PENENTUAN JUMLAH *SPANNING-TREE* PADA GRAF BERARAH DENGAN METODE PENUKARAN SISI DAN MATRIKS *IN-DEGREE*

skripsi

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

oleh

Risky Samodra Raharjo
4150405515

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SEMARANG
2009

PERNYATAAN KEASLIAN TULISAN

Dengan ini saya menyatakan bahwa skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya tidak terdapat karya yang diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam skripsi ini dan disebutkan dalam daftar pustaka.

Semarang,

Risky Samodra Raharjo
NIM. 4150405515

PENGESAHAN

Skripsi ini telah dipertahankan di hadapan sidang Panitia Ujian Skripsi Fakultas Ilmu Pengetahuan Alam Universitas Negeri Semarang pada tanggal 25 Agustus 2009.

Panitia:

Ketua,	Sekretaris,
Dr. Kasmadi Imam S., M.S	Drs. Edy Soedjoko, M.Pd
NIP. 130781011	NIP. 131693657
	Penguji,
	Mulyono, S.Si, M.Si
	NIP. 132158717
Penguji/Pembimbing I	Penguji/Pembimbing II
Drs. Amin Suyitno, M.Pd	Isnaini Rosyida, S.Si, M.Si
NIP. 130604211	NIP. 132205927

MOTTO DAN PERSEMBAHAN

MOTTO

- **Janganlah engkau mengerjakan apa yang engkau sendiri meragukannya.**

- **Meskipun sakit tetapi selama masih ada jiwa, harapan tetap masih ada.**

PERSEMBAHAN

- **Special thank's for all My Family; Bapak, Ibu dan Eyangku tercinta serta adek-adekku tersayang.**

- **Keluarga besar Salatiga; Bapak Priyono, Ibu Dewi, Mbk piet dan dek Yaya.**

- **My life spirit, Hanifah dan sahabatku Marom**

- **My favorite dosen Bapak Amin Suyitno dan Ibu Isnaini Rosyida yang telah membimbing saya selama pembuatan skripsi.**

- **Semua teman-temanku.**

- **Almamaterku.**

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan limpahan rahmat dan hidayah-Nya, sehingga penulis memperoleh kekuatan untuk menyelesaikan skripsi ini. Dalam kesempatan ini penulis menghaturkan terima kasih yang tak terhingga kepada:

1. Prof. Dr. H. Sudijono Sastroatmodjo, M.Si., Rektor Universitas Negeri Semarang yang telah memberikan fasilitas-fasilitas kepada penulis.
2. Dr. Kasmadi Imam S., M.S, Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang.
3. Drs. Edy Soedjoko, M.Pd, Ketua Jurusan Matematika Universitas Negeri Semarang
4. Drs. Amin Suyitno, M.Pd, Dosen Pembimbing I yang penuh keikhlasan mengarahkan dan membimbing penulis dalam menyusun skripsi ini dari awal hingga akhir.
5. Isnaini Rosyida, S.Si, M.Si, Dosen Pembimbing II yang penuh keikhlasan mengarahkan dan membimbing penulis dalam menyusun skripsi ini dari awal hingga akhir.
6. Bapak dan Ibu dosen Jurusan Matematika yang telah memberikan bekal ilmu dan pengetahuan selama kuliah.
7. Bapak Susilo Raharjo, B.Sc dan Ibu Mugiyati, kedua orang tuaku yang telah dengan sabar dan ikhlas mencurahkan waktu untuk mendidik, memberi kasih sayang, menasihati, dan membimbing penulis.

8. Teman-teman Matematika Angkatan 2005 yang telah memberikan dukungannya hingga terselesaikannya skripsi ini.

Semoga Allah SWT senantiasa memberikan balasan atas bantuan dan amal baiknya. Penulis berharap semoga skripsi ini dapat bermanfaat bagi para pembaca.

Semarang, Agustus 2009

Penulis

ABSTRAK

Raharjo, Risky Samodra. 2009. *Penentuan Jumlah Spanning-tree pada Graf Berarah dengan Menggunakan Metode Penukaran Sisi dan Matriks In-degree*. Skripsi. Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang. Pembimbing I: Drs. Amin Suyitno, M. Pd dan Pembimbing II: Isnaini Rosyida, S. Si, M. Si.

Kata kunci : *Spanning-tree,* Graf Berarah, Matriks *in-degree.*

Graf memiliki banyak konsep. Salah satu diantaranya adalah konsep pohon (*tree*). Pohon adalah graf terhubung yang tidak memuat siklus. Konsep pohon merupakan konsep yang paling penting dan populer karena konsep ini mampu mendukung penerapan graf dalam berbagai bidang ilmu. Sedangkan *Spanning-tree* adalah sebuah pohon pada graf G yang memuat semua titik di G. Dari setiap graf dapat dibentuk paling sedikit sebuah *spanning-tree*. Permasalahan dalam skripsi ini adalah bagaimana menentukan jumlah *spanning-tree* pada graf berarah, baik dengan menggunakan metode penukaran sisi maupun dengan matriks *in-degree.*

Metode yang digunakan dalam penelitian ini adalah studi pustaka. Langkah pertama yang dilakukan dalam penelitian ini adalah menemukan masalah. Kemudian merumuskan masalah, selanjutnya dengan menggunakan analisis pemecahan yang dapat dilakukan dengan dua cara yaitu menggunakan metode penukaran sisi dan menggunakan matriks *in-degree* sehingga tercapai tujuan penulisan skripsi.

Pada pembahasan, untuk menentukan banyaknya *spanning-tree* pada graf berarah G dengan metode penukaran sisi dapat dilakukan dengan cara: membuat *spanning-tree* awal, kemudian menambahkan *chord* dan menghapus *branch* maka akan terbentuk *spanning-tree* baru. Untuk selanjutnya, dengan langkah yang sama maka akan terbentuk *spanning-tree* yang lain dengan berdasar pada *spanning-tree* awal. Sedangkan untuk menentukan banyaknya *spanning-tree* dengan matriks *in-degree* dapat digunakan teorema: nilai kofaktor k_{qq} dari K(G) adalah sama dengan banyaknya *arborescence* pada G dengan titik v_q sebagai *root. Arborescence* pada G juga merupakan *spanning arborescence.*

Berdasarkan hasil penelitian tersebut penulis menyarankan kepada pembaca untuk melakukan penelitian lebih lanjut dalam menentukan jumlah *spanning-tree* pada graf berarah yang bukan *arborescence.*

DAFTAR ISI

BAB I

PENDAHULUAN

1. LATAR BELAKANG

Masalah jembatan Konigsberg adalah masalah yang pertama kali dimodelkan menggunakan graf. Konigsberg adalah sebuah kota di sebelah timur Prussia (Jerman sekarang) dimana terdapat Sungai Pregel dan merupakan tempat tinggal duke of Prussia pada abad ke-16 (tahun 1736). Kota tersebut saat ini bernama Kaliningrad dan merupakan pusat ekonomi dan industri utama di Rusia Barat. Sungai Pregel membagi kota menjadi empat daratan dengan mengalir mengitari pulau Kneiphof kemudian bercabang menjadi dua buah anak sungai.

Sekitar abad kedelapan belas, pada jembatan Konigsberg dibangun tujuh jembatan yang menghubungkan keempat daratan tersebut. Masalah yang dihadapi adalah mungkinkah seseorang berjalan mengelilingi kota yang dimulai dan diakhiri pada tempat yang sama, dengan melintasi ketujuh jembatan masing-masing tepat satu kali. Leonhard Euler, ahli matematika dari Swiss yang pertama kali menulis mengenai upaya pemecahan masalah jembatan Konigsberg yang sangat terkenal di Eropa. Dari tulisan Euler tersebut muncul cabang dari matematika yang saat ini dikenal sebagai “Teori Graf”.

Dewasa ini teori graf semakin banyak dikembangkan oleh para ahli matematika dan dipelajari oleh ahli di bidang lain seperti ekonomi, sosial, fisika, kimia, biologi, teknik dan komputer. Secara kasar, graf adalah suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat. Dalam kehidupan sehari-hari, graf digunakan untuk menggambarkan berbagai macam 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, pewarnaan peta, jaringan listrik, dan lain-lain.

Graf memiliki banyak konsep. Salah satu diantaranya adalah konsep pohon (*tree*). Konsep pohon merupakan konsep yang paling penting dan populer karena konsep ini mampu mendukung penerapan graf dalam berbagai bidang ilmu. Aplikasi yang menggunakan konsep pohon diantaranya adalah pembangunan jalan dan rel kereta api, pembuatan jaringan komputer, dan lain-lain. Di dalam konsep pohon sendiri, terdapat banyak jenis pohon yang dapat digunakan untuk mencari solusi dari masalah masalah real. Salah satunya adalah pohon pembangun atau lebih dikenal dengan *Spanning-tree.*

Spanning-tree pada graf G adalah sebuah pohon di G yang memuat semua titik di G. Dari suatu graf terhubung, dapat ditemukan *spanning-tree*. Dalam kajian ini hanya terbatas pada penentuan jumlah *spanning-tree* pada graf berarah. Salah satu metode yang digunakan untuk menentukan jumlah *spanning-tree* dari graf berarah yaitu metode penukaran sisi, akan tetapi metode penukaran sisi tidak memungkinkan untuk digunakan jika graf G=(V,E) tersebut mempunyai titik dan sisi dalam jumlah besar.

Berdasarkan latar belakang di atas, penulis akan menyajikan cara yang lebih efisien dalam menentukan jumlah *spanning-tree* untuk graf berarah, dengan menggunakan matriks *in-degree*.

2. Rumusan Masalah

Yang menjadi permasalahan sebagai berikut.

2.1. Bagaimana menentukan jumlah *spanning-tree* pada graf berarah dengan metode penukaran sisi?

2.2. Bagaimana menentukan jumlah *spanning-tree* pada graf berarah dengan matriks *in-degree*?

3. Pembatasan Masalah

Yang dimaksud *spanning-tree* pada graf berarah adalah *arborescence*.

4. Tujuan dan Manfaat Kegiatan

4.1. Tujuan Kegiatan

Untuk menentukan jumlah *spanning-tree* pada graf berarah dengan menggunakan metode yang lebih efisien yang belum pernah diajarkan selama dalam perkuliahan.

4.2. Manfaat Kegiatan

4.2.1 Bagi penulis

Sesuai dengan tujuan di atas, diharapkan penelitian ini dapat memberikan manfaat kepada penulis agar dapat mengembangkan kemampuan berpikir dalam matematika diskrit terutama untuk mengkaji konsep dari pohon (*tree*) khususnya pohon pembangun (*spanning-tree*) yang banyak diterapkan dalam berbagai bidang ilmu.

