IMPLEMENTASI ALGORITMA EDMONDS KARP DALAM PENCARIAN ALIRAN MAKSIMUM PADA JARINGAN LISTRIK

SKRIPSI
disajikan sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains
Progam Studi Matematika

Oleh
Nunik Sutrisni
4111413027

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SEMARANG

2019

PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuann peraturan perundang-undangan.

Semarang, 15 Agustus 2019

METERAI TEMPEL

TGL. 20

89EC1AFF949479748

6000

ENAM RIBU RUPIAH

Nunik Sutrisni

NIM 4111413027

PENGESAHAN

Skripsi yang berjudul *Implementasi Algoritma Edmonds Karp dalam Pencarian Aliran Maksimum pada Jaringan Listrik* karya Nunik Sutrisni 4111413027 telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada tanggal 21 Agustus 2019.

Panitia:

Ketua

Dr. Sugianto, M.Si.
NIP.196102191993031001

Sekretaris

Drs. Arief Agoestanto, M.Si.
NIP.196807221993031005

Ketua Penguji

Dr. Mulyono, M.Si.
NIP. 197009021997021001

Anggota Penguji/Pembimbing I

Dr. Isnaini Rosyida, S.Si., M.Si
NIP. 197302191998022001

Anggota Penguji/Pembimbing II

Dr. Tri Sri Noor A., S.Si., M.Si.
NIP. 197706142008122002

MOTTO DAN PERSEMBAHAN

MOTTO

1. Jangan katakan aku tak dapat. Katakan aku dapat, dan cobalah.
2. Alam dan kita adalah satu. Ketika kita percaya kepada alam, maka alam akan melindungi kita.
3. Be the best of me!

PERSEMBAHAN

1. Ibu Sunarsih dan Bapak Suyono atas doa, kasih sayang dan dukungan yang tak terhingga.
2. Untuk Wahyu Sutristyono, adikku tercinta atas dukungannya.
3. Untuk teman-teman seperjuangan Nana, Agnes, Eka, Esta, Santi dan Iin terimakasih waktunya.
4. Untuk teman-teman seorganisasi The MATe, terimakasih kekeluargaannya.
5. Teman-teman mahasiswa matematika angkatan 2013
6. Untuk semua yang telah memberikan dukungan.

KATA PENGANTAR

Alhamdulillah penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat, taufik, serta hidayah-Nya sehingga penulis dapat menyelesaikanskripsi dengan judul “Implementasi Algoritma Edmons Karp dalam Pencarian Aliran Maksimum pada Jaringan Listrik”. Penulis menyadari bahwa tanpa bantuan serta dukungan dari berbagai pihak, skripsi ini tidak akan terselesaikan denbgan baik dan lancar. Oleh karena itu, pada kesempatan kali ini penulis ingin mengucapkan rasa terima kasih kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang.
2. Dr. Sugianto, M.Si., Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.
4. Dr. Isnaini Rosyida, S.Si., M.Si., selaku Dosen Pembimbing I yang telah memberikan bimbingan, motivasi, waktu dan pengarahan selama penyusunan skripsi ini.
5. Dr. Tri Sri Noor Asih, S.Si., M.Si., selaku Dosen Pembimbing II yang telah memberikan bimbingan, motivasi, waktu dan pengarahan selama penyusunan skripsi ini.
6. Dr. Mulyono, M.Si., selaku Dosen Penguji yang telah memberikan penilaian dan saran dalam perbaikan skripsi ini serta telah memberikan bimbingan dan arahan.

7. Drs. Sugiman, M.Si., selaku dosen wali yang telah memberikan arahan dan bimbingan selama masa kuliah.
8. Ibu dan Bapak tercinta yang selalu memberikan doa serta memberikan dukungan baik secara moral maupun spiritual.
9. Segenap keluarga besar yang telah memberikan dukungannya.
10. Mahasiswa Matematika angkatan 2013 yang telah memberikan dorongan dan motivasi.
11. Semua pihak yang telah membantu terselesaikannya penulisan skripsi ini.

Penulis hanya bisa berharap semoga skripsi ini dapat memberikan manfaat serta nilai tambah bagi pembaca, khususnya mahasiswa Prodi Matematika

Semarang, 15 Agustus 2019

Penulis

ABSTRAK

Sutrisni, Nunik. 2019, *Implementasi Algoritma Edmons Karp dalam Pencarian Aliran Maksimum pada Jaringan Listrik.* Skripsi. Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang, Pembimbing Utama Dr. Isnaini Rosyida, S.Si., M.Si. dan Pembimbing Pendamping Dr. Tri Sri Noor Asih, S.Si., M.Si.

Kata kunci: Aliran Maksimum, Algoritma Edmonds Karp, Algoritma BFS, Matlab.

Penelitian ini membahas tentang sebuah permasalahan pencarian aliran maksimum pada jaringan listik, dalam kasus ini di Kota Tegal Wilayah Distribusi Kebasen 11. Permasalahan mencari aliran maksimum pada jaringan listrik tersebut bertujuan untuk memaksimalkan arus yang dapat mengalir pada jaringan, sehingga tidak menimbulkan kelebihan atau kekurangan arus listrik.

Permasalahan yang diambil oleh penelitian ini adalah bagaimana tahapan algoritma Edmonds Karp dalam pencarian aliran maksimum, bagaimana mencari aliran maksimum di Kota Tegal Wilayah Distribusi Kebasen 11 dengan algoritma Edmonds Karp, dan bagaimana pencarian aliran maksimum dengan bantuan tool software Matlab.

Tahapan algoritma Edmonds karp dalam pencarian aliran maksimum didalamnya menggunakan algoritma BFS untuk pencarian lintasan penambah. Algoritma BFS bekerja dengan mengunjungi semua tetangga dari simpul yang masuk antrian, sampai bertemu dengan simpul tujuan. Hasil perhitungan manual algoritma Edmonds Karp dengan bantuan tool software Matlab mendapatkan hasil yang sama.

Hasil analisis dari penelitian ini diharapkan PT. PLN Persero dalam menentukan aliran maksimum pada jaringan listrik, khususnya jaringan menengah menggunakan perhitungan dari penelitian ini, beserta program hasil penelitian.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring dengan perkembangan zaman dan teknologi yang semakin maju, hampir setiap kebutuhan dalam ilmu pengetahuan dan teknologi membutuhkan peranan matematika. Matematika dibutuhkan untuk menunjang eksistensi ilmu-ilmu lain seperti fisika, kimia, biologi, ekonomi, teknik dan lain sebagainya. Tidak heran mengapa matematika dijuluki *"mathematic is a queen, but also a servant"*, matematika sebagai ratu ilmu, tetapi juga sekaligus pelayan untuk ilmu-ilmu lain (Bell, 1952). Kehidupan manusia sekarang tidak dapat dipisahkan dengan peranan matematika dalam memecahkan berbagai masalah kehidupan nyata. Teori graf merupakan salah satu peranan matematika untuk menyelesaikan masalah-masalah dunia nyata dengan menggunakan visualisasi dan diagram.

Secara umum graf adalah suatu diagram yang memuat informasi tertentu, jika diinterpretasikan secara tepat. Dalam kehidupan sehari-hari graf digunakan untuk menggambarkan berbagai struktur yang ada. Tujuannya adalah sebagai visualisasi objek-objek agar lebih mudah dimengerti. Beberapa contoh graf yang sering dijumpai dalam kehidupan sehari-hari antara lain, struktur organisasi, bagan alir pengambilan mata kuliah, rangkaian listrik, dll. (Siang, 2002:187)

Jaringan merupakan salah satu contoh graf dan juga salah satu kajian dalam riset operasi. Jaringan adalah sebuah graf berarah sederhana yang mempunyai sumber dan tujuan yang setiap sisinya mempunyai kapasitas. Jaringan yang sering dijumpai dalam kehidupan sehari-hari adalah jaringan pendistribusian barang, jaringan aliran air, jaringan aliran listrik, jaringan telepon, dan jaringan komputer. Manusia terus mengembangkan metode untuk memudahkan dalam menyelesaikan penyelesaian jaringan, contohnya mencari rute terpendek, menyusun jadwal agar waktu yang digunakan lebih optimal, dan mencari lintasan agar pendistribusian barang lebih maksimal tetapi tidak membutuhkan biaya banyak. Permasalahan-permasalahan tersebut biasa disebut *network flow*. Di dalam jaringan terdapat beberapa model yang bisa digunakan untuk membantu memecahkan masalah-masalah yang merupakan aplikasi *network flow*, di antaranya adalah model distribusi kendali, model rentang jaringan minimum, model rute terpendek, dan jaringan aliran maksimum.

Salah satu masalah *network flow* yang sering terjadi pada kehidupan sehari-hari adalah mencari aliran maksimum. Jaringan aliran maksimum merupakan permasalahan optimasi yang memiliki tujuan untuk memaksimalkan jumlah aliran yang mengalir pada sebuah jaringan yang hanya memiliki satu sumber (*source*) dan satu tujuan (*sink*). Contoh aliran maksimum pada sebuah jaringan adalah jaringan listrik, pipa saluran, dan jalur lalu lintas dalam sebuah sistem jaringan yang tertutup. Kapasitas pada setiap jaringan akan membatasi jumlah arus atau aliran yang melewatinya. Arus listrik yang mengalir dari sumber tegangan menuju saklar lampu melalui kawat penghantar. Optimasi pembebanan pada kawat penghantar adalah memaksimalkan batasan besar arus yang dilalukan melewati penghantar sesuai dengan KHA dan kondisi sekitarnya, sebab apabila berlebihan akan mengakibatkan pelunakan pada titik tumpu penghantar, pelunakan pada titik tumpu ikatan penghantar, berkurangnya jarak aman, kerusakan pada isolasi (Ryu, 2018)

Sebagai contoh suatu kabel dengan kapasitas 20 ampere akan segera terbakar jika memaksa kabel itu dilewati arus 100 ampere pada tingkat tegangan yang sama. Oleh karena itu perlu memaksimalkan arus yang mengalir pada kawat agar tidak ada gangguan jaringan listrik. Model aliran maksimum mempunyai tujuan untuk memaksimalkan jumlah arus yang melewati jaringan dalam sebuah sistem jaringan. Hal ini sangat umum terjadi pada bidang transportasi, produksi, komunikasi, dan distribusi.

Aplikasi masalah aliran maksimum yang akan dibahas dalam skripsi ini adalah jaringan listrik. Penulis memilih jaringan listrik karena jaringan listrik merupakan kebutuhan utama dimana setiap manusia pasti menggunakan listrik untuk berbagai hal sehingga ketergantungan manusia amat tinggi terhadap pemanfaatan listrik untuk kehidupan sehari-hari. Aliran arus listrik yang tidak maksimum dapat menyebabkan kerusakan pada perangkat yang menggunakan energi listrik. Hal ini terjadi karena perangkat yang menggunakan listrik memakai satuan volt, dimana volt/ tegangan berbanding lurus dengan arus. Selain arus yang berbanding lurus dengan tegangan, tahanan dalam listrik juga berbanding lurus dengan tegangan. Jadi arus merupakan salah satu faktor yang dapat menyebabkan rusaknya perangkat yang menggunakan energi listrik. Oleh karena itu dibutuhkan cara untuk menanggulanginya yaitu dengan mencari arus maksimal yang harus dialirkan pada jaringan melalui gardu induk. Berdasarkan permasalahan tersebut aplikasi aliran maksimum dapat digunakan dengan mencari aliran arus maksimum pada jaringan listrik.

