

Nonblocking Minimal Spanning Switch

Abraham Ranardo Sumarsono (13507056)¹⁾

1) Jurusan Teknik Informatika, STEI ITB, Bandung 40116, email: if17056@students.if.itb.ac.id

Abstract – Makalah ini membahas tentang prinsip kerja ‘nonblocking minimal spanning switch’. Prinsip kerja dari ‘nonblocking minimal spanning switch’ itu sendiri memiliki algoritma tersendiri. Algoritma yang digunakan dalam kasus ini digunakan untuk mengatur ‘switching’ di dalam pertukaran jalur telepon. Pertukaran jalur telepon itu sendiri merupakan salah satu proses yang penting dalam dunia telekomunikasi. Sehingga peran dari ‘nonblocking minimal spanning switch’ juga penting di dalam perkembangan dunia telekomunikasi. Prinsip dasar dari algoritma ‘nonblocking minimal spanning switch’ memiliki persamaan dengan prinsip yang digunakan oleh ‘topological sort’.

Kata Kunci: switch/saklar, teori graf, pertukaran jalur telepon, nonblocking, topological sorting

1. PENDAHULUAN

Di dalam bidang telekomunikasi, pertukaran jalur telepon merupakan sebuah sistem yang terdiri dari komponen elektronik yang menghubungkan panggilan telepon dengan penerimanya. Dengan adanya sistem pertukaran jalur telepon yang baik, masyarakat dapat membuat panggilan telepon, membuat koneksi, dan meneruskan informasi auditorial dengan baik.

Pada dulunya, proses pertukaran jalur telepon ini membutuhkan koneksi antara penelepon yang diatur dengan sebuah saklar/switch’ elektromekanikal yang besar dan mahal, yang biasa disebut ‘switch Strowger’. Pada tahun 1940-an dan 1950-an, para insinyur yang bekerja di Laboratorium Bell memulai sebuah penelitian ilmiah tentang bagaimana cara mengurangi ukuran serta harga bahan pada ‘switch’ Strowger tersebut. Hal ini dibutuhkan untuk meningkatkan tingkat efisien dan efektif dalam proses pertukaran jalur telepon itu sendiri.

Pada dasarnya, ‘switch’ telepon memiliki properti matematika, yakni untuk setiap satu masukan ke dalam ‘switch’, terdapat persis satu keluaran khusus untuk masukan tersebut. Para insinyur di Laboratorium Bell sudah banyak mencoba berbagai teori matematika untuk mengurangi jumlah bahan yang dibutuhkan untuk menghubungkan sebuah kombinasi masukan ke sebuah kombinasi keluaran.

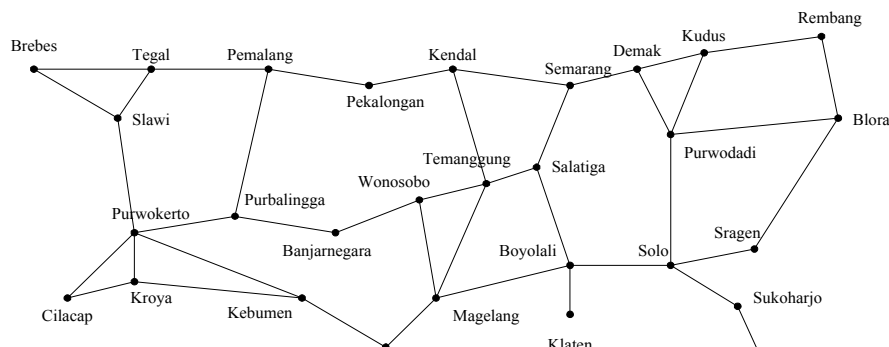
Algoritma yang pertama kali berhasil dikembangkan untuk tujuan ini adalah algoritma ‘nonblocking minimal spanning switch’.

Untuk itu, penulis membuat makalah ini, agar algoritma ‘nonblocking minimal spanning switch’ ini dapat dipahami. Dengan terwujudnya tujuan ini, perkembangan dunia telekomunikasi diharapkan akan menjadi lebih baik.

