



UNIVERSITAS INDONESIA

**ALGORITMA PELABELAN TOTAL BUSUR AJAIB PADA
GRAF LINGKARAN, KIPAS, DAN RODA**

SKRIPSI

**ARUMELLA SURGANDINI
0606067282**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI SARJANA MATEMATIKA
DEPOK
JULI 2010**



UNIVERSITAS INDONESIA

**ALGORITMA PELABELAN TOTAL BUSUR AJAIB PADA
GRAF LINGKARAN, KIPAS, DAN RODA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana sains

**ARUMELLA SURGANDINI
0606067282**

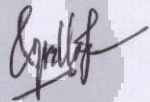
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI SARJANA MATEMATIKA
DEPOK
JULI 2010**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk
telah saya nyatakan dengan benar.

Nama : Arumella Surgandini

NPM : 0606067282

Tanda Tangan : 

Tanggal : 12 Juli 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Arumella Surgandini

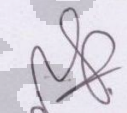
NPM : 0606067282

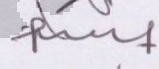
Program Studi : Sarjana Matematika

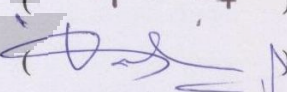
Judul Skripsi : Algoritma Pelabelan Total Busur Ajaib pada Graf
Lingkaran, Kipas, dan Roda

Telah berhasil dipertanyakan di hadapan Dewan Penguji dan diterima sebagai persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Matematika, Fakultas Ilmu Pengetahuan Alam Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dra. Denny Riama Silaban, M.Kom ()

Pembimbing : Dra. Siti Aminah, M.Kom ()

Penguji : Rahmi Rusin, S.Si., M.ScTech ()

Penguji : Dra. Rustina ()

Ditetapkan di : Depok

Tanggal : 12 Juli 2010

KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kepada Allah swt. atas semua rahmat dan karunia yang telah diberikan sehingga penulis dapat menyelesaikan skripsi ini. Penulis menyadari bahwa penyelesaian skripsi ini tidak terlepas dari bantuan dan bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

- (1) Dra. Denny Riama Silaban, M.Kom selaku pembimbing I, dan Dra. Siti Aminah, M.Kom selaku pembimbing II. Terima kasih yang teramat banyak untuk semua nasihat, bantuan, masukan dan dorongan semangat yang luar biasa yang telah diberikan kepada penulis untuk menyelesaikan tugas akhir ini.
- (2) Dr. Kiki A. Sugeng, Dhian Widya, S.Si, M.Kom, Dra. Yahma Wisnani, M.Kom, Prof. Dr. Djati Kerami, dan Drs. Suryadi MT, MT yang telah hadir dan memberikan saran serta masukan bagi penulis pada SIG 1, SIG 2, dan kolokium.
- (3) Dra. Suarsih Utama selaku pembimbing akademik penulis selama menyelesaikan masa studi. Terima kasih atas semua bantuan dan nasihatnya.
- (4) Dr. Yudi Satria, MT selaku Ketua Departemen, Rahmi Rusin, S.Si, M.ScTech selaku Sekretaris Departemen, dan Dr. Dian Lestari selaku koordinator pendidikan yang telah banyak membantu proses penyelesaian tugas akhir ini. Mila Novita, S.Si, M.Si selaku koordinator mahasiswa yang telah banyak membantu penulis selama kuliah dan dalam mengurus beasiswa.
- (5) Seluruh staf pengajar di Departemen Matematika UI atas ilmu pengetahuan yang telah kalian berikan.
- (6) Seluruh karyawan di Departemen Matematika UI atas bantuan yang telah diberikan.
- (7) Orang tua yang tercinta yang telah memberikan bantuan dukungan material dan moral.

- (8) Brilyan selaku adik yang juga telah memberikan semangat dan dukungan kepada penulis terutama selama penyusunan skripsi ini.
- (9) Tami, Alfa, dan Milla selaku teman-teman seperjuangan yang telah berjuang bersama dengan sangat kompak.
- (10) Farah dan Yuko. Terima kasih atas bantuannya dalam proses penyusunan skripsi.
- (11) Lena, Indah, Widya, Nadya, Annisa, Rita, Kiki, Lee, Yuri, Dian, Putri Helmet, Nurgi, Rahmanita, Ita, Yunita, Beki, Reza, Rifza, Lathief, Sevenhot, dan Indra selaku teman-teman seperjuangan dalam penyusunan skripsi. Terima kasih atas semangat dan dukungannya.
- (12) Tino, Teguh, dan teman-teman yang sudah lulus semester lalu (Rizqi, Inne, Anggha, Mei) serta teman-teman angkatan 2006 yang lainnya. Terima kasih atas semangat, dukungan, serta pengalaman selama perkuliahan yang tak terlupakan.
- (13) Kak Intan terima kasih atas saran, semangat, dan dukungan serta kesediaannya mendengarkan cerita tentang lika-liku perjuangan menyusun skripsi.
- (14) Kak Yanu dan Novi terima kasih atas kesediaannya menjawab pertanyaan-pertanyaan seputar pemrograman yang sangat membantu penulis.
- (15) Arif 07 terima kasih atas bantuannya sewaktu pendaftaran kolokium.
- (16) Kepada semua teman-teman di Matematika UI, khususnya angkatan 2007 dan 2008 terima kasih atas semangat dan dukungannya.

Kepada semua pihak yang telah membantu penulis dalam pengerjaan skripsi ini, yang namanya tidak bisa disebutkan satu-persatu, penulis ucapkan terima kasih. Akhir kata, penulis mohon maaf jika terdapat kesalahan atau kekurangan dalam skripsi ini. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Penulis
2010

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Arumella Surgandini
NPM : 0606067282
Program Studi : Sarjana Matematika
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

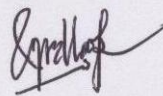
Algoritma Pelabelan Total Busur Ajaib pada Graf Lingkaran, Kipas, dan Roda beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 12 Juli 2010

Yang menyatakan



(Arumella Surgandini)

ABSTRAK

Nama : Arumella Surgandini
Program Studi : Matematika
Judul : Algoritma Pelabelan Total Busur Ajaib pada Graf Lingkaran,
Kipas, dan Roda

Misalkan G adalah graf dengan himpunan simpul $V = V(G)$ dan himpunan busur $E = E(G)$, dimana $|V(G)|$ dan $|E(G)|$ menyatakan banyaknya simpul dan busur pada G . Suatu pemetaan λ dari $V \cup E$ ke himpunan bilangan asli $\{1, 2, 3, \dots, |V(G)| + |E(G)|\}$ disebut pelabelan total busur ajaib jika λ merupakan pemetaan bijektif sedemikian sehingga $\forall xy \in E(G)$, bobot busur $\lambda(x) + \lambda(y) + \lambda(xy) = k$, untuk suatu konstanta k . Konstanta k disebut sebagai konstanta ajaib dari λ . Algoritma-algoritma pelabelan sembarang graf secara umum adalah bersifat *NP-complete*. Dalam skripsi ini akan dibangun algoritma pelabelan total busur ajaib pada graf lingkaran C_n , kipas f_n , dan roda W_n . Dengan menggunakan algoritma-algoritma tersebut dapat dihasilkan semua pelabelan total busur ajaib pada graf yang terkait (jika ada). Algoritma-algoritma ini kemudian diimplementasikan dalam bentuk program. Sebagai hasil implementasi dilakukan simulasi yang memberikan banyaknya pelabelan total busur ajaib yang mungkin dan berbeda dari graf lingkaran, kipas, dan roda untuk setiap nilai k yang mungkin. Simulasi banyaknya pelabelan total busur ajaib pada graf lingkaran dilakukan untuk $n \leq 12$, sedangkan pada graf kipas dan roda dilakukan untuk $n \leq 10$.

Kata Kunci : pelabelan total busur ajaib, graf lingkaran, graf kipas, graf roda
xv+78 halaman ; 65 gambar; 4 tabel
Daftar Pustaka : 6 (2001-2010)

ABSTRACT

Name : Arumella Surgandini
Study Program : Mathematics
Title : Edge Magic Total Labeling Algorithms on Cycle, Fan, and Wheel Graphs

Let G be a graph with the set of vertices $V = V(G)$ and the set of edges $E = E(G)$ where $|V(G)|$ and $|E(G)|$ are the number of vertices and edges of G respectively. A mapping λ from $V \cup E$ to the set $\{1, 2, 3, \dots, |V(G)| + |E(G)|\}$ is called an edge magic total labeling if λ is a bijection such that $\forall xy \in E(G)$ the weight of edge $\lambda(x) + \lambda(y) + \lambda(xy) = k$, for some constant k . A constant k is called the magic constant of λ . It is known that the labeling algorithms of any graph is NP-Complete. This *skripsi* gives explains edge magic total labeling algorithms on cycle C_n , fan f_n , and wheel W_n graphs. The algorithms generate all non isomorphic labelings on those graphs for all possible values of k . The implementation of the algorithms and simulation for some value of n ($n \leq 12$ for cycle, $n \leq 10$ for fan and wheel) are also given.