Banyak algoritma yang telah dikembangkan untuk penyelesaian aliran maksimum, misalnya algoritma Djikstra, algoritma Ford-fulkerson, algoritma Preflow dan algoritma Edmons Karp. Latifah (2010) telah membahas penerapan algoritma Prim dan Kruskal pada jaringan distribusi air PDAM. Sedangkan Farizal (2006) telah membahas aliran maksimum menggunakan algoritma Ford-Fulkerson. Penelitian tersebut menyimpulkan algoritma Ford-Fulkerson memiliki iterasi yang cukup banyak sehingga kurang efisien. Diperlukan algoritma yang lebih berkembang dari Ford Fulkerson yaitu algoritma Edmons Karp yang merupakan perkembangan lebih lanjut dari algoritma Ford Fulkerson. Hal inilah yang membuat penulis mempertimbangkan untuk menerapkan algoritma Edmonds Karp untuk mencari aliran maksimum dari beberapa simpul pada jaringan listrik dan simulasinya menggunakan software Matlab.

1.2. Rumusan Masalah

Dari latar belakang di atas diperoleh permasalahan yang timbul dalam penyusunan skripsi ini, antara lain:

1. Bagaimana representasi jaringan listrik dalam bentuk network?
2. Bagaimana aliran maksimum dari titik sumber di Gardu Induk ke titik tujuan di tiang listrik Jl. Jend Ahmad Yani pada jaringan listrik Kota Tegal Wilayah distribusi Kebasen 11 dengan Algoritma Edmonds Karp?
3. Bagaimana hasil aliran maksimum yang diproses dalam tool software Matlab?

1.3. Batasan Masalah

Dalam skripsi ini, terdapat batasan masalah sebagai berikut:

1. Data yang diolah adalah data berupa jaringan distribusi listrik dan daftar kapasitas pada masing-masing kabel yang diambil dari jaringan listrik Kota Tegal Wilayah distribusi Kebasen 11.
2. Pada peta jaringan listrik, tiang listrik menunjukkan simpul dan kabel yang menghubungkan antar tiang listrik menunjukkan sisi sedangkan kapasitas pada masing-masing sisi merupakan kapasitas kabel dalam satuan ampere.
3. Kapasitas kabel dalam satuan ampere diambil dari laporan aset jaringan tegangan menengah yang disesuaikan dengan luas penampang kabel dan kemampuan hantar listrik.
4. Perancangan program Algoritma Edmonds Karp menggunakan bantuan tools software Matlab.
5. Pencarian aliran maksimum pada jaringan listrik menggunakan Algoritma Edmonds Karp di mana dalam memlilih lintasan penambah terpendek menggunakan Algoritma BFS (*Breadth First Search*).
6. Informasi yang dihasilkan adalah nilai maksimal yang telah dihitung menggunakan Algoritma Edmonds Karp.

1.4. Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1) Mengetahui representasi jaringan listrik dalam bentuk network.
2) Dapat mensimulasikan perhitungan aliran maksimum dengan Algoritma Edmonds Karp untuk mendapatkan nilai maksimumnya.

3) Dapat menggunakan tools Matlab untuk menentukan aliran maksimum pada suatu jaringan.

1.5. Manfaat

Manfaat yang diharapkan dari penyusunan skripsi ini adalah:

1) Bagi peneliti

   Dapat mengetahui bagaimana menentukan aliran maksimum dengan algoritma Edmonds Karp.

2) Bagi universitas

   Menambah arsip dan referensi untuk bahan penelitian selanjutnya.

3) Bagi mahasiswa

   Dapat dipakai sebagai bahan acuan bagi mahasiswa yang ingin melanjutkan penelitian pencarian aliran arus maksimum dengan algoritma yang berbeda.

1.6.Sistematika Skripsi

Dalam penulisan skripsi ini secara garis besar dibagi menjadi tiga bagian pokok, yaitu bagian awal, bagian isi, dan bagian akhir.

1.6.1. Bagian Awal

Bagian awal skripsi memuat : (1) Halaman sampul, (2) Halaman judul, (3) Pernyataan keaslian tulisan, (4) Halaman pengesahan, (5) Motto dan persembahan, (6) Abstrak, (7) Kata pengantar, (8) Daftar isi, (9) Daftar gambar, (10) Daftar lampiran

1.6.2. Bagian Isi

(1) Bab I : Pendahuluan

Mengemukakan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan skripsi.

(2) Bab II : Landasan Teori

Berisi uraian teoritis atau teori teori yang mendasari pemecahan tentang masalah-masalah yang berhubungan dengan judul skripsi. Pada bab ini dibagi menjadi beberapa sub bab yaitu teori graf, terminologi dasar, jaringan, aliran maksimum, algoritma Edmonds Karp

(3) Bab III : Metode Penelitian

Berisi tentang metode metode yang digunakan dalam penelitian yang meliputi menemukan masalah, merumuskan masalah, metode pengambilan data, analisis data dan pemecahan masalah, dan penarikan kesimpulan

(4) Bab IV : Hasil Penelitian dan Pembahasan

Berisi semua hasil penelitian dan pembahasan mengenai analisis algoritma Edmonds Karp untuk mencari arus maksimum

(5) Bab V : Penutup

Bab ini berisi tentang simpulan dan saran-saran yang diberikan peneliti berdasarkan simpulan yang diambil.

1.6.3. Bagian Akhir

Bagian akhir berisi tentang daftar pustaka dan lampiran-lampiran yang mendukung skripsi.

BAB II

LANDASAN TEORI

2.1 Terminologi Dasar dalam Graf

Teori graf merupakan pokok bahasan yang banyak penerapannya pada masa kini. Pemakaian teori graf telah banyak dirasakan dalam berbagai ilmu, antara lain: optimisasi jaringan, ekonomi, psikologi, genetika, riset operasi, dan lain-lain. Graf digunakan untuk mempresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek sebagai simpul atau titik, sedangkan hubungan antara objek dinyatakan dengan garis. Makalah pertama tentang teori graf ditulis pada tahun 1736 oleh seorang matematikawan Swiss yang bernama Leonard Euler yang telah menggunakan teori graf untuk menyelesaikan masalah jembatan Konigsberg (sekarang, bernama Kaliningrad).

2.1.1 Definisi graf

Graf adalah kumpulan simpul (*nodes/ vertex*) yang dihubungkan satu sama lain melalui sisi (*edge*) atau busur (*arc*). Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan simpul) yang dihubungkan oleh garis-garis (melambangkan sisi) atau garis berpanah (melambangkan busur). Graf G didefinisikan sebagai pasangan himpunan (V, E) yang dalam hal ini, V adalah himpunan berhingga dan tidak kosong dari simpul-simpul (*nodes/vertex*) sedangkan E adalah himpunan sisi (*edges atau arcs*) yang menghubungkan sepasang simpul, atau dapat ditulis dengan notasi $G = (V, E)$ (Munir, 2010:356).

Sebuah graf dimungkinkan tidak mempunyai sisi sama sekali, tetapi simpul harus ada. Graf yang hanya mempunyai satu buah simpul tanpa sebuah sisi dinamakan graf trivial. Simpul pada graf dapat dilabeli dengan huruf, seperti $a, b, c, \ldots, v, w, \ldots$ atau dengan bilangan asli $1, 2, 3, \ldots$ atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul u dengan simpul v, maka e dapat ditulis sebagai (u, v) atau dinyatakan dengan lambang $e_1, e_2, \ldots$. Dengan kata lain, jika e adalah sisi yang menghubungkan simpul u dengan simpul v, maka e dapat ditulis sebagai $e = (u, v)$. Secara geometri graf dapat digunakan sebagian sekumpulan simpul didalam bidang dwimatra (bidang dua dimensi) yang dihubungkan dengan sekumpulan garis (sisi). Sebuah sisi paralel adalah dua sisi yang berbeda yang menghubungkan simpul yang sama, sedangkan loop adalah sisi yang hanya berhubungan dengan satu simpul ujung (Siang, 2002: 187).

Pada Gambar 1 memperlihatkan 3 buah graf, G_1, G_2 dan G_3. G_1 adalah graf dengan himpunan simpul V dan himpunan sisi E sebagai berikut

$V = \{1, 2, 3, 4\}$

$E = \{(1,2), (1,3), (2,3), (2,4), (3,4)\}$

G_2 adalah graf dengan himpunan simpul V dan himpunan sisi E adalah:

$V = \{1, 2, 3, 4\}$

$E = \{(1,2), (2,3), (1,3), (2,4), (3,4), (3,4)\} \Rightarrow \text{himpunan sisi ganda}$

$= (e_{1,} e_{2,} e_{3,} e_{4,} e_{5,} e_{6,} e_7 \}$

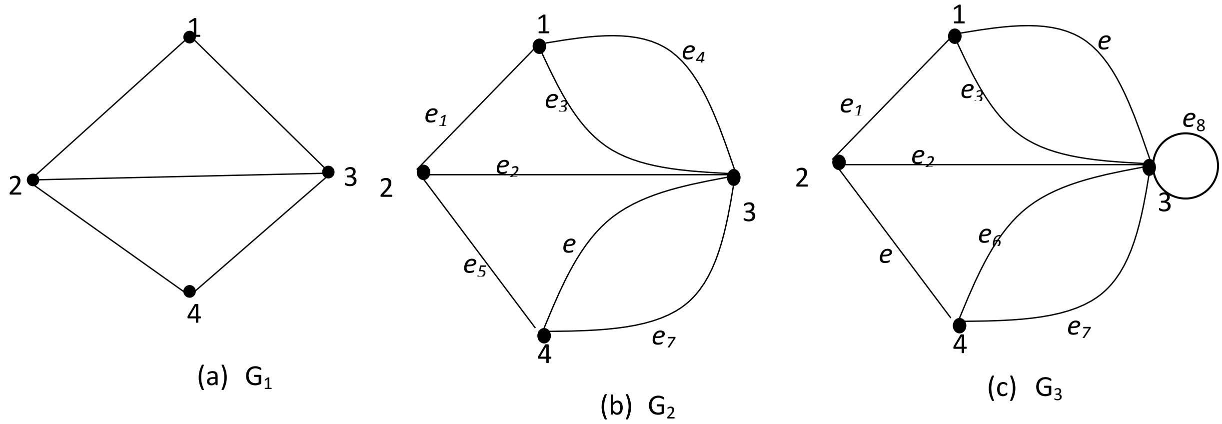
G_3 adalah graf dengan himpunan simpul V dan himpunan sisi E adalah:

$V = \{1, 2, 3, 4\}$

$E = \{(1,2), (2,3), (1,3), (2,4), (3,4), (3,4)\}$ ⇒ himpunan sisi ganda

$= (e_{1,} e_{2,} e_{3,} e_{4,} e_{5,} e_{6,} e_7 \}$

Pada G_1 sisi $e_3 = (1, 3)$ dan sisi $e_4 = (1, 3)$ dinamakan sisi ganda (*multiple edges* atau *pararel edges*) karena kedua sisi ini menghubungkan dua buah simpul yang sama, yaitu simpul 1 dan simpul 3. Pada G_2, sisi $e_8 = (3, 3)$ dinamakan gelung atau kalung karena berawal dan berakhir pada simpul yang sama. (Munir, 2010)



Gambar 2.1 Tiga buah graf (G_1) graf sederhana, (G_2) graf ganda, (G_2) graf semu

2.1.2 Jenis-Jenis Graf

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

Berdasarkam ada tidaknya gelang atau sisi ganda suatu graf, maka graf digolongkan menjadi dua jenis :

a. Graf sederhana (*simple graph*)

Graf yang tidak mengandung gelang maupun sisi ganda dinamakan graf sederhana. Pada gambar 2.1, G_1 adalah contoh graf sederhana yang mempresentasikan jaringan komputer. Simpul menyatakan komputer, sedangkan sisi adalah saluran telepon untuk berkomunikasi. Saluran telepon dapat beroprasi pada dua arah.

