Implementasi algoritma *generic preflow push* dalam pencarian aliran maksimum pada jaringan listrik

Finda Mardikasari[1], Purwanto[2], dan Mahmuddin Yunus[3]
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI MALANG
E-mail: finda.mardikasari@gmail.com

ABSTRAK : *Maksimum Flow Problem* merupakan suatu permasalahan dalam pencarian aliran maksimum pada suatu jaringan yang memiliki titik sumber (*source*) dan titik tujuan (*sink*). Algoritma *Generic Preflow Push* merupakan algoritma yang bekerja bertahap dengan menggunakan aliran semu atau *preflow* dan mendorong menuju titik yang paling dekat dengan titik tujuan (*sink*) dengan tujuan mendapatkan nilai *excess* yang besar menggunakan metode *push/*relabel. Pada *Maximum Flow Problem* algoritma ini terdiri dari 4 langkah yaitu konstruksi jaringan sisa, inisialisasi *preflow* dan label titik, pencarian titik aktif, dan *push/relabel*. Dalam menyelesaikan permasalahan pencarian aliran maksimum dengan menggunakan algoritma *Generic Preflow Push* dibutuhkan proses yang panjang. Oleh sebab itu, untuk mempermudah pencarian rute, diimplementasikan ke dalam program dengan menggunakan bahasa pemrograman *Borland Delphi* 7.0.

Kata kunci : *Maximum Flow Problem*, Algoritma *Generic Preflow Push*, *Borland Delphi* 7.0.

ABSTRACT : *Maximum Flow Problem* is an issue in search of maximum flow in a network that has source vertex and sink vertex. *Generic Preflow Push* algorithm is algorithm that works step by step using apparent flow referred to as *preflow* and push it to the vertex closest to the sink vertex in order to get a large *excess* value using *push/relabel*. *Generic Preflow Push* algorithm on *maximum flow problem* contains with 4 steps which are construction of residual network, *preflow* and vertex label initialization, searching of active vertex, and *push/relabel*. In solving the *maximum flow problem* using *Generic Preflow Push* algorithm needs a long process. . Therefore, to simplify the search of route, it is implemented into a computer program using Borland Delphi 7.0.

Keywords : *Maximum Flow Problem, Generic Preflow Push* algorithm, *Borland Delphi* 7.0.

Teori Graph merupakan salah satu bagian dari ilmu matematika yang dapat digunakan dalam menyelesaikan suatu masalah. Misalnya, dengan menggunakan teori graph maka permasalahan seperti penjadwalan, menghitung jarak terpendek, menghitung biaya minimum dari pendistribusian suatu barang, juga dalam

1. Finda Mardikasari adalah mahasiswa jurusan Matematika FMIPA Universitas Negeri Malang
2. Purwanto adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang
3. Mahmuddin Yunus adalah dosen jurusan Matematika FMIPA Universitas Negeri Malang

menyelesaikan penghitungan kapasitas maksimum dalam suatu arus atau jaringan dapat diselesaikan. Aliran jaringan atau *network flow* dalam teori graph merupakan graph berarah yang tiap sisinya mempunyai kapasitas dan sifat-sifat tertentu. Permasalahan yang berkaitan dengan *flow network* salah satunya adalah *maximum flow problem*. *Maximum flow problem* didefinisikan sebagai suatu permasalahan untuk mencari aliran maksimum pada jaringan yang memiliki satu sumber (*source*) dan satu tujuan (*sink*)[1]. Salah satu contoh permasalahan mengenai *maximum flow problem* adalah pencarian aliran arus maksimum pada jaringan listrik.

Algoritma ini bekerja dengan menggunakan aliran semu atau *preflow* pada jaringan awal dan mendorongnya menuju titik yang paling dekat dengan titik tujuan (*sink*) dengan tujuan untuk mendapatkan nilai *excess* yang besar. Namun jika hal tersebut tidak dapat terjadi maka aliran semu atau *preflow* didorong kembali menuju titik yang paling dekat dengan titik sumber (*source*). Aliran semu atau *preflow* akan menjadi *flow* yang kemudian merupakan *flow* maksimum apabila nilai *excess* untuk semua titik bernilai nol kecuali titik yang menjadi titik sumber (*source*) dan titik tujuan (*sink*)[2].

Berdasarkan uraian diatas, akan dibahas mengenai Algoritma *Generic Preflow Push* pada *Maximum Flow Problem*.

HASIL YANG DIHARAPKAN

1. Mengetahui langkah-langkah algortima *Generic Preflow Push* yang digunakan dalam pencarian aliran maksimum pada jaringan dalam *maximum flow problem*.
2. Dapat mengetahui analisa hasil dari algoritma *Generic Preflow Push* dalam menyelesaikan permasalahan aliran maksimum.
3. Mengetahui implementasi algoritma *Generic Preflow Push* dalam bahasa pemrograman *Borlan Delphi 7.0*.