2. TEORI DASAR

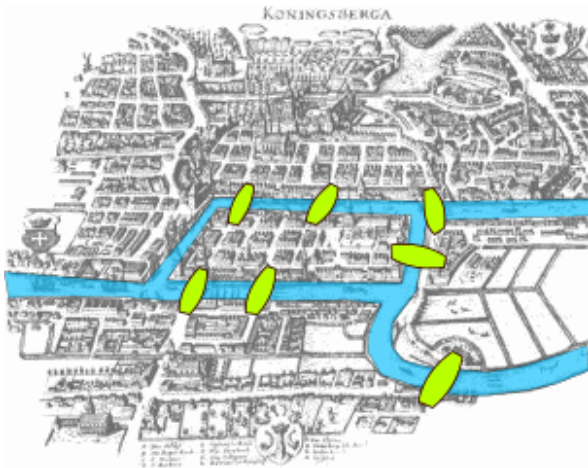
2.1. Teori Graf

Di dalam bidang matematika dan ilmu komputer, teori graf adalah suatu pembelajaran tentang graf. Yang dimaksud graf di sini adalah sebuah struktur matematika yang digunakan untuk merepresentasikan hubungan antara objek-objek diskrit dari suatu koleksi objek diskrit. Graf pada konteks ini merujuk kepada sebuah koleksi dari simpul (atau titik) dan sisi yang menghubungkan simpul-simpul tersebut. Dalam graf yang memenuhi syarat, dimana biasanya *tidak berarah*, sebuah garis dari titik *A* ke titik *B* dianggap sama dengan garis dari titik *B* ke titik *A*. Dalam *graf berarah*, garis tersebut memiliki arah. Pada dasarnya, sebuah graf digambarkan dengan bentuk diagram sebagai himpunan dari titik-titik (sudut atau simpul) yang digabungkan dengan kurva (garis atau sisi). Representasi graf terlihat seperti yang digambarkan pada Gambar 1. [2]

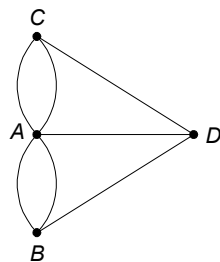


Gambar 1. Representasi Graf dengan kota-kota di Jawa Tengah sebagai simpulnya

Sejarah graf dimulai ketika Leonhard Euler menulis sebuah makalah tentang “Seven Bridges of Königsberg”. Makalah tersebut diterbitkan pada tahun 1736 dan diakui sebagai makalah pertama yang di sejarah teori graf. Pada makalah tersebut, Euler merepresentasikan jembatan yang menghubungkan antar kota sebagai sisi dan kota-kotanya sebagai simpul. Representasi graf tersebut seperti yang terlihat pada Gambar 2 dan Gambar 3. [1]



Gambar 2. Jembatan Königsberg



Gambar 3. Graf yang merepresentasikan Jembatan Königsberg

Pada Gambar 3, graf tersebut memiliki dua titik yang dihubungkan oleh dua sisi. Sisi ini dinamakan sisi ganda. Selain sisi ganda, di dalam teori graf terdapat istilah gelang. Gelang di sini adalah suatu sisi yang menghubungkan titik yang sama.

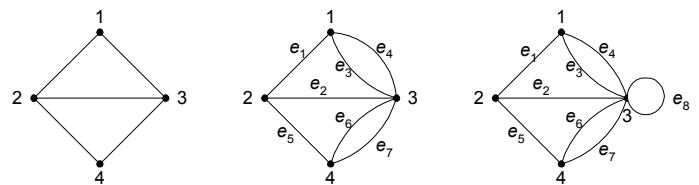
Graf dapat dikelompokkan menjadi beberapa kategori (jenis) bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi kalang, berdasarkan jumlah simpul, atau berdasarkan orientasi arah pada sisi. [1]