Pada graf sederhana, sisi adalah pasangan tak-terurut (*unordered pairs*). Jadi, menuliskan sisi (u, v), sama saja dengan (v, u). Graf sederhana $G = (V, E)$ yang terdiri dari himpunan tidak kosong simpul-simpul dan E adalah himpunan pasangan tak terurut yang berbeda yang disebut sisi.

b. Graf tak sederhana

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak sederhana. Ada dua macam graf tak sederhana, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda adalah graf yang mengandung sisi ganda. Sisi ganda yang menghubungkan sepasang simpul bisa lebih dari dua buah. G_2 pada Gambar 2.1 adalah graf ganda. Sisi ganda dapat diasosiasikan sebagai pasangan tak terurut yang sama. Setiap graf sederhana adalah graf ganda, tertapi tidak setiap graf.

Graf semu adalah graf yang mengandung gelang (*loop*). G_3 adalah graf semu(termasuk bila memiliki sisi ganda sekalipun). Sisi gelang pada G_3 dapat dianggap sebagai saluran telepon tambahan yang menghubungkan komputer dengan dirinya sendiri.

Jumlah simpul pada graf disebut sebagai kardinalitas graf, dan dinyatakan dengan $n = |V|$, dan jumlah sisi kita nyatakan dengan $m = |E|$. Sebagai contoh, G_1mempunyai $n = 4$, dan $m = 4$, sedangkan G_2 mempunyai $n = 3$ dan $m = 4$.

Sisi pada graf mempunyai orientasi arah. Berdasarkan orientaei arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:

Berdasarkan jumlah simpul pada suatu graf, maka secara umum graf dapat digolongkan menjadi dua jenis :

a. Graf berhingga (*limited graph*)

   Graf berhingga adalah graf yang jumlah simpulnya *n* berhingga.

b. Graf tak berhingga (*unlimited graph*)

   Graf tak berhingga adalah graf yang jumlah simpulnya *n* tidak berhingga banyaknya (Purwanto, 2006).

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas dua jenis:

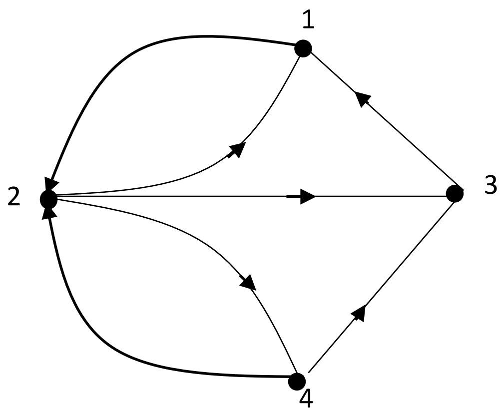
a. Graf tak berarah (*undirected graph*)

   Graf yang sisinya tidak mempunyai orientasi arah disebut sebagai graf tak berarah. Pada graf tak berarah urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(u, v) = (v, u)$ adalah sisi yang sama. Pada jaringan telepon, sisi pada graf berarah menyatakan bahwa saluran telepon dapat beroperasi pada dua arah.

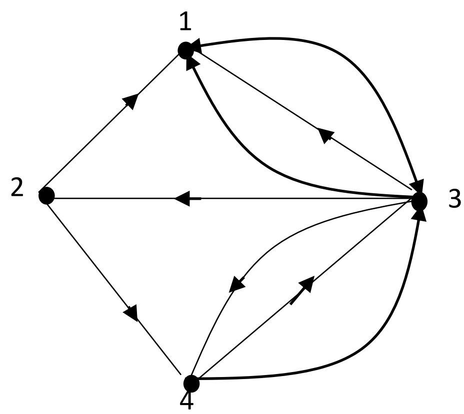
b. Graf berarah (*directed graph* atau *digraph*)

   Graf yang setiap sisinya diberikan orientasi arah disebut graf berarah. Sisi berarah biasa disebut dengan busur (*arc*). Pada graf berarah (u, v) dan

(v, u) menyatakan dua buah busur yang berbeda, dengan kata lain $(u, v) \neq (v, u)$ untuk busur (u, v), simpul u dinamakan simpul asal (*initial vertex*) dan simpul *v* dinamakan simpul terminal (*terminal vertex*). Graf berarah sering dipakai untuk menggambarkan aliran proses, peta lalu lintas suatu kota (jalan searah atau dua arah). Pada graf berarah, gelang diperbolehkan tetapi sisi ganda tidak (Munir, 2010). G_4 dan G_5 merupakan contoh grah berarah. Definisi graf dapat diperluas sehinga mencakup graf-ganda berarah (*directed multigraph*). Pada graf-ganda berarah, gelang dan sisi ganda diperbolehkan ada. G_5 pada Gambar 2.2 (b) adalah contoh graf-ganda berarah



(a) G_4



(b) G_5

Gambar 2.2 (a) graf berarah, (b) graf ganda berarah

Suatu graf berarah G terdiri dari: himpunan titik $V(G): \{v_1, v_2, \ldots, v_n\}$, himpunan garis $E(G): \{e_1, e_2, \ldots, e_n\}$, dan suatu F yang mengawankan setiap garis $E(G)$ ke suatu pasangan berurutan titik $(v_i,\ v_j)$. Jika $e_k = (v_i,\ v_j)$ adalah suatu garis dalam G, maka v_i disebut titik awal e_k dan v_j disebut titik akhir e_k. Arah garis adalah dari v_i ke v_j. Jumlah garis yang keluar dari titik v_i disebut derajat keluar titik v_i (simbol $d^+(v_i)$), sedangkan jumlah garis yang menuju ke titik v_i disebut derajat masuk titik v_i, yang disimbolkan dengan $d^-(v_i)$. Titik terasing adalah titik dalam G dimana derajat keluar dan derajat masuk adalah 0. Titik pendan adalah titik dalam G dimana jumlah derajat masuk dan derajat keluarnya adalah 1. Dua garis dikatakan paralel jika keduanya mempunyai titik awal dan titik akhir yang sama (Siang, 2002:226-227).

Graf yang himpunan sisinya merupakan himpunan kosong disebut sebagai graf kosong dan ditulis sebagia N_n yang dalam hal ini n adalah jumlah simpul. Loop berarah adalah sisi yang hanya berhubungan dengan satu simpul yang mempunyai arah tertentu.

2.1.3 Ketetanggaan (*adjacent*)

Dua buah simpul dikatakan bertetangga jika keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain v_j bertetangga dengan v_i jika (v_j, v_i) adalah sebuah sisi pada graf $E(G)$, Secara formal dinyatakan:

v_j bertetangga dengan v_i jika terdapat $e \epsilon E(G)$ sedemikian hingga $e = (v_j, v_i)$.

2.1.4 Bersisian (*incidency*)

Untuk sembarang sisi $e = (v_j, v_k)$, sisi e dikatakan bersisian dengan simpul v_j, atau sisi e bersisian dengan $v_{k.}$

2.1.5 Simpul Terpencil (*isolated vertex)*

Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengannya. Atau, dapat juga dinyatakan simpul terpencil adalah simpul yang tidak satupun bertetangga dengan simpul lainnya.

2.1.6 Graf Kosong *(Null Graph* atau *Empty Graph)*

Graf yang himpunan sisinya merupakan himpunan kosong disebut sebagai graf kosong dan ditulis N_n, yang dalam hal ini n adalah jumlah simpul

2.1.7 Derajat (*degree*)

Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Dinotasikan dengan $d(v)$, simpul yang mempunyai loop dihitung dua sisi yang bersisian dengannya, sehingga derajat dari simpul yang memuat loop adalah dua. Sedara umum, jika terdapat g buah loop dan e buah sisi bukan loop yang bersisian dengan sisi v adalah $d(v) = 2g + e$. Alasan mengapa loop mengkontribusikan dua untuk derajat simpulnya adalah karena loop direpresentasikan sebagai (v, v), dan simpul v versisian dua kali pada sisi (v, v).

Simpul yang berdarajat satu disebut **anting-anting** (*pendant vertex)*. Dengan kata lain, anting-anting hanya bertetangga dengan sebuah simpul.

Pada graf berarah, derajat suatu simpul dibedakan menjadi dua macam untuk mencerminkan jumlah busur dengan simpul tersebut sebagai simpul asal dan jumlah busur dengan simpul tersebut sebagai simpul terminal.

Derajat simpul v dinyatakan dengan $d_{in}(v)$ dan $d_{out}(v)$ yang dalam hal ini

$d_{in}(v)$ = derajat masuk (*in degree*) = jumah busur yang masuk ke simpul v.

$d_{out}(v)$ = derajat keluar (*out degree*) = jumah busur yang keluar ke simpul v.

Dan $d(v) = d_{in}(v) + d_{out}(v)$

2.1.8 Graf bagian (*Subgraf*)

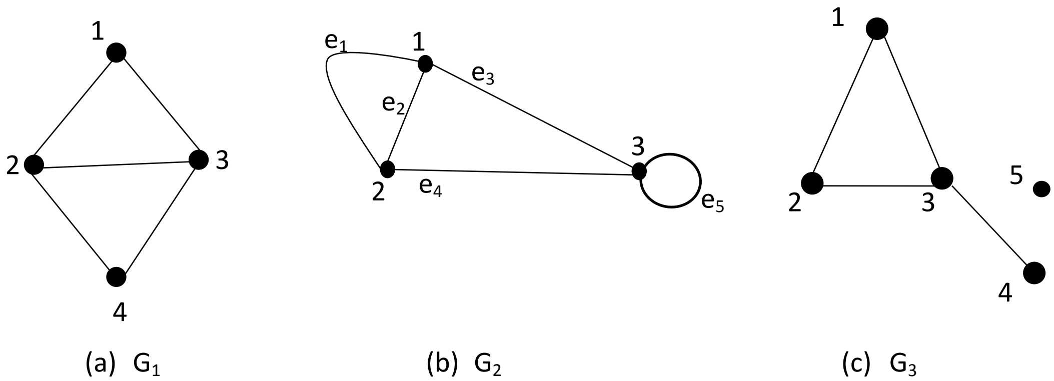
Graf bagian (subgraf) dari G adalah suatu graf yang setiap titiknya adalah anggota $V(G)$ dan setiap sisinya adalah anggota $E(G)$, atau jika $V(H) \subseteq V(G)$dan

$E(H) \subseteq E(G)$. Jika H suatu graf bagian dari G dan $V(H) = V(G)$, maka H disebut graf bagian rentangan (*spaning subgraf*) dari G.

2.1.9 Terhubung (*connected*)

Dua buah simpul v_1 dan simpul v_2 pada graf $G(V, E)$ disebut terhubung jika terdapat lintasan dari v_1 ke v_2 dan terdapat lintasan dari v_2 dan v_1. Jika setiap pasang simpul di dalam graf terhubung, maka graf tersebut dapat dikatakan sebagai graf terhubung.

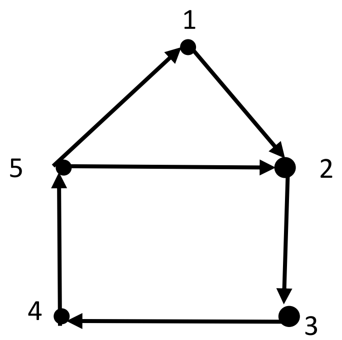
Graf tak berarah G disebut graf terhubung jika untuk setiap pasang simpul v_i dan v_j dalam himpunan V terdapat lintasan dari v_i ke v_j (yang juga harus berarti ada lintasan dari v_j ke v_i) jika tidak, maka G disebut graf tak terhubung. Sebagai catatan, setiap simpul terhubung dengan dirinya sendiri. Jadi, graf yang hanya terdiri atas satu simpul saja (tidak ada sisi), juga dikatakan graf terhubung. G_1 dan G_2 pada gambar 2.3 adalah graf terhubung, sedangkan G_3 tidak terhubung. (Munir, 2010:371).