HASIL DAN PEMBAHASAN

- ***Maximum Flow Problem***

  Network flow problem sering dirumuskan pada graph terhubung. Permasalahan yang mendasar pada *network flow problem* ini adalah *maximum flow problem*. Unsur-unsur yang terdapat pada permasalahan ini adalah suatu graph terhubung $D = (V, E)$, fungsi kapasitas tak negatif yang merupakan kapasitas setiap sisi, suatu titik sumber $s \in V$ dan suatu titik tujuan $t \in V$. Sehingga *maximum flow problem* dapat dideskripsikan sebagai suatu masalah untuk mencari arus maksimum yang dapat mengalir pada sebuah network yang hanya memiliki satu *source* dan *sink*[3].

  Dalam *maximum flow problem* terdapat istilah – istilah yang sering digunakan, diantaranya sebagai berikut:

  1. *Network N*

     Network adalah digraph berbobot yang memiliki suatu titik sumber dan satu titik tujuan. Pada titik sumber, tidak terdapat sisi masuk, sedangkan pada titik tujuan tidak terdapat sisi keluar, bobot tiap sisi pada suatu *network* adalah kapasitas (C) sisi tersebut[4].

2. Flow (F)

Misalkan $G = (V, E)$ adalah suatu graph dengan s, t dua titik tetap. $c: E \rightarrow N$ suatu pemetaan yang disebut fungsi kapasitas di G dan $N = (G, s, t, c)$adalah suatu jaringan. Suatu fungsi $f : E \rightarrow R$ adalah arus di N. Sehingga didefinisikan *Flow* (F) merupakan suatu bilangan tak negatif yang didefinisikan pada tiap sisi pada suatu *network* yang memenuhi $Fij < Cij$ untuk sebarang sisi (i, j) pada *network* tersebut. Setiap arus (*flow*) yang ada dalam *network*, harus memenuhi sebuah batasan yaitu arus yang masuk pada suatu simpul harus sama dengan arus yang keluar pada simpul tersebut, kecuali pada *source*, yang arus keluarnya lebih besar dari arus masuk, dan *sink*, yang arus masuknya lebih besar dari arus keluar.

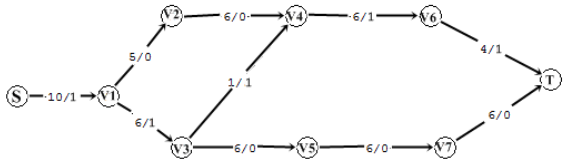
3. Residual Network

Residual network dari G_f didefinisikan sebagai $N_f = (G_f, \mathrm{s}, \mathrm{t}, c_f)$ dimana :

$G_f = (\mathrm{V}, E_f)$, dengan E_f adalah sisi yang dialiri $flow\ f$

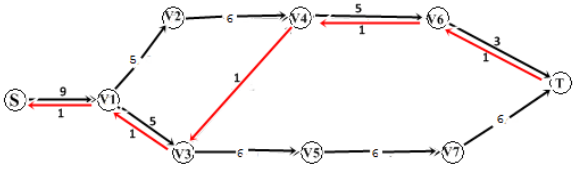
$c_f = c(u, v) - f(u, v)$ yang merupakan kapasitas sisi residu

Perhatikan bahwa lintasan s, v_1, v_3, v_4, v_6, t pada graph berikut telah dialiri *flow* sebesar 1:



Gambar 1. Graph R

Sehingga *residual network* diperoleh:



Gambar 2. *Residual Network*

Nilai flow (*flow value*) adalah jumlah semua flow yang meninggalkan titik sumber (s). $G_f = (V, E_f)$ disebut *residual network* relatif jika dan hanya jika himpunan titik V di G_f sama dengan himpunan titik V di G dan untuk setiap sisi $e = (u, v)$ di G, maka G_f mempunyai sisi:

1) Jika $f(e) < c(e)$ maka $\exists\ e' = (u, v), e' \in E_f$ dengan kapasitas sisa $c_f(e') = c(e) - f(e)$

2) Jika $f(e) > 0$ maka $\exists\ e' = (v, u), e' \in E_f$ dengan kapasitas sisa $c_f(e') = f(e)$

- **Algoritma *Generic Preflow Push* pada *Maximum Flow Problem***

Algoritma *Generic Preflow Push* merupakan salah satu algoritma yang digunakan untuk menyelesaikan permasalahan *maximum flow*. Algoritma *Generic Preflow Push* disebut juga sebagai Algoritma Preflow Push (Herniawaty:2004). Algoritma ini bekerja dengan beberapa tahap secara berulang. Tahapan tersebut terdiri dari operasi dasar, *push* dan *relabel*[2]..

Algortima *Generic Preflow Push* dimulai dengan penginisialisasian *preflow* dengan besar nilainya sama dengan kapasitas sisi, untuk setiap sisi yang keluar dari titik sumber (*source*) dan bernilai nol untuk sisi yang lain. Kemudian dilakukan pelabelan untuk setiap titik pada jaringan tersebut dengan $d(s) = n$, $d(t) = 0$, dan $d(v) = 0$ untuk $v \in V - \{s\}$.