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis, yakni graf sederhana dan graf tak-sederhana. Graf sederhana merupakan graf yang tidak mengandung gelang maupun sisi ganda. Sedangkan, graf tak-sederhana adalah graf yang mengandung sisi ganda atau gelang. [1]

Berdasarkan jumlah simpul pada suatu graf, maka secara umum graf dapat digolongkan menjadi 2 jenis, yakni graf berhingga dan graf tak-berhingga. Yang dimaksud graf berhingga di sini adalah graf dengan jumlah simpul sebanyak n , dengan n merupakan bilangan bulat yang terhingga. Sedangkan, graf tak-berhingga adalah graf yang memiliki jumlah simpul yang tidak berhingga. [1]

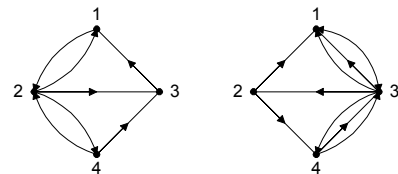
Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis, yakni graf berarah dan graf tak-berarah. Graf tak-berarah adalah graf yang sisinya tidak mempunyai orientasi arah. Sedangkan, graf berarah adalah graf yang setiap sisinya diberikan orientasi arah. [1]

Contoh dari graf sederhana dan tak-sederhana adalah sebagai berikut:



Gambar 4. (a) graf sederhana, (b) dan (c) graf tak-sederhana

Sedangkan, contoh dari graf berarah adalah sebagai berikut:



Gambar 5. (a) graf berarah, (b) graf ganda-berarah

2.2. Switch

Di bidang elektronika, ‘switch’ atau saklar adalah sebuah komponen elektronika yang bisa memutuskan suatu sirkuit elektrik, mengganggu arus yang sedang mengalir ataupun membaginya dari satu konduktor ke konduktor lainnya. Bentuk yang paling dikenal dari sebuah ‘switch’ atau saklar adalah sebuah alat elektromekanikal yang dioperasikan secara manual dengan satu atau lebih set kontak elektrik. Setiap set kontak elektrik bisa berada di salah satu ataupun kedua kondisi: baik itu tertutup yang berarti kontak menyentuh dan arus listrik bisa mengalir, ataupun terbuka yang berarti kontak terpisah dan tidak bersifat konduksi. [9]



Gambar 6. Contoh simbol sebuah ‘switch’

Sejak adanya logika digital di tahun 1950, istilah tersebut telah menyebar ke varietas dari alat aktif digital seperti transistor dan gerbang logika yang

berfungsi untuk mengubah kondisi keluaran antara dua level logika atau menghubungkan garis sinyal yang berbeda, dan bahkan komputer, saklar jaringan, yang berfungsi untuk memberikan hubungan antara 'port' yang berbeda di dalam jaringan komputer. Persamaan dari semua penggunaan ini adalah bahwa mereka semua adalah alat yang mengontrol sebuah kondisi biner, yakni nyala atau mati, tertutup atau terbuka, ataupun terhubung atau tak-terhubung.

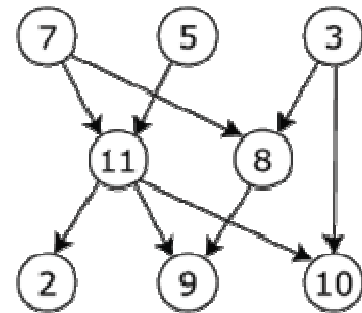
2.3. Topological Sort

Di dalam teori graf, 'topological sort' atau pengurutan topologi adalah sebuah pengurutan linier dari simpul-simpul yang masing-masingnya datang dari sebelum semua simpul yang memiliki sisi. Pengurutan ini hanya bisa dilakukan pada 'directed acyclic graph' atau graf asiklus berarah. Semua graf asiklus berarah memiliki satu atau lebih pengurutan topologi. Pengurutan topologi kadang-kadang juga dikenal sebagai pengurutan 'ancestral' atau 'ancestral ordering'.