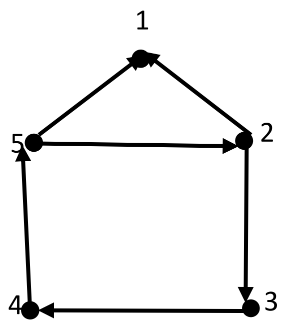
Gambar 2.3. Graf terhubung dan graf tak terhubung

Pada graf berarah $G(V, E)$ dikatakan terhubung jika graf tak-berarahnya terhubung (graf tak-berarah dari G diperoleh dengan menghilangkan arahnya).

Keterhubungan dua buah simpul pada graf berarah $G(V, E)$ dibedakan menjadi terhubung kuat dan terhubung lemah. Dua simpul, u dan v, pada graf berarah G disebut terhubung kuat (*strongly connected*) jika terdapat lintasan berarah dari u ke v dan juga lintasan berarah dari v ke u. Pada Gambar 2.4 (a) simpul 1 dan simpul 3 terhubung kuat karena terdapat lintasan dari 1 ke 3 (yaitu 1, 2, 3), begitu juga terdapat lintasan dari 3 ke 1 (yaitu 3, 2, 1),



Gambar 2.4 (a) Terhubung lemah



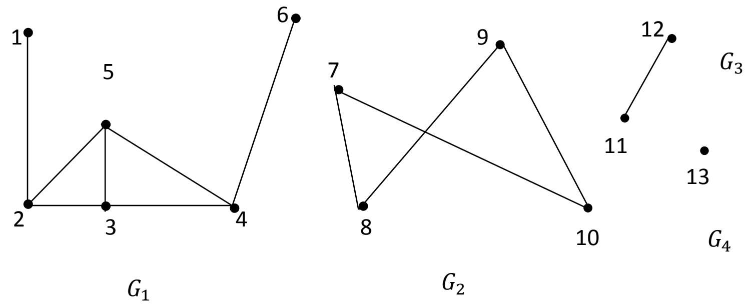
Gambar 2.4 (b) Terhubung kuat

Jika u dan v tidak terhubung kuat tetapi terhubung pada graf tidak berarahnya, maka u dan v dikatakan terhubung lemah (*weakly connected*). Pada Gambar 2.4 (b) simpul 1 dan simpul 3 terhubung lemah karena hanya terdapat lintasan dari 3 ke 1 (yaitu 3, 4, 5, 1), tetapi tidak ada lintasan dari 1 ke 3.

Kedua hal diatas (terhubung kuat dan terhubung lemah) melahirkan definisi graf terhubung kuat. Graf berarah G disebut graf terhubung kuat (*strongly connected graph*) apabila untuk setiap pasang simpul sembarang v_i dan v_j di G terhubung kuat. Kalau tidak, G disebut terhubung lemah. Graf pada Gambar 2.4 (a) adalah graf terhubung kuat, karena untuk sembarang sepasang simpul di dalam graf terdapat lintasan, sedangkan pada Gambar 2.4 (b) adalah terhubung lemah karena tidak semua pasangan simpul mempunyai pasangan dari dua arah (Munir, 2010:372).

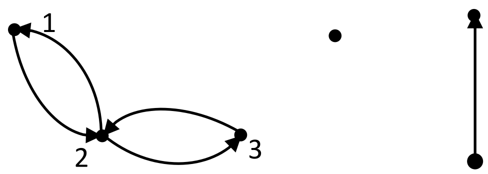
2.1.10 Komponen Graf

Komponen graf (*connected component*) adalah jumlah maksimum pada graf terhubung dalam graf G. Ini berarti setiap komponen terhubung di dalam graf G saling lepas (*disjoint*). Graf G di bawah ini mempunyai 4 buah komponen.



Gambar 2.5. Graf G yang mempunya 4 komponen, yaitu G_1, G_2, G_3 dan G_4

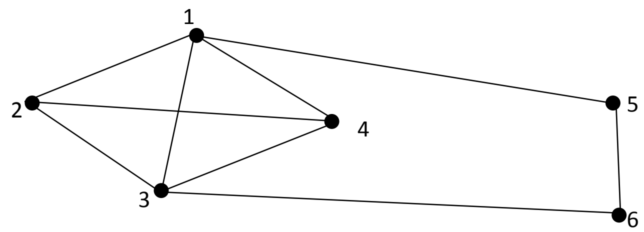
Pada graf berarah, komponen terhubung kuat (*strongly connected component*) adalah jumlah maksimum upagraf yang terhubung kuat. Sedangkan definisi dari upagraf, misalkan $G = (V, E)$ adalah sebuah graf, $G_1 = (V_1, E_1)$ adalah upagraf (*subgraph*) dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$. Graf pada Gambar 2.6 dibawah ini mempunyai dua komponen terhubung kuat, yaitu upagaraf dengan simpul 1, 2, 3 dan upagraf yang hanya mempunyai satu simpul, 4.



Gambar 2.6. Graf berarah G yang mempunyai mempunyai 2 buah komponen terhubung kuat

2.1.11 Lintasan (*Path)*, Jalan (*Walk*), Jejak (*Trail*), Sirkuit (*Circuit*), dan Siklus (*Cycle*)

Lintasan yang panjangnya n dari simpul v_0 ke v_n dalam graf G adalah barisan berselang-selang simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \ldots, e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G. Pada Gambar 2.7 lintasan 1, 2, 4 dan 3 adalah lintasan dengan barisan sisi (1, 2), (2, 4), dan (4, 3). Istilah lain untuk lintasan adalah jalur. Simpul dan sisi yang dilalui di dalam lintasan boleh berulang. Sebuah lintasan dikatakan lintasan sederhana (*simple path*) jika semua simpulnya berbeda (setiap sisi yang dilalui hanya satu kali).

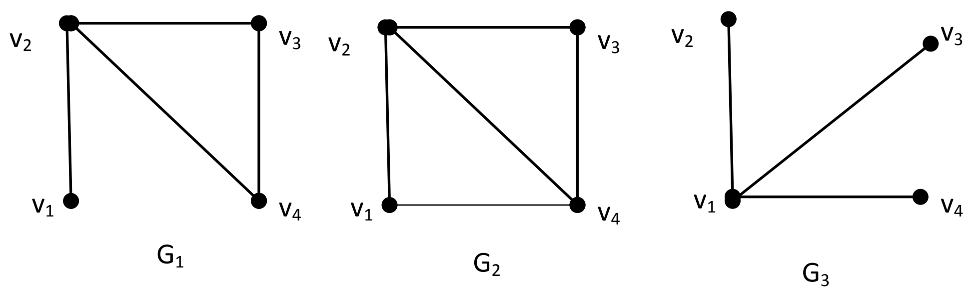


Gambar 2.7. Graf dengan lintasan

Lintasan terbuka (*open walk*) adalah lintasan yang tidak berawal dan berakhir pada simpul yang sama (Munir, 2010:370). Pada Gambar 2.7 lintasan 1, 2, 3, dan 4 merupakan lintasan terbuka.

Lintasan terrtutup (*closed walk*) adalah lintasan yang berawal dan berakhir pada simpul yang sama (Munir, 2010:370). Lintasan tertutup disebut juga siklus/ cycle. Pada gambar 2.7 lintasan 1, 2, 4, dan 1 merupakan lintasan tertutup.

Lintasan dan siklus dalam graf berarah sama dengan lintasan dan siklus dalam graf tak berarah. Hanya saja dalam graf tak berarah, perjalanan yang dilakukan harus mengikuti arah garis. Untuk membedakan dengan graf tak berarah, maka lintasan dan siklus dalam graf berarah disebut lintasan berarah dan siklus berarah. Suatu graf berarah yang tidak memuat sirkuit berarah disebut graf berarah asklik. Panjang lintasan adalah jumlah sisi dalam lintasan tersebut. Lintasan 1, 2, 3, 4 pada Gambar 2.7 memiliki panjang 3. Lintasan yang berawal dan berakhir pada simpul yang sama disebut siklus (*Cycle*) atau sirkuit (*Circuit*). Lintasan yang melalui setiap simpul di dalam graf tepat satu kali disebut lintasan hamilton, dan sirkuit hamilton adalah graf sirkuit yang mengunjungi tiap simpul pada graf terhubung G tepat satu kali, kecuali simpul awal (yang jugamerupakan simpul akhir) dilewati dua kali.



Gambar 2.8. Contoh Sirkuit Hamilton dan Lintasan Hamilton

Pada Gambar 2.8, graf G_1 merupakan contoh lintasan Hamilton $W = \{v_1, v_2, v_3, v_4\}$, graf G_2 adalah contoh sirkuit Hamilton dengan rute $W = \{v_4, v_1, v_2, v_3, v_4\}$, sedangkan graf G_3 tidak memiliki lintasan Hamilton dan sirkuit Hamilton.

Jalan (*Walk*) adalah suatu barisan yang suku-sukunya berupa simpul dan rusuk yang diurutkan secara bergantian sedemikian hingga rusuk ujung e_i adalah simpul

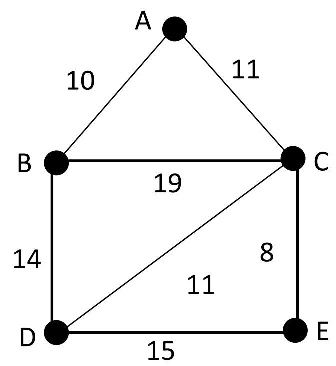
v_{i-1} dan v_i jika dimisalkan graf G dengan rusuk $V = \{v_1, v_2, v_3, \dots, v_k\}$ dan simpul $E = \{e_1, e_2, e_3, \dots, e_k\}$, yang membentuk barisan berhingga $W = \{v_1, e_1, v_2, e_2, v_3, e_3, \dots, v_k, e_k\}$. Jalan tanpa rusuk berulang disebut jejak (*Trail*).

2.1.12 Graf Berbobot

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak dua buah kota, biaya perjalanan antara dua buah kota, waktu tempuh pesan dari sebuah simpul komunikasi ke simpul komunikasi lain dan sebagainya. Graf berbobot juga sering dikaitkan dengan istilah graf berlabel.

Syarat-syarat dalam graf berbobot.

- Jika setiap busur mempunyai nilai yang menyatakan hubungan antara 2 buah simpul, maka busur tersebut dinyatakan memiliki bobot.
- Bobot sebuah busur dapat menyatakan panjang sebuah jalan dari 2 buah titik, jumlah rata-rata kendaraan perhari yang melalui sebuah jalan, dll.



Gambar 2.9 Graf berbobot

Istilah lain yang sering dikaitkan dengan graf berbobot adalah graf berlabel. Namun graf berlabel sesungguhnya lebih luas lagi definisinya. Label tidak hanya diberikan pada sisi, tetapi juga pada simpul. Sisi diberi label berupa bilangan tak-negatif, sedangkan simpul diberi label berupa data lain. Misalnya pada graf yang memodelkan kota-kota, simpul diberi nama kota-kota, sedangkan label pada sisi menyatakan jarak antara kota-kota.