Langkah selanjutnya adalah dilakukannya algoritma *push* dan *relabel* secara berulang. Ketika algoritma *push* bekerja maka aliran semu atau *preflow* didorong dari titik v ke titik w sehingga menaikkan $f(v,w)$ dan $e(w)$ sebesar $\delta = \min\{e(v), c_f(v,w)\}$ dan secara bersamaan menurunkan $f(w,v)$ dan $e(v)$ sebesar δ. Suatu dorongan atau *push* dikatakan *satured* apabila setelah dilakukan *push* diperoleh $c_f(v,w)$ dan *unsatured* jika sebaliknya. Algoritma *push* dapat berlaku apabila untuk setiap titik v, $v \in$ titik aktif , $c_f(v,w) > 0$ dan $d(v) = d(w) + 1$.

Namun jika syarat tersebut tidak dipenuhi maka algoritma *relabel* bekerja. Untuk setiap titik v, $v \in$ titik aktif , $c_f(v,w) > 0$ dan $d(v) \leq d(w)$ maka *relabel* $d(v) = \min\{d(w) + 1\}$.

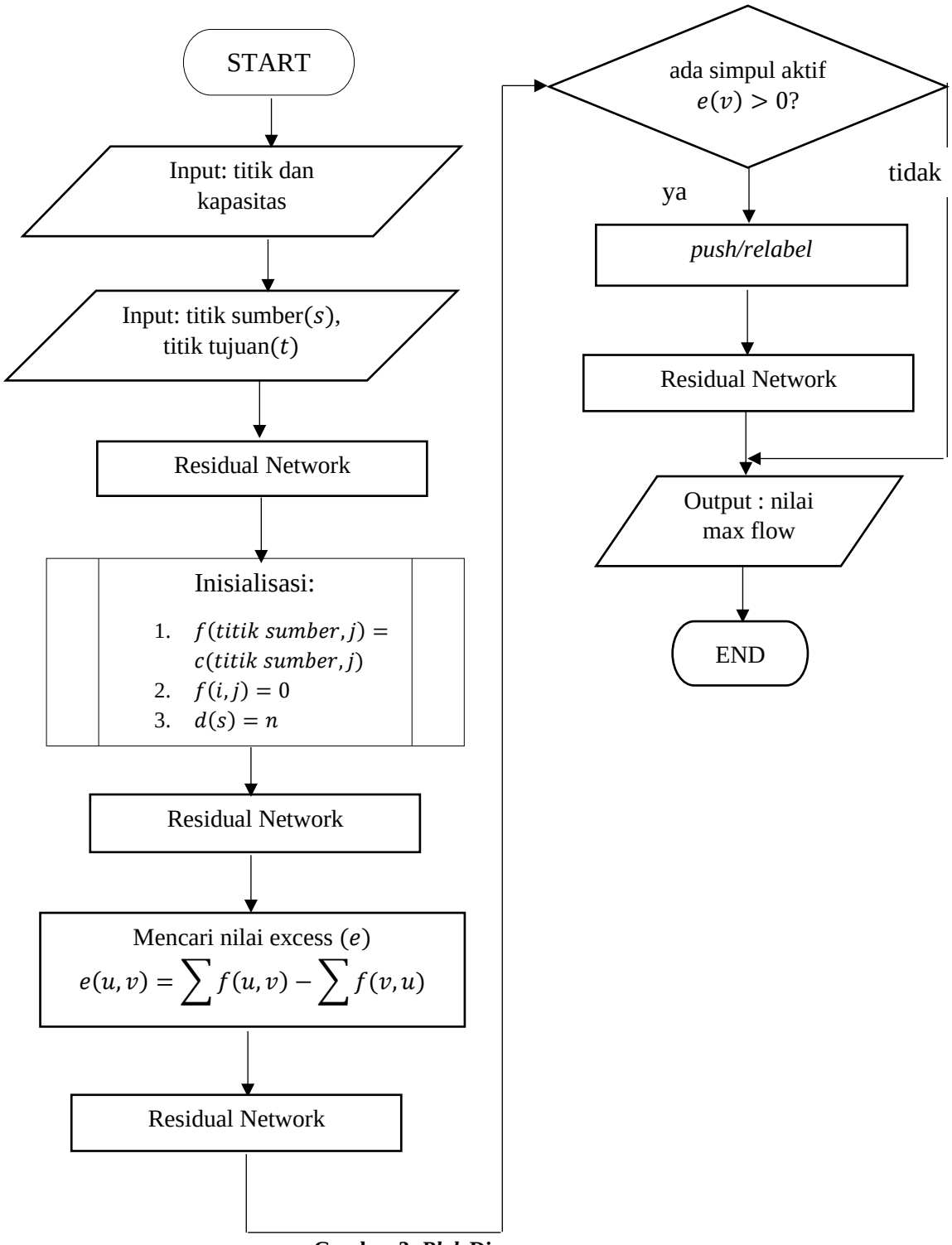
Adapun langkah-langkah formal dari algoritma *Generic Preflow Push* sebagai berikut:

1. Konstruksi jaringan sisa $G_f = (V, E_f)$
2. Inisialisasi *preflow f* dan label titik d
   a. $f(s,v) = c(s,v), \forall (s,v) \in E$
   b. $f(v,w) = 0, \forall (v,w) \in E$ dengan $v \neq s$
   c. $d(s) = n$ dan $d(v) = 0, \forall v \neq s$
3. Jika v titik aktif maka operasi dasar *push* dan *relabel* dapat diterapkan pada v. Jika tidak maka BERHENTI dan f telah maksimum.
4. Pilih titik aktif dan lakukan algoritma *push relabel* secara berulang-ulang hingga tidak ditemukan lagi simpul aktif (kembali ke langkah 3).

Sedangkan langkah-langkah algoritma *Push Relabel* sebagai berikut:

1. Jika v titik aktif, $c_f(v,w) > 0$ dan $d(v) = d(w) + 1$ berlaku maka lakukan algoritma *push* yaitu meningkatkan arus pada $f(v,w)$ dan $e(w)$ dan menurunkan arus pada $f(w,v)$ dan $e(v)$ sebesar δ, dengan $\delta = \min\{e(v), c_f(v,w)\}$ kemudian BERHENTI.
2. Jika algoritma *push* (langkah1) tidak berlaku maka lakukan algoritma *Relabel*, yaitu dengan meningkatkan label titik sebesar $d(v) = \min\{d(w) + 1 \mid c_f(v,w) > 0\}$, kemudian BERHENTI.

Berikut ini adalah *Block* Diagram dari algoritma *Generic Preflow Push* pada *Maximum Flow Problem*.

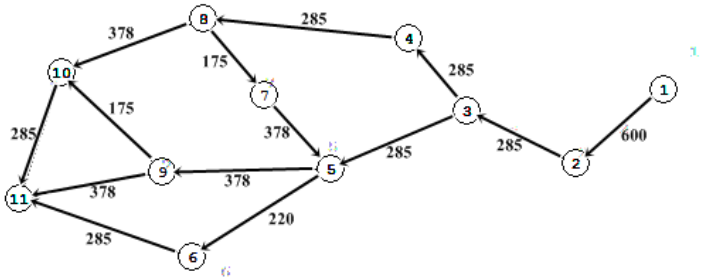


Gambar 3. *Blok* Diagram

- **Contoh Penerapan**

  Salah satu penyulang yang ada di wilayah UPJ Pacitan adalah penyulang Teuku Umar yang mengalirkan arus listrik untuk konsumen-konsumen wilayah sekitar kota Pacitan. Setiap harinya mengalirkan arus

listrik sebesar 202,67 ampere. Mengingat masyarakat yang tinggal di sekitar kota jauh lebih banyak dibandingkan dengan di wilayah lainnya, maka kebutuhan terhadap listrik juga jauh lebih besar. Oleh karena itu, perlu adanya pemaksimalan aliran listrik untuk mendukung aktifitas masyarakat yang tinggal di sekitar kota. Berikut peta jaringan listriknya:

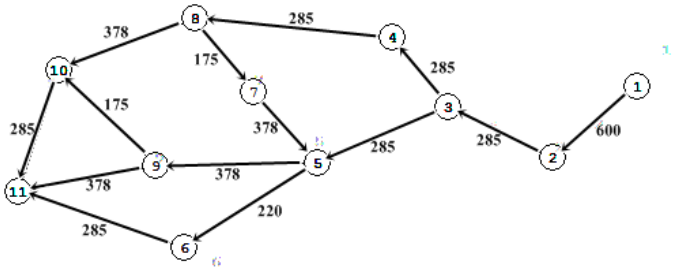


Gambar 4. Peta Jaringan Listrik

Titik-titik yang terdapat pada peta jaringan di atas merupakan representasi dari tiang-tiang listrik yang berada pada suatu wilayah penyulang. Titik dengan nomor 1 merupakan titik sumber (*source*) dan titik dengan nomor 11 merupakan titik tujuan (*sink*). Sedangkan bobot sisi yang terdapat pada peta jaringan tersebut merupakan representasi dari besarnya kemampuan kuat hantar arus yang dimiliki oleh kabel-kabel penghantar.