Lebih formalnya dapat dilihat dari definisi pengurutan parsial. Definisi sebuah pengurutan parsial adalah suatu hubungan R pada simpul-simpul dari graf asiklus berarah, seperti xRy , jika dan hanya jika ada tepat sebuah jalan/sisi yang berarah dari x ke y . Sehingga, sebuah pengurutan topologi adalah sebuah ekstensi linier dari pengurutan parsial ini. Dengan kata lain, sebuah urutan keseluruhan yang sesuai dengan pengurutan parsial.

Salah satu aplikasi dari pengurutan topologi adalah pada penjadwalan sebuah sekuensi dari pekerjaan. Algoritma pengurutan topologi dipelajari pertama kali pada awal tahun 1960 di dalam konteks dari PERT teknik untuk menjadwalkan di dalam pengaturan beberapa proyek. Pekerjaan-pekerjaan digambarkan sebagai simpul, dan terdapat sisi dari x ke y jika pekerjaan x harus diselesaikan sebelum pekerjaan y bisa dimulai. Sebagai contohnya, ketika mencuci pakaian, kita harus mengisi ember dengan air sebelum membilas pakaian. Selanjutnya, pengurutan topologi memberikan sebuah urutan dimana dengan urutan tersebut, pekerjaan bisa dijalankan.

Di ilmu komputer, aplikasi dari pengurutan topologi dapat dilihat pada penjadwalan instruksi, pengurutan evaluasi formula suatu sel ketika menghitung formula di 'spreadsheet', menentukan urutan kompilasi pekerjaan untuk dilakukan di dalam membuat suatu 'file'. Dan menyelesaikan masalah ketergantungan simbol di suatu koneksi atau 'linker'.



Gambar 7. Contoh sebuah graf asiklus berarah

Sebagai contoh, graf asiklus berarah pada gambar 6. Memiliki banyak solusi pengurutan topologi. Beberapa diantaranya adalah sebagai berikut:

- 7,5,3,11,8,2,10,9
- 7,5,11,2,3,10,8,9
- 3,7,8,5,11,10,9,2
- 3,5,7,11,10,2,8,9

Algoritma pengurutan topologi biasanya memiliki waktu pengerjaan sebesar jumlah dari simpul ditambah dengan jumlah dari sisi, atau dengan notasi $O(|\Sigma \text{simpul}| + |\Sigma \text{sisi}|)$.

Salah satu algoritma pengurutan topologi, yang dicetuskan oleh Kahn (1962), bekerja dengan memilih simpul di urutan yang sama dengan pengurutan topologi secara 'ultimate' atau terakhir. Pertama-tama, temukan sebuah 'list' atau daftar dari simpul-simpul awal yang tidak punya sisi masukan. Lalu, masukkan mereka ke dalam sebuah himpunan S . Selanjutnya, dirumuskan dengan algoritma berikut:

```

L ← List kosong yang akan memuat elemen yang
terurut
S ← Himpunan dari semua simpul yang tidak
memiliki sisi masukan
while S tidak kosong do
    pindahkan sebuah simpul n dari S
    masukkan simpul n ke dalam L
    for each simpul m dengan sebuah sisi
    masukan e dari n ke m do
        pindahkan sisi e dari graf
        if m tidak punya sisi masukan
        lainnya then
            masukkan m ke dalam S
if graf memiliki sisi then
    output(pesan error)
else
    output(hasil pengurutan topologi: L)
  
```

Jika graf tersebut merupakan graf asiklus berarah, maka solusi termuat di dalam list L (solusi yang ada tidak unik). Di lain pihak, graf memiliki setidaknya satu siklus dan mustahil untuk melakukan pengurutan topologi pada graf tersebut.