2.2 Jaringan (Network)

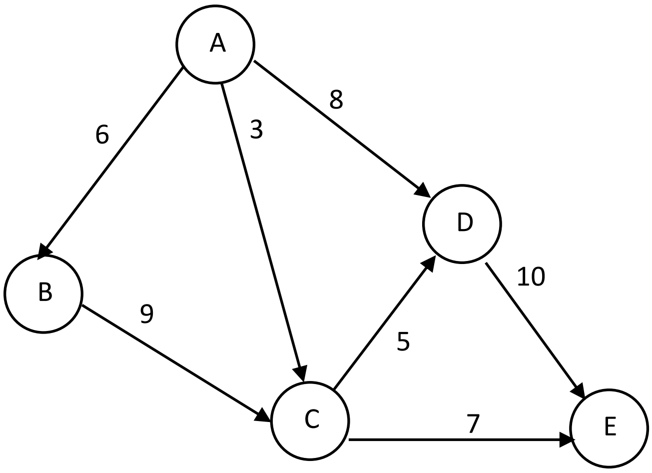
Network $N(V, E)$ adalah sebuah graf berarah yang terdiri dari himpunan titik V dan himpunan busur E yang tiap sisinya memiliki kapasistas atau bobot dan pada tiap sisi tersebut terdapat arus (*flow*) yang mengalir antara 2 simpul yang mengapit sisi tersebut. Jumlah arus yang mengalir pada tiap sisi harus lebih kecil atau sama dengan kapasistas sisi tersebut. Pada aplikasinya, sebuah graf berarah sering disebut *network.* Setiap arus (*flow*) yang ada dalam network harus memenuhi sebuah batasnya yaitu arus yang masuk pada suatu simpul harus sama dengan arus yang keluar pada simpul tersebut, kecuali pada titik sumber (*source*), yang keluarnya lebih besar dari arus masuk, dan titik tujuan (*sink*), yang arus masuknya lebih besar dari arus keluar sebuah *network* biasanya digunakan untuk memodelkan sistem lalu lintas, saluran pipa, dan saluran listrik.

Jaringan transportasi adalah sebuah graf berarah yang sederhana dengan setiap sisi mempunyai kapasitas dengan sejumlah syarat sebagai berikut:

(1) Terdapat satu simpul di dalam graf itu yang tidak mempunyai sisi masuk disebut dengan sumber.

(2) Terdapat satu simpul di dalam graf itu yang tidak mempunyai sisi keluar disebut dengan tujuan.

(3) Pembobot setiap sisi $C(i,j)$ dari suatu sisi berarah (i,j) nmerupakan sebuah bilangan real non negatif disebut dengan kapasitas sisi (i,j). (Johnsonbaugh, 1986).



Gambar 2.10. Jaringan Transportasi

Pada Gambar 2.10 pembobot setiap $sisi\ (A,B) = 6, (A,D) = 8, (A,C) = 3, (B,C) = 9, (C,D) = 5, (C,E) = 7, (D,E) = 10.$

(4) Arus (*flow*) pada network, harus:

- Arus yang mengalir $\leq$ kapasitas sisi yang dialiri.
- Untuk setiap titik pada graf, kecuali titik sumber dan titik tujuan berlaku selisih arus yang keluar dan arus yang masuk sama dengan nol.
- Jumlah arus yang keluar dari titik sumber sama dengan jumlah arus yang masuk ke titik tujuan.

Network banyak dipakai dalam banyak hal untuk kegunaan yang berbeda-beda. Jaringan transportasi, jaringan listrik dan jaringan telekomunikasi adalah contoh-contoh dimana *network* ditemukan dalam kehidupan sehari-hari. Representasi *network* juga dipakai dalam produksi, distribusi, *project planning*, penempatan fasilitas, dan *financial planning*. Suatu *network* diperlukan karena memberi gambaran visual dan bantuan konseptual yang lebih jelas untuk mengetahui hubungan antar komponen dalam sistem yang sering dijumpai dalam banyak kasus. Banyak permasalahan pada *network* yang sebenarnya berbentuk *linear programing.* Macam-macam permasalahan *network* antara lain:

a. *Shortest-Path Problem*
b. *Minimum Spanning Tree Problem.*
c. *Maximum Flow Problem*
d. Minimum Cost Flow Problem

2.3 Aliran Maksimum (*Maximum Flow Problem*)

Maximum flow problem dideskripsikan sebagai masalah pencarian untuk mencari arus maksimum yang dapat mengalir pada sebuah *network* yang hanya memiliki satu sumber dan satu tujuan. *Maximum flow* adalah sebuah model yang dapat digunakan untuk mengetahui nilai maximum seluruh arus di dalam sebuah sistem jaringan. Contoh *maximum flow* pada sebuah jaringan adalah jaringan listrik, saluran pipa, dan lintasan lalu lintas. Kapasitas pada setiap jaringan akan membatasi jumlah arus atau aliran yang melewatinya. *Maximum flow* mempunyai tujuan untuk memaksimalkan jumlah arus yang melewati jaringan dalam sebuah sistem jaringan. Hal ini tentunya sangat umum terjadi pada bidang transportasi, produksi, komunikasi, dan distribusi (Fakhri, 2008)

Pada Aliran Maksimum, sering dijumpai istilah sebagai berikut:

- *Network*

  Seperti terdapat dalam 2.2.

- *Flow (F)*

  Misalkan N adalah sebuah jaringan dengan titik sumber s dan titik tujuan t. Jika v adalah sebuah titik di N, maka himpunan semua busur N yang keluar dari titik v (meninggalkan v) dilambangkan dengan $O(v)$ dan himpunan semua busur N yang menuju titik v dilambangkan dengan $I(v)$.

  Pada suatu jaringan $N = (V(N), E(N))$ adalah sebuah graf berarah sederhana terhubung lemah yang setiap unsurnya dikaitkan dengan bilangan real non negatif, dengan simpul awal s, simpul akhir t dan kapasitas dari busur (i,j) adalah $C(i,j)$, arus (flow) pada jaringan berarah merupakan fungsi yang mengawankan busur (i,j) di N ke sebuah bilangan non negatif yang memenuhi syarat-syarat berikut:

  (i) $0 \leq F(i,j) \leq C(i,j)$ untuk semua $(i,j) \in E(N)$

  (ii) $\sum_{(i,j)\in O(s)} F(i,j) = \sum_{(i,j)\in I(t)} F(j,i)$

  (iii) $\sum_{(i,j)\in O(x)} F(i,j) = \sum_{(i,j)\in I(x)} F(j,i)\ \forall x \epsilon V(N) - \{s,t\}$

  Syarat (i) menyatakan arus pada busur tidak pernah melampaui kapasitas busur dan persamaan tersebut disebut kendala kapasitas untuk masalah arus maksimum. Syarat (ii) menyatakan bahwa totoal nilai $flow$ yang keluar dari titik sumber s sama dengan total nilai $flow$ yang sampai di titik tujuan. Nilai yang selanjutnya disebut “nilai $flow\ f$” dari s ke t pada jaringan N. Syarat

(iii) menyatakan bahwa untuk setiap titik antara pada N berlaku total $flow$ yang meninggalkan x sama dengan total $flow$ yang menuju titik x

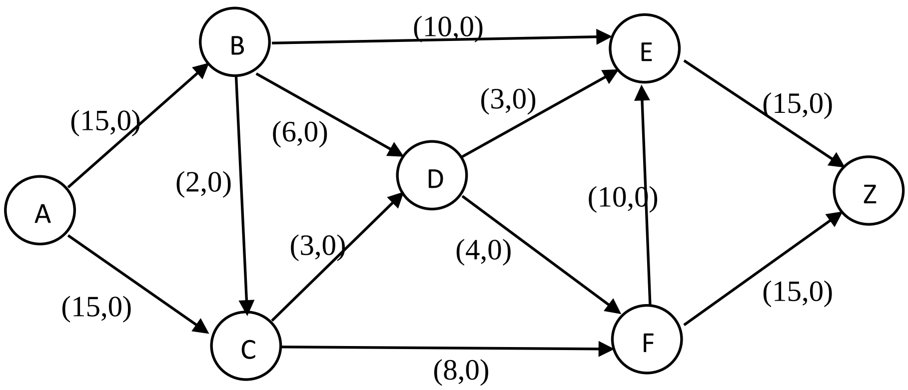
- *Residual Network*

  Residual Network merupakan jaringan yang mempunyai jumlah simpul yang sama dengan jaringan awal dengan ketentuan pelabelan sisinya adalah sebagai berikut: $C'(i,j) = C(i,j) - F(i,j)$

Prosedur utnuk mencari maksimum flow adalah sebagai berikut:

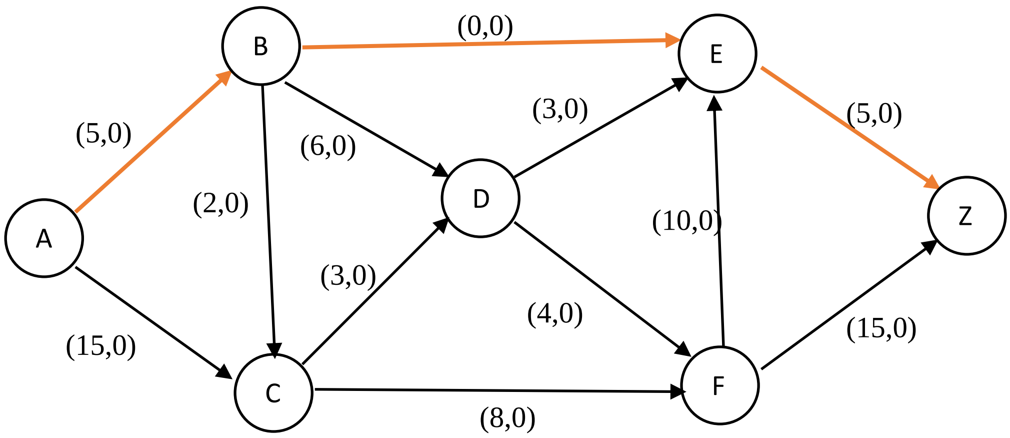
1. Cari dan temukan path dari titik sumber ke titik lokasi tujuan yang memiliki arah dengan aliran kapasitas yang lebih besar dari nol untuk seluruh segitiga di dalam path. Jika tidak ada path yang tersedia, berarti optimal solution telah tercapai.
2. Cari di aliran kapasitas yang paling kecil (S_f) di dalam path yang terpilih di Step 1. Lakukan perubahan aliran di dalam jaringan dengan mengirimkan sejumlah (S_f).
3. Untuk path yang terpilih di Step 1, kurangkan seluruh arus kapasitas dengan (S_f) di simpul arah masuk dan tambahkan di arus balik simpul sebesar (S_f)
4. Ulangi Step 1

Pencarian aliran maksimum dapat diselesaikan dengan beberapa cara diantaranya adalah dengan *max-flow min-cut Theorm,* algoritma Djikstra dan algoritma Ford-Fulkerson. Sebagai contoh tentukan aliran maksimal pada Gambar 11 dibawah ini:



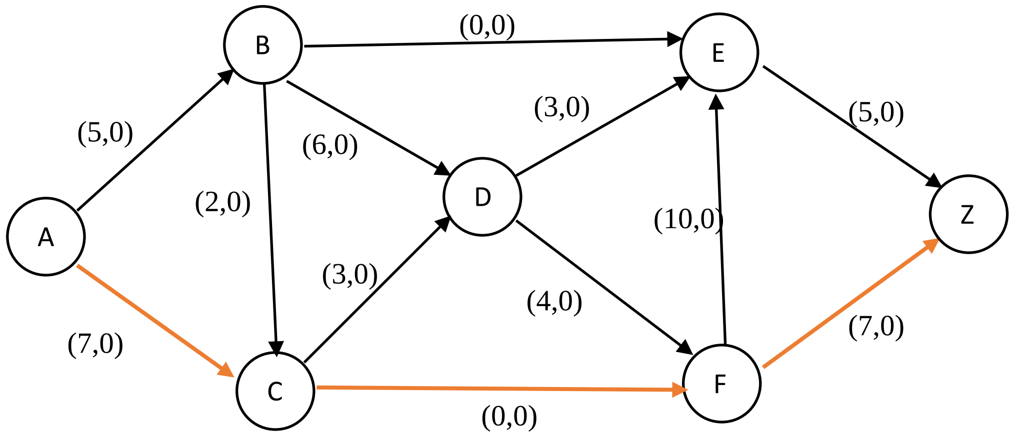
Gambar 2.11. Graf awal (Farizal, 2013)

Pilih lintasan dari sumber ke tujuan untuk memulai iterasi pertama, misalkan kita pilih lintasan $A - B - E - Z$. Tentukan nilai terkecil dari lintasan ini yaitu $min\,\{15, 10, 15\} = 10$, ditemukan pada sisi $B – E$. Sehingga nilai yang digunakan $F = 10$. Nilai kapasitas $B - E$ telah 0, s ehingga untuk pencarian berikutnya sisi $B - E$ sudah tidak dapat dilewati lagi.



