.nesabamedia.com/pengertian-manfaat-dan-macam-macam-jaringan-komputer/> Diakses pada 3 Desember 20:08
- [4] <https://www.techopedia.com/definition/26122/end-node> Diakses pada 3 Desember 22:54
- [5] <https://www.techopedia.com/definition/12938/network-protocols> Diakses pada 3 Desember 23:07
- [6] <https://www.pluralsight.com/blog/software-development/spanning-tree-protocol-tutorial> Diakses pada 4 Desember 10:48
- [7] <http://www.certiology.com/wp-content/uploads/2014/04/Network-Topology.jpg> Diakses pada 4 Desember 11:30
- [8] [https://osrg.github.io/ryu-book/en/html/\\_images/fig16.png](https://osrg.github.io/ryu-book/en/html/_images/fig16.png) Diakses pada 4 Desember 11:30
- [9] <http://ecomputernotes.com/images/Datagram-packet-switching.gif> Diakses pada 4 Desember 11:30
- [10] [https://www.ictlounge.com/Images/bridge\\_example\\_large.gif](https://www.ictlounge.com/Images/bridge_example_large.gif) Diakses pada 4 Desember 11:30
- [11] <http://docwiki.cisco.com/w/images/b/bb/CT844105.jpg> Diakses pada 4 Desember 11:30
- [12] <http://www.optinetsystems.com/wp-content/uploads/2013/05/network-management-edit.jpg> Diakses pada 4 Desember 11:30

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2017



Aries Tri Sutrisno KA  
13515116