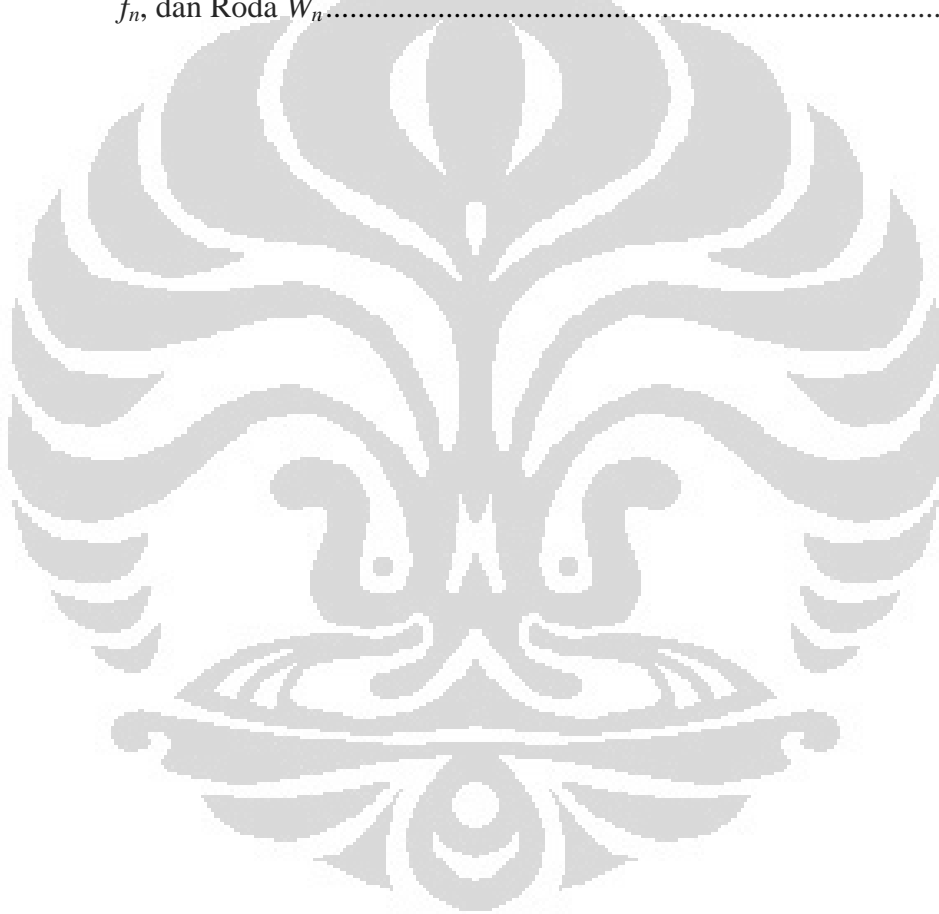
Keywords : Edge magic total labeling, cycle graph, fan graph, wheel graph
xv+78 pages ; 65 pictures; 4 tables
Bibliography : 6 (2001-2010)

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vii
ABSTRAK	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	3
1.3 Tujuan Penulisan	3
1.4 Pembatasan Masalah.....	3
1.5 Sistematika Penulisan.....	3
BAB 2 LANDASAN TEORI	4
2.1 Teori Graf	4
2.2 Jenis-jenis Graf.....	7
2.3 Pelabelan pada Graf.....	9
2.4 Hasil yang Telah Diketahui	15
BAB 3 ALGORITMA PTBA	17
3.1 Algoritma PTBA pada Graf Lingkaran C_n	18
3.2 Algoritma PTBA pada Graf Kipas f_n	30
3.3 Algoritma PTBA pada Graf Roda W_n	43
BAB 4 IMPLEMENTASI DAN SIMULASI	57
4.1 Implementasi dan Simulasi Algoritma PTBA pada Graf Lingkaran C_n	57
4.2 Implementasi dan Simulasi Algoritma PTBA pada Graf Kipas f_n	62
4.3 Implementasi Algoritma PTBA pada Graf Roda W_n	67
BAB 5 KESIMPULAN	73
DAFTAR PUSTAKA	75

DAFTAR TABEL

Tabel 4.1 Banyaknya PTBA yang berbeda pada graf Lingkaran C_3 sampai C_{12} berdasarkan kostanta ajaib k ($\frac{5n+3}{2} \leq k \leq \frac{7n+3}{2}$)	60
Tabel 4.2 Banyaknya PTBA yang berbeda pada graf Kipas f_3 sampai f_{10} berdasarkan kostanta ajaib k ($\frac{11n^2+7n-4}{4n-2} \leq k \leq \frac{25n^2-13n-2}{4n-2}$)	65
Tabel 4.3 Banyaknya PTBA yang berbeda pada graf Roda W_3 sampai W_{10} berdasarkan kostanta ajaib k ($\frac{11n+17}{4} \leq k \leq \frac{25n+7}{4}$)	70
Tabel 5.1 Ringkasan hasil simulasi algoritma PTBA pada graf lingkaran C_n , Kipas f_n , dan Roda W_n	73



DAFTAR GAMBAR

Gambar 2. 1	Contoh Graf G_1 dengan $ V =3$ dan $ E =5$	4
Gambar 2. 2	(a) Graf H_1 dan H_2 (b) Graf H_1 isomorfik terhadap graf H_2	6
Gambar 2. 3	Graf (a) dan (b) tidak isomorfik dikarenakan pemberian label yang berbeda	7
Gambar 2. 4	(a) Graf P_4 (b) Graf C_4	8
Gambar 2. 5	(a) Graf f_4 (b) Graf W_4	8
Gambar 2. 6	(a) Pelabelan simpul (b) Pelabelan busur (c) Pelabelan total pada graf C_5	9
Gambar 2. 7	PTBA pada graf C_6 dengan $k = 17$	10
Gambar 2. 8	(a) Pelabelan λ dan (b) pelabelan λ' pada C_5	14
Gambar 2. 9	Kasus rotasi pada C_5 dengan $k = 14$	14
Gambar 2. 10	Kasus refleksi pada C_4 dengan $k = 14$	15
Gambar 3. 1	Penamaan simpul dan busur pada graf C_6	19
Gambar 3. 2	Kasus rotasi pada C_4	19
Gambar 3. 3	Kasus refleksi pada C_4	20
Gambar 3. 4	Kasus refleksi pada graf C_4 (a) dan C_5 yang dapat dihindari dengan syarat $\lambda v_2 < \lambda(vn)$	21
Gambar 3. 5	Tahap pelabelan v_1 (a), v_2 (c) dan e_1 (c).....	22
Gambar 3. 6	Tahap mengganti label v_2 apabila label busur busur e_1 tidak tersedia ($\lambda(e_1) < 1$ atau $\lambda(e_1) > 2n$)	23
Gambar 3. 7	Tahap <i>backtracking</i> ke v_1 apabila label e_1 tidak tersedia	23
Gambar 3. 8	Tahap pelabelan v_n (a), e_{n-1} (b), dan e_n (c).....	24
Gambar 3. 9	Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ pada fungsi <i>initializeCycle</i> , melabel v_1 (a), v_2 (b), dan e_1 (c).....	26
Gambar 3. 10	Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi <i>initializeCycle</i> , mengganti label v_2 (a), melabel kembali e_1 (b), melabel v_3 dan e_2 (c)	26
Gambar 3. 11	Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi <i>extendCycle</i> , melabel v_4 , e_3 (a) dan melabel v_5 , e_4	27
Gambar 3. 12	Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi <i>extendCycle</i> , semua pelabelan yang mungkin untuk $\lambda(v_4) = 5$ (a) dan $\lambda(v_4) = 8$ (b)	28
Gambar 3. 13	Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi <i>extendCycle</i> , semua pelabelan yang mungkin untuk $\lambda(v_3) = 4$	28

