

ANALISIS KINERJA JARINGAN SWITCHING BATCHER-BANYAN

Chairunnisa FR. Tanjung, M. Zulfin

Konsentrasi Teknik Telekomunikasi, Departemen Teknik Elektro
Fakultas Teknik Universitas Sumatera Utara (USU)
Jl. Almamater, Kampus USU Medan 20155 INDONESIA
e-mail:chairunnisan@gmail.com

Abstrak

Batcher banyan network adalah jaringan *self-route* yang mampu bergerak serentak pada routing paket data dalam slot waktu dari satu set terminal masukan untuk set terminal output tanpa kendali terpusat. Routing yang melalui network tersebut di tentukan dari alamat yang terdapat dalam header pada setiap data. Dalam penelitian ini dianalisis kinerja jaringan *switching Banyan* yang menambahkan *Batcher* didepannya. *Probabilitas blocking* pada jaringan *Banyan*, untuk *input/output* 2,4,8,16,32,64,128,256 maka jumlah tingkat *switching* yang dibutuhkan adalah 1,3,5,7,9,11,13,15. Untuk hasil *crosspoint* pada *Banyan switch* untuk *input/output* 2,8,16,32 maka nilai *crosspoint* adalah 1,12,32,80, untuk *Batcher switch* dengan *input/output* 2,8,16,32 maka hasil *crosspoint* adalah 1,24,80,240 dan untuk *Batcher-banyan switch* dengan *input/output* 2,8,16,32 maka hasil *crosspoint* adalah 2,54,112,320. Apabila ditambahkan dengan menggunakan jaringan *Batcher* di depan jaringan *Banyan* maka *Probabilitas blocking* (P_b) = 0.

Kata Kunci: *Banyan, Batcher, Batcher-Banyan switch*

1. Pendahuluan

Batcher banyan network adalah jaringan *self-route* yang mampu bergerak serentak pada routing paket data dalam slot waktu dari satu set terminal masukan untuk set terminal output tanpa kendali pusat. Jaringan *batcher-banyan* ini terdiri dari *batcher* menyortir jaringan setelah itu diikuti oleh *route* jaringan. Dimana fungsi dari jaringan penyortir *batcher* adalah untuk mengatur paket yang masuk pada urutan menaik ataupun menurun sesuai dengan alamatnya.

Dalam *route* paket data dari setiap terminal masukan ke terminal tujuan mungkin akan mengalami kemacetan pada paket internalnya. Hal itu dapat terjadi bila dua atau lebih paket data disalurkan melalui *link internal* yang sama pada waktu yang sama pula. Tetapi dengan adanya jaringan *banyan* ini secara internal *non-blocking* terjadi jika dalam slot waktu tertentu dan tidak lebih dari satu paket masuk maka setiap *output* banyan dan paket akan diatur menaik atau menurun ketika tiba di *input banyan*. Oleh karena itu sangatlah mungkin untuk membangun jaringan *non-blocking* dan menggabungkan *batcher* mensortir jaringan dengan *banyan* jaringan routing.

2. Switching

Komponen utama dari sistem *switching* atau sentral adalah seperangkat *circuit* masukan dan keluaran yang disebut dengan *inlet* dan *outlet*. Fungsi utama dari sistem *switching* adalah membangun jalan listrik diantara sepasang *inlet* dan *outlet* tertentu, dimana *hardware* yang digunakan untuk membangun koneksi seperti itu disebut matriks *switching* atau *switching network*[1]. Maka untuk menghitung tingkat elemen *switching* (S) dapat menggunakan Persamaan (1) dan untuk menghitung *probabilitas blocking* (P_b) dapat menggunakan Persamaan (2).

$$S = 2 \log_2 N - 1 \dots\dots\dots(1)$$

$$P_{k+1} = 1 - \left(1 - \frac{pk}{b}\right)^b \dots\dots\dots(2)$$

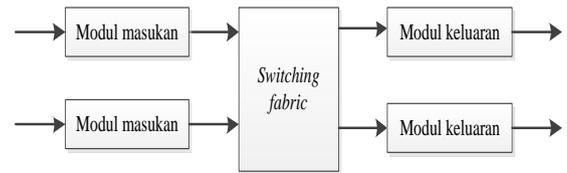
Secara umum digunakan tiga teknik *switching*, yaitu *circuit switching*, *packet switching*, dan *message switching*. Tetapi yang sering digunakan adalah *circuit switching* dan *packet switching*. Pada *circuit switching*, jalur antara sumber dan tujuan harus telah disediakan sebelum komunikasi terjadi dan koneksi ini harus tetap dijaga sampai pesan mencapai tujuannya. Setiap koneksi yang dibangun melalui jaringan

circuit switching mengakibatkan dibangunnya kanal komunikasi fisik diantara terminal sumber dengan terminal tujuan. Kanal komunikasi ini digunakan secara khusus selama terjadi koneksi. Jaringan *circuit switching* juga menyediakan kanal dengan laju yang tetap.