4.2.2 Bagi pembaca

Untuk menambah ilmu pengetahuan terutama dalam hal menentukan jumlah *spanning-tree* pada suatu graf berarah.

BAB II

KAJIAN TEORI

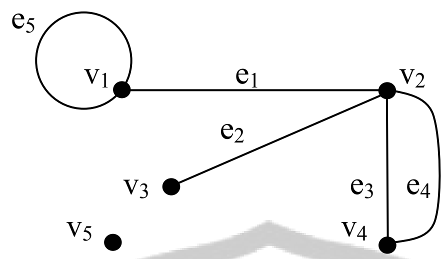
1. Teori Graf

Definisi 1.1

G disebut graf jika G terdiri dari 2 himpunan berhingga yaitu himpunan tak kosong V(G) yang elemen-elemennya disebut titik dan himpunan (mungkin kosong) E(G) yang elemen-elemennya disebut sisi, sedemikian hingga setiap elemen e dalam E(G) adalah sebuah pasangan tak terurut dari titik-titik di V(G). V(G) disebut himpunan titik dari G dan E(G) disebut himpunan sisi dari G.

Setiap sisi berhubungan dengan satu atau dua titik. Titik-titik tersebut dinamakan **titik ujung**. Sisi yang dua titik ujungnya sama disebut ***loop***, sebagai contoh sisi e_5 pada gambar 1. Titik yang tidak mempunyai sisi yang berhubungan dengannya disebut **titik terasing**, sebagai contoh titik v_5 (gambar 1) disebut titik terasing karena pada titik v_5 tidak terdapat sisi yang berhubungan dengan titik tersebut. Jika dua buah titik yang sama dihubungkan oleh lebih dari satu sisi disebut **sisi rangkap,** sebagai contoh sisi e_3 dan e_4, karena kedua sisi tersebut berhubungan dengan dua titik yang sama yaitu v_2 dan v_4. Graf yang tidak memuat *loop* dan sisi rangkap disebut **graf sederhana** dan disebut **graf tak sederhana** jika memuat loop atau sisi rangkap.

Contoh 1.1



Gambar 1. Graf G

Keterangan:

- Gambar 1. Graf G merupakan graf dengan $V = \{v_1,v_2,...,v_5\}$dan $E = \{e_1,e_2,..,e_5\}$.

Definisi 1.2

Dua buah titik dikatakan **bertetangga** bila keduanya terhubung langsung. Secara formal dinyatakan, v_j bertetangga dengan v_k jika $\exists e \in E$ sedemikian sehingga $e = (v_j,v_k)$.

Contoh 1.2

Perhatikan gambar 1 di atas, titik v_1 dan v_2 merupakan dua titik yang bertetangga. Sedangkan titik v_1 dan v_4 merupakan dua titik yang tidak saling bertetangga.

Definisi 1.3

Jika sebuah titik v_i merupakan titik ujung dari suatu sisi e_j, maka sisi e_j dikatakan **terkait/insiden** dengan titik v_i.

Contoh 1.3

Perhatikan gambar1 di atas, sisi e_1, e_2, e_3 dan e_4 adalah sisi yang terkait dengan titik v_2.

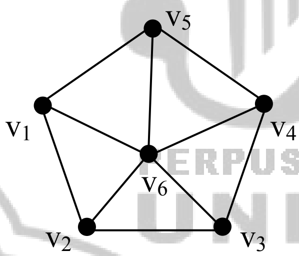
Definisi 1.4

Sebuah graf K disebut **graf bagian (*subgraf*)** dari graf G, dinotasikan $K \subseteq G$, jika $V(K) \subseteq V(G)$ dan $E(K) \subseteq E(G)$.

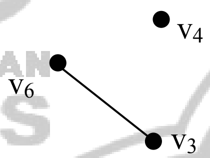
Dari definisi *subgraf*, ada beberapa hal yang dapat diturunkan :

a. Sebuah titik dalam graf G merupakan *subgraf* G.
b. Sebuah sisi dalam G bersamaan dengan kedua titik ujungnya merupakan *subgraf* G.
c. Setiap graf merupakan *subgraf* dari dirinya sendiri.
d. Dalam *subgraf* berlaku sifat transitif: Jika H adalah *subgraf* G dan G adalah *subgraf* K, maka H adalah *subgraf* K.

Contoh 1.4



Gambar 2.a Graf G_1

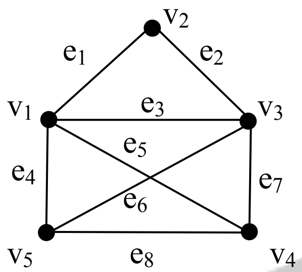


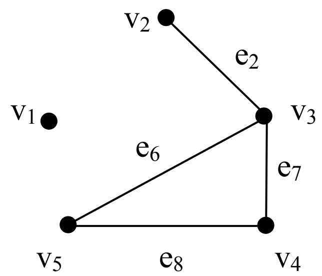
Gambar 2.b Graf G_2

Definisi 1.5

Sub *graf* $G_1 = (V_1,E_1)$ dari $G = (V,E)$ dikatakan ***spanning-subgraf*** jika $V_1 = V$.

Contoh 1.5





Gambar 3.a Graf G   Gambar 3.b *Spanning*-*subgraf* H

Keterangan:

- Gambar 3.a. Graf G, $V(G) = (v_1,v_2,v_3,v_4,v_5)$ dan

  $E(G) = (e_1,e_2,e_3,e_4,e_5,e_6,e_7,e_8)$.
- Gambar 3.b. *Spanning*-*subgraf* H, $V(H) = (v_1,v_2,v_3,v_4,v_5)$ dan

  $E(H) = (e_2,e_6,e_7,e_8)$.
- Gambar 3.b merupakan contoh *spanning-subgraf H* dari graf G, karena $V(H) = V(G)$.

Jalan (*walk*), Jejak (*trail*), Lintasan (*path*) dan Circuit

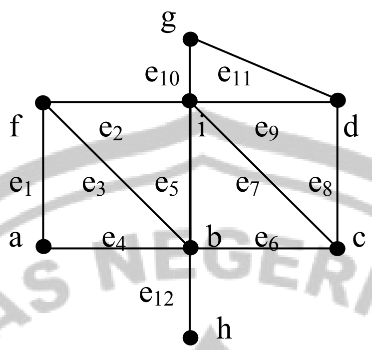
Definisi 1.6

Misalkan G adalah sebuah graf. Sebuah **jalan (*walk*)** di G adalah sebuah barisan berhingga (tak kosong) $W = v_0\ e_1\ v_1\ e_2\ ...\ e_k\ v_k$ yang suku-sukunya bergantian titik dan sisi, sehingga v_{i-1} dan v_i merupakan titik-titik akhir dari sisi e_i, $1 \leq i \leq k$. Titik v_0 dan v_k disebut titik awal dan titik akhir dari *walk* W. Titik $v_1,v_2,...,v_{k-1}$ disebut **titik internal**.

Walk W disebut tertutup jika v_i=v_j. Banyaknya sisi pada *walk* W disebut panjang *walk* W.

Contoh 1.6

Untuk lebih jelasnya perhatikan graf G pada gambar 4, W_1= a e_1 f e_2 i e_2 f e_3 b disebut jalan (*walk*) dengan panjang W_1= 4.



Gambar 4. Graf G

Definisi 1.7

Jika semua sisi pada *walk* W berbeda maka W disebut **jejak (*trail*)**. Sedangkan jejak tertutup yang semua titik internalnya berbeda disebut ***siklus***.

Contoh 1.7

Perhatikan graf G pada gambar 4, W_2 = a e_4 b e_6 c e_7 i e_5 b disebut jejak, dengan panjang jejak W_2 = 4 dan W_3 = a e_1 f e_3 b e_4 a disebut *siklus*.

Definisi 1.8

Jika semua titik dan sisi pada *walk* W berbeda, maka disebut **lintasan (*path*).**

Contoh 1.8

Perhatikan graf G pada gambar 4, W_4 = a e_4 b e_6 c e_7 i disebut lintasan, dengan panjang lintasan W_4 = 3 .

Definisi 1.9

Circuit adalah jejak tertutup.

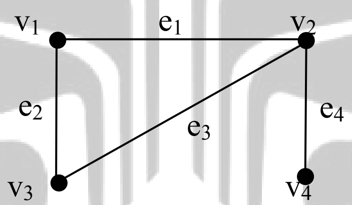
Contoh 1.9

Perhatikan graf G pada gambar 4, W_5 = a e_4 b e_6 c e_7 i e_5 b e_3 f e_1 a disebut *circuit*.

Definisi 1.10

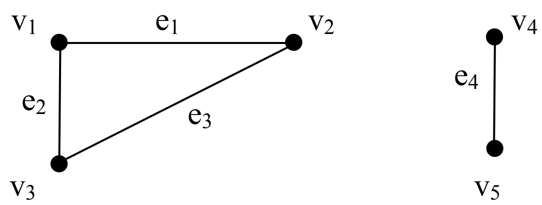
Misal G graf, komponen graf G adalah graf bagian terhubung maksimum dalam graf G. Sebuah graf G dikatakan **terhubung** jika untuk setiap dua titik berbeda u dan v di G, terdapat suatu lintasan yang menghubungkan titik u dan titik v. Sebaliknya, disebut **graf tak terhubung**. Gambar 5.a di bawah ini memuat contoh graf G_1 yang merupakan graf terhubung.

Contoh 1.10



Gambar 5.a Graf terhubung G_1

Apabila suatu graf tidak terhubung, maka graf tersebut terdiri atas beberapa komponen yang masing-masing komponennya adalah suatu graf terhubung atau suatu titik terasing. Sebagai contoh perhatikan graf pada gambar 5.b di bawah ini. Graf G tersebut merupakan graf tidak terhubung yang memiliki 2 buah komponen.



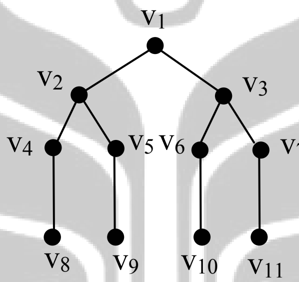
Gambar 5.b Graf tidak terhubung dengan 2 komponen

Definisi 1.11

Tree **(pohon)** adalah graf terhubung yang tidak memuat siklus.