Berikut penyelesaian pencarian aliran maksimum pada jaringan listrik dengan menggunakan algoritma *Generic Preflow Push*:

L1. Konstruksi jaringan sisa $G_f = (V, E_f)$ dari $G = (V, E)$



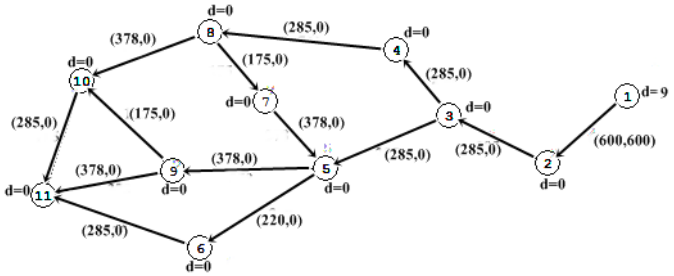
Gambar 5. Konstruksi Jaringan

L2. Inisialisasi preflow f dan label jarak d

i) $f(s, v) = c(s, v)$

ii) $f(v, w) = 0$, $\forall (v, w) \in E$ dengan $v \neq s$

iii) $d(s) = n$ dan $d(v) = 0$



Gambar 6. Inisialisasi Jaringan

L3. Karena ditemukan titik aktif yaitu titik 2 maka dilakukan *push/relabel*.
L4. Lakukan operasi *Push/Relabel*

Push/Relabel pertama:

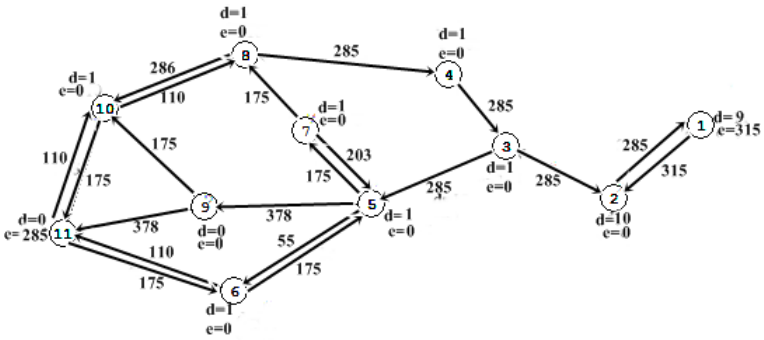
- Karena hanya ada 1 titik yang aktif, maka pilih titik 2. Karena tidak ada sisi yang memenuhi $d(v) \leq d(w) + 1$ pada titik 2, maka dilakukan algoritma *relabel*:$d(2) = \min\{d(1) + 1, d(3) + 1\} = \min\{10,1\} = 1$. Oleh karena itu, sisi yang memenuhi adalah $(2,3)$.
- Dilakukan algoritma *push* $\delta = \min\{e_f(2), c_f(2,3)\} = \min\{600,285\} = 285$
  $f(2,3) \leftarrow f(2,3) + \delta \leftarrow 0 + 285 \leftarrow 285$
  $f(3,2) \leftarrow f(3,2) - \delta \leftarrow 0 - 285 \leftarrow -285$ (diabaikan)
  $e_f(2) \leftarrow e_f(2) - \delta \leftarrow 600 - 285 \leftarrow 315$
  $e_f(3) \leftarrow e_f(3) + \delta \leftarrow 0 + 285 \leftarrow 285$
- Konstruksi *residual network*
  Untuk sisi $e_2 = (2,3) \in E_f$
  - Karena $f(e_2) = c(e_2)$ maka $\nexists\, e_2 = (2,3) \in E_f$
  - $f(e_2) > 0$ maka $\exists e'_2 = (3,2) \in E_f$ dengan $c_f(e'_2) = f(e_2) = 285$

  Kemudian lakukan berulang kali hingga tidak ditemukan titik aktif. Pada permasalahan ini dilakukan *push/relabel* sebanyak 10 kali dan berikut *push /relabel* kesepuluh:

Push/Relabel kesepuluh:

- Pilih titik 10. Karena tidak ada sisi yang memenuhi $d(v) \leq d(w) + 1$ pada titik 10, maka dilakukan algoritma *relabel*:$d(10) = \min\{d(11) + 1\} = \min\{1\} = 1$. Oleh karena itu, pilih sisi yang memenuhi misal $(10,11)$.
- Dilakukan algoritma *push* $\delta = \min\{e_f(10), c_f(10,11)\} = \min\{110,285\} = 110$
  $f(10,11) \leftarrow f(10,11) + \delta \leftarrow 0 + 110 \leftarrow 110$
  $f(11,10) \leftarrow f(11,10) - \delta \leftarrow 0 - 110 \leftarrow -110$ (diabaikan)
  $e_f(10) \leftarrow e_f(10) - \delta \leftarrow 110 - 110 \leftarrow 0$
  $e_f(11) \leftarrow e_f(11) + \delta \leftarrow 0 + 110 \leftarrow 110$
- Konstruksi *residual network*
  Untuk sisi $e_{11} = (10,11) \in E_f$
  - Karena $f(e_{11}) < c(e_{11})$ maka $\exists\, e'_{10} = (6,11) \in E_f$ dengan $c(e'_{10}) = c(e_{10}) - f(e_{10}) = 285 - 110 = 175$
  - $f(e_{11}) > 0$ maka $\exists\, e'_{11} = (11,10) \in E_f$ dengan $c_f(e'_{11}) = f(e_{11}) = 110$