Pada hubungan *circuit switching*, koneksi biasanya terjadi secara fisik bersifat *point to point*. Kerugian terbesar dari teknik ini adalah penggunaan jalur yang bertambah banyak untuk penambahan jumlah node sehingga biaya akan semakin meningkat dan pengaturan *switching* menjadi sangat kompleks. Kelemahan yang lain adalah munculnya *idle time* bagi jalur yang tidak digunakan, yang akan menambah inefisiensi. *Circuit switching* mentransmisikan data dengan kecepatan yang konstan sehingga untuk menggabungkan dengan jaringan lain yang berbeda kecepatan terus akan sulit.

Pemecahan yang baik yang bisa digunakan untuk mengatasi persoalan di atas adalah dengan metode *packet switching*. Dengan pendekatan ini, pesan yang dikirim dipecah-pecah dengan besar tertentu dan pada tiap pecahan data ditambahkan informasi kendali. Informasi kendali ini, dalam bentuk yang paling minim, digunakan untuk membantu proses pencarian *route* dalam suatu jaringan sehingga pesan dapat sampai ke alamat tujuan[2].

Rancangan elemen *switching* yang dibutuhkan adalah rancangan yang dapat meneruskan paket data secara cepat, dapat dikembangkan dengan skala yang lebih besar dan dapat secara mudah untuk diimplementasikan. Suatu elemen *switching* dapat digambarkan sebagai suatu elemen jaringan yang menyalurkan paket data dari terminal masukan menuju terminal keluaran. Kata terminal dapat berarti sebagai suatu titik yang terdapat pada elemen *switching*. Dari pengertian diatas dapat disimpulkan bahwa *switching* adalah proses transfer data dari terminal masukan menuju terminal keluaran. Gambar 1 memperlihatkan elemen *switching* terdiri dari tiga komponen dasar yaitu: modul masukan, *switching fabric*, dan modul keluaran[3].



Gambar 1. Tipe elemen *switching*

Ketiga komponen *switch* tersebut dijelaskan sebagai berikut[3] :

1. Modul masukan

Modul masukan akan menerima paket yang datang pada terminal masukan. Modul masukan akan menyaring paket yang datang tersebut berdasarkan alamat yang terdapat pada *header* dari paket tersebut. Alamat tersebut akan disesuaikan dengan daftar yang terdapat pada *virtual circuit* yang terdapat pada modul masukan. Fungsi ini juga dilakukan pada modul keluaran. Fungsi lain dilaksanakan pada modul masukan adalah sinkronisasi, pengelompokan paket menjadi beberapa kategori, pengecekan *error* dan beberapa fungsi lainnya sesuai dengan teknologi yang ada pada *switching* tersebut.

2. *Switching fabric*

Switching fabric melakukan fungsi *switching* dalam arti sebenarnya yaitu meroutekan paket dari terminal masukan menuju terminal keluaran. *Switching fabric* terdiri atas jaringan transmisi dan elemen *switching*. Jaringan transmisi lain ini bersifat pasif dalam arti bahwa hanya sebagai saluran saja. Pada sisi lain elemen *switching* melaksanakan fungsi seperti *internal routing*.

3. Modul keluaran

Modul keluaran berfungsi untuk menghubungkan paket ke media transmisi dan ke berbagai jenis teknologi seperti kontrol *error*, data *filtering*, tergantung pada kemampuan yang terdapat pada modul keluaran tersebut.

3. *Batcher-Banyan Switch*

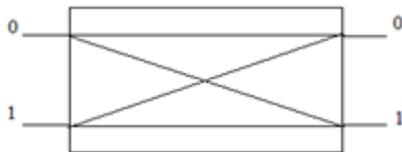
Jaringan *banyan* adalah sebuah jaringan *switching* bertingkat (*Multistage Interconnection Network / MIN*), yang biasanya terdiri dari sejumlah *elemen switching* yang digabungkan ke dalam beberapa tingkat yang diinterkoneksi

oleh seperangkat *link* dengan jalur antara sumber dengan tujuan.