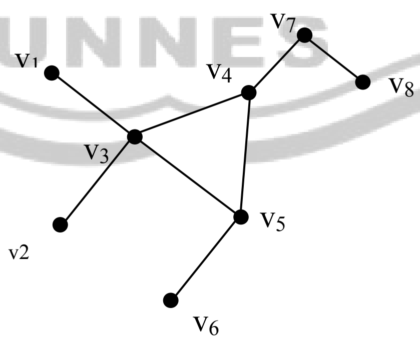
Contoh 1.11

Gambar 6.a di bawah ini memuat contoh pohon. Karena graf tersebut terhubung dan tidak memuat *siklus*.



Gambar 6.a *Tree*

Sedangkan gambar 6.b di bawah ini bukan merupakan suatu pohon, karena *walk* $v_3\ v_4\ v_5\ v_3$ merupakan siklus.



Gambar 6.b

<u>Teorema 1</u>

Misalkan G adalah suatu graf dengan n buah titik dan tepat n-1 sisi. Bila G tidak memuat siklus, maka G adalah pohon.

Bukti :

Diketahui graf G dengan n titik dan n-1 sisi. Karena telah diketahui G tidak mempunyai siklus, maka tinggal ditunjukan bahwa G terhubung. Andaikan G adalah tidak terhubung. Maka G mempunyai paling sedikit $K \geq 2$ komponen. Sebut $G_1, G_2, \ldots, G_k$ adalah komponen-komponen dari G, dengan masing-masing komponen mempunyai $n_1, n_2, \ldots, n_k$ titik, dimana $n_1 + n_2 + \ldots + n_k = n$. Setiap komponen tidak mempunyai siklus, sehingga masing-masing komponen akan memuat sisi paling banyak n_i-1, untuk $i = 1,2,\ldots,k$. Dengan demikian jumlah seluruh sisi dari graf G adalah:

$$(n_1 - 1) + (n_2 - 1) + \ldots + (n_k - 1) = n - k$$

Karena G mempunyai paling sedikit $K \geq 2$ komponen, maka hal ini bertentangan dengan pemisalan bahwa G mempunyai n-1 sisi. Jadi pemisalan bahwa G tak terhubung adalah tidak benar. G adalah graf yang terhubung dan tidak mempunyai siklus. Dengan demikian terbukti bahwa G adalah suatu *tree*.

(Terbukti).

Definisi 1.12

Misal G sebuah graf. Sebuah pohon di G yang memuat semua titik G disebut **pohon pembangun (*spanning-tree*)** dari G.

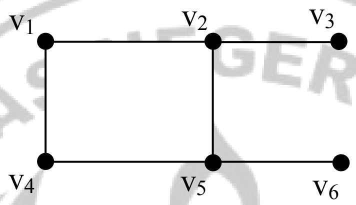
Definisi 1.12.1

Branch adalah sebuah sisi yang terdapat dalam sebuah *spanning-tree.*

Definisi 1.12.2

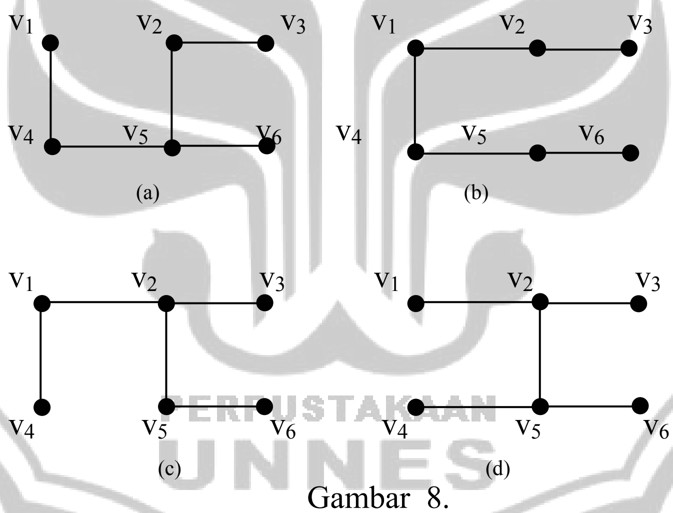
Chord adalah sebuah sisi yang tidak terdapat dalam sebuah *spanning-tree*, tetapi berada dalam graf G.

Contoh 1.12



Gambar 7. Graf G

Spanning-*tree* dari graf G di atas ditunjukan pada gambar 8 (a)-(d).



Gambar 8.

Definisi 1.13

Misal v sebarang titik pada graf G. **Derajat** titik v adalah banyaknya sisi yang terkait dengan titik tersebut dan sisi suatu *loop* dihitung dua kali. Derajat titik v dinotasikan d(v) atau d_G (v). **Derajat total** G adalah jumlah derajat semua titik dalam G.

Contoh 1.13

Perhatikan gambar 6.b dapat dilihat bahwa derajat pada titik v_3 ($d(v_3)$) sebanyak 4 buah. Sedangkan $d(v_4) = d(v_5) = 3$, $d(v_7) = 2$ dan $d(v_1) = d(v_2) = d(v_6) = d(v_8) = 1$. Sehingga derajat totalnya adalah $\sum_{i=1}^{8} d(v_i) = 1 + 1 + 4 + 3 + 3 + 1 + 2 + 1 = 16$.

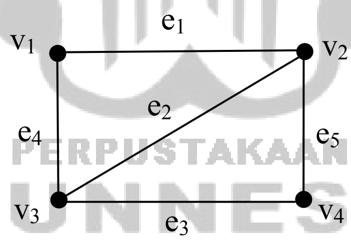
2. Jenis-jenis graf

2.1 Berdasarkan jenis sisinya, graf dibedakan dalam 2 kategori yaitu Graf Berarah (*Directed Graph*) dan Graf Tak Berarah (*Undirected Graph*).

Definisi 2.1.1

Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah. Urutan pasangan titik pada graf tak berarah tidak diperhatikan, jadi sisi (*u*, *v*) sama dengan (*v*, *u*). Graf tak berarah *G* sering disebut dengan *graf G* saja.

Contoh 2.1.1



Gambar 11.a Graf Tak Berarah G

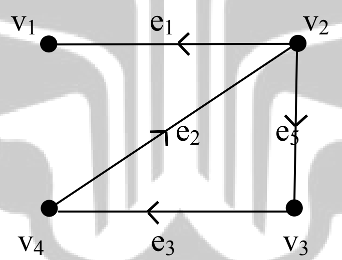
Definisi 2.1.2

Graf berarah adalah graf yang setiap sisinya merupakan pasangan terurut dari dua titik. Pada graf berarah, sisi (u,v) tidak sama dengan (v,u). Untuk sisi (u,v), titik u merupakan titik awal dan titik v merupakan titik akhir. Lintasan dan circuit pada graf berarah memiliki konsep dasar yang sama pada graf tak berarah, yang membedakan hanyalah ada tidaknya orientasi arah pada tiap-tiap sisi pada lintasan dan circuit tersebut. Lintasan pada graf berarah sering disebut juga **lintasan berarah**, sedangkan circuit pada graf berarah disebut **circuit berarah.**

Pada graf berarah terdapat dua jenis derajat yaitu derajat masuk atau sering disebut ***in-degree*** (simbol $d_{in}(v_i)$) adalah banyaknya sisi yang menuju titik v_i dan derajat keluar atau ***out degree*** (simbol $d_{out}(v_i)$) adalah banyaknya sisi yang keluar dari titik v_i.

Sebuah titik v pada graf berarah G disebut **titik ujung** jika $d_{in}(v) + d_{out}(v) = 1$.

Contoh 2.1.2



Gambar 11.b Graf Berarah G

Keterangan :

- Salah satu contoh lintasan berarah dan circuit berarah pada gambar 11.b yakni:

  Lintasan berarah: v_3 e_3 v_4 e_2 v_2 e_1 v_1

  Circuit berarah: v_3 e_3 v_4 e_2 v_2 e_5 v_3

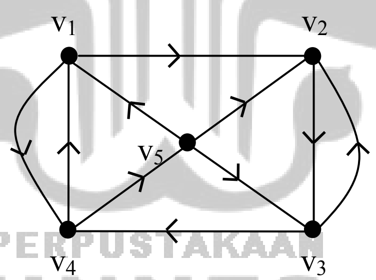
- Graf pada gambar 11.b merupakan graf berarah, dengan $d_{in}(v_1) = d_{in}(v_2) = d_{in}(v_3) = d_{in}(v_4) = 1$, sedangkan $d_{out}(v_1) = 0$, $d_{out}(v_2) = 2$, $d_{out}(v_3) = d_{out}(v_4) = 1$. Karena $d_{in}(v_1) + d_{out}(v_1) = 1$ maka v_1 adalah titik ujung.

2.2 Berdasarkan arah sisinya, dalam graf berarah dikenal 2 jenis keterhubungan yaitu terhubung kuat dan terhubung lemah.

Definisi 2.2.1

Sebuah graf berarah G dikatakan **terhubung kuat** jika setiap dua titik u dan v di G dihubungkan dengan lintasan berarah. Gambar 12.a. graf G_1 pada contoh 2.2.1 merupakan graf terhubung kuat, karena setiap dua titik pada graf G_1 dapat dihubungkan dengan lintasan berarah. Maka graf berarah G_1 adalah graf terhubung kuat.

Contoh 2.2.1

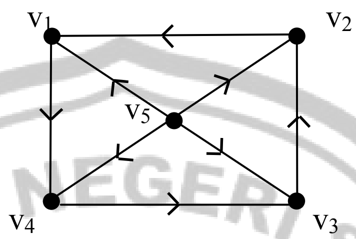


Gambar 12.a Graf G_1

Definisi 2.2.2

Sebuah graf berarah G dikatakan **terhubung lemah** jika G tidak terhubung kuat, tetapi graf tak berarah dari G terhubung. Sebagai contoh lihat graf G_2 pada gambar 12.b. Dalam graf G_2 tidak ada lintasan berarah yang menghubungkan v_1 ke v_5. Akan tetapi, jika semua arah sisi dihilangkan (sehingga G_2 menjadi graf tidak berarah), maka G_2 merupakan graf yang terhubung. Jadi G_2 merupakan graf yang terhubung lemah.