Gambar 3. 14	Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi <code>extendCycle(4)</code> , melabel e_5 (a) dan PTBA yang diperoleh untuk C_5 dengan $k = 14$ (b).....	29
Gambar 3. 15	Graf kipas dengan 8 simpul (f_7).....	31
Gambar 3. 16	Kasus refleksi pada graf f_3	31
Gambar 3. 17	Kasus refleksi pada graf f_4 (a) dan f_5 (b).....	32
Gambar 3. 18	Kasus rotasi pada graf f_3	33
Gambar 3. 19	Tahap pelabelan c (a), v_1 (b), dan cv_1 (c).....	33
Gambar 3. 20	Tahap apabila label cv_1 tidak tersedia	34
Gambar 3. 21	Tahap pelabelan v_2 (a), e_1 (b), dan cv_2 (c).....	35
Gambar 3. 22	Tahap pelabelan v_n (a), e_{n-1} (b), dan cv_n (c).....	36
Gambar 3. 23	Contoh penerapan algoritma PTBA pada graf f_4 dengan $k = 20$ pada fungsi <code>initializeFan</code> , melabel c , v_1 (a) dan cv_1 (b).....	38
Gambar 3. 24	Contoh penerapan algoritma PTBA pada graf f_4 dengan $k = 20$ pada fungsi <code>extendFan</code> , melabel v_2 (a), e_1 (b) dan cv_2 (c)	39
Gambar 3. 25	Contoh penerapan algoritma PTBA pada graf f_4 dengan $k = 20$ pada fungsi <code>extendFan</code> , melabel v_3 , e_2 dan cv_3	40
Gambar 3. 26	Contoh penerapan algoritma PTBA pada graf f_4 dengan $k = 20$ pada fungsi <code>extendFan</code> melabel v_4 , e_3 (a), <i>backtracking</i> ke v_3 (b), melabel v_3 , e_2 , cv_3 , v_4 , e_3 (c), dan melabel cv_4 (d).....	41
Gambar 3. 27	PTBA yang berbeda pada graf f_4 dengan $k = 20$	42
Gambar 3. 28	Graf roda dengan 5 simpul (W_4).....	43
Gambar 3. 29	Kasus rotasi pada graf W_4	43
Gambar 3. 30	Kasus refleksi pada graf W_4	44
Gambar 3. 31	Tahap pelabelan c (a), v_1 (b), dan cv_1 (c).....	45
Gambar 3. 32	Tahap apabila label cv_1 tidak tersedia	46
Gambar 3. 33	Tahap pelabelan v_2 (a), e_1 (b), dan cv_2 (c).....	47
Gambar 3. 34	Tahap pelabelan v_n (a), e_{n-1} (b), cv_n (c), dan e_n (d)	48
Gambar 3. 35	Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi <code>initializeWheel</code> , melabel c , v_1 (a) dan cv_1 (b).....	51
Gambar 3. 36	Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi <code>extendWheel</code> , melabel v_2 , e_1 , cv_2 (a) dan v_3 , e_2 , cv_3 (b) 51	
Gambar 3. 37	Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi <code>extendWheel</code> , penggantian label v_3 (a) dan melabel v_3 , e_2 , cv_3 dengan label yang lainnya (b).....	52
Gambar 3. 38	Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi <code>extendWheel</code> , melabel v_4 , e_3 , cv_4 (a), penggantian label v_4 (b), melabel v_4 , e_3 , cv_4 (c), dan <i>backtracking</i> ke v_3 untuk mengganti label simpul tersebut (d)	53
Gambar 3. 39	Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi <code>extendWheel</code> , melabel v_3 , e_2 , cv_3 (a) dan v_4 , e_3 (b)	54

Gambar 3. 40	Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi <i>extendWheel</i> , <i>backtracking</i> ke v_3 untuk mengganti label simpul tersebut (a) dan melabel v_3, e_2, cv_3 (b)	54
Gambar 3. 41	Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi <i>extendWheel</i> , melabel v_4, e_3, cv_4 (a), penggantian label v_4 (b), melabel v_4, e_3, cv_4 (c), dan melabel e_4 (d).....	55
Gambar 3. 42	PTBA yang berbeda pada graf W_4 dengan $k = 16$	56
Gambar 4. 1	Tampilan pada <i>Command Window</i> saat fungsi <i>cycle</i> dipanggil....	58
Gambar 4. 2	Tampilan <i>Command Window</i> saat program dipanggil untuk mencari PTBA pada graf C_5 dengan $k = 14$	58
Gambar 4. 3	PTBA yang diperoleh untuk graf C_5 dengan $k = 14$ yang disimpan di berkas keluaran C-5-14.txt	59
Gambar 4. 4	PTBA pada C_3 dengan $k = 9$ (a), $k = 10$ (b), $k = 11$ (c),	59
Gambar 4. 5	Grafik hasil simulasi program algoritma PTBA pada graf lingkaran untuk $n=3,4,\dots,12$	62
Gambar 4. 6	Tampilan pada <i>Command Window</i> saat fungsi <i>fan</i> dipanggil	63
Gambar 4. 7	Tampilan <i>Command Window</i> saat program dipanggil untuk masalah PTBA pada graf f_4 dengan $k = 20$	63
Gambar 4. 8	PTBA yang diperoleh untuk graf f_4 dengan $k = 20$ yang disimpan di berkas keluaran f-4-20.txt.....	64
Gambar 4. 9	Grafik hasil simulasi program algoritma PTBA pada graf kipas untuk $n = 3, 4, \dots, 10$	67
Gambar 4. 10	Tampilan pada <i>Command Window</i> saat fungsi <i>wheel</i> dipanggil...	68
Gambar 4. 11	Tampilan <i>Command Window</i> saat program dipanggil untuk masalah PTBA pada graf W_4 dengan $k = 16$	68
Gambar 4. 12	PTBA yang diperoleh untuk graf W_4 dengan $k = 16$ yang disimpan di berkas keluaran W-4-16.txt	69
Gambar 4. 13	Grafik hasil simulasi program algoritma PTBA pada graf roda untuk $n = 3,4,\dots,10$	72

DAFTAR LAMPIRAN

Lampiran 1	PTBA yang berbeda pada graf f_3 dengan $12 \leq k \leq 18$	76
Lampiran 2	PTBA yang berbeda pada graf W_4 dengan $16 \leq k \leq 26$	77
Lampiran 3	Listing Program Algoritma PTBA pada Graf Lingkaran C_n	CD
Lampiran 4	Listing Program Algoritma PTBA pada Graf Kipas f_n	CD
Lampiran 5	Listing Program Algoritma PTBA pada Graf Roda W_n	CD



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teori graf adalah bagian dari matematika diskrit. Teori graf pertama kali diperkenalkan pada abad ke-18 oleh seorang matematikawan Swiss yang bernama Leonhard Euler. Ia menggunakan graf untuk menyelesaikan masalah jembatan Königsberg. Graf dapat digunakan untuk menyelesaikan masalah di beberapa bidang ilmu. Banyak permasalahan akan lebih mudah dimengerti dan diselesaikan apabila dapat direpresentasikan dalam bentuk graf.

Suatu graf $G(V(G),E(G))$ terdiri atas suatu himpunan tak-kosong dan berhingga $V(G)$ yang anggotanya disebut simpul, dan suatu himpunan berhingga $E(G)$ dengan anggota yang disebut busur, dimana busur menghubungkan dua simpul (boleh sama) dan simpul-simpul tersebut disebut titik-titik ujung (*endpoint*) dari busur. Banyak anggota pada himpunan simpul $V(G)$ atau V dinyatakan sebagai $|V|$. Banyaknya busur pada suatu graf G dinyatakan sebagai $|E|$. Graf G disebut graf berhingga apabila $|V|$ berhingga.

Terdapat jenis-jenis graf yang memiliki ciri tertentu sehingga graf tersebut diberi nama khusus, seperti graf lintasan, lingkaran, kipas, roda, dll. Graf lintasan (*path graph*), P_n , adalah graf yang mempunyai n simpul dengan busur $v_1v_2, v_2v_3, \dots, v_{n-1}v_n$. Graf lingkaran (*cycle graph*), C_n ($n \geq 3$) adalah graf yang diperoleh dari graf lintasan P_n dengan diberi tambahan busur v_nv_1 . Graf kipas (*fan graph*), f_n ($n \geq 2$) adalah suatu graf yang diperoleh dengan menghubungkan semua simpul dari graf lintasan P_n ke satu simpul lain yang disebut simpul pusat (*center*). Sedangkan graf roda (*wheel graph*), W_n ($n \geq 3$) adalah suatu graf yang diperoleh dari graf lingkaran C_n dengan menghubungkan semua simpul ke satu simpul lain yang disebut simpul pusat (*center*).

Suatu pelabelan dari graf $G(V,E)$ (*graph labeling*) adalah suatu pemetaan bijektif dari $V \cup E$ ke himpunan bilangan asli. Apabila daerah asal dari pemetaan hanya himpunan simpul, maka pelabelan disebut pelabelan simpul. Apabila daerah asal hanya himpunan busur, maka pelabelan disebut pelabelan busur. Apabila

daerah asal merupakan gabungan dari himpunan simpul dan busur, maka pelabelan disebut pelabelan total.