Secara lebih khusus, jaringan banyan $N \times N$ menggunakan elemen-elemen $(N/2)\log N$. Oleh karena itu, jaringan tidak bisa *nonblok*, permutasi *input* ke *output* bisa dibangun dari yang tidak bisa dirutekan secara konkrit dengan jaringan. Oleh sebab itu, *buffer* penghalus harus terletak di dalam jaringan untuk mencapai angka kehilangan paket serendah mungkin. Maka dapat dihitung jumlah nilai pada *crosspoint* untuk *Banyan switch* dengan menggunakan persamaan (1)[4].

$$N_x = 2N \log_2 N[\text{crosspoint}] \leftrightarrow N_x = \frac{N}{2} \log_2 N \dots (3)$$

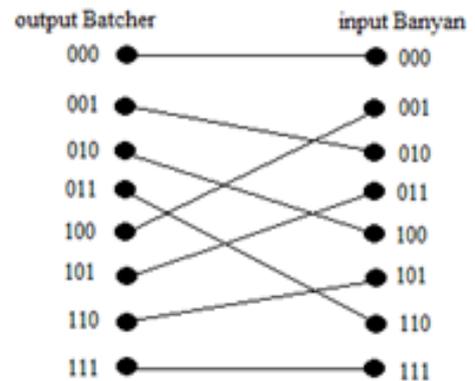
Salah satu karakteristik dari jaringan banyan adalah bahwa jaringan ini mampu melakukan perutean sendiri (*self-routing*), dimana bit-bit alamat keluaran yang terdapat pada *header* paket dapat menentukan sendiri kemana perutean akan dilakukan[5]. Apabila sebuah paket tiba pada masukan jaringan *banyan*, elemen *switching* pertama merutekan paket keluaran sebelah atas jika bit pertama pada alamat tujuan adalah 0 dan merutekan ke keluaran sebelah bawah jika nilainya 1. Elemen *switching* jaringan *banyan* diperlihatkan pada Gambar 2.



Gambar 2. Elemen *switching* jaringan *banyan*

Sebelum memasuki jaringan *Banyan*, paket-paket akan dilewatkan melalui jaringan *shuffle* yang berfungsi menetapkan *port input* baru bagi paket-paket ketika memasuki jaringan *Banyan*[6], kemudian meneruskan paket tersebut melalui *output* bagian atas jika bit tersebut bernilai 0 dan akan melalui *output* bagian bawah jika bit tersebut bernilai 1. Jika ada lebih dari satu paket tiba pada suatu elemen *Banyan* pada waktu yang bersamaan dan bit-nya bernilai sama sehingga harus diteruskan melalui *port output* yang sama pula, maka akan timbul masalah *collision* (tabrakan). Masalah tabrakan ini merupakan tantangan utama dalam perancangan *switch self-routing*. *Banyan network* yang terdiri dari elemen-elemen *switch 2x2* ditata sedemikian rupa sehingga dijamin tidak akan terdapat

collision pada jalur sejumlah paket yang telah terurut naik berdasarkan nomor *port* tujuannya[7]. Pola link *shuffle* adalah seperti pada Gambar 3[6].



Gambar 3. Pola link *Shuffle*

Meskipun *Banyan network* dapat menemukan jalur tanpa ada tabrakan, hal itu dapat terjadi jika syarat keterurutan paket (berdasarkan nomor *port output*) terpenuhi. Untuk mengeliminasi syarat ini, diperlukan ‘sesuatu’ yang ditambahkan sebelum *Banyan network* untuk mengurutkan terlebih dahulu paket berdasarkan nomor *port output*-nya. *Batcher network* adalah *network* yang dirancang untuk melakukan hal seperti itu. *Batcher network* diletakkan didepan *Banyan network*, maka akan didapat *Batcher-banyan fabric* yang memiliki sifat *nonblocking* (tidak akan terjadi tabrakan) selama paket-paket ditujukan pada *port output* yang berbeda-beda.