Algoritma alternatif lainnya untuk pengurutan topologi, berdasarkan pada pencarian kedalaman,

ditemukan oleh Cormen, Leiserson, dan Rivest (1990) di dalam sebuah makalah oleh Tarjan (1976). Pertama-tama, 'looping' ke dalam simpul-simpul di dalam graf, pada urutan apapun, untuk menginisiasi sebuah pencarian kedalaman untuk simpul manapun yang belum pernah dikunjungi oleh pencarian sebelumnya. Pengurutan topologi yang diinginkan adalah kebalikan dari urutan 'post' dari pencarian ini. Dengan itu, kita dapat membangun sebuah urutan, seperti sebuah list simpul, dengan menambahkan setiap simpul ke awal dari list ketika pencarian kedalaman dilakukan pada simpul tersebut dan telah kembali dari memproses semua anak dari simpul tersebut. Karena setiap sisi dan simpul dikunjungi sekali, maka algoritma tersebut dijalankan dalam waktu linier.

3. PEMBAHASAN

3.1. Nonblocking Minimal Spanning Switch

Nonblocking minimal spanning switch adalah sebuah alat yang mengimplementasikan sebuah saklar yang bisa menghubungkan sejumlah N masukan dengan N keluaran dengan berbagai kombinasi karena tidak adanya blok dan melakukan pekerjaan tersebut dengan komponen sedikit (untuk pengeluaran biaya yang minimum). Sebuah sistem saklar mengatur hubungan antara sejumlah sumber informasi dan sejumlah lainnya penerima informasi. Istilah umumnya untuk 'nonblocking minimal spanning switch' adalah pertukaran saluran telepon.

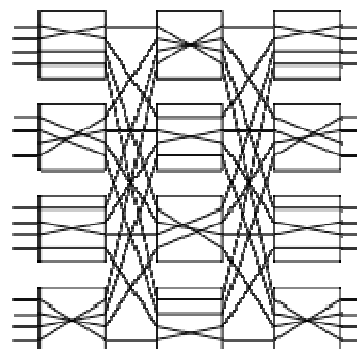
3.2. Saklar 'Crossbar'

Saklar 'crossbar' memiliki properti untuk bisa menghubungkan N masukan ke N keluaran dengan kombinasi satu-satu apapun, sehingga alat ini bisa menghubungkan pemanggil apapun kepada penerima apapun yang tidak sibuk, seperti istilah 'nonblocking'. Pertukaran saluran yang 'nonblocking' selalu bisa meneruskan semua panggilan (kepada penerima yang tidak sibuk). Hal ini bisa memaksimalkan pemasukan sebuah perusahaan telepon, karena 'nonblocking' adalah fitur yang diinginkan untuk sisi ekonomis dan kepuasan konsumen.

Meskipun demikian, sebuah saklar 'crossbar' bekerja dengan tingkat biaya penggunaan N^2 saklar sederhana. Untuk jumlah N yang besar, seperti yang terlihat pada dunia nyata, membawa tingkat biaya yang mahal. Lebih jauh lagi, untuk pertukaran jalur telepon, saklar 'crossbar' memiliki masalah fisik. Tidak hanya saklar tersebut membutuhkan ruang yang banyak tetapi juga saklar tersebut memiliki kontak saklar yang bisa menjadi sangat panjang tergantung jumlah saklar sederhana yang menyusun, sehingga menjadi tidak bisa diandalkan. Para insinyur juga menyadari bahwa pada suatu sisi dari saklar 'crossbar' hanya membuat sebuah koneksi/hubungan. Kontak lainnya di dua sisi lainnya tidak berguna. Hal ini menyatakan bahwa terdapat bahan saklar yang terbuang pada komponen

saklar 'crossbar' sehingga menjadi tidak efisien dan efektif.

Salah satu jalan untuk meminimalisasi masalah ini adalah dengan membangun saklar 'crossbar' dari saklar 'crossbar' yang lebih kecil. Pada gambar di bawah ini adalah sebuah 16x16 saklar 'crossbar' yang dibangun dari 12 buah 4x4 saklar 'crossbar'.



Gambar 8. Contoh nonblocking minimal spanning switch yang terdiri dari saklar 'crossbar' sebanyak 12 buah.

3.3. Saklar 3 Bagian

Pendekatan selanjutnya adalah dengan memecah saklar 'crossbar' menjadi 3 bagian dari saklar 'crossbar' yang lebih kecil. Bagian-bagian tersebut adalah bagian masukan, bagian tengah dan bagian keluaran. Dengan saklar yang lebih kecil, semakin bisa diandalkan, semakin mudah untuk dibuat, dan semakin murah pengeluaran biaya.