Dalam pelabelan total, jumlah dari semua label yang terkait dengan elemen graf disebut bobot. Bobot dapat dihitung untuk simpul dan busur. Penjumlahan dari label simpul dan label busur yang hadir pada simpul tersebut disebut bobot simpul. Penjumlahan dari label busur dan label simpul-simpul yang dihubungkan dengan busur tersebut disebut bobot busur. Dalam pelabelan dikenal pelabelan ajaib. Graf G dikatakan memiliki pelabelan ajaib jika bobot untuk setiap simpul dan/atau busur sama dengan suatu konstanta k . Konstanta k disebut konstanta ajaib.

Dalam skripsi ini hanya akan dibahas pelabelan total busur ajaib. Pelabelan total busur ajaib (PTBA) pada graf G adalah suatu pemetaan bijektif dari $V \cup E$ ke himpunan bilangan asli sedemikian sehingga bobot untuk setiap busurnya sama dengan k . Pada PTBA didefinisikan pelabelan dual, yaitu suatu PTBA yang dapat diperoleh dari PTBA yang sudah diketahui. Dual dari pelabelan dual adalah pelabelan awal yang diketahui.

Terdapat hasil-hasil penelitian yang telah dipublikasikan mengenai eksistensi dari PTBA. Hasil-hasil penelitian tersebut dapat dilihat di (Gallian, 2009) dan (Slamin et al., 2002). Jika diberikan suatu graf, batas k yang mungkin agar graf tersebut memiliki PTBA dapat diperoleh dengan perhitungan matematis. Pembuktian yang dilakukan dalam penelitian mengenai eksistensi dari pelabelan hanya untuk k tertentu saja dan bukan untuk semua k yang mungkin. Pendekatan lain yang dapat dilakukan untuk memperoleh semua pelabelan bagi setiap k yang mungkin adalah dengan menggunakan algoritma seperti pada Baker dan Sawada (2008). Mereka memberikan algoritma pelabelan total simpul ajaib (PTSA) untuk graf lingkaran dan roda. Dengan algoritma pelabelan tersebut dapat dicari pelabelan berbeda untuk setiap k yang mungkin (jika ada). Untuk mencari pelabelan yang berbeda perlu diperhatikan pelabelan yang isomorfik. Pelabelan isomorfik adalah pelabelan yang membuat graf berlabel isomorfik. Jika diobservasi, pelabelan isomorfik pada suatu graf merupakan hasil rotasi atau refleksi dari label graf tersebut. Oleh karena itu, pelabelan isomorfik harus dihindari agar diperoleh pelabelan yang berbeda.

Algoritma pelabelan sembarang graf secara umum adalah *NP-complete*. Oleh karena itu pembangunan algoritma pelabelan biasanya dilakukan untuk kelas graf tertentu. Menggunakan ide yang dilakukan oleh Baker dan Sawada dalam membangun algoritma PTSA untuk graf lingkaran, dapat dibangun algoritma PTBA untuk graf lingkaran. Kemudian ide tersebut akan dikembangkan untuk membangun algoritma-algoritma PTBA pada graf yang diperoleh dengan menghubungkan satu simpul lain ke setiap simpul dari graf lintasan atau lingkaran.

1.2 Permasalahan

Bagaimanakah cara membangun algoritma untuk memperoleh semua PTBA yang berbeda untuk kelas graf lingkaran dan kelas graf yang diperoleh dari graf lingkaran dan lintasan?

1.3 Tujuan Penulisan

Membangun algoritma PTBA pada kelas graf lingkaran dan kelas graf yang diperoleh dari graf lingkaran dan lintasan. Algoritma-algoritma ini akan diimplementasikan dalam bentuk program MATLAB. Sebagai hasil implementasi dilakukan simulasi yang memberikan banyaknya PTBA yang berbeda dari graf terkait untuk setiap nilai k yang mungkin dengan n tertentu.

1.4 Pembatasan Masalah

Dalam skripsi ini hanya dibatasi pada graf lingkaran, kipas, dan roda.

1.5 Sistematika Penulisan

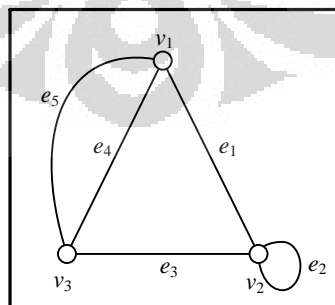
Skripsi ini terbagi dalam 5 bab. Pada Bab 2, diberikan definisi-definisi dan konsep dasar dalam teori graf dan pelabelan graf. Bab 3 diberikan algoritma pelabelan total busur ajaib pada graf lingkaran, kipas, dan roda. Bab 4 akan diberikan implementasi dan simulasi dari algoritma yang dibangun pada Bab 3. Terakhir, pada Bab 5 akan diberikan kesimpulan.

BAB 2 LANDASAN TEORI

Pada bab ini akan diberikan beberapa definisi dan konsep dasar dari teori graf dan pelabelan graf yang akan digunakan pada bab-bab selanjutnya. Pada umumnya definisi-definisi yang terdapat pada bab ini diambil dari buku *Introduction to Graph Theory* karangan Douglas B. West (West, 2001).

2.1 Teori Graf

Suatu graf $G(V(G), E(G))$ terdiri atas suatu himpunan simpul $V(G)$ yang tak-kosong dan berhingga dan suatu himpunan busur $E(G)$ yang berhingga, dimana busur menghubungkan dua simpul (boleh sama) dan simpul-simpul tersebut disebut **titik-titik ujung** (*endpoint*) dari busur. Himpunan busur dapat berupa himpunan kosong (disebut **graf kosong**). Untuk penyederhanaan, $G=(V(G), E(G))$ akan dinotasikan dengan G , $V(G)$ dengan V , dan $E(G)$ dengan E . Banyaknya simpul dan busur pada suatu graf masing-masing dinyatakan sebagai $|V|$ dan $|E|$. Apabila banyaknya simpul berhingga maka graf G disebut **graf berhingga**. Suatu graf G disebut **graf berarah** apabila G terdiri dari himpunan simpul V dan himpunan busur E , dimana setiap busur adalah pasangan terurut dari simpul. Jika setiap busur pada graf G bukan merupakan pasangan terurut maka graf G disebut **graf tak berarah**.



Gambar 2.1 Contoh Graf G_1 dengan $|V| = 3$ dan $|E| = 5$

Graf dapat direpresentasikan dalam bentuk gambar. Simpul pada graf digambarkan sebagai bulatan kecil sedangkan busur digambarkan sebagai garis yang menghubungkan sembarang dua simpul. Simpul biasanya dinyatakan dengan huruf kecil u, v, \dots atau v_1, v_2, \dots sedangkan busur dinyatakan dengan e_1, e_2, \dots atau dengan kedua ujungnya v_1v_2, v_1v_3, \dots . Pada Gambar 2.1 diberikan contoh graf dengan $V = \{v_1, v_2, v_3\}$ dan $E = \{e_1, e_2, e_3, e_4, e_5\}$.

Pada graf G , dua simpul dikatakan **bertetangga** (*adjacent*) apabila terdapat satu atau lebih busur yang menghubungkan kedua simpul tersebut. Suatu busur dikatakan **hadir** (*incident*) pada suatu simpul apabila simpul tersebut merupakan salah satu titik ujung dari busur tersebut. Busur yang memiliki titik ujung yang sama disebut **gelung** (*loop*). Dua busur disebut **busur ganda** (*multiple edges*) apabila kedua busur ini memiliki pasangan titik ujung yang sama. Sebagai contoh pada Gambar 2.1, v_1 dan v_2 bertetangga, e_1 hadir pada simpul v_1 dan v_2 , e_2 merupakan gelung, e_4 dan e_5 merupakan busur ganda. Suatu graf yang tidak memiliki gelung dan busur ganda disebut **graf sederhana** (*simple graph*).

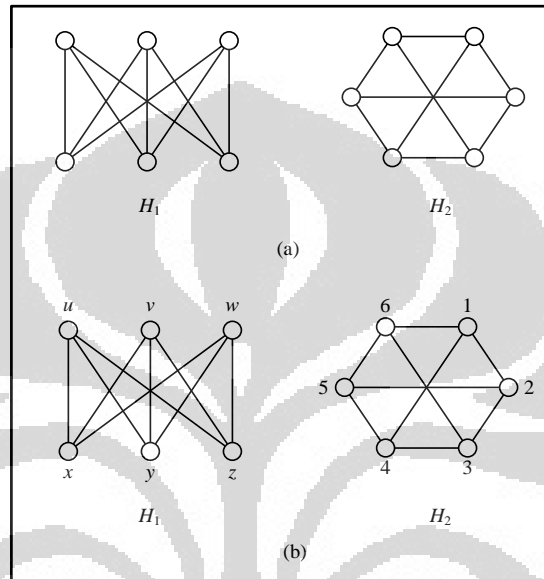
Derajat (*degree*) dari simpul v pada graf G , ditulis $d_G(v)$ atau $d(v)$, adalah banyaknya busur yang hadir pada simpul v , untuk gelung dihitung dua kali. Didefinisikan $\Delta(G)$ sebagai derajat maksimum dan $\delta(G)$ sebagai derajat minimum dari graf G . Suatu graf G disebut **graf teratur** (*regular graph*) jika $\Delta(G) = \delta(G)$. Graf G disebut **graf teratur berderajat r** (*r-regular*) jika $\Delta(G) = \delta(G) = r$. Simpul dengan $d(v) = 0$ disebut **simpul terpencil** (*isolated vertex*) dan simpul dengan $d(v) = 1$ disebut **simpul terminal** (*terminal vertex*). Pada Gambar 2.1, G_1 memiliki $d(v_1) = d(v_3) = 3$, $d(v_2) = 4$, sehingga G_1 bukan graf teratur karena $\Delta(G_1) = 4$, $\delta(G_1) = 3$.