Sehingga diperoleh *residual network* dan *preflow* terbaru sebagai berikut:

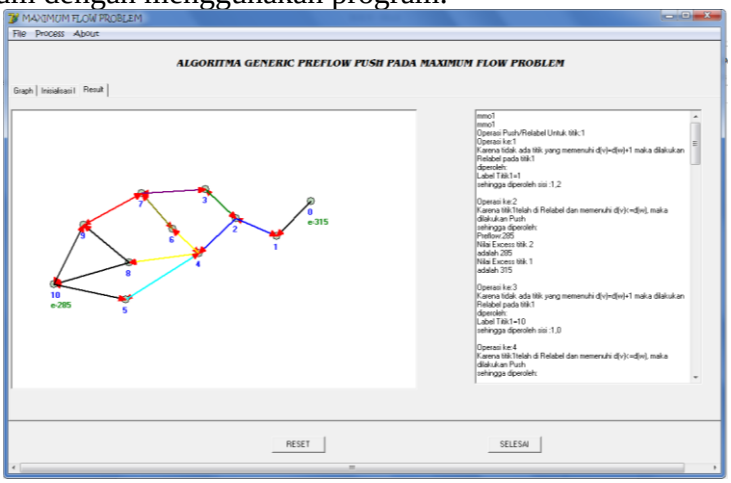


Gambar 7. *Residual Network*

Karena $e(v) = 0, \forall v \in V - \{s, t\}$ artinya tidak ada lagi titik aktif pada jaringan G_f, maka algoritma BERHENTI. Jadi didapatkan arus maksimum sebesar 285 ampere. Oleh karena itu, PT. PLN (Persero) dapat mengalirkan arus listrik pada jaringan Teuku Umar sebesar 285 ampere guna menjaga kestabilan listrik sehingga dapat meminimalisasi gangguan listrik yang dapat mengakibatkan terhambatnya aktifitas masyarakat yang tinggal di sekitar kota Pacitan.

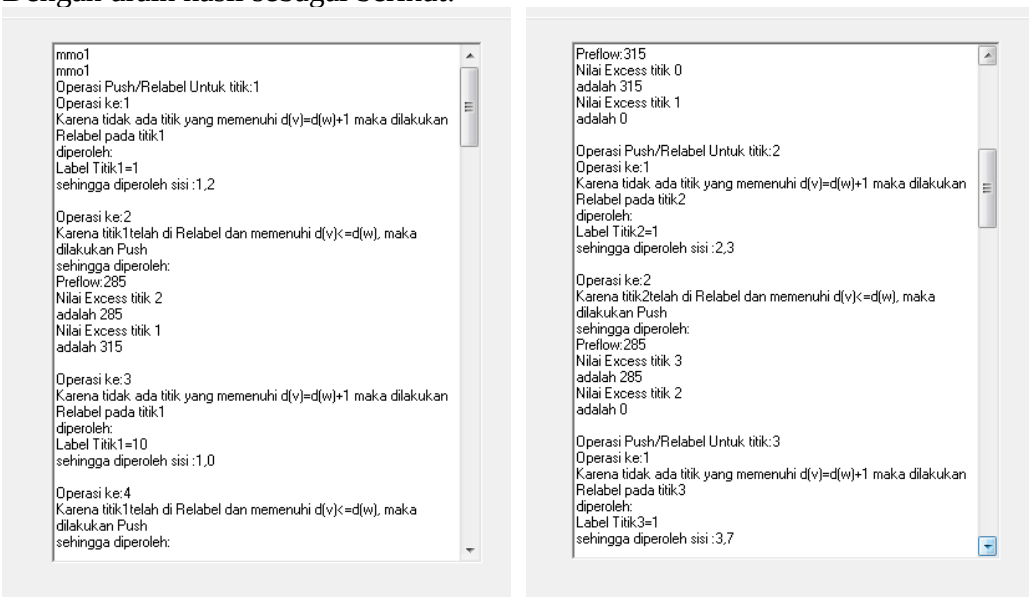
Algoritma *Generic Preflow Push* diimplementasikan ke dalam program komputer menggunakan bahasa pemrograman *Borland Delphi* 7.0. Untuk menjalankan programnya dimulai dengan menginputkan data seperti banyaknya titik dengan titik 0 sebagai titik sumber (*source*) dan salah satu titik lainnya sebagai titik tujuan (*sink*), kapasitas sisi, dan inisialisasi titik yang digunakan sebagai titik sumber (*source*) dan titik tujuan (*sink*). Data tersebut diproses sesuai dengan langkah-langkah algoritma *Generic Preflow Push*, yaitu konstruksi jaringan sisa, inisialisasi *preflow* dan label titik, pencarian titik aktif, dan *push/relabel*. Output dari proses tersebut adalah urutan titik-titik yang dikenai proses *push/relabel*, nilai maksimum yang dicari, dan jaringan sisa akhir.