Contoh 2.2.2

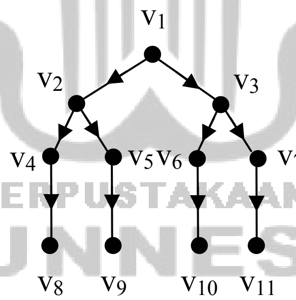


Gambar 12.b Graf G_2

Definisi 1.14

Sebuah pohon pada graf berarah yang tidak memuat siklus berarah disebut **pohon berarah**. Sedangkan sebuah titik pada pohon berarah yang memiliki derajat masuk 0 disebut ***root.***

Contoh 1.14



Gambar 13. Pohon Berarah dengan *root* v_1

Definisi 1.15

Misal G=(V,A) adalah graf berarah dan $r \in V$ adalah sebuah titik yang disebut root.

Arborescence dengan akar r adalah sebuah subgraf T=(V,F) dari G yang tidak memuat sepasang sisi berlawanan sedemikian hingga kondisi berikut terpenuhi.

a) Jika arah dari tiap sisi diabaikan, maka T adalah sebuah *spanning-tree.*

b) Terdapat sebuah lintasan dari r ke setiap titik yang lain v∈*V.*

(Weisstein, Eric W; 1990)

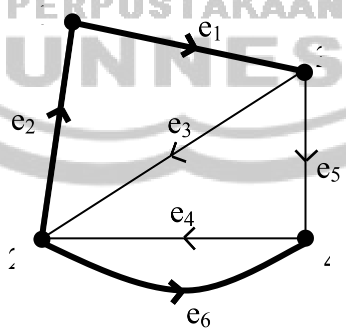
Contoh 1.15

Gambar 13 pada contoh 1.14 merupakan *arborescence* dengan v_1 sebagai *root.*

Definisi 1.16

Spanning-tree pada graf terhubung berarah dengan n buah titik, analog dengan *spanning-tree* pada graf tak berarah yaitu terdiri dari n-1 buah sisi. ***Spanning arborescence*** pada graf terhubung berarah adalah *spanning-tree* yang *arborescence.* Sehingga *spanning-arborescence* merupakan *arborescence.* Subgraf yang dicetak tebal pada digraf G (gambar 14) merupakan contoh dari s*panning arborescence* (Narsingh Deo; 1997).

Contoh 1.16



Gambar 14. Digraf G

3. Teori Matriks

Definisi 3.1

Matriks adalah susunan segi empat siku-siku dari bilangan-bilangan yang diatur dalam baris dan kolom. Bilangan-bilangan tersebut dinamakan entri dalam matriks. Matriks ditulis sebagai berikut :

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Susunan di atas disebut sebuah matriks dengan ukuran (*ordo*) m kali n (mxn) karena memiliki m baris dan n kolom. Matriks yang berordo n kali n (nxn) sering disebut **matriks persegi**. Matriks lazimnya dinotasikan dengan sebuah huruf besar, dan entri-entri di dalam matriks ditulis dengan huruf kecil (a_{ij}, b_{ij} dst) dimana i menyatakan baris ke i dan j menyatakan kolom ke j dari entri tersebut. Sedangkan di dalam teori matriks terdapat beberapa operasi pada matriks antara lain: penjumlahan matriks dan perkalian matriks.

Penjumlahan Matriks

Definisi 3.1.1

Jika A dan B adalah sembarang dua matriks yang ordonya sama, maka A+B adalah matriks yang diperoleh dengan menambahkan bersama-sama entri yang bersesuaian dalam kedua matriks tersebut. Matriks yang berukuran berbeda tidak dapat ditambahkan.

Contoh 3.1.1

Diketahui matriks A = $\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$ dan B = $\begin{bmatrix} 0 & 3 \\ 1 & 4 \end{bmatrix}$, Hitung A + B!

Maka A + B = $\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} + \begin{bmatrix} 0 & 3 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 1+0 & 2+3 \\ 2+1 & 5+4 \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 3 & 9 \end{bmatrix}$

Perkalian Matriks

Definisi 3.1.2

Jika A adalah matriks m x r dan B adalah matriks r x n, maka hasil kali AB didefinisikan sebagai matriks C dengan ordo m x n yang entri-entrinya dihitung dari elemen-elemen dari A dan B menurut

$$c_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj} \quad \text{dengan i = 1,2, . . .m; j = 1,2,. . . .r.}$$

Dalam hasil kali matriks AB, A disebut pengali dan B pengali belakang. Hasil kali AB dapat ditentukan jika jumlah kolom pada matriks A sama dengan jumlah baris pada matriks B.

Contoh 3.1.2

Diketahui matriks A = $\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$ dan B = $\begin{bmatrix} 0 & 3 \\ 1 & 4 \end{bmatrix}$, Hitung A x B!

Maka A x B = $\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} \text{x} \begin{bmatrix} 0 & 3 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 1x0+2x1 & 1x3+2x4 \\ 2x0+5x1 & 2x3+5x4 \end{bmatrix} = \begin{bmatrix} 2 & 11 \\ 5 & 26 \end{bmatrix}$

Definisi 3.2

Matriks persegi disebut **matriks segitiga atas**, jika semua entri di bawah diagonal utama adalah nol. Sedangkan disebut **matriks segitiga bawah**, jika semua entri di atas diagonal utamanya adalah nol.

Contoh 3.2

Bentuk umum matriks segitiga atas yang berordo 4x4.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

Bentuk umum matriks segitiga bawah yang berordo 4x4.

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

4. Representasi Graf dengan matriks

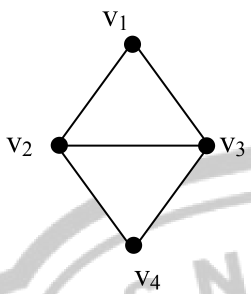
Terdapat beberapa jenis matriks yang dapat digunakan untuk merepresentasikan graf, antara lain:

4.1 Matriks Ketetanggaan (*adjacency matriks*)

4.1.1 Pada Graf Tak Berarah

Misal G adalah sebuah graf dengan n titik, $n \geq 1$. Matriks ketetanggaan G adalah matriks $A=[a_{ij}]$ yang merupakan matriks bujur sangkar berukuran n x n. Entri a_{ij} menyatakan banyaknya sisi yang menghubungkan titik v_i dan titik v_j. Contoh sebuah graf dengan matriks ketetanggaannya disajikan pada gambar 15 di bawah ini.

Contoh 4.1.1



(a)

$$
A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}
\begin{array}{c}
\begin{array}{cccc} v1 & v2 & v3 & v4 \end{array} \\
\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}
\end{array}
$$

(b)

Gambar 15.(a) Graf G (b) Matriks ketetanggan graf G

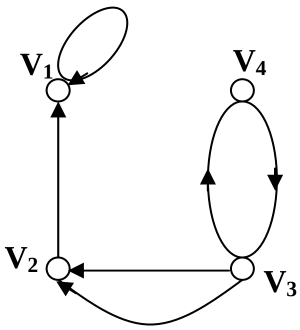
4.1.2 Pada Graf Berarah

Misal G adalah sebuah graf berarah dengan n buah titik. Matriks X dengan ordo (nxn) merupakan matriks ketetanggaan dari G yang didefinisikan sebagai berikut.

$X = [x_{ij}]$, dengan x_{ij} banyaknya sisi berarah yang menghubungkan dari titik v_i menuju titik v_j.

Bila G tidak mengandung sisi rangkap. Maka entri-entri pada matriks X adalah 0 dan 1. Jika graf berarah G mengandung sisi rangkap, entri-entri pada matriks X merupakan bilangan bulat non negatif. Sebagai contoh representasi graf berarah dalam matriks ketetanggaan dapat dilihat pada contoh 4.1.2 di bawah ini.

Contoh 4.1.2



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

a. Graf G b. Matriks *ketetanggaan graf G*

Gambar 16

Matriks ketetanggaan untuk graf sederhana dan tidak berarah selalu simetri, selain itu diagonal utamanya selalu nol karena tidak ada *loop*. sedangkan untuk graf berarah matriks ketetanggannya belum tentu simetri (akan simetri jika berupa graf lengkap).

4.2 Penyajian Graf dengan Notasi Matriks Keterkaitan (*Incidence Matriks*)

4.2.1 Pada Graf Tak Berarah

Misalkan G adalah sebuah graf dengan n titik, e sisi, dan tidak memuat *loop*. Definisikan sebuah matriks $A=[a_{ij}]$ berukuran nxe dengan n menyatakan baris dan e menyatakan kolom, di mana elemen matriksnya untuk graf sederhana sebagai berikut.

$$a_{ij} = \begin{cases} 1, \text{ jika sisi ke-j terkait dengan titik } v_i \\ 0, \text{ jika lainnya} \end{cases}$$

matriks semacam ini disebut matriks insidensi.

Sedangkan untuk graf tak sederhana, elemen matriksnya sebagai berikut.

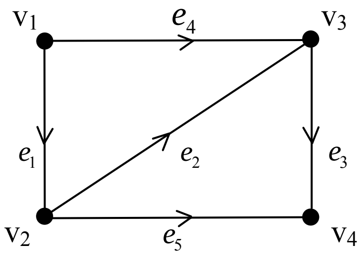
$$a_{ij} = \begin{cases} 0, \text{ jika sisi } e_j \text{ tidak terkait dengan titik } v_i \\ 1, \text{ jika } e_j \text{ terkait dengan } v_i \text{ dan } e_j \text{ bukan loop} \\ 2, \text{ jika } e_j \text{ terkait dengan } v_i \text{ dan } e_j \text{ adalah loop.} \end{cases}$$

4.2.2 Pada Graf Berarah

Matriks *exhaustive incidence* $A_e=[a_{ij}]$ untuk semua graf yang menyajikan hubungan antara sebuah titik dengan sisi yang menghubungkannya. Untuk graf terhubung berarah elemen matriks *exhaustive incidence* $A_e=[a_{ij}]$ di mana;

$$a_{ij} = \begin{cases} 0, \text{ jika sisi j tidak terkait dengan titik i} \\ 1, \text{ jika sisi j terkait dengan titik i dan arah sisinya keluar dari i} \\ -1, \text{ jika sisi j terkait dengan titik i dan arah sisinya masuk ke i} \end{cases}$$

Apabila diadakan penghapusan terhadap baris yang bersesuaian dengan sembarang titik pada matriks *exhaustive incidence*, maka terbentuk sub matriks (n-1)xm. Titik yang bersesuaian dengan baris yang dihapus tersebut disebut titik acuan (*verteks reference*), sedangkan matriks berukuran (n-1)xm yang terjadi ini disebut matriks keterkaitan (*matriks incidence*) dengan notasi A.