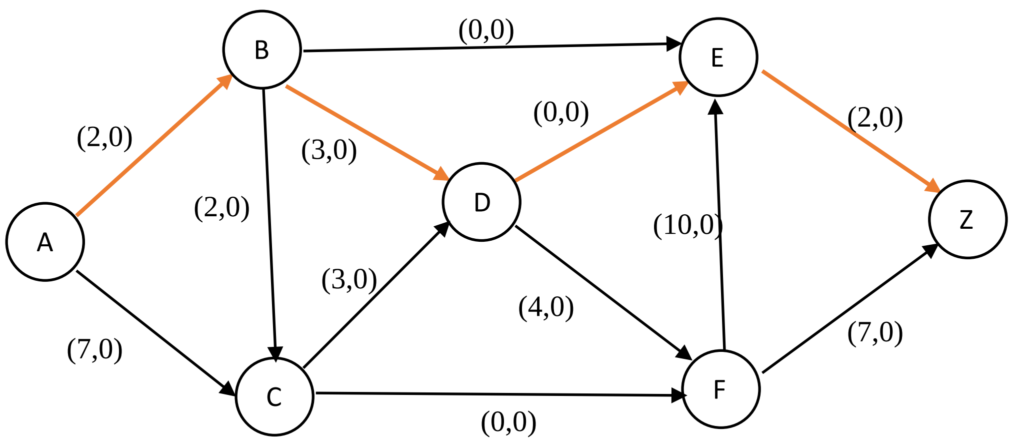
Gambar 2.12. Lintasan A-B-E-Z

Lintasan kedua pada Gambar 2.13 yaitu $A - C - F - Z$. Nilai terkecil dari lintasan ini adalah $min\,\{15, 8, 15\} = 8$. Jadi nilai yang digunakan$F = 8$. Nilai kapasitas $C - F$ menjadi 0, sehingga untuk pencarian berikutnya sisi $C - F$ sudah tidak ditelusuri lagi.



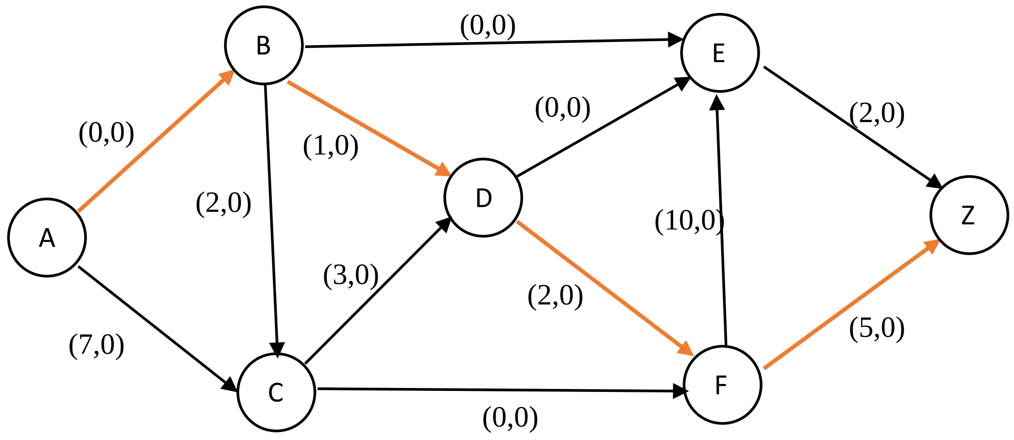
Gambar 2.13. Lintasan A-C-F-Z

Lintasan ketiga yaitu $A - B - D - E - Z$ dengan nilai kapasitas terkecil yaitu $min\ \{5, 6, 3, 5\} = 3$. Jadi nilai F yang digunakan adalah $F = 3$. Sehingga kapasitas sisi $D - E$ telah 0, sehingga untuk pencarian berikutnya sisi $D - E$ sudah tidak dapat dilewati lagi. Gambar 2.14 merupakan iterasi ketiga dengan nilai kapasitas masing-masing sisi.



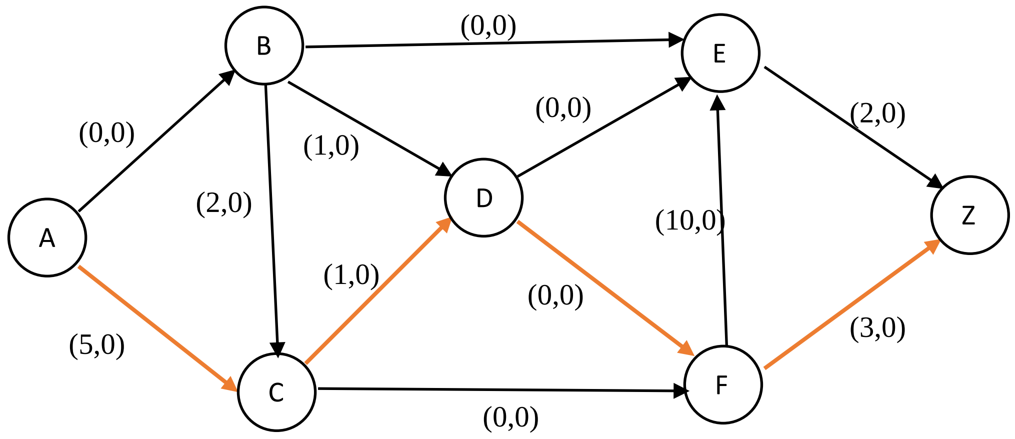
Gambar 2.14. Lintasan A-B-D-E-Z

Lintasan keempat yaitu $A - B - D - E - Z$. Nilai terkecil dari lintasan ini adalah $min\,\{2, 3, 4, 7\} = 2$. Jadi nilai yang digunakan $F = 2$. Karena nilai kapasitas $A - B$ adalah 2, maka lintasan ini menjadi 0, sehingga untuk pencarian berikutnya sisi $A - B$ sudah tidak ditelusuri lagi.



Gambar 2.15. Lintasan $A - B - D - F - Z$

Lintasan $A - C - D - F - Z$ merupakan lintasan kelima dari Gambar 2.16, nilai terkecil dari lintasan ini adalah $min\,\{7, 3, 2, 5\} = 2$. Jadi nilai F yang digunakan adalah $F = 2$. Sehingga kapasitas sisi $D - F$ telah 0, sehingga untuk pencarian berikutnya sisi $D - F$ sudah tidak dapat dilewati lagi.



Gambar 2.16. Lintasan $A - C - D - F - Z$

Karena nilai kapasitas sisi $A - B,\ D - E,\ D - F$ sudah 0, maka tidak ada lintasan yang dapat dilewati lagi untk menuju ke Z. Sehingga pencarian berhenti. Graf dari Gambar 2.11 memiliki *maximum flow* sebesar:

$$Lintasan\,1\ +\ lintasan\,2\ +\ lintasan\,3\ +\ lintasan\,4\ +\ lintasan\,5$$

$$=\ 10\ +\ 8\ +\ 3\ +2\ +\ 2\ =\ 25.$$

Secara umum *Maximum flow* dapat dijelaskan sebagai berikut:

a. Semua aliran yang melalui sebuah jaringan yang berarah dan terhubung dari simpul awal ke simpul akhir.
b. Simpul awal disebut sumber (*source*) dan simpul akhir disebut tujuan (*sink*). Simpul sisa yang lain dinamakan simpul antara.
c. Aliran hanya diperbolehkan ke arah yang ditunjukkan oleh anak panah dalam suatu cabang. Pada simpul awal, semua cabang meninggalkan simpul, sedangkan pada simpul akhir, semua cabang mengarah masuk ke simpul.
d. Tujuan maximum flow adalah memaksimalkan aliran dari sumber ke tujuan. Jumlah yang diangkut ini bisa dikatakan jumlah yang meninggalkan sumber atau jumlah yang sampai pada tujuan.
e. Kapasitas setiap jaringan akan membatasi jumlah arus atau aliran yang melewatinya. Sebagai contoh, sebuah pipa dengan kapasitas 9 kubik akan pecah apabila kita memaksa pipa itu dilewati oleh air sebesar 30 kubik pada tingkat kapasitas yang sama.

Maximum flow sering diterapkan dalam kehidupan sehari-hari, berikut contoh aplikasi *maximum flow* dikehidupan sehari-hari:

a. Maksimasi aliran minyak

b. Maksimasi aliran air
c. Maksimasi aliran kendaraan
d. Maksimasi aliran listrik
e. Maksimasi jaringan distribusi barang.

2.4Algoritma *Edmonds-Karp*

Algoritma *Edmonds-Karp* pertama kali diperkenalkan oleh seorang ilmuan Rusia, Dinic pada tahun 1970, dan dipopulerkan oleh Jack Edmonds dan Richard Karp pada tahun 1972 (Dey, 1993). Dalam algoritma *Edmonds-Karp,* lintasan penambah yang dipilih merupakan lintasan penambah terpendek. Untuk memilih lintasan penambah terpendek yang dimaksud, digunakan algoritma *Breadth First Search* (BFS). Sehingga hanya diperlukan sedikit iterasi dalam pencarian *maximum flow* dalam suatu permasalahan yang ada dibandingkan dengan algoritma *Ford Fulkerson.* Algoritma ini merupakan penyempurnaan dari algoritma *Ford Fulkerson.*

Adapun langkah-langkah dari algoritma Edmonds-Karp adalah sebagai berikut:

1) Input graf $G = (V_i, E_j)$, kapasitas setiap sisi $C(i, j)$ dan inisialisasi *flow* untuk busur (i, j) sebesar nol $F(i, j) = 0$
2) Identifikasi suatu lintasan penambah pada *residual network* dengan menggunakan algoritma BFS (*Breadth First Search*)

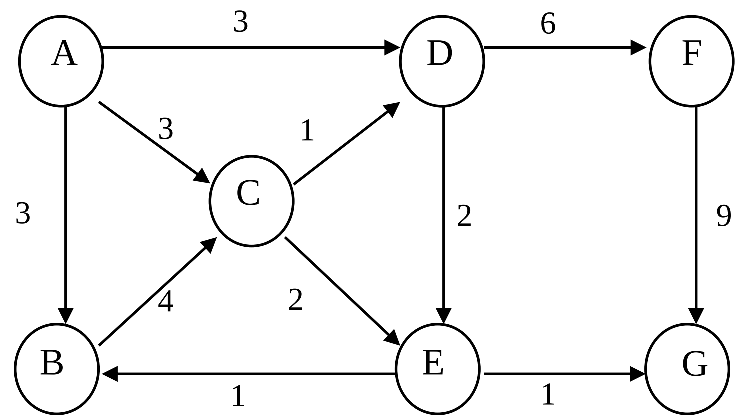
   Yang dimaksud dengan lintasan penambah (*augmenthing path*) adalah suatu lintasan berarah dari simpul awal ke tujuan pada setiap sisi dalam lintasan memiliki kapasitas yang positif.