Sebuah sistem telepon hanya harus membuat sebuah koneksi/hubungan satu-satu. Hal ini berarti jumlah masukan dan jumlah keluaran sama pada setiap subsaklar. Misalnya, kita mau membuat sebuah saklar 'crossbar' 16x16. Salah satu rancangannya adalah 4 subsaklar pada sisi masukan, masing-masing 4 masukan, sehingga totalnya 16 masukan. Selanjutnya, di sisi keluaran, kita bisa juga memiliki 4 subsaklar dengan masing-masing 4 masukan. Hal ini bergantung kepada jumlah kabel yang digunakan, karena kabel juga memiliki biaya yang cukup mahal. Jumlah kabel yang paling sedikit yang bisa menghubungkan dua subsaklar adalah sebuah kabel. Sehingga, setiap subsaklar masukan akan memiliki sebuah kabel untuk setiap subsaklar tengah. Begitu pula setiap subsaklar tengah juga memiliki paling tidak satu kabel untuk setiap subsaklar keluaran.

Pertanyaannya adalah berapa banyak subsaklar tengah yang dibutuhkan, dan berapa banyak jumlah kabel yang seharusnya menghubungkan bagian masukan ke bagian tengah. Karena saklar telepon simetris (penerima telepon dan penelepon bisa ditukar-tukar), logika yang sama akan berlaku untuk bagian keluaran, dan subsaklar tengah akan menjadi 'kotak persegi', memiliki jumlah yang sama dengan masukan ataupun keluaran.

Jumlah dari subsaklar tengah bergantung pada algoritma yang digunakan untuk mengalokasikan hubungan kepada subsaklar tersebut. Algoritma dasar untuk mengatur saklar 3 bagian adalah dengan mencari subsaklar tengah yang memiliki kabel yang tidak digunakan ke saklar masukan dan keluaran yang membutuhkan. Setelah subsaklar tengah yang dapat digunakan ditemukan, subsaklar tersebut dihubungkan ke masukan dan keluaran yang benar dapat dilakukan.

Secara teori, pada contoh, hanya 4 saklar sentral yang dibutuhkan, masing-masing dengan tepat satu koneksi/hubungan ke setiap saklar masukan dan satu koneksi/hubungan ke setiap saklar keluaran. Hal ini disebut 'minimal spanning switch'.

Faktanya, jika terdapat satu panggilan per saklar masukan, dan hal ini menghasilkan satu panggilan per saklar keluaran, sehingga 16 panggilan pertama pada saklar ini akan memblok sampai dengan 15 panggilan tambahan yang membutuhkan koneksi/hubungan yang sama.

Untuk alasan ini, saklar 16x16 dengan 4 subsaklar masukan dan 4 subsaklar keluaran diperkirakan membutuhkan 7 subsaklar tengah. Untuk alasan ini juga, pengaturan saklar ini dinamakan saklar '2n-1', dimana n adalah jumlah dari subsaklar masukan.

3.4. Algoritma

Kunci dari algoritma yang digunakan adalah dengan mengatur ulang koneksi/hubungan di subsaklar tengah sehingga koneksi/hubungan baru dapat dibuat. Algoritma ini menggunakan prinsip kerja seperti algoritma pengurutan topologi.