Elemen *switching* yang terdapat pada *Batcher network* bekerja dengan cara yang sangat berbeda dengan yang terdapat pada *Banyan network*. Elemen *switching* tersebut melakukan perbandingan menyeluruh terhadap angka pada *self-routing header*, kemudian meneruskan paket yang memiliki angka lebih tinggi ke satu *port output* sementara yang lebih rendah ke *port output* yang lain. Jika ternyata kedua nilai sama, maka dilakukan pemilihan secara acak. Ada dua jenis elemen *switch* pada *Batcher network* yaitu elemen yang melakukan pengurutan ‘naik’ (meneruskan paket dengan nomor lebih tinggi ke *port* sebelah atas) dan elemen yang melakukan pengurutan ‘turun’ (meneruskan paket dengan nomor lebih tinggi ke *port* sebelah bawah)[6]. Maka jumlah nilai pada setiap tingkat dan *crosspoint* pada

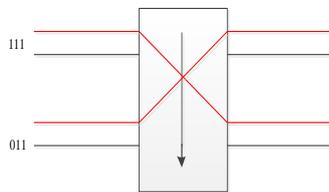
Batcher switch dengan menggunakan persamaan (2) dan (3)[4].

$$M = \sum_{i=1}^{\log_2 N} = \frac{1}{2} \log_2 N [\log_2 N + 1] \dots \dots \dots (4)$$

$$N_x = \frac{NM}{2} \dots \dots \dots (5)$$

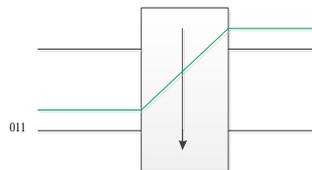
Teknik penyortiran yang dilakukan oleh *batcher* dapat di jelaskan seperti dibawah ini[4]:

1. Jika dua *input* yang muncul, maka *input* yang memiliki alamat tertinggi akan meneruskannya ke *port output* atau dengan mengikuti tanda panah. Penyortiran tahap I diperlihatkan pada Gambar 4.



Gambar 4. Penyortiran tahap I

2. Jika hanya satu alamat *input* yang muncul, maka akan meneruskannya, tetapi tidak mengikuti tanda panah yang ada. Penyortiran tahap II diperlihatkan pada Gambar 5.



Gambar 5. Penyortiran tahap II

Jaringan *batcher-banyan* terdiri dari *batcher* menyortir jaringan, diikuti oleh *route* jaringan *banyan*. Fungsi jaringan penyortiran *batcher* adalah untuk mengatur paket masuk dalam urutan menaik atau menurun sesuai dengan alamat yang akan dituju mereka. Fungsi jaringan *banyan* adalah untuk *route* paket ke terminal *output* yang mereka inginkan.

Jaringan *switch batcher-banyan* yang dikonfigurasi menurut temuan sekarang terdiri dari empat *input* dan empat *output*. *Input* ke dan *output* jaringan *switch batcher-banyan* adalah data. Tahap pertama dari jaringan *batcher-banyan* terdiri dari jaringan penyortir *batcher*. Jaringan penyortir *batcher* dirangkai untuk mensortir data yang datang ke dalam urutan

menaik. Jaringan penyortir *batcher* bisa dirangkai untuk mensortir data yang datang ke dalam urutan menurun. Maka jumlah nilai pada *crosspoints* dengan menggunakan persamaan (4)[4].

$$N_x = \frac{N}{4} \log_2 N [\log_2 N + 3] \dots \dots \dots (6)$$

4. Kinerja *Batcher-Banyan switch*

4.1 Perhitungan pada *banyan switch, batcher switch dan batcher-banyan switch*

Perhitungan yang di lakukan disini adalah perhitungan *Banyan switch, Batcher switch dan Batcher-Banyan switch* yang dimulai dari topologi 2x2, 8x8, 16x16, 32x32. Maka untuk perhitungan pada setiap tingkat dan *crosspoint* pada *banyan switch, batcher switch dan batcher-banyan switch* dihitung dengan persamaan (2),(3) dan (4). Untuk hasil yang didapat diperlihatkan pada Tabel 1.