Suatu **jalan** (*walk*) adalah sederetan $v_0, e_1, v_1, \dots, e_k, v_k$ dari simpul dan busur sedemikian sehingga untuk $1 \leq i \leq k$, busur e_i memiliki titik ujung v_{i-1} dan v_i .

Jalur (*trail*) adalah jalan dengan busur tidak berulang. Jalan- u, v atau jalur- u, v memiliki simpul pertama u dan simpul terakhir v , u dan v adalah titik ujung.

Lintasan- u, v (*u,v-path*) adalah jalur yang deretan simpul-simpulnya tidak pernah berulang, dimana simpul yang mempunyai derajat 1 pada lintasan adalah u dan v (titik ujungnya); simpul yang lain disebut **simpul dalam** (*internal vertex*). Panjang dari suatu jalan (jalur ataupun lintasan) adalah banyaknya busur pada jalan

(jalur ataupun lintasan) tersebut. Jalan (jalur ataupun lintasan) disebut **tertutup** jika titik ujungnya sama. Suatu graf tak berarah G disebut **graf terhubung** (*connected graph*) jika G memiliki suatu lintasan- u,v untuk setiap $u,v \in V(G)$ (kebalikannya, G disebut **graf tak terhubung**). Jika G memiliki suatu lintasan- u,v , maka u terhubung dengan v di G .



Gambar 2.2 (a) Graf H_1 dan H_2 (b) Graf H_1 isomorfik terhadap graf H_2

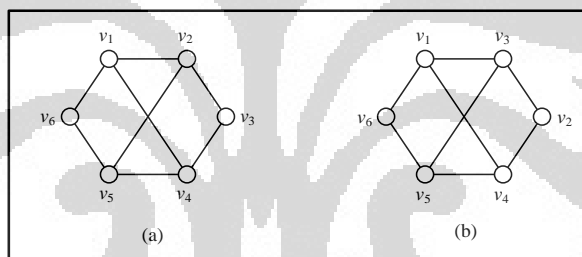
Dua graf dapat saling isomorfik walaupun bentuknya berbeda. Suatu isomorfisma dari graf G ke graf H adalah suatu pemetaan bijektif $f: V(G) \rightarrow V(H)$ sedemikian sehingga $uv \in E(G)$ jika dan hanya jika $f(u)f(v) \in E(H)$. Jika terdapat suatu isomorfisma dari G ke H maka disebut “ G isomorfik terhadap H ”, ditulis $G \cong H$. Isomorfisma graf adalah masalah *NP*.

Pada Gambar 2.2(a) diberikan dua graf terhubung yaitu, H_1 dan H_2 yang sama-sama memiliki $|V| = 6$ dan $|E| = 9$. Dua graf ini memiliki bentuk yang berbeda tetapi H_1 dan H_2 isomorfik. Untuk membuktikan $H_1 \cong H_2$, pertama berikan nama u, v, w, x, y, z untuk setiap simpul pada graf H_1 dan $1, 2, 3, 4, 5, 6$ untuk setiap simpul pada graf H_2 . Kemudian tentukan pemetaan bijektif $f: V(H_1) \rightarrow V(H_2)$ dan periksa apakah pemetaan bijektif tersebut mempertahankan hubungan kebertetanggaan (*adjacency relation*). Setelah diberi nama untuk setiap simpul seperti pada Gambar 2.2(b), ternyata pemetaan bijektif f dari H_1 ke H_2 yang me-

metakan u, v, w, x, y, z ke 1, 3, 5, 2, 4, 6 berturut-turut mempertahankan hubungan kebertetanggaan. Jadi H_1 isomorfik terhadap H_2 .

Pemberian label pada dua buah graf dapat mempengaruhi isomorfisma antara dua graf tersebut. Dua graf yang isomorfik apabila diberi label berbeda maka ada kemungkinan dua graf tersebut menjadi tidak isomorfik. Hal ini tergantung pada ada atau tidaknya pemetaan bijektif dari himpunan simpul salah satu graf berlabel tersebut ke himpunan simpul graf yang lainnya, yang mempertahankan hubungan kebertetanggaan. Pada Gambar 2.3 merupakan contoh pemberian label berbeda pada dua buah graf yang sama (a) dan (b) yang mengakibatkan dua graf tersebut tidak isomorfik.

Dalam skripsi ini akan digunakan graf berhingga, sederhana, tak berarah, dan terhubung. Selanjutnya akan dibahas jenis-jenis graf yang digunakan dalam skripsi ini.



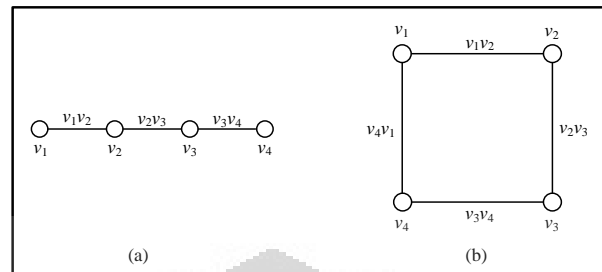
Gambar 2.3 Graf (a) dan (b) tidak isomorfik dikarenakan pemberian label yang berbeda

2.2 Jenis-jenis Graf

Graf lintasan (*path graph*), P_n , adalah graf yang mempunyai n simpul dengan busur $v_1v_2, v_2v_3, \dots, v_{n-1}v_n$. Sehingga $|V| = n$ dan $|E| = n - 1$. Simpul v_1 disebut sebagai simpul awal dan v_n adalah simpul akhir. Semua simpul berderajat 2 kecuali untuk simpul awal dan simpul akhir berderajat 1. Pada Gambar 2.4(a) diberikan graf lintasan dengan 4 simpul (P_4).

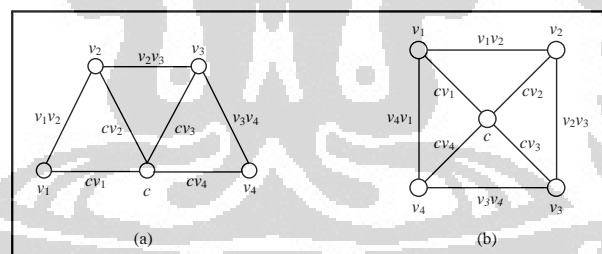
Graf lingkaran (*cycle graph*), C_n ($n \geq 3$) adalah graf yang diperoleh dari graf lintasan P_n dengan diberi tambahan busur antara simpul awal dan simpul akhir (busur v_nv_1). Jadi graf lingkaran adalah graf teratur karena semua simpulnya

memiliki derajat 2. Dalam graf lingkaran C_n berlaku $|V| = |E|$. Pada Gambar 2.4(b) diberikan graf lingkaran dengan 4 simpul (C_4).



Gambar 2.4 (a) Graf P_4 (b) Graf C_4

Graf kipas (*fan graph*), f_n ($n \geq 2$) adalah suatu graf yang diperoleh dengan menghubungkan semua simpul dari suatu graf lintasan P_n ke satu simpul lain yang disebut simpul pusat (*center*). Sehingga f_n terdiri dari $n + 1$ simpul yaitu c , v_1, v_2, \dots, v_n dan $2n-1$ busur yaitu $cv_1, cv_2, \dots, cv_n, v_1v_2, v_2v_3, \dots, v_{n-1}v_n$. Graf kipas f_4 yang diperoleh dengan menghubungkan semua simpul dari P_4 ke simpul pusat (c) terdapat pada Gambar 2.5(a).