Berikut ini adalah penyelesaian permasalahan pencarian aliran arus maksimum dengan menggunakan program.

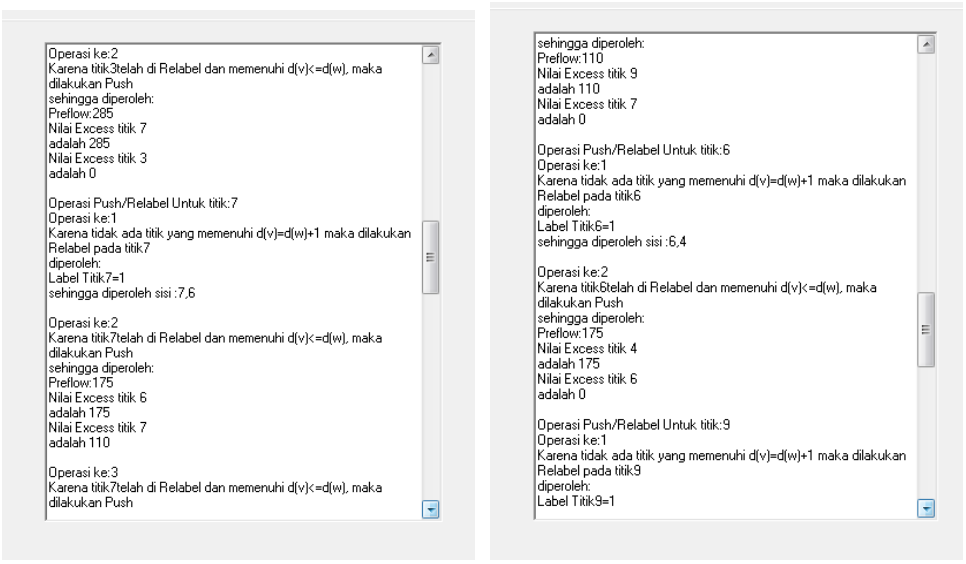


Gambar 8. Tampilan Hasil

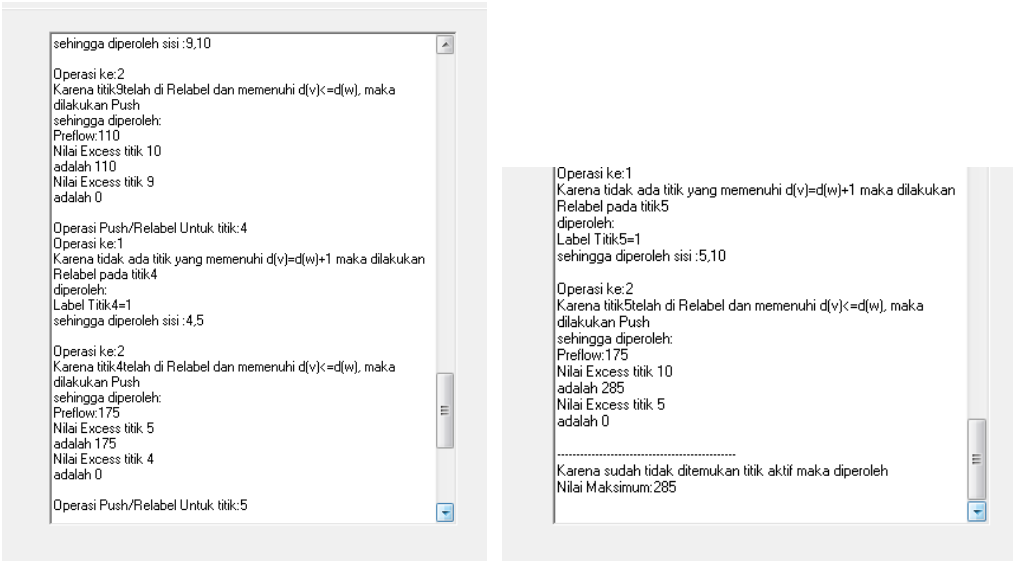
Dengan urain hasil sebagai berikut:



Gambar 9. Uraian Hasil



Gambar 10. Lanjutan 1 Uraian Hasil



Gambar 11. Lanjutan 2 Uraian Hasil

Berdasarkan uraian hasil diatas dapat diketahui bahwa nilai maksimum yang dapat diperoleh adalah285 dengan melakukan *relabel* terhadap titik 1,3,7,6,9,4,5 kemudian dilakukan *push* terhadap titik-titik tersebut dengan titik lain yang memenuhi kondisi *push* hingga nilai *excess* tiap titik selain titik sumber (*source*) dan titik tujuan (*sink*) bernilai nol. Sehingga tersisa nilai *excess* untuk titik sumber (*source*) sebesar 315 dan nilai *excess* untuk titik tujuan (*sink*) sebesar 285sesuai dengan graph hasil yang digambarkan dan diperoleh nilai maksimum sebesar 285.