3) Jika telah diperoleh suatu lintasan penambah, maka tentukan kapasitas residu lintasan penambah tersebut yang dinotasikan dengan Δ. Kapasitas residu adalah kapasitas sisi yang minimum pada lintasan penambah.
4) Tentukan *residual network* dengan tahapan sebagai berikut:
   - Tambahkan *flow* sebesar Δ ke setiap busur pada lintasan penambah.
   - Mengurangkan kapasitas sebesar Δ ke setiap busur pada lintasan penambah.
5) Jika masih ada lintasan penambah yang lain, ulangi langkah 2. Jika tidak ada lintasan penambah yang lain, hitung aliran pada setiap busur.

Adapun langkah-langkah untuk mencari lintasan penambah menggunakan algoritma *Breadth First Search* adalah sebagai berikut:

a. Masukkan simpul awal kedalam antrian (*Queue)*. Antrian ini berisi simpul yang akan dicari simpul-simpul yang bertetangga dengan simpul tersebut.
b. Periksa semua simpul yang bertetangga dengan simpul awal, dengan bantuan memasangkan simpul-simpul yang bertetangga dengan simpul awal (*Parent Map*).
c. Periksa apakah simpul yang bertetangga sudah berpasangan dengan simpul tujuan.
d. Jika simpul sudah berpasangan dengan simpul tujuan, pencarian selesai.
e. Jika simpul belum berpasangan dengan simpul tujuan, masukkan simpul yang bertetangga (simpul anak) ke dalam antrian untuk dicari simpul yang bertetangga lagi.
f. Ulangi pencarian dari langkah kedua sampai menemukan simpul tujuan.

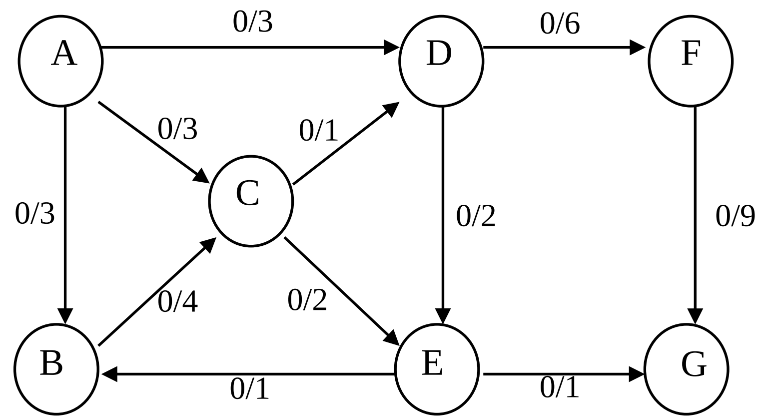
Sebagai contoh diberikan sebuah *network* dengan 7 titik, *source* A, *sink* G dan kapasitas diberikan sebagai berikut:



Gambar 2.17. *Network* dengan 7 titik

Dalam pasangan F/C yang tertulis pada sisi-sisi, F adalah flow dalam satu sisi, dan C adalah kapasitas.

- Langkah 1 : Inisiasi flow untuk setiap busur (i,j) pada N sebesar nol $(F(i,j) = 0)$



Gambar 2.18. *Network* dengan 7 titik dengan $F(i,j) = 0)$

- Langkah 2 : identifikasi suatu lintasan penambah dengan menggunakan algoritma BFS

1. lintasan penambah 1

Antrian (Queue)	Simpul Tetangga	Parent Map (Path)
A	(A, B) (A, C) (A, D)	$A - B$ $A - C$ $A - D$
B C D	(B, C) (C, D) (C, E) (D, E) (D, F)	$A - B - C$ $A - C - D$ $A - C - E$ $A - D - E$ $A - D - F$
$\boldsymbol{E}$	$(\boldsymbol{E}, \boldsymbol{G})$	$\boldsymbol{A - C - E - G}$

Maka ditemukan jalur tambahan yaitu $A - C - E - G$

2. Lintasan penambah 2

Antrian (Queue)	Simpul Tetangga	Parent Map (Path)
A	(A, B) (A, C) (A, D)	$A - B$ $A - C$ $A - D$
B C D	(B, C) (C, D) (C, E) (D, E) (D, F)	$A - B - C$ $A - C - D$ $A - C - E$ $A - D - E$ $A - D - F$
$\boldsymbol{F}$	$(\boldsymbol{F}, \boldsymbol{G})$	$\boldsymbol{A - D - F - G}$

Maka ditemukan jalur tambahan yaitu $A - D - F - G$

1. Lintasan penambah 3

Antrian (Queue)	Simpul Tetangga	Parent Map (Path)
A	(A, B) (A, C)	$A - B$ $A - C$
B C	(B, C) (C, D) (C, E)	$A - B - C$ $A - C - D$ $A - C - E$
D E	(D, E) (D, F) (E, D)	$A - C - D - E$ $A - C - D - F$ $A - C - E - D$
$\boldsymbol{F}$	$(\boldsymbol{F}, \boldsymbol{G})$	$\boldsymbol{A - C - D - F - G}$

Maka ditemukan jalur tambahan yaitu A - C- D – F – G.

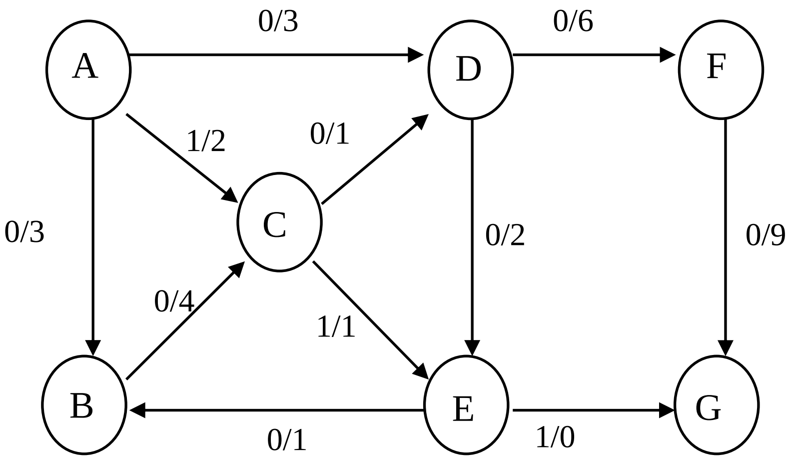
- Langkah 3: Jika telah diperoleh suatu lintasan penambah, maka tentukan kapasitas residu lintasan penambah tersebut yang dinotasikan dengan Δ, dan
- Langkah 4: Tambahkan *flow* sebesar Δ ke setiap busur pada lintasan penambah tersebut.

1. Pada jalur: $A - C - E - G$.

Dari setiap sisi dari jalur ini pada Gambar 2.17, ditemukan nilai terkecil yang dimiliki oleh sisi tersebut, sehingga nilai yang digunakan menjadi $F = 1$.

Flow : 1
Max_flow : 1 +
$Min\ (C\ (A, C), C(C, E), C(E, G)\ =$
$Min\ (3, 2, 1) = \min(3, 2, 1) = 1$
Tambah flow sebesar 1dan kurang kapasitas sebesar 1 ke setiap sisi pada lintasan

$A - C - E - G$



Gambar 2.19. *Hasil jaringan A – C – E - G*

2. Pada jalur: A – D – F – G.

Dari setiap sisi dari jalur ini, ditemukan nilai terkecil yang dimiliki oleh sisi tersebut, sehingga nilai yang digunakan menjadi $F = 3$.

Flow : 3

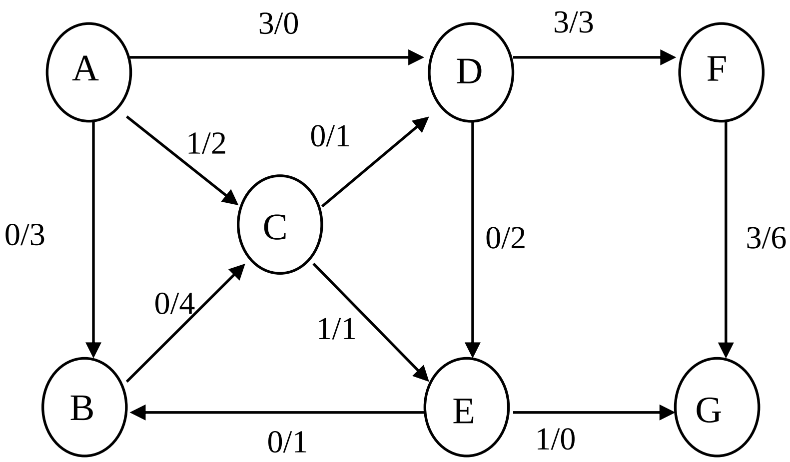
Max_flow : 1 + 3 +

$Min\ (C\ (A, D), C(D, F), C(F, G)\ =$

$Min\ (3, 6, 9)\ =\ min\ (3, 6, 9)\ =\ 3$

Tambah flow sebesar 3 dan kurang kapasitas sebesar 3 ke setiap sisi pada lintasan

A – D – F - G



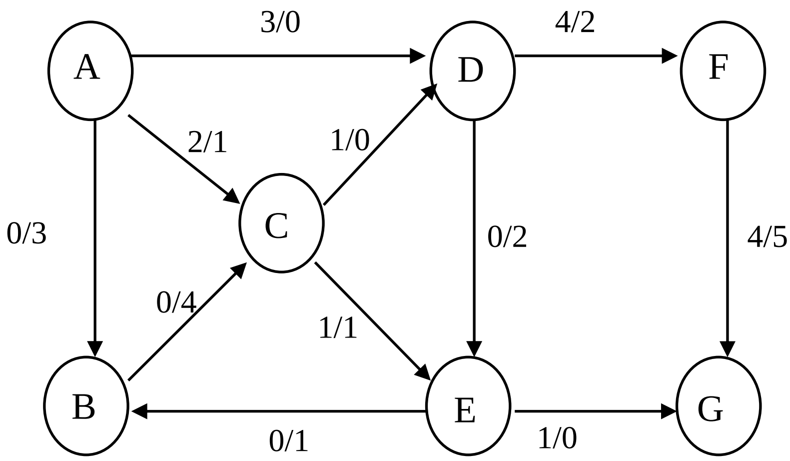
Gambar 2.20. *Hasil jaringan A – D – F - G*

1. Pada jalur $A - C - D – F – G$

Dari setiap sisi dari jalur ini, ditemukan nilai terkecil yang dimiliki oleh sisi tersebut, sehingga nilai yang digunakan menjadi $F = 1$.