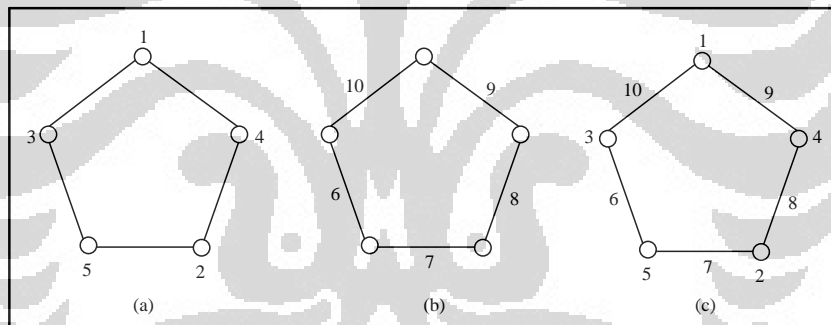
Gambar 2.5 (a) Graf f_4 (b) Graf W_4

Graf roda (*wheel graph*), W_n ($n \geq 3$) adalah suatu graf yang diperoleh dengan menghubungkan semua simpul dari suatu graf lingkaran C_n ke satu simpul lain yang disebut simpul pusat (*center*). Jadi W_n terdiri dari $n + 1$ simpul yaitu c , v_1, v_2, \dots, v_n dan $2n$ busur yaitu $cv_1, cv_2, \dots, cv_n, v_1v_2, v_2v_3, \dots, v_{n-1}v_n, v_nv_1$. Graf roda W_4 yang diperoleh dengan menghubungkan semua simpul dari C_4 ke simpul pusat (c) terdapat pada Gambar 2.5(b).

Jenis-jenis graf yang akan digunakan dalam skripsi ini adalah graf lingkaran, kipas, dan roda. Pada bagian selanjutnya akan dibahas definisi pelabelan yang digunakan dalam skripsi ini.

2.3 Pelabelan pada Graf

Pelabelan pada graf merupakan suatu pemetaan bijektif λ yang memetakan setiap elemen graf ke himpunan bilangan asli. Himpunan bilangan asli yang merupakan daerah hasil disebut **label**. Apabila hanya simpul yang dilabel, maka pelabelan disebut **pelabelan simpul**. Apabila hanya busur yang diberi label, maka pelabelan disebut **pelabelan busur**. Apabila simpul dan busur keduanya dilabel, maka pelabelan disebut **pelabelan total**. Pada Gambar 2.6 diberikan contoh pelabelan pada graf C_5 dengan pelabelan simpul (a), pelabelan busur (b), dan pelabelan total (c).



Gambar 2. 6 (a) Pelabelan simpul (b) Pelabelan busur (c) Pelabelan total pada graf C_5

Jumlah dari semua label yang terkait dengan elemen graf disebut **bobot**. Bobot dapat dihitung untuk simpul dan busur. Apabila λ adalah pelabelan total, **bobot simpul** x atas pelabelan λ ($w_\lambda(x)$) adalah jumlah dari label simpul dan label busur yang hadir pada simpul tersebut,

$$w_\lambda(x) = \lambda(x) + \sum_{y \in N(x)} \lambda(xy).$$

dimana $N(x)$ merupakan himpunan simpul yang bertetangga dengan simpul x . Sedangkan **bobot busur** xy atas pelabelan λ ($w_\lambda(xy)$) adalah jumlah dari label busur dan label simpul-simpul yang dihubungkan dengan busur tersebut,

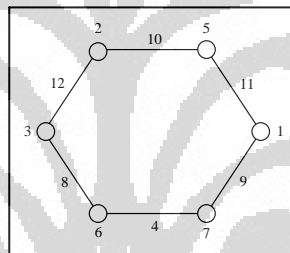
$$w_\lambda(xy) = \lambda(xy) + \lambda(x) + \lambda(y).$$

Dalam pelabelan diperkenalkan pelabelan ajaib. Graf G dikatakan memiliki **pelabelan ajaib** jika graf G dapat dilabel sehingga bobot untuk setiap simpul dan/atau busur sama dengan suatu konstanta k . Konstanta k disebut **konstanta ajaib**.

Dalam skripsi ini hanya akan dibahas pelabelan total busur ajaib. **Pelabelan total busur ajaib** (PTBA) pada graf G adalah suatu pemetaan bijektif λ dari $V \cup E$ ke himpunan bilangan asli $\{1, 2, \dots, |V| + |E|\}$ dengan sifat untuk setiap busur xy ,

$$w_\lambda(xy) = \lambda(x) + \lambda(y) + \lambda(xy) = k$$

untuk suatu konstanta k (Slamin et al., 2002). Suatu graf disebut **total busur ajaib** jika graf tersebut memiliki pelabelan total busur ajaib. Pada Gambar 2.7 diberikan contoh PTBA pada graf C_6 dengan $k = 17$.



Gambar 2.7 PTBA pada graf C_6 dengan $k = 17$

Jika diberikan suatu graf, batas k yang mungkin agar graf tersebut memiliki PTBA dapat diperoleh dengan perhitungan matematis. Misalkan $M = |V| + |E|$, S_v adalah jumlah dari semua label simpul, dan S_e adalah jumlah semua label busur. Oleh karena label merupakan bilangan asli dari $1, 2, \dots, M$, dapat diperoleh jumlah dari semua label pada graf C_n , yaitu

$$S_v + S_e = \sum_{i=1}^M i \quad (2.1)$$

Karena bobot setiap busur xy adalah $w_\lambda(xy) = \lambda(x) + \lambda(y) + \lambda(xy) = k$, jumlahkan semua bobot busur sehingga dapat diketahui bahwa label busur dihitung sekali dan label simpul dihitung dua kali

$$2S_v + S_e = |E|k. \quad (2.2)$$

Substitusi (2.1) ke (2.2)

$$S_v + \sum_{i=1}^M i = |E|k. \quad (2.3)$$

Karena S_v akan minimum jika seluruh simpul dilabel dengan v label terkecil dan akan maksimum jika seluruh simpul dilabel dengan v label terbesar maka

$$\sum_{i=1}^v i \leq S_v \leq \sum_{i=e+1}^M i \quad (2.4)$$

dari (2.3) dan (2.4) diperoleh

$$\begin{aligned} \sum_{i=1}^M i + \sum_{i=1}^v i &\leq |E|k \leq \sum_{i=1}^M i + \sum_{i=e+1}^M i \\ \sum_{i=1}^M i + \sum_{i=1}^v i &\leq |E|k \leq 2 \sum_{i=1}^M i - \sum_{i=1}^e i \end{aligned} \quad (2.5)$$

Banyaknya simpul C_n adalah $|V|=n$ dan banyaknya busur adalah $|E|=n$ dengan demikian $M = 2n$ sehingga dari persamaan (2.5) diperoleh batas k untuk C_n , yaitu

$$\begin{aligned} \sum_{i=1}^{2n} i + \sum_{i=1}^n i &\leq nk \leq 2 \sum_{i=1}^{2n} i - \sum_{i=1}^n i \\ \frac{2n(2n+1)}{2} + \frac{n(n+1)}{2} &\leq nk \leq 2 \frac{2n(2n+1)}{2} - \frac{n(n+1)}{2} \\ \frac{5n+3}{2} &\leq k \leq \frac{7n+3}{2}. \end{aligned} \quad (2.6)$$

Misalkan akan dicari batas konstanta ajaib (k) untuk graf C_4 . Substitusi $n = 4$ ke (2.6) sehingga diperoleh batas k yang mungkin adalah

$$\begin{aligned} \frac{5(4)+3}{2} &\leq k \leq \frac{7(4)+3}{2} \\ \frac{23}{2} &\leq k \leq \frac{31}{2} \\ 11.5 &\leq k \leq 15.5. \end{aligned}$$

Konstanta ajaib (k) untuk graf C_4 adalah semua bilangan asli yang ada pada interval $11.5 \leq k \leq 15.5$.