Gambar 17. Graf G

Sebagai contoh dari graf G pada gambar 17, bentuk matriks *exhaustive incidence* sebagai berikut :

$$A_e = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 \end{pmatrix} \end{array}$$

Sedangkan bentuk matriks keterkaitannya dipengaruhi oleh titik acuan *(vertek reverence)* yang dipilih, bentuk-bentuk matriks keterkaitannya antara lain:

a) Dengan mengambil v_1 sebagai titik acuannya maka bentuk matriks keterkaitannya adalah:

$$A = \begin{array}{c} \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 \end{pmatrix} \end{array}$$

b) Dengan mengambil v_2 sebagai titik acuannya maka bentuk matriks keterkaitannya adalah:

$$A = \begin{array}{c} \\ v_1 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 \end{pmatrix} \end{array}$$

c) Dengan mengambil v_3 sebagai titik acuannya maka bentuk matriks keterkaitannya adalah:

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 \end{pmatrix} \end{array}$$

d) Dan jika v_4 yang diambil sebagai titik acuannya maka bentuk matriks keterkaitannya adalah:

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & -1 & 0 \end{pmatrix} \end{array}$$

4.3 Penyajian Graf dengan matriks *in-degree*

Suatu Matriks *in-degree*, K(G) dari sebuah graf terhubung berarah G = (V,E) tanpa *loop* dengan V = {v_1, v_2, ... , v_n} dan E = {e_1, e_2, ..., e_n} adalah sebuah matriks dengan ukuran n x n yang mempunyai sifat :

$$K(G) = \begin{cases} d_{in}(v_i), & \text{jika } i = j \\ -\,x_{ij}, & \text{jika } i \neq j \text{ maka } x_{ij} \text{ adalah entri dari matriks ketetanggaan, yang diberi tanda negatif.} \end{cases}$$

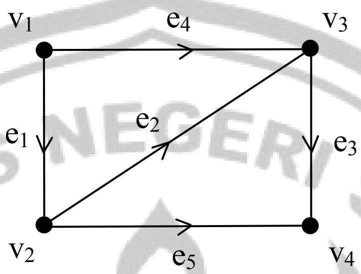
Apabila diadakan penghapusan terhadap sebuah baris dan kolom yang bersesuaian dengan *root* v_i, maka akan diperoleh submatrik K_{ii} dengan mengeluarkan baris-i dan kolom-i dari matriks K(G).

Contoh 4.3

Diketahui : Graf $G=(V,E)$ dimana $V=\{v_1,v_2,v_3,v_4\}$ dan $E=\{e_1, e_2, e_3, e_4, e_5\}$

Ditanya: : Matriks *in-degree* K(G) dan $K_{i\,j}$ untuk i=1 dan j=1

Jawab :



Gambar 18. Graf terhubung berarah

Bentuk matriks *in-degree* K(G) adalah :

$$K(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} \text{v1} & \text{v2} & \text{v3} & \text{v4} \end{array} \\ \begin{pmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 2 \end{pmatrix} \end{array}$$

Ambil i = 1, berarti mengeluarkan baris 1 dari matriks K(G)

j = 1, berarti mengeluarkan kolom 1 dari matriks K(G), sehingga matriks K_{ij} nya adalah sebagai berikut :

$$K(G) = \begin{array}{c|cccc} & v1 & v2 & v3 & v4 \\ \hline v_1 & 0 & -1 & -1 & 0 \\ v_2 & 0 & 1 & -1 & -1 \\ v_3 & 0 & 0 & 2 & -1 \\ v_4 & 0 & 0 & 0 & 2 \end{array} \dashrightarrow K_{11} = \begin{bmatrix} 1 & -1 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

Definisi 1.17

Himpunan bilangan-bilangan bulat {1, 2, …, n} adalah susunan bilangan-bilangan bulat menurut suatu aturan tanpa menghilangkan atau mengulangi bilangan tersebut.

Definisi 1.18

Sebuah invers (*inversion*) dikatakan terjadi dalam permutasi $(j_1, j_2, \dots, j_n)$ jika sebuah bilangan bulat yang lebih besar mendahului sebuah bilangan bulat yang lebih kecil. Jumlah invers seluruhnya yang terjadi dalam permutasi dapat diperoleh sebagai berikut:

a. Carilah banyaknya bilangan bulat yang lebih kecil dari j_1 dan membawa j_1 dalam permutasi tersebut.
b. Carilah banyaknya bilangan bulat yang lebih kecil dari j_2 dan membawa j_2 dalam permutasi tersebut.
c. Teruskanlah proses perhitungan ini untuk j_3, ..., j_{n-1}. Jumlah bilangan-bilangan ini akan sama dengan jumlah invers seluruhnya dalam permutasi tersebut.

Definisi 1.19

Misalkan A adalah matriks kuadrat. Fungsi determinan dinyatakan oleh det, dan didefinisikan det(A) sebagai semua hasil kali elementer bertanda dari A. Jumlah det(A) kita namakan determinan A. Determinan A sering ditulis secara simbolis sebagai:

$$\text{Det (A)} = |A| = \sum \pm a_{1j_1} a_{2j_2} \cdots a_{nj_n}$$

dimana $\sum$ menunjukan bahwa suku-suku tersebut harus dijumlahkan terhadap semua permutasi (j_1, j_2, ..., j_n) dan simbol (+) atau (–) dapat dipilih dalam masing-masing suku sesuai dengan apakah permutasi itu genap atau ganjil.

Sebuah permutasi dinamakan **genap** jika jumlah invers seluruhnya adalah sebuah bilangan bulat yang genap (simbol (+)) dan dinamakan **ganjil** jika jumlah invers seluruhnya adalah sebuah bilangan bulat ganjil (simbol (-)).

Contoh 1.19

a. Diketahui matriks A= $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$.

Daftarkanlah semua hasil kali elementer bertanda dari matriks tersebut!

Penyelesaian:

Karena setiap hasil kali elementer mempunyai dua faktor, dan karena setiap faktor berasal dari baris yang berbeda, maka hasil kali elementer dapat dituliskan dalam bentuk $a_{1\_}a_{2\_}$ di mana titik kosong menandakan nomor kolom. Karena tidak terdapat dua faktor dalam hasil kali tersebut dari kolom yang sama, nomor kolom haruslah $\underline{1}$ $\underline{2}$ atau $\underline{2}$ $\underline{1}$. Maka hasil kali elementer hanyalah a_{11} a_{22} dan a_{12} a_{21}.

Tabel 1.1. Berikut mengklasifikasikan berbagai permutasi dari matiks A di atas.

Hasil kali elementer	Permutasi terasosiasi	Banyak inversi	Genap atau ganjil	Hasil kali elementer bertanda
$a_{11}a_{22}$ $a_{12}a_{21}$	(1,2) (2,1)	0 1	Genap Ganjil	$+a_{11}a_{22}$ $-a_{12}a_{21}$

Jadi det(A) = $+\, a_{11}a_{22} - a_{12}a_{21}.$

b. Diketahui matriks B = $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$

Daftarkanlah semua hasil kali elementer bertanda dari matriks tersebut!

Penyelesaian:

Karena setiap hasil kali elementer mempunyai tiga faktor, yang masing-masing berasal dari baris yang berbeda, maka hasil kali elementer dapat dituliskan dalam bentuk $a_{1\_}\, a_{2\_}\, a_{3.}$ Karena tidak terdapat dua faktor dalam hasil kali tersebut berasal dari kolom yang sama, maka nomor-nomor kolom tersebut harus membentuk permutasi himpunan {1,2,3}. Disini permutasi 3! = 6 menghasilkan daftar hasil kali elementer berikut.

$a_{11}a_{22}a_{33}$ $a_{12}a_{21}a_{33}$ $a_{13}a_{21}a_{32}$

$a_{11}a_{23}a_{32}$ $a_{12}a_{23}a_{31}$ $a_{13}a_{22}a_{31}$

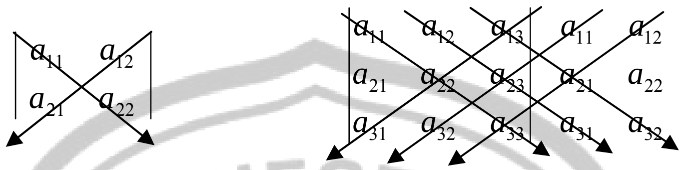
Tabel 1.2. Berikut mengklasifikasikan berbagai permutasi dari matriks B di atas.

Hasil kali elementer	Permutasi terasosiasi	Banyak inversi	Genap atau ganjil	Hasil kali elementer bertanda
$a_{11}a_{22}a_{33}$	(1,2,3)	0	Genap	$a_{11}a_{22}a_{33}$
$a_{11}a_{23}a_{32}$	(1,3,2)	1	Ganjil	$-a_{11}a_{23}\,a_{32}$
$a_{12}a_{21}a_{33}$	(2,1,3)	1	Ganjil	$-a_{12}a_{21}a_{33}$
$a_{12}a_{23}a_{31}$	(2,3,1)	2	Genap	$a_{12}a_{23}a_{31}$
$a_{13}a_{21}a_{32}$	(3,1,2)	2	Genap	$a_{13}a_{21}a_{32}$
$a_{13}a_{22}a_{31}$	(3,2,1)	3	Ganjil	$-a_{13}a_{22}a_{31}$

Jadi det(B) = $a_{11}\,a_{22}\,a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{11}a_{23}\,a_{32} - a_{12}a_{21}a_{33}$ $-a_{13}a_{22}a_{31}$.

Seperti yang ditunjukkan oleh contoh di atas, matriks A yang berukuran nxn mempunyai n! hasil kali elementer. Hasil kali elementer tersebut adalah hasil kali yang berbentuk $a_{1j_1}a_{2j_2}\cdots a_{nj_n}$, dimana $(j_1, j_2, \ldots, j_n)$ adalah permutasi himpunan {1, 2, ... , n}. Yang kita artikan dengan *hasil kali elementer bertanda A* adalah hasil kali elementer $a_{1j_1}a_{2j_2}\cdots a_{nj_n}$ dikalikan dengan +1 atau -1.

Metode yang digunakan untuk mencari determinan suatu matriks persegi berordo 2x2 atau 3x3 seperti contoh di atas terlalu rumit. Untuk memudahkan dalam menghitung determinan matriks persegi digunakan metode SARRUS (khusus matriks berordo ≤ 3).