Program yang merupakan implementasi dari algoritma *Generic Preflow Push* pada *Maximum Flow Problem* ini telah diuji coba dengan menggunkan 7, 8, 9, 11, dan 27 titik.

PENUTUP

Kesimpulan

Langkah-langkah algoritma *Generic Preflow Push* untuk menyelesaikan permasalahan aliran maksimum dapat dipahami sebagai berikut. Algortima *Generic Preflow Push* dimulai dengan mengkonstruksi jaringan sisa setelah jaringan tersebut dialiri aliran sebesar nol ke seluruh sisi yang ada pada jaringan tersebut. Kemudian pada jaringan tersebut dilakukan penginisialisasian *preflow* terhadap sisi yang keluar dari titik sumber (*source*) sebesar kapasitas sisi, dan *preflow* pada sisi lainnya sebesar nol. Selain penginisialisasian *preflow*, dilakukan penginisialisasian label titik. Untuk titik sumber (*source*) inisialisasi label titik sebesar banyaknya titik yang ada pada jaringan tersebut dan untuk titik tujuan (*sink*) serta titik lainnya diberikan inisialisasi label nol. Langkah selanjutnya adalah mencari nilai *excess* untuk mendapatkan titik aktif sehingga dapat dilakukan *push/relabel* untuk mendapatkan arus maksimum. *Push/relabel* akan terus dilakukan hingga tidak ditemukan lagi titik aktif pada jaringan tersebut. Ketika titik aktif sudah tidak ditemukan dan nilai *excess* untuk setiap titik selain titik sumber (*source*) dan titik tujuan (*sink*) sama dengan nol, maka inisialisasi

preflow menjadi *flow* maksimum yang dicari dan algoritma *Generic Preflow Push* berhenti. Jika tidak ditemukan maka jaringan tersebut telah maksimum dan algoritma *Generic Preflow Push* berhenti.

Algortima *Generic Preflow Push* sangat membantu dalam pencarian nilai aliran maksimum pada suatu jaringan. Algortima ini bekerja dengan mempertahankan *preflow* pada jaringan asal dan mendorong *flow* dan *excess* suatu titik ke titik di dekatnya hingga mencapai titik tujuan (*sink*), apabila hal ini tidak dapat terjadi maka akan dikembalikan ke titik terdekat dengan titik sumber (*source*). Ketika algoritma ini berhenti maka *preflow* berubah menjadi *flow* maksimum untuk jaringan tersebut.

Implementasi program dari algoritma *Generic Preflow Push* ini adalah program Delphi dengan inputnya adalah input titik, input kapasitas, input titik *source*, dan input titik *sink*. Hasil yang diperoleh berupa nilai aliran maksimum jaringan tersebut. Implementasi program sangat membantu walaupun masih ada kelemahan yaitu tidak dapat menampilkan kapasitas residu pada graph hasil. User hanya dapat mengamati sisi-sisi graph mana saja yang telah diproses.

Saran

Algoritma *Generic Preflow Push* merupakan salah satu algoritma yang dapat digunakan untuk menyelesaikan permasalahan pencarian nilai aliran maksimum pada suatu jaringan. Berdasarkan uraian yang telah dipaparkan pada bab-bab sebelumnya, dapat disimpulkan bahwa algoritma *Generic Preflow Push* digunakan untuk menyelesaikan permasalahan pencarian nilai aliran maksimum pada suatu jaringan dengan mencari titik aktif yang kemudian digunakan untuk proses *push/relabel* hingga tidak ditemukan titik aktif pada jaringan tersebut. Oleh karena itu, algoritma ini dapat digunakan sebagai alternative dalam pencarian nilai aliran maksimum pada suatu jaringan.

Selain digunakan untuk mencari nilai aliran maksimum pada suatu jaringan, algoritma ini dapat dikembangkan untuk memperoleh nilai maksimum pada suatu jaringan dengan kondisi tertentu. Misalkan jaringan tersebut disisipi sisi baru atau sisi yang ada dihapuskan. Begitu juga untuk implementasi pada programnya.

DAFTAR PUSTAKA

[1] Farizal, T. 2013. *Pencarian Aliran Maksimum dengan Algoritma Ford – Fulkerson (Studi Kasus pada Jaringan Listrik Kota Tegal)*. Skripsi tidak diterbitkan. Semarang : Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.

[2] Goldberg, A.V. dan R. E. Tarjan.1988. A New Approach to the Maximum-Flow Problem. *Journal of the Association for Computing Machinery,* Vol.4.

[3] Atallah, M. J. dan Marina B. 2010. *Algorithms and Theory of Computation Handbook, Second Edition, General Concepts and Techniques*. USA: Taylor and Francis Group, LLC.

[4] Park, J. 2015. *Network Flow Problems*. Stanford University.