Langkah awal dari algoritma ini adalah mencari sebuah subsaklar tengah yang memiliki koneksi masukan dan keluaran yang dibutuhkan. Jika sebuah subsaklar tengah itu ditemukan, kemudian dialokasikan. Tetapi pencarian ini kadang-kadang gagal, karena jumlah dari subsaklar tengah lebih kecil di 'minimal spanning switch' daripada di saklar '2n-1'. Jika sebuah subsaklar dengan koneksi saklar masukan yang dibutuhkan tidak bisa ditemukan. Salah satu subsaklar pasti memiliki koneksi/hubungan ke saklar masukan yang dibutuhkan, yang lainnya pasti punya koneksi/hubungan ke subsaklar keluaran yang dibutuhkan. Subsaklar ini haruslah ada, karena untuk setiap input di 'minimal spanning switch', terdapat paling sedikit sebuah kabel dari subsaklar masukan ke subsaklar tengah. Karena keseluruhan saklar adalah sebuah system saluran telepon (penerima telepon dan penelepon bisa ditukar-tukar), saklar juga bersifat simetris, sehingga terdapat sebuah kabel yang bebas dari salah satu subsaklar tengah ke subsaklar keluaran yang dibutuhkan.

Secara konseptual, algoritma tersebut harus menyusun ulang hubungan di antara dua subsaklar tengah ini,

sebut saja A dan B. Prinsipnya adalah untuk menjaga semua hubungan yang ada dan telah melewati A dan B, dengan tujuan untuk menghindari pemutusan panggilan, dan juga membawa dua kabel A atau B ke subsaklar masukan dan keluaran yang dibutuhkan.

Dengan penggambaran baku, hubungan diagram A dan B sebenarnya bergantung satu sama lain. Masukan A bergantung pada masukan B. Begitu juga dengan keluaran A dan B.

Hubungan yang melewati A dan B diletakkan di dalam sebuah list/daftar yang juga meliputi koneksi baru yang diinginkan.

Pertama-tama, setiap koneksi yang hanya memiliki satu masukan atau satu keluaran ditelusuri pada superposisi A dan B. Dengan kata lain, dalam penggambaran menggunakan kertas dan pensil, setiap titik hanya dapat dilewati oleh satu garis. Pada hal ini, titik dianalogikan dengan masukan atau keluaran dari subsaklar. Sedangkan, garis dianalogi sebagai kabel yang menghubungkan masukan tersebut dengan keluaran.

Mulai dari beberapa masukan atau keluaran, lalu ditelusuri sampai ke sebuah keluaran, lalu ditelusuri kembali koneksi lainnya dari keluaran yang ditemukan tersebut ke masukan lainnya, dan terus begitu, hingga tidak menemukan koneksi lainnya.

Setiap kali penelusuran dari masukan ke keluaran, koneksi tersebut dialokasikan ke subsaklar A. Ketika menelusuri pada arah sebaliknya, koneksi tersebut dialokasikan ke B.

Setelah semua koneksi dengan satu masukan atau satu keluaran hilang, yang tersisa adalah tinggal graf-graf siklus dari koneksi. Sekali lagi, lakukan penelusuran untuk setiap graf, menghubungkan koneksi ke subsaklar, dan memindahkan setiap koneksi dari daftar koneksi.

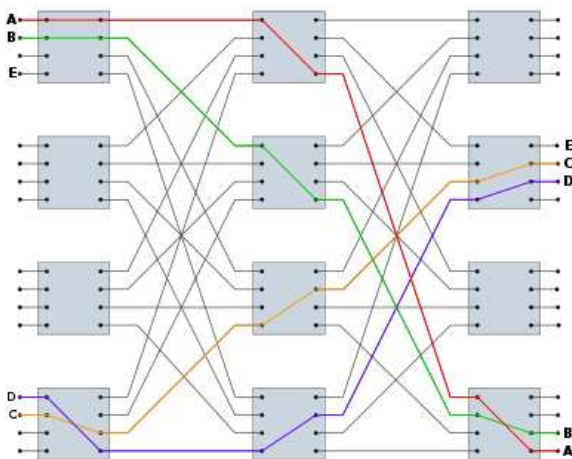
Terdapat lebih sedikit penelusuran yang dilakukan jika semua graf asiklus telah ditelusuri dan dipindahkan sebelum graf-graf siklus ditelusuri dan dipindahkan. Dengan cara tersebut, kita tidak perlu mengecek masukan atau keluaran lebih dari dua kali, dan tidak perlu mencatat masukan dan keluaran mana saja yang telah diperiksa.