Graf kipas f_n memiliki simpul sebanyak $|V| = n + 1$ dan busur sebanyak $|E| = 2n - 1$. Dalam menghitung jumlah semua bobot busur pada graf kipas, label simpul v dihitung $d(v)$ kali dan label busur dihitung sekali sehingga

$$\begin{aligned} d(c)\lambda(c) + d(v_1)\lambda(v_2) + d(v_2)\lambda(v_2) + \dots + d(v_{n-1})\lambda(v_{n-1}) + \\ 2\lambda(v_n) + S_e &= |E|k. \\ n\lambda(c) + 2\lambda(v_1) + 3\lambda(v_2) + 3\lambda(v_3) + \dots + 3\lambda(v_{n-1}) + 2\lambda(v_n) + \\ S_e &= |E|k. \\ n\lambda(c) + 3\lambda(v_2) + 3\lambda(v_3) + \dots + 3\lambda(v_{n-1}) + 2\lambda(v_1) + 2\lambda(v_n) + \\ S_e &= |E|k. \end{aligned} \quad (2.7)$$

Batas bawah untuk k diperoleh dengan melabel semua simpul dengan $|V|$ label terkecil, yaitu $1, 2, \dots, n + 1$ dimana semakin besar derajat dari simpul maka semakin kecil labelnya sehingga

$$\begin{aligned}
n + 3 \sum_{i=2}^{n-1} i + 2 \sum_{i=n}^{n+1} i + \sum_{i=n+2}^{3n} i &\leq (2n-1)k \\
n + 3 \left(\frac{n^2-n-2}{2} \right) + 2(2n+1) + \frac{8n^2-2}{2} &\leq (2n-1)k \\
\frac{2n+3n^2-3n-6+8n+4+8n^2-2}{2} &\leq (2n-1)k \\
\frac{11n^2+7n-4}{2} &\leq (2n-1)k \\
\frac{11n^2+7n-4}{4n-2} &\leq k. \tag{2.8}
\end{aligned}$$

Batas atas untuk k bisa diperoleh dengan cara sebaliknya. Simpul-simpul dilabel $|V|$ label terbesar, yaitu $2n, 2n+1, \dots, 3n$ dimana semakin besar derajat dari simpul maka semakin besar labelnya sehingga

$$\begin{aligned}
3n^2 + 3 \sum_{i=2n+2}^{3n-1} i + 2 \sum_{i=2n}^{2n+1} i + \sum_{i=1}^{2n-1} i &\geq (2n-1)k \\
3n^2 + 3 \left(\frac{5n^2-9n-2}{2} \right) + 2(4n+1) + \frac{4n^2-2n}{2} &\geq (2n-1)k \\
\frac{6n^2+15n^2-27n-6+16n+4+4n^2-2n}{2} &\geq (2n-1)k \\
\frac{25n^2-13n-2}{2} &\geq (2n-1)k \\
\frac{25n^2-13n-2}{4n-2} &\geq k. \tag{2.9}
\end{aligned}$$

Jadi dari (2.8) dan (2.9) diperoleh batas k untuk f_n , yaitu

$$\frac{11n^2+7n-4}{4n-2} \leq k \leq \frac{25n^2-13n-2}{4n-2}. \tag{2.10}$$

Contohnya, batas k untuk graf f_4 adalah

$$\begin{aligned}
\frac{11(4^2) + 7(4) - 4}{4(4) - 2} &\leq k \leq \frac{25(4^2) - 13(4) - 2}{4(4) - 2} \\
\frac{200}{14} &\leq k \leq \frac{346}{14} \\
14.29 &\leq k \leq 24.71
\end{aligned}$$

Jadi konstanta ajaib (k) yang mungkin untuk graf f_4 adalah semua bilangan asli yang terdapat pada interval $14.29 \leq k \leq 24.71$, yaitu 15, 16, ..., 24.

Untuk menghitung batas k pada W_n , dilakukan hal yang sama saat menghitung batas k pada f_n . Banyaknya simpul W_n adalah $|V| = n+1$ dan banyaknya busur adalah $|E| = 2n$. Jumlah dari semua bobot busur pada graf W_n adalah

$$\begin{aligned}
d(c)\lambda(c) + d(v_1)\lambda(v_1) + d(v_2)\lambda(v_2) + \dots + d(v_{n-1})\lambda(v_{n-1}) + \\
d(v_n)\lambda(v_n) + S_e = |E|k
\end{aligned}$$

$$n \lambda(c) + 3 \sum_{i=1}^n \lambda(v_i) + S_e = |E|k. \quad (2.11)$$

Batas bawah untuk k diperoleh dengan melabel semua simpul dengan $|V|$ label terkecil, yaitu $1, 2, \dots, n+1$ dimana semakin besar derajat dari simpul maka semakin kecil labelnya sehingga

$$\begin{aligned} n + 3 \sum_{i=2}^{n+1} i + \sum_{i=n+2}^{3n+1} i &\leq 2nk \\ n + \frac{3n^2+9n}{2} + 4n^2 + 3n &\leq 2nk \\ \frac{11n^2+17n}{2} &\leq 2nk \\ \frac{11n+17}{4} &\leq k. \end{aligned} \quad (2.12)$$

Batas atas untuk k bisa diperoleh dengan cara sebaliknya. Simpul-simpul dilabel $|V|$ label terbesar, yaitu $2n+1, 2n+2, \dots, 3n+1$ dimana semakin besar derajat dari simpul maka semakin besar labelnya sehingga

$$\begin{aligned} n(3n+1) + 3 \sum_{i=2n+1}^{3n} i + \sum_{i=1}^{2n} i &\geq 2nk \\ 3n^2 + n + \frac{15n^2+3n}{2} + \frac{4n^2+2n}{2} &\geq 2nk \\ \frac{25n^2+7n}{2} &\geq 2nk \\ \frac{25n+7}{4} &\geq k. \end{aligned} \quad (2.13)$$

Jadi dari (2.12) dan (2.13) diperoleh batas k untuk W_n adalah

$$\frac{11n+17}{4} \leq k \leq \frac{25n+7}{4}. \quad (2.14)$$

Sebagai contoh, akan dicari batas k untuk W_4 . Substitusi $n=4$ ke (2.14) sehingga diperoleh batas k yang mungkin adalah

$$\begin{aligned} \frac{11(4)+17}{4} &\leq k \leq \frac{25(4)+7}{4} \\ \frac{61}{4} &\leq k \leq \frac{107}{4} \\ 15.25 &\leq k \leq 26.75 \end{aligned}$$

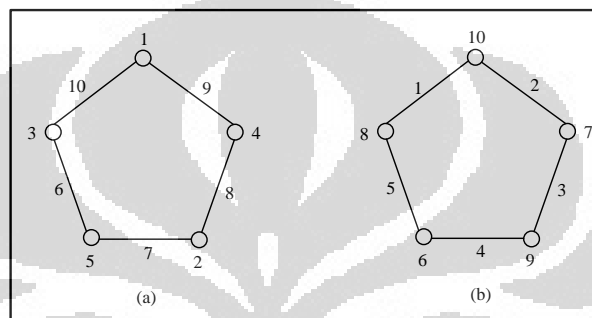
Jadi konstanta ajaib (k) yang mungkin untuk graf W_4 adalah semua bilangan asli yang terdapat pada interval $15.25 \leq k \leq 26.75$, yaitu $16, 17, \dots, 26$. Batas-batas k yang diperoleh di atas, nanti akan digunakan untuk menghitung batas k dari kelas graf yang digunakan.

Pada pelabelan total busur ajaib didefinisikan **pelabelan dual**. Misalkan pelabelan $\lambda : V \cup E \rightarrow \{1, 2, \dots, M\}$ merupakan pelabelan total busur ajaib pada graf G . Definisikan pelabelan $\lambda' : V \cup E \rightarrow \{1, 2, \dots, M\}$ dengan

$$\lambda'(x) : M + 1 - \lambda(x), \forall x \in V$$

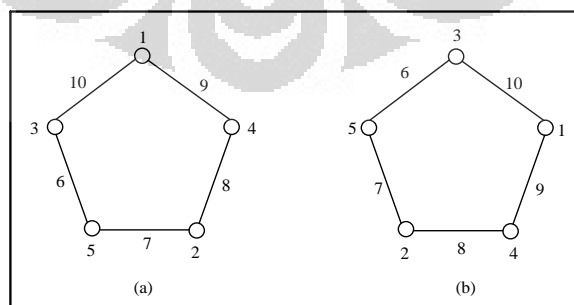
$$\lambda'(xy) : M + 1 - \lambda(xy), \forall xy \in E$$

maka pelabelan λ' disebut **dual** dari λ (MacDougall et al., 2002). Gambar 2.8 merupakan contoh PTBA pada C_5 dengan $k = 14$ (a) dan dualnya dengan $k = 19$ (b).



Gambar 2.8 (a) Pelabelan λ dan (b) pelabelan λ' pada graf C_5

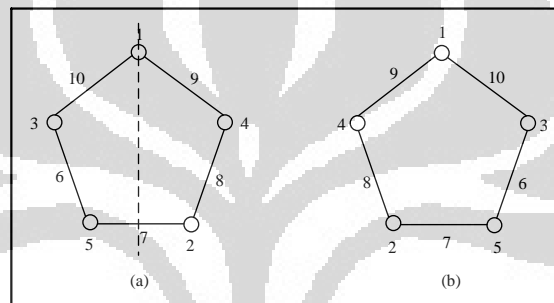
Untuk mencari banyaknya pelabelan yang berbeda perlu diberikan definisi pelabelan yang isomorfik. Misalkan $G = (V, E)$ dan $H = (W, F)$ adalah graf dengan pelabelan simpul $l : V \rightarrow L$, $l' : W \rightarrow L$ dimana L adalah himpunan label. Graf G dan H adalah graf berlabel yang isomorfik jika terdapat fungsi bijektif $f : V \rightarrow W$ sedemikian sehingga $\forall v, w \in V$, $vw \in E$ jika dan hanya jika $f(v)f(w) \in F$ dan $\forall v \in V$, $l(v) = l'(f(v))$ (Graph Isomorphism).