Tabel 1. Nilai *Banyan switch, Batcher switch dan Batcher-banyan switch* pada topologi 2x2, 8x8,16x16 dan 32x32

Log ₂ N (n)	jumlah Input/output (N)	Banyan switch Nilai crosspoint (Nx)	Batcher switch		Batcher-Banyan switch Nilai crosspoint (Nx)
			Nilai setiap Tingkat (M)	Nilai crosspoint (N x)	
1	2	1	1	1	2
3	8	12	6	24	36
4	16	32	10	80	112
5	32	80	15	240	320

4.2 Perhitungan *probabilitas blocking* untuk jaringan *banyan*.

Probabilitas blocking untuk *switching banyan* dengan berbagai nilai *input/output* (2ⁿ) dapat dicari. Perhitungan *probabilitas blocking* untuk berbagai nilai *input/output* jika dihitung dengan Persamaan (1), nilai *probabilitas blocking*-nya seperti Tabel 2. Dan untuk menghitung *probabilitas blocking*-nya dapat dilihat pada Tabel 2 berikut:

Tabel 2. Probabilitas blocking pada jaringan banyan

$\log_2 N$ (n)	Jumlah Input/output (N)	Jumlah switch tiap tingkatan (N/2)	Jumlah tingkat yang dibutuhkan ($2\log_2 N - 1$)	Probabilitas blocking (pb)
1	2	1	1	0,75
2	4	2	3	0,516547
3	8	4	5	0,399249
4	16	8	7	0,327107
5	32	16	9	0,277804
6	64	32	11	0,241803
7	128	64	13	0,214282
8	256	128	15	0,192521

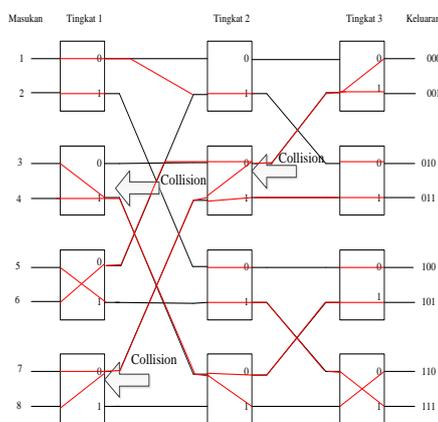
Tabel 2 menunjukkan bahwa terjadi probabilitas blocking pada jaringan banyan. Semakin besar jumlah tingkat elemen switching maka probabilitas blocking yang terjadi juga akan semakin berkurang.

4.3 Aplikasi jaringan banyan dan jaringan batcher

Contoh aplikasi jaringan Banyan ukuran 8x8 dengan koneksi input ke output dengan data yang diberikan seperti berikut:

- 1 – 3 : (000 – 010)
- 5 – 8 : (100 – 111)
- 2 – 5 : (001 – 100)
- 6 – 2 : (101 – 001)
- 3 – 7 : (010 – 110)
- 7 – 1 : (110 – 000)
- 4 – 6 : (011 – 101)
- 8 – 4 : (111 – 011)

Dengan adanya penambahan jaringan Batcher diperlihatkan pada Gambar 7.

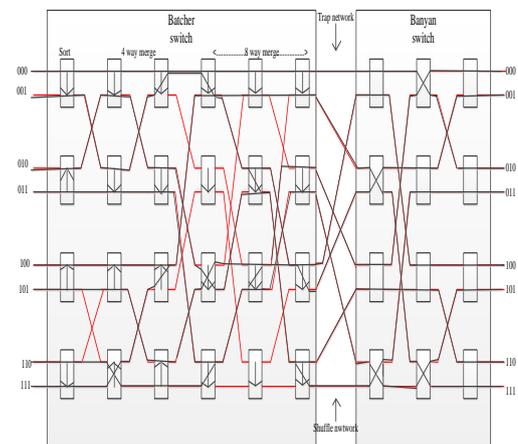


Gambar 7. Contoh aplikasi jaringan banyan 8x8 dengan input/output 1-3, 2-5, 3-7, 4-6, 5-8, 6-2, 7-1, 8-4

Dari gambar 7 dapat diperoleh data yang mengalami collision adalah:

- 3 – 7 : (010 – 110) collision dengan
- 4 – 6 : (011 – 101)
- 6 – 2 : (101 – 001) collision dengan
- 7 – 1 : (110 – 000)
- 7 – 1 : (110 – 000) collision dengan
- 8 – 4 : (111 – 011)

Untuk menghindari collision dan internal blocking yang terjadi pada jaringan Banyan, maka dilakukan penyortiran dengan menambahkan jaringan Batcher di depan jaringan Banyan. Penyortiran oleh jaringan Batcher diperlihatkan pada Gambar 8.