A ataupun B dapat memiliki berbagai arah penelusuran, karena A dan B memiliki jumlah koneksi yang sama, yakni satu kabel untuk setiap subsaklar masukan dan keluaran.

Setelah koneksi-koneksi dialokasikan ke dalam daftar di dalam program, kemudian saklar dapat di-program ulang, secara fisik dipindahkan koneksinya.

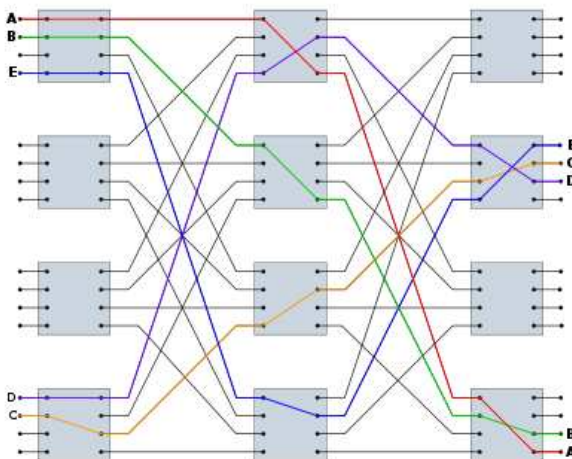
Saklar elektronik yang modern sengaja didesain untuk menerima data baru dan diolah dengan menekan sebuah tombol. Hasilnya adalah koneksi-koneksi berpindah secara cepat, tanpa mengganggu hubungan telepon yang menyala. Pada saklar yang lebih tua daripada saklar elektronik, terkadang terdengar bunyi ketika terjadi perpindahan koneksi.

Sebagai contoh, gambar 9 adalah saklar 16x16 yang terdiri dari 5 saluran, yakni A, B, C, D dan E. Pada gambar tersebut, jalur yang membentuk saluran A, B, C, dan D memblok jalur pembentuk saluran E.



Gambar 9. Saklar 16x16 dengan 5 saluran

Dengan menggunakan algoritma tersebut, pemblokiran ini dapat dihindari. Algoritma tersebut dapat memberikan banyak variasi solusi dari masalah blok di sini. Salah satunya adalah gambar 10.



Gambar 10. Salah satu solusi yang dihasilkan algoritma ini

4. KESIMPULAN

Algoritma 'nonblocking minimal spanning switch' ini dapat menghasilkan berbagai macam solusi. Sehingga solusi yang ditawarkan di sini merupakan solusi yang tidak unik.

Algoritma 'nonblocking minimal spanning switch' merupakan algoritma yang penting dalam perkembangan dunia telekomunikasi. Sehingga, dengan mengembangkan algoritma tersebut, kita dapat memajukan dunia telekomunikasi baik dari segi ekonomis maupun kenyamanan penggunaan.

DAFTAR REFERENSI

- [1] Munir, Rinaldi, *Struktur Diskrit*, ITB, 2008.
- [2] http://en.wikipedia.org/wiki/Graph_theory (tanggal akses : 31/12/08 - waktu akses: 14:28 WIB)
- [3] http://en.wikipedia.org/wiki/Nonblocking_Minimal_Spanning_Switch (tanggal akses : 31/12/08 - waktu akses: 14:29 WIB)
- [4] http://en.wikipedia.org/wiki/Telephone_switch (tanggal akses : 31/12/08 - waktu akses: 14:30 WIB)
- [5] <http://www.economicexpert.com/a/Nonblocking:Minimal:Spanning:Switch.htm> (tanggal akses : 31/12/08 - waktu akses: 14:30 WIB)
- [6] <http://www.economicexpert.com/a/Topological:sort.htm> (tanggal akses : 31/12/08 - waktu akses: 14:31 WIB)
- [7] http://en.wikipedia.org/wiki/Topological_sorting (tanggal akses : 31/12/08 - waktu akses: 14:31 WIB)
- [8] <http://en.wikipedia.org/wiki/Switch> (tanggal akses: 31/1/09 - waktu akses: 17:45 WIB)