Flow : 1
Max_flow : 1 + 3 + 1 +
$Min\ (C\ (A,C), C(C,D), C(D,F), C(F,G)\ =$
$Min\ (3,4,1\ 4,7)\ =\ min\ (3,4,1,4,7)\ =\ 1$
Tambah flow sebesar 1 dan kurangi kapasitas sebesar 1 ke setiap sisi pada lintasan

$A - C - D – F – G$



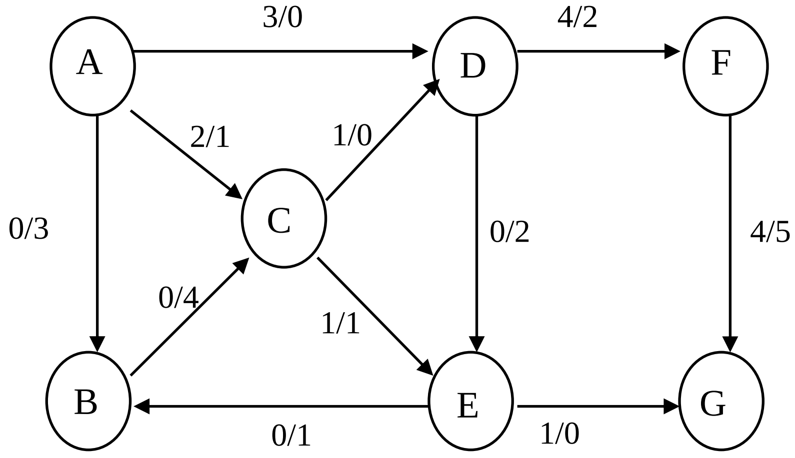
Gambar 2.21. *Hasil jaringan A –C – D – F – G*

Jika masih ada lintasan penambah yang lain, ulangi langkah 3 sampai dengan langkah 5. Jika tidak ada lintasan penambah yang lain, hitung aliran pada setiap busur.

Karena sudah tidak ditemukan lagi jalur penambah dengan Algoritma BFS, maka langkah sudah berakhir. Sehingga dapat dihitung max flow dari graf diatas adalah:

$$\max\_flow = 1 + 3 + 1 = 5$$

Dengan hasil graf sebagai berikut:



Gambar 2.22. *Hasil network*

2.5 Matlab

Matlab merupakan sebuah singkatasn dari Matrix Laboratory, yang pertama kali dikenalkan oleh University of New Mexico dan University of Stanford pada tahun 1970. Softwere ini pertama kali memang digunakan untuk keperluan analisis numerik, aljabar linier dan teori tentang matriks. Bahasa yang digunakan dalam saebuah sistem MATLAB interaktif yang elemen data dasarnya adalah *array* yang tidak membutuhkan pengaturan dimensi. Hal ini memungkinkan penyelesaian banyak masalah komputasi teknik, terutama yang berhubungan dengan formulasi *matriks* dan vektor (Away, 2006).

Saat ini, kemampuan dan fitur yang dimiliki oleh Matlab sudah jauh lebih lengkap dengan ditambahkannya toolbox-toolbox. Beberapa manfaat yang didapatkan dari Matlab antara lain:

- Perhitungan matematika
- Komputasi numerik

- Simulasi dan pemodelan
- Visualisasi dan analisis data
- Pembuatan grafik untuk keperluan sains dan teknik
- Pengembangan aplikasi, misalnya dengan memanfaatkan GUI.

Matlab dapat dipandang sebagai kalkulator dengan fitur yang lengkap. Kita pernah menggunakan kalkulator dengan fasilitas minimal, misalnya hanya terdapat fasilitas penambahan, pengurangan, perkalian dan pembagian. Kalkulator yang lengkap lagi adalah kalkulator scientific dimana fasilitas yang diberikan tidak hanya yang disebutkan di atas, melainkan sudah ada fungsi-fungsi trigonometri, bilangan kompleks, akar kuadrat dan logaritma. Matlab mirip dengan kalkulator tersebut, tetapi dengan fitur-fitur yang lengkap diantaranya dapat digunakan untuk memprogram, aplikasi berbasis GUI dan lengkap dengan toolbox yang dapat dimanfaatkan untuk memecahkan masalah sains dan teknik.

Matlab juga menyediakan beberapa fasilitas, antara lain sebagai berikut.

1. Manipulasi mudah untuk membentuk matriks
2. Sejumlah rutin terpasang di dalam yang bisa dimodifikasi dan dikembangkan
3. Fasilitas canggih untuk mendapatkan gambar dimensi dua atau tiga
4. Kemudahan untuk menuliskan program yang singkat, sederhana, dan dikembangkan sesuai kebutuhan.

BAB V

PENUTUP

5.1. Simpulan

1. Representasi graf berupa 46 tiang listrik yang diasumsikan sebagai 46 titik, kabel listrik diasumsikan sebagai busur berarah yang berjumlah 63, dan kapasitas arus kabel bersatuan ampere diasumsikan sebagai kapasitas setiap busur.
2. Pencarian aliran maksimum dengan algoritma Edmonds Karp pada jaringan listrik Kota Tegal diperoleh aliran maksimum sebesar 1300 Ampere dengan 8 kali iterasi.
3. Pencarian aliran maksimum dengan bantuan tool software Matlab menghasilkan aliran maksimum yang sama yaitu sebesar 1300 Ampere.

5.2. Saran

1. Pencarian aliran maksimum menggunakan tools software Matlab belum menghasilkan hasil yang sempurna terutama pada tampilan graf akhir. Diharapkan peneliti selanjutnya dapat membuat program dengan algoritma yang sesuai dan menyempurnakan program.
2. Peneliti lain dapat mengembangkan dan menerapkan algoritma Edmonds Karp dalam permasalahan lainnya, misalkan untuk mencari rute terpendek, *travelling salesman problem*, maupun masalah *knapsack.*

3. Untuk pihak PLN diharapkan perhitungan ini bisa menjadi alternatif perhitungan aliran maksimum pada jaringan listrik tegangan menengah sehingga dapat menghemat biaya dan alat.

DAFTAR PUSTAKA

Ahuja, R.K, & Orlin, J.B. 1989. A Fast and Simple Algorithm for The Maximum Flow Problem, *JSTOR Operation Research.* 37:748-759.

Aini, K. 2010. *Penyelesaian Masalah Aliran Maksimum dengan Menggunakan Algoritma Djikstra dan Algoritma Ford Fulkerson*. Skripsi. Riau:Universitas Islam Negeri Sultan Syarif Kasim.

Amelia, M. A. 2007. *Optimasi Rus Maksimum pada Jaringan*. Skripsi. Yogyakarta: Universitas Sanata Dharma Yogyakarta.

Andersen, J. 2004. *Discrete Mathematics with Combinatorics* (2nd ed). New Jersey: Precticle Hall

Away, G. A. 2006. *The Sortcut of Matlab. Informatika*. BI-obses. Bandung. Informatika Bandung.

Bell, E. T .1952. Mathematics: Queen and Servant of Science. London: G. Bell & Sons

Dey, T.K. 2009. *Edmonds-Karp Algorithm (Advanced Algorithms (CSE 794)).*http://www.cse.ohio-state.edu/~tamaldey/course/794/ek-algo.pdf. [21-3-2017]

Fakhri. 2008. Penerapan Algoritma Djikstra dalam Pencarian Solusi Maximum Flow Problem. Strategi Algoritmik.

Farizal, T. 2013. *Pencarian Aliran Maksimal Dengan Algoritma Ford-Fulkerson*. Skripsi. Semarang: Fakultas MIPA, Universitas Negeri Semarang.

Fathimatuzzahro. 2013. *Penyelesaian Aliran Maksimum Menggunakan Edmonds-Karp Algorithm.* Skripsi. Malang: Universitas Negeri Malang.

Handoyo, A. B. 2010. *Aplikasi Algoritma Network Flow untuk Manajemen Pendistribusian Minyak*. Makalah. Bandung: ITB.

Jain, C. & Garg, D. 2012. Improved Edmond Karps Algorithm for Network Flow Problem. *International Journal of Computer Applications (0975-8887.* Tersedia di http://citeseerx.ist.psu.edu [diakses 16-10-2017]

Johnsonbaugh, R. 1986. *Discrate Mathematics* (8th ed). New York: Macmilian Publishing Company.

Kumar, D. & Manjunatha, A.S. 2014. Implementing Map Reduce Based Edmonds Karp Algorithm to Determine Maximum Flow in Large Network Graph. *International Journal of Recent Research in Mathematics Computer Science and Information Technology,* (73-78). Tersedia di www.paperpublications.org [diakses 16-10-2017]

Kumar, M. K. K., Khan, A. R., Ahmed, M. M., Arefin, S. & Uddin, S. 2016. Modified Edmonds Karp Algorithm to Solve Maximum Flow Problems. *Open Journal of Applied Sciences,* 6, 131-140. Tersedia di http://www.scirp.org/journal/ojapps [diakses 16-10-2017]

Kumar, S. & Gupta,P. 2001. An Incremental Algorithm for the Maximum Flow Peoblem. *Journal of Mathematical Modelling and Algorithm 2: 1-16.* Tersedia di https://www.academia.edu/14241896/ [diakses 20-10-2017]

Kyi, M. T. & Naing, L. L. 2018. Application of Ford Fulkerson Algorithm to Maximum Flow in Water Distribution Pipeline Network. *International Journal of Scientific and Research Publications Vol.8 2250-3153.* Tersedia di http://www.ijsrp.org/research-paper-1218/ijsrp-p8441.pdf [diakses 16-12-2018]

Latifal, U. & Sugiharti, E. 2015. Penerapan Algoritma Prim dan Kruskal pada Jaringan Distribusi Air PDAM Tirta Moedal Cabang Semarang Utara. *Unnes Journal of Mathematics.* Tersedia di http://journal.unnes.ac.id/sju/index.php/ujm

Madanella, E.D.M. 2007. *Analisis Penggunaan Algoritma Pencarian Melebar (BFS) dan Algoritma Pencarian Mendalam (DFS) dalam Teori Graf.* Skripsi. Bandung: ITB.

Munir, R. 2010. *MATEMATIKA DISKRIT*. Bandung: Informatika Bandung.

Nisa’, F. 2015. *Algoritma Ford Fulkerson untuk Memaksimumkan Flow dalam Pendistribusian Barang.* Skripsi. Malang: Universitas Islam Negeri Maulana Malik Ibrahim.

Prasetyo, B. & Hidayah, M. R. 2014. Penggunaan Metode Depth First Search (DFS) dan Breadth First Search (BFS) pada Strategi Game Kamen Rider Decade Versi 0.3. *Scientific Journal of Informatics vol.1.*

Rahma, I. N. Permanasar, Y. & Respitawulan. 2016. Aplikasi Aliran Maksimum pada Jaringan Listrik Menggunakan Metode Ford Fulkerson. *Prosiding Matematika.* Bandung: Universitas Islam Bandung.

Ramadhan, R. & Darmaji. 2013. Penentuan Kuat Arus Maksimum Jaringan Listrik Menggunakan Algoritma Augmenting Path. *Jurnal Sains dan Seni Pomits Vol.2,* 2337-3520. Tersedia di digilib.its.ac.id/ITS-Undergraduate [diakses 16-10-2017]

Ryu,S. 2018. *Optimasi Sistem Distribusi.* Online. Tersedia di https://www.scribd.com/document/377901254/Optimasi-Sistem-Distribusi-docx [diakses 10-1-2019]

Siang, J. 2002. *Matematika Diskret dan Aplikasinya pada Ilmu Kompter*. Yogyakarta: Andi.

Suswanto, D. 2009. *Sistem Distribusi Tenaga Listrik* (1st ed). Online. Tersedia di https://www.academia.edu/6754861/ [diakses 10-1-2019]

Ulya, R. 2013. *Menentukan Aliran Maksimum dengan Algoritma Ford-Fulkerson dan Preflow Push*. Skripsi. Semarang: Universitas Negeri Semarang.

Wilf, H. S. 1994. *Algorithm and Complexity* (internet edition). Online. Tersedia di http://www.math.upenn.edu. [diakses 17-10-2017]

Zai, D. Budiati, H. & Berutu, S. S. 2016. Simulasi Rute Terpendek Lokasi Pariwisata Di Nias Dengan Metode Breadth First Search dan Tabu Search. *Jurnal InFact Vol.1 NO.2.* Tersedia di http://www.e-jurnal.ukrimuniversity.ac.id [diakses 17-10-2017]