Gambar 8. Penyortiran oleh jaringan Batcher

Tabel 3. Hasil Analisis menggunakan Batcher-Banyan switch

No	Alamat input-output	Collision	Alamat input setelah penambahan jaringan Batcher	Probabilitas blocking (Pb)
1	000 - 010	Tidak ada	000	0
2	001 - 100	Tidak ada	100	0
3	010 - 110	Ada	110	0
4	011 - 101	Ada	011	0
5	100 - 111	Tidak ada	001	0
6	101 - 001	Ada	101	0
7	110 - 000	Ada	010	0
8	111 - 011	Ada	111	0

Tabel 3 menunjukkan bahwa setelah penambahan batcher di depan jaringan banyan maka terlihat probabilitas blockingnya bernilai

nol. Sehingga dapat memperlihatkan bahwa tidak ada terjadinya *probabilitas blocking* dan perubahan alamat *input* dengan penambahan jaringan *Batcher* di depan jaringan *Banyan* dengan mengambil contoh beberapa koneksi *input-output* jaringan *Banyan* yang menyebabkan terjadinya *internal blocking* pada ukuran *switch* 8x8.

5. Kesimpulan

Dari hasil analisis yang dilakukan maka dapat di ambil kesimpulan sebagai berikut:

1. Perubahan jumlah *input/output* pada jaringan *Banyan* sebanding dengan jumlah tingkat elemen *switching* yang dibutuhkan. Dimana jika jumlah *input/output* jaringan *Banyan* bertambah, maka tingkat elemen *switching* pada jaringan *Banyan* akan semakin bertambah. Untuk *input/output* 2,4,8,16,32,64,128,256 dan seterusnya maka jumlah tingkat *switching* yang dibutuhkan adalah 1,3,5,7,9,11,13,15 dan seterusnya.
2. Perubahan jumlah tingkat elemen *switching* berbanding terbalik dengan nilai *probabilitas blocking*. Didapat bahwa *probabilitas blocking* pada jaringan *switching Banyan* terus berkurang jika jumlah tingkat elemen *switching* semakin besar. Untuk jaringan *Banyan* yang dibangun dengan jumlah 13 tingkatan dan 15 tingkatan maka nilai *probabilitas blocking* adalah 0,214282 dan 0,192521.
3. Terjadi perubahan hasil pada jaringan *Banyan* setelah diletakkan jaringan *Batcher* di depannya. Terlihat bahwa jaringan *Batcher* dapat mengatasi masalah *Internal Blocking* yang terjadi pada jaringan *Banyan*. Sehingga tidak ada tubrukan yang terjadi dari *input* ke *output*.

6. UcapanTerimaKasih

Penulis mengucapkan terima kasih kepada Isroi Tanjung, ST dan Fausiah selaku orang tua penulis, Ir. M. Zulfin, MT selaku dosen pembimbing, juga Rahmad Fauzi, ST, MT dan Ali Hanafiah Rambe, ST, MT selaku dosen penguji penulis yang sudah membantu penulis dalam menyelesaikan paper ini, serta teman-teman penulis yang sudah memberikan dukungan selama pembuatan paper ini.

7. DaftarPustaka

- [1]. Zulfin, M. 2008, “Dasar *Switching*: Buku Ajar Teknik Penyambungan”. Medan.
- [2]. Dally, william J. 2004, “*Principles and Practices of Interconnection Networks*”, Morgan Kauffman Publishers: San Francisco.
- [3]. Chen, Thomas M. dan Stephen S. Liu, 1995. “*ATM Switching System*”. Artech House: Michigan University.
- [4]. Dobrota, virgil. 2009, “*Switching and Routing System*”. Technical University of Cluj-Napoca: Rumania.
- [5]. Tarihoran, parlindungan. 2009, “Analisa Kinerja Jaringan *Vertically Stocked Optical Banyan (VSOB)*”. Departemen Teknik Elektro Universitas Sumatera Utara, Medan.
- [6]. Zulfin, M. 2012, “Analisa Kinerja *Switch Batcher-Banyan*”. Edisi I volume I. JiTEKH. STTH Harapan. Medan.
- [7]. Arif, fazmah. 2003, “Jaringan Komputer: Diktat kuliah. Sekolah Tinggi Teknologi Telkom. Bandung. Hal.108-112.