(a) (b)

Gambar 19

Determinan matriks berordo 2 pada gambar 19.(a), diperoleh dengan mengalikan entri-entri pada panah yang mengarah ke kanan dan mengurangkan hasil kali entri-entri pada panah-panah yang mengarah ke kiri. Sedangkan determinan untuk matriks berordo 3 pada gambar 19.(b), diperoleh dengan menyalin kembali kolom pertama dan kolom kedua menjumlahkan hasil kali entri-entri pada panah-panah yang mengarah ke kanan dan mengurangkan jumlah hasil kali panah-panah yang mengarah ke kiri maka diperoleh determinan matriks tersebut.

Teorema 2

Jika A merupakan matriks segitiga bawah ataupun segitiga atas maka det(A) berupa hasil kali entri-entri pada diagonal utama, yaitu:

$$\det(A) = a_{11}\, a_{22} \ldots a_{nn}.$$

Bukti:

Misal A adalah matriks segitiga bawah, akan dibuktikan det(A) = $a_{11}\, a_{22} \ldots a_{nn}$.

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Satu-satunya hasil kali elementer A yang tidak sama dengan nol adalah $a_{11}\ a_{22}\ \ldots\ a_{nn}$. Untuk melihat hal ini, tinjau hasil kali elementer khas $a_{1j_1}\, a_{2j_2} \cdots a_{nj_n}$. Karena $a_{12} = a_{13} = \ldots = a_{1n} = 0$, maka harus dipunyai $j_1 = 1$, agar hasil kali elementernya tak nol. Jika $j_1 = 1$, maka $j_2 \neq 1$, karena tidak ada dua faktor yang berasal dari kolom yang sama.

Selanjutnya karena $a_{23} = a_{24} = \ldots = a_{2n} = 0$, maka kita harus mempunyai $j_2=2$, agar mempunyai hasil kali elementer tak nol. Dengan mengulangi langkah tersebut, maka diperoleh $j_3 = 3$, $j_4 = 4$, ... , $j_n = n$. Karena $(j_1, j_2, \ldots, j_n)$ merupakan permutasi genap maka $a_{11}\ a_{22}\ \ldots\ a_{nn}$ dikalikan oleh +1 dalam membentuk hasil kali elementer bertanda tersebut, maka diperoleh

$$\det(A) = a_{11}\, a_{22} \ldots a_{nn}.$$

Suatu argumen yang serupa dengan argumen yang baru saja disajikan dapat diterapkan pada sebarang matriks segitiga. **(Terbukti).**

<u>Teorema 3</u>

Misalkan A, B dan C adalah matriks-matriks nxn yang berbeda hanya satu kolom, misalnya kolom ke-r dari C dapat diperoleh dengan menjumlahkan entri-entri yang bersesuaian pada kolom ke-r dari A dan B.

Maka

$$\det(C) = \det(A) + \det(B)$$

Hasil yang sama berlaku untuk baris.

Bukti:

Tanpa mengurangi keumuman, maka dipilih matriks A, B dan C berordo 2x2.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{dan} \quad B = \begin{bmatrix} a_{11} & b_{12} \\ a_{21} & b_{22} \end{bmatrix}$$

Maka

$$\begin{aligned} \det(A) + \det(B) &= (a_{11}.a_{22} - a_{12}.a_{21}) + (a_{11}.b_{22} - b_{12}.a_{21}) \\ &= a_{11}.a_{22} + a_{11}.b_{22} - a_{12}.a_{21} - b_{12}.a_{21} \\ &= a_{11}(a_{22} + b_{22}) - a_{21}(a_{12} + b_{12}) \\ &= \det \begin{bmatrix} a_{11} & (a_{12} + b_{12}) \\ a_{21} & (a_{22} + b_{22}) \end{bmatrix} \\ &= \det(C) \quad \textbf{(Terbukti).} \end{aligned}$$

Definisi 1.20

Misalkan matriks A = $[a_{ij}]$ berordo nxn, dan A_{ij} suatu submatriks dari A yang berordo (n-1) x (n-1) di mana baris ke i dan kolom ke j dihilangkan. Kofaktor a_{ij} adalah $(-1)^{i+j} . |M_{ij}|$.

Contoh 1.20

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 1 \end{bmatrix} \dashrightarrow A_{32} = \begin{vmatrix} 2 & 4 \\ 5 & 7 \end{vmatrix} = -6$$

Jadi kofaktor a_{32} adalah $(-1)^{3+2} \cdot (-6) = 6$.

BAB III

METODE PENELITIAN

Langkah-langkah yang dilakukan dalam penelitian ini sebagai berikut.

1. Penemuan Masalah

Metode ini merupakan tahapan pertama dalam penelitian yaitu dengan pencarian ide atau gagasan materi dari bidang kajian yang dipilih dan dijadikan permasalahan untuk dikaji pada penelitian ini.

2. Perumusan Masalah

Tahap ini dimaksudkan untuk memperjelas permasalahan yang telah ditemukan yaitu bagaimana menentukan jumlah *spanning-treee* pada suatu graf berarah?

3. Studi Pustaka

Studi pustaka adalah metode pengumpulan data dengan cara mencari informasi melalui buku-buku, koran, majalah, internet dan literatur lainnya. Dalam hal ini pengumpulan data dilakukan dengan cara membaca, mencari data di internet, dan mempelajari buku-buku literatur dan sumber bacaan lainnya yang berkaitan dengan obyek pembahasan.

4. Analisis Pemecahan Masalah

Untuk menentukan jumlah *spanning-tree* dari graf berarah, dapat dilakukan dengan 2 cara yaitu.

4.1 Menggunakan metode penukaran sisi (*edge exchange*).

Suatu graf berarah G = (V,E) yang terhubung sederhana dapat ditentukan jumlah *spanning-treenya* dengan metode penukaran sisi. Adapun langkah-langkah untuk menentukan jumlah *spanning-tree* dari suatu graf berarah adalah:

a. Pilih sebuah titik pada graf G sebagai *root*.
b. Buat *spanning-tree* awal dari graf G sebut graf T_1, dimana *spanning-tree* yang terbentuk merupakan *spanning-arborescence.*
c. Untuk selanjutnya untuk membentuk *spanning-tree* yang lain dengan berdasar pada *spanning-tree* awal/graf T_1 dari graf G.
d. Tambahkan sebuah *chord* pada *spanning-tree* awal/graf T_1.
e. Hapus *branch* pada graf T_1 agar tidak terjadi sirkuit sehingga akan terbentuk *spanning-tree* yang baru, akan tetapi *spanning-tree* baru yang terbentuk harus memuat lintasan berarah dari *root* ke semua titik yang lain.
f. Kemudian ulangi langkah ke-4 dan ke-5, untuk membentuk *spanning-tree* yang lain sampai diperoleh semua *spanning-tree* yang berbed.

Akan tetapi jika graf G = (V,E) tersebut merupakan sisi dan titik dalam jumlah besar sehingga tidak memungkinkan menggunakan metode penukaran sisi (*edge exchange*), maka jumlah dan bentuk *spanning-treenya* dapat dicari dengan menggunakan matriks *in-degree.*

4.2 Menggunakan matriks *in-degree.*

Adapun langkah-langkah yang dilakukan untuk menentukan jumlah *spanning-tree* pada graf terhubung berarah adalah sebagai berikut:

a. Pilih sebuah titik v_q sebagai *root,* misal G graf berarah, representasikan graf G ke dalam bentuk matriks *in-degree* (sebut matriks K(G)).
b. Lakukan penghapusan sebuah baris dan kolom matriks K(G). Maka akan diperoleh matriks K_{qq}, di mana matriks tersebut diperoleh dengan mengeluarkan baris ke-q dan kolom ke-q sembarang dari matriks K(G).
c. Hitung determinan matriks K_{qq}, kemudian untuk menentukan jumlah *spanning-tree* dari graf terhubung berarah yaitu dengan mengalikan kofaktor k_{qq} dengan determinan matriks K_{qq}.

5. Penarikan Simpulan

Tahap ini merupakan tahap terakhir dari penelitian. Setelah menganalisis dan memecahkan masalah berdasarkan studi pustaka dan pembahasannya kemudian dibuat sebagai simpulan sebagai jawaban dari permasalahan yang telah dirumuskan sebelumnya.

BAB IV

PENENTUAN JUMLAH *SPANNING-TREE* PADA GRAF BERARAH

1. Menggunakan Metode Penukaran Sisi (*Edge Exchange*)

Berikut ini akan dikaji bagaimana mendapatkan *spanning-tree* dari graf berarah. Yang dimaksud dengan *spanning-tree* pada pembahasan ini adalah *spanning-arborescence*

Untuk mendapatkan *spanning-tree* dari graf berarah, terlebih dahulu memilih sebuah titik pada suatu graf berarah sebagai *root.* Kemudian buat *spanning-tree* dari suatu graf terhubung berarah yang memuat lintasan berarah dari *root* yang terpilih ke semua titik pada *spanning-tree* tersebut. Misal T_1 adalah *spanning-tree* awal dari graf terhubung tersebut, maka untuk memperoleh *spanning-tree* yang lain dapat dilakukan metode penukaran sisi sebagai berikut :

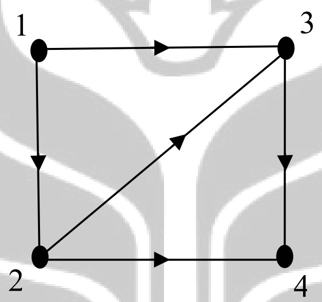
- Langkah 1 : tambahkan *chords* pada T_1 (*spanning-tree* awal)
- Langkah 2 : hapus *branch* di T_1 agar tidak terjadi siklus, sehingga terbentuk *spanning-tree* yang baru. Akan tetapi *spanning-tree* yang terbentuk merupakan *spanning arborescence.*
- Langkah 3 : selanjutnya ulangi langkah ke-1 dan ke-2 untuk membentuk *spanning-tree* yang lain dengan berdasar pada T_1, sampai diperoleh semua *spanning-tree* yang berbeda.

(Narsingh Deo; 1997)

Metode inilah yang disebut penukaran sisi (*Edge Exchange*). Sesuai dengan metode tersebut, maka dapat dicari semua *spanning-tree* dari graf G, dengan bertumpu pada graf G dan *spanning-tree* pertama (T_1) yang telah terbentuk.

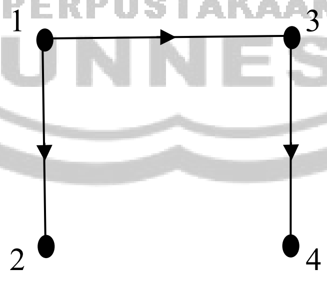
Contoh 1.1

Dipunyai sebuah graf G terhubung berarah, dengan 5 buah *chords* yaitu *chords* (1,2),(1,3),(2,3),(2,4) dan (3,4) pada graf G adalah seperti gambar 20 di bawah ini.



Gambar 20. Graf G

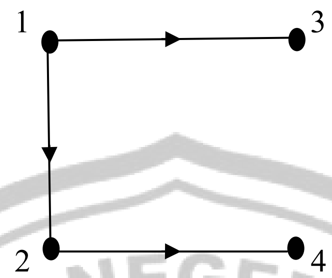
Di pilih titik 1 sebagai *root* maka dari graf G dapat dibentuk *spanning-tree* pertama (T_1) sebagai berikut:



Gambar 21. Graf T_1

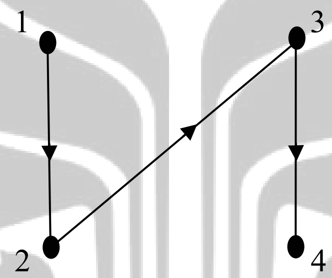
Pada graf T_1 (gambar 21) dipunyai 3 buah *branch* antara lain: *branch* (1,2), (1,3) dan (3,4). Sedangkan *spanning-tree* yang lain dapat dibuat dengan metode penukaran sisi yaitu:

- Dengan menambahkan *chord* (2,4) dan menghapus *branch* (3,4) dari T_1 maka didapat *spanning-tree* kedua adalah:



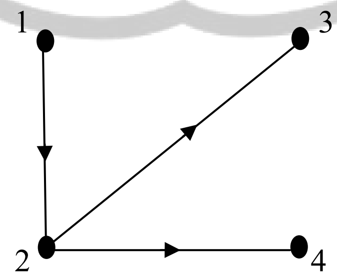
Gambar 22. Graf T_2

- Dengan menambahkan *chord* (2,3) dan menghapus *branch* (1,3) dari T_1 maka didapat *spanning-tree* ketiga adalah:



Gambar 23. Graf T_3

- Dengan menambahkan *chord* (2,3), (2,4) dan menghapus *branch* (1,3),(3,4) dari T_1 maka didapat *spanning-tree* keempat adalah:



Gambar 24. Graf T_4

Pada dasarnya metode penukaran sisi yang telah dikaji di atas, merupakan metode yang sangat sederhana. Akan tetapi metode tersebut tidak memungkinkan digunakan jika menentukan jumlah *spanning-tree* pada sebuah graf yang memiliki sisi dan titik dalam jumlah besar, maka jumlah dan bentuk *spanning-treenya* dapat dicari dengan menggunakan matriks *in-degree.*

2. Menggunakan Matriks *In-d*egree.

Pada Sub-bab ini akan dikaji penentuan jumlah *spanning-tree* pada graf berarah. *Spanning-tree* pada graf berarah dengan n buah titik sama dengan *spanning-tree* pada graf tak berarah, yang terdiri dari n-1 sisi berarah.

Menentukan banyaknya *spanning-tree* pada graf tak berarah sama halnya dengan menentukan banyaknya *spanning arborescence* pada graf berarah sederhana. Teorema-teorema berikut akan mengkaji bagaimana menentukan jumlah *spanning-tree* pada graf berarah.

Teorema 4

Pada *arborescence* terdapat sebuah lintasan berarah dari *root* menuju setiap titik yang lain. Sebaliknya jika G graf berarah sederhana tanpa siklus, G disebut *arborescence* jika terdapat sebuah titik v di G sedemikian hingga dapat di buat lintasan berarah dari v ke setiap titik yang lain, maka G *arborescence.*

Bukti:

(a) Andaikan tidak ada lintasan berarah dari *root* R ke sebuah titik v_i, artinya titik v_i berderajat masuk 0 (v_i *root*). Hal ini kontradiksi dengan G *arborescence*. Jadi terdapat lintasan berarah dari *root* R ke semua titik yang lain di G.

(b) Jelas G tidak memuat siklus maka G adalah pohon. Sehingga tinggal dibuktikan G memuat tepat satu *root*. Karena terdapat sebuah lintasan berarah dari v ke titik yang lain, maka derajat masuk v sama dengan 0, jadi v *root* dari G. Andaikan terdapat *root* yang lain di G sebut u maka yang terjadi, tidak bisa dibuat lintasan dari u ke v. Sehingga kontradiksi, jadi haruslah G memuat sebuah *root*.

Pada graf berarah diketahui adanya *in-degree* (derajat masuk) dan *out-degree* (derajat keluar), bergantung pada orientasi arah yang diberikan pada tiap sisinya. Akan tetapi pada permasalahan ini, lebih menitik beratkan pada derajat masuk suatu graf berarah.

Suatu Matriks *in-degree*, K(G) dari sebuah graf terhubung berarah G = (V,E) tanpa *loop* dengan V = $\{v_1, v_2, \ldots, v_n\}$ dan E=$\{e_1, e_2, \ldots, e_n\}$ adalah sebuah matriks dengan ukuran nxn yang mempunyai sifat :

$$K(G) = \begin{cases} d_{in}(vi), & \text{jika } i = j \\ -\ x_{ij}, & \text{jika } i \neq j \text{ maka } x_{ij} \text{ adalah entri dari matriks ketetanggaan, yang diberi tanda negatif.} \end{cases}$$

Apabila diadakan penghapusan terhadap sebuah baris dan kolom yang bersesuaian dengan *root* v_i, maka akan diperoleh submatrik K_{ii} dengan mengeluarkan baris-i dan kolom-i dari matriks K(G).

<u>Teorema 5</u>

G graf berarah sederhana dengan n buah titik dan n-1 sisi berarah adalah *arborescence* dengan v_1 sebagai *root*, jika dan hanya jika kofaktor k_{11} dari K(G) adalah 1.

Bukti:

a. Akan dibuktikan syarat perlu ($\dashrightarrow$)

Misal G adalah *arborescence* dengan n buah titik dan v_1 sebagai *root.* Beri label titik-titik di G dengan $v_1, v_2, v_3, \ldots, v_n$ sehingga titik-titik yang berada di sepanjang lintasan berarah yang dimulai dari v_i, diberi indeks naik.

Karena v_1 berderajat masuk sama dengan nol maka entri-entri pada kolom pertama sama dengan nol. Entri-entri pada kolom pertama sebagai berikut:

$k_{11} = 0$ (karena $d_{in}(v_1) = 0$)

$k_{21} = 0$ (karena tidak ada sisi berarah dari v_2 ke v_1)

$k_{31} = 0$ (karena tidak ada sisi berarah dari v_3 ke v_1)

$\vdots \qquad \vdots$

$k_{n1} = 0$ (karena tidak ada sisi berarah dari v_n ke v_1).

Sehingga semua entri pada kolom pertama dari matriks K(G) bernilai nol. Sedangkan entri yang lain dalam K(G) adalah

$$k_{ij} = 0, \qquad i > j$$

(karena semua titik pada lintasan berarah dari v_i ke v_j diberi label menaik, maka tidak ada sisi berarah dari v_i ke v_j dengan $i > j$ sehingga $k_{ij}=0$ untuk $i > j$).

$$k_{ij} = - x_{ij}, \qquad i < j$$

(karena semua titik pada lintasan berarah dari v_i ke v_j diberi label menaik, artinya: terdapat sisi berarah dari v_i ke v_j dengan $i < j$ sehingga $k_{ij} = - x_{ij}$, untuk $i < j$).

$$k_{ii} = 1, \qquad i > 1$$

Andaikan $k_{ii} > 1$, artinya: terdapat paling sedikit 2 sisi berarah yang menuju ke v_i. Misalkan sisi tersebut berasal dari titik v_{i-1} dan v_{i-2}. Sedangkan dari v_1 diketahui terdapat lintasan berarah dari v_1 ke v_{i-1} dan v_1 ke v_{i-2}. Sehingga kontradiksi dengan G *arborescence.*

Maka K matriks dari *arborescence* dengan *root* v_1 berlaku:

$$K(G) = \begin{bmatrix} 0 & -x_{12} & -x_{13} & \cdots & -x_{1n} \\ 0 & 1 & -x_{23} & \cdots & -x_{2n} \\ 0 & 0 & 1 & \cdots & -x_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Dengan menghapus baris pertama dan kolom pertama dari K(G) diperoleh submatriks K_{11} yang merupakan matriks segitiga atas. Berdasarkan teorema 2 diperoleh det $K_{11} = 1$ sehingga kofaktor k_{11} adalah 1.

b. Akan dibuktikan syarat cukup (◄- - - -)

Misal G adalah graf berarah sederhana dengan n buah titik dan n-1 sisi dengan kofaktor k_{11} dari matriks K(G) adalah 1, maka det $K_{11} = 1$. Karena det $K_{11} \neq 0$, maka setiap kolom dalam K_{11} paling sedikit memuat sebuah entri bukan nol.

Oleh karena itu,

$$d_{in}(v_i) \geq 1, \qquad \text{untuk } i = 2, 3, \ldots, n.$$

Tidak mungkin terjadi $d_{in}(v_i) > 1$. Andaikan $k_{ii} > 1$, artinya: terdapat paling sedikit 2 sisi berarah yang menuju ke v_i. Misalkan sisi tersebut berasal dari titik v_{i-1} dan v_{i-2}. Sedangkan dari v_1 diketahui terdapat lintasan berarah dari v_1 ke v_{i-1} dan v_1 ke v_{i-2}. Sehingga G memuat siklus dari v_1 ke v_i Hal ini kontradiksi dengan G hanya memuat n-1 sisi berarah.

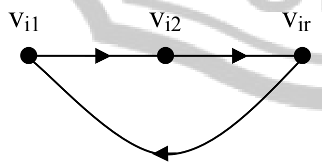
Sehingga

$$d_{in}(v_i) = 1, \qquad \text{untuk } i = 2, 3, \ldots, n.$$

dan

$$d_{in}(v_1) = 0.$$

Akan dibuktikan G tidak memuat siklus. Andaikan terdapat Siklus di G, yang melalui titik $v_{i_1}, v_{i_2}, \ldots, v_{i_r}$. Maka jumlah kolom $i_1, i_2, \ldots, i_r$ dalam K_{11} adalah nol.



$$\begin{array}{c|cccc} & v_{i1} & v_{i2} & \cdots & v_{ir} \\ \hline v_{i1} & 1 & -1 & \cdots & 0 \\ v_{i2} & 0 & 1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ v_{ir} & -1 & 0 & \cdots & 1 \end{array}$$

Gambar 25

sehingga r kolom pada K_{11} adalah bergantung linier, Akibatnya det K_{11} = 0. Hal ini sebuah kontradiksi dengan det K_{11} = 1, sehingga G tidak memuat siklus.

Karena G memuat n-1 sisi dan tidak memuat siklus maka G adalah pohon. Karena $d_{in}(v_1) = 0$ dan $d_{in}(v_i) = 1$ untuk i = 2, 3, ... , n, maka G haruslah *arborescence* dengan *root* v_1. **(terbukti).**

Teorema 6

Misal K(G) adalah matriks *in-degree* dari graf berarah sederhana. Maka nilai dari kofaktor k_{qq} dari K(G) adalah sama dengan banyaknya *arborescence* pada G dengan titik v_q sebagai *root.*

Bukti:

Misal K(G) adalah matriks *in-degree* dari graf berarah sederhana yang terdiri dari n buah vektor kolom untuk setiap matriks berordo n.

Sehingga

$$K(G) = [\ p_1, p_2, \dots , p_r, \dots , p_n]$$

Pilih vektor kolom p_r dari K(G) yang memuat entri $k_{ii}>1$, sehingga p_r dapat dibentuk menjadi $(p_i + p_i')$ dengan p_i dan p_i' adalah vektor kolom yang masing-masing memuat entri $k_{ii} = 1$.

Sehingga

$$K(G) = [\ p_1, p_2, \dots , (p_i + p_i'), \dots , p_n]$$

berdasarkan teorema 3, maka diperoleh

$$\det K(G) = \det [\ p_1, p_2, \dots , p_i, \dots , p_n] + \det [\ p_1, p_2, \dots , p_i', \dots , p_n].$$

Setiap vektor kolom p_j yang memuat entri $k_{ii} > 1$ dibentuk menjadi $(p_j + p_j')$ dengan p_j dan p_j' adalah vektor kolom yang memuat entri $k_{ii} = 1$. Sedangkan vektor kolom p_q, $q \neq j$ digunakan untuk membentuk kofaktor K_{qq}. Masing-masing kofaktor bersesuaian dengan sebuah subgraf g dari G yang memenuhi:

1) Setiap titik di g punya derajat masuk tepat satu, kecuali v_q.
2) g memuat n-1 titik dan n-1 sisi.

Jadi $K_{qq}\ (G) = \sum_{g} \det K_{qq}\ (g)$.

Dari teorema 5, diperoleh:

Det $K_{qq} = 1$, jika dan hanya jika g *arborescence* berakar di q

$= 0$ yang lain.

Jadi nilai dari kofaktor k_{qq} dari K(G) adalah sama dengan banyaknya *arborescence* pada G dengan titik v_q sebagai root. **(terbukti).**

Untuk lebih jelasnya lihat contoh 2.1 dibawah ini.

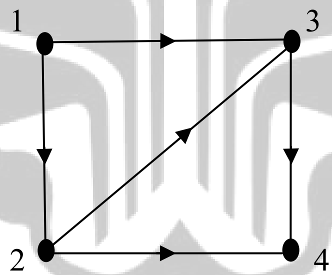
Contoh 2.1

Diketahui : Graf G = (V,E) adalah graf berarah sederhana dengan V = {1,2,3,4} dan E = { (1,2), (1,3), (2,3), (2,4), (3,4) }

Ditanya : banyaknya *arborescence* pada graf G.

Jawab :

Lihat graf berarah G pada gambar 26 di bawah ini, pilih titik 1 sebagai *root*.



Gambar 26. Graf G

Representasikan graf G di atas ke dalam bentuk matriks *in-degree*.

$$K(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \end{array}$$

Apabila diadakan penghapusan terhadap sebuah baris dan kolom yang bersesuaian dengan *root* v_1, maka akan di peroleh submatrik K_{11} dengan mengeluarkan baris-1 dan kolom-1 dari matriks K(G). Sehingga submatriks yang terbentuk sebagai berikut.

$$K_{11} = \begin{bmatrix} 1 & -1 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

Maka determinan dari K_{11} adalah:

$$|K_{11}| = \begin{vmatrix} 1 & -1 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{vmatrix}$$

$$= 1.2.2 + (-1).(-1).0 + (-1).0.0) - ((-1).2.0 + 1.(-1).0 + (-1).0.2)$$

$$= 4 - 0$$

$$= 4$$

Jadi determinan K_{11} adalah 4, sehingga kofaktor k_{11} dari K(G) adalah $(-1)^{1+1}.\det K_{11} = 1.4 = 4$.

Berdasarkan teorema 6, misal K(G) adalah matriks in-degree dari graf berarah sederhana. Maka nilai dari kofaktor k_{qq} dari K(G) adalah sama dengan banyaknya *arborescence* pada graf G dengan titik v_q sebagai *root.* Sehingga banyaknya *arborescence* yang diperoleh dari contoh 1 dengan v_1 sebagai *root* adalah 4.

Karena pada *arborescence* memuat sebuah lintasan berarah dari *root* ke setiap titik yang lain di G, maka dalam *arborescence* memuat semua titik di G. Sehingga *arborescence* juga merupakan *spanning-arborescence.* Jadi banyaknya *spanning arborescence* dari contoh 1 adalah 4.

Dengan berdasarkan pembuktian dari teorema 6, dari contoh 2.1 dapat diketahui bentuk dari setiap *arborescence* yang dihasilkan.

$$\det K(G) = \begin{array}{cc} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{vmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 2 \end{vmatrix} \end{array}$$

$$= \begin{array}{cc} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{vmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 2 \end{vmatrix} \end{array} + \begin{array}{cc} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{vmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 2 \end{vmatrix} \end{array}$$

$$= \begin{array}{cc} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{vmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \end{array} + \begin{array}{cc} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{vmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \end{array} +$$

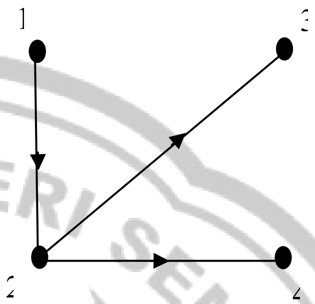
$$\begin{array}{cc} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{vmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \end{array} + \begin{array}{cc} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{vmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \end{array}$$

$$= 1 + 1 + 1 + 1$$

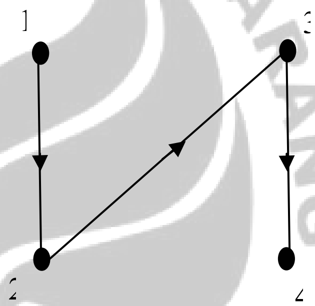
$$= 4$$

Jadi banyaknya *arborescence* yang terbentuk adalah jumlah determinan dari masing-masing matriks kofaktor K_{11} yaitu sama dengan 4. Berdasarkan perhitungan diatas diperoleh 4 buah matriks yang entri-entrinya dapat direpresentasikan menjadi *arborescence* sebagai berikut.

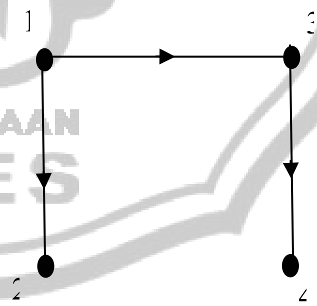
1)
$$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



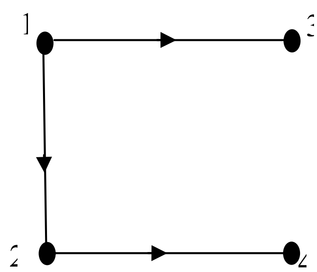
2)
$$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



3)
$$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



4)
$$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



BAB V
KESIMPULAN DAN SARAN

1. KESIMPULAN

Berdasarkan pembahasan, dapat disimpulkan:

1.1 Metode penukaran sisi (*edge exchange*) dapat digunakan untuk menemukan *spanning-tree* dari graf berarah sederhana G, dengan cara: membuat sebuah *spanning-tree* awal, kemudian menambahkan *chord* dan menghapus *branch* maka akan terbentuk *spanning-tree* baru. Untuk selanjutnya, dengan langkah yang sama maka akan terbentuk *spanning-tree* yang lain dengan berdasar pada *spanning-tree* awal.

1.2 Penentuan banyaknya *spanning-tree* pada graf berarah sederhana dapat dicari dengan menggunakan matriks *in-degree* (K(G)) dengan teorema: banyaknya *arborescence* pada graf G dengan *root* v_q sama dengan kofaktor k_{qq} dari K(G).

2. SARAN

Berkaitan dengan hasil penelitian, ada beberapa hal yang perlu mendapat perhatian yaitu penelitian ini hanya mengkaji *spanning-tree* pada *arborescence.* untuk itu perlu penelitian lebih lanjut untuk menentukan jumlah *spanning-tree* pada graf berarah yang bukan *arborescence.*

DAFTAR PUSTAKA

Anton, Howard. 2004. *Aljabar Liniear Elementer*. Jakarta: Erlangga.

Bambang, Sumantri. 1995. *Dasar-dasar Matenatika Diskrit.* Jakarta: PT Gramedia Pustaka Utama.

Budayasa K. 1997. *Matematika Diskrit 1*. Surabaya: IKIP Surabaya.

Even, S. 1979. Graph Algorithms. Computer Science Press. Tersedia di:

http://www.cs.technion.ac.il/~cs234141 [24 Februari 2009].

Hadley G. 1961. *Linier Algebra*. Adison-Wesley PublishingCo.

Mayeda, Wataru. 1972. *Graph Theory*. Wiley-Interscience: United State of America.

Munir, Rinaldi. 2001. *Matematika Diskrit*. Bandung:Informatika.

Munir, Rinaldi. 2003. *Matematika Diskrit*. Bandung: Informatika

Manohar R. 1975. *Discrete Mathematical Structure with Application to Computer Science.*

Narsingh Deo. 1997. *Graph Theory With Aplication to Engineering and Computer Science*. Prentice-Hall of India.

Siang, J. 2002. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer.* Yogyakarta: Andi.

Suryadi D, S.Harini Machmudi. 1996. *Teori dan Soal Pendahuluan Aljabar Linier*. ghalia Indonesia.

Sutarno, H. 2005. *Matematica Diskrit*. Malang: IKIP Malang.

Weisstein, Eric W. *Minimum Cost Arborescences.* Tersedia di: http://mathworld.wolfram.com/Arborescence.pdf [29Agustus 2009].