Gambar 2.9 Kasus rotasi pada graf C_5

Jika diobservasi, untuk memperoleh pelabelan yang isomorfik terhadap pelabelan λ dapat diperoleh dengan melakukan rotasi atau refleksi terhadap pelabelan λ . Kasus dimana suatu pelabelan diperoleh dengan melakukan rotasi terhadap pelabelan yang ada disebut **kasus rotasi** (*rotational symmetry*). Pada Gambar 2.9 diberikan kasus rotasi dimana pelabelan pada graf (b) diperoleh dengan merotasi label pada graf (a).

Kasus dimana suatu pelabelan diperoleh dengan melakukan refleksi terhadap pelabelan yang ada disebut **kasus refleksi** (*reflective symmetry*). Pelabelan dari hasil rotasi atau refleksi adalah sama dengan pelabelan awal.



Gambar 2. 10 Kasus refleksi pada graf C_5

Pada Gambar 2.10 diberikan contoh kasus refleksi pada graf C_5 dimana garis putus-putus merupakan sumbu refleksi. Pelabelan pada graf (b) diperoleh dengan merefleksikan label pada graf (a).

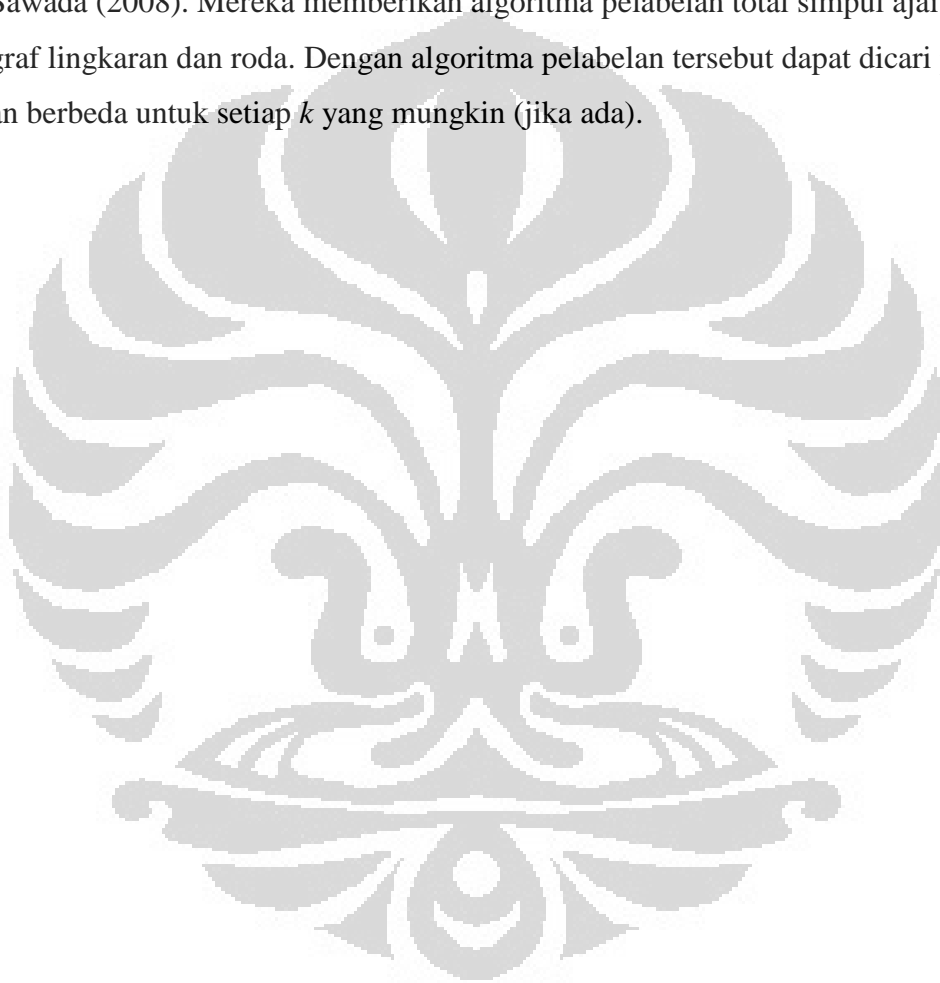
Seperti yang diperlihatkan pada Gambar 2.9 dan 2.10, terdapat kasus rotasi dan refleksi pada graf lingkaran C_n . Kedua kasus ini juga terdapat pada graf kipas dan roda. Namun, kasus rotasi pada graf kipas f_n hanya terdapat untuk $n \leq 3$.

2.4 Hasil yang Telah Diketahui

Terdapat hasil-hasil penelitian yang telah dipublikasikan mengenai eksistensi dari PTBA. Menurut Galian (2009) Kotzig dan Rosa membuktikan bahwa C_n memiliki PTBA untuk $n \geq 3$, Enomoto dkk. membuktikan bahwa W_n tidak memiliki PTBA untuk $n \equiv 3 \pmod{4}$, dan Phillips dkk. membuktikan bahwa W_n memiliki PTBA untuk $n \equiv 0,1 \pmod{4}$. Slamini dkk (2002) membuktikan bahwa setiap graf f_n memiliki PTBA dengan $k = 4n+2$ untuk $n \geq 2$ dan W_n memiliki PTBA

dengan $k = 5n + 2$ untuk $n \equiv 6 \pmod{8}$. Hasil-hasil lain untuk PTBA dapat dilihat di (Gallian, 2009).

Dalam penelitian mengenai eksistensi dari pelabelan, pembuktian yang dilakukan hanya untuk k tertentu saja dan bukan untuk semua k yang mungkin. Semua nilai k yang mungkin dapat diperoleh dengan perhitungan matematis. Pendekatan yang dapat dilakukan untuk memperoleh semua pelabelan bagi setiap k yang mungkin adalah dengan menggunakan algoritma seperti pada Baker dan Sawada (2008). Mereka memberikan algoritma pelabelan total simpul ajaib untuk graf lingkaran dan roda. Dengan algoritma pelabelan tersebut dapat dicari pelabelan berbeda untuk setiap k yang mungkin (jika ada).



BAB 3 ALGORITMA PTBA

Algoritma pelabelan sembarang graf secara umum adalah *NP-complete*. Oleh karena itu pembangunan algoritma pelabelan biasanya dilakukan untuk kelas graf tertentu. Seperti yang dilakukan oleh Baker dan Sawada (2008). Mereka membangun algoritma pelabelan total simpul ajaib (PTSA) untuk graf lingkaran dan roda. Dengan algoritma tersebut dapat dicari PTSA yang mungkin dan berbeda untuk graf terkait (jika ada).

Menggunakan ide yang dilakukan oleh Baker dan Sawada dalam membangun algoritma PTSA untuk graf lingkaran, dapat dibangun algoritma PTBA untuk graf lingkaran. Kemudian ide tersebut dapat dikembangkan untuk membangun algoritma-algoritma PTBA pada graf yang diperoleh dengan menghubungkan satu simpul lain ke setiap simpul dari graf lintasan atau lingkaran. Seperti graf kipas dan graf roda.

Ide dari algoritma PTBA secara umum adalah melabel simpul dan busur dari suatu graf secara iteratif yang menjamin setiap busur memiliki bobot yang sama. Algoritma PTBA akan menghasilkan semua PTBA yang mungkin yang berbeda (tidak isomorfik). Pelabelan isomorfik adalah pelabelan yang membuat graf berlabel isomorfik. Jika di observasi, pelabelan isomorfik pada suatu graf merupakan hasil refleksi atau rotasi dari label graf tersebut seperti yang telah dijelaskan pada Subbab 2.3.

Masukan dari algoritma PTBA adalah nilai dari n dan k . Algoritma PTBA akan melabel semua simpul dan busur dari suatu graf secara iteratif dengan label yang merupakan bilangan asli $1, 2, \dots, |V| + |E|$. Setiap label digunakan tepat satu kali. Algoritma ini menjamin bobot setiap busur sama dengan k . Jika $\lambda(x)$ dan $\lambda(xy)$ masing-masing menyatakan label dari simpul x dan busur xy maka beberapa kondisi yang harus dipenuhi adalah $0 < \lambda(x) \leq |V| + |E|$, $0 < \lambda(xy) \leq |V| + |E|$, dan $w_\lambda(xy) = \lambda(x) + \lambda(y) + \lambda(xy) = k$. Label simpul akan diberikan sesuai dengan label yang tersedia, sedangkan label busur ditentukan dengan pengurangan konstanta ajaib k dengan jumlah label dari simpul-simpul yang dihubungkan oleh busur tersebut. Jika label busur yang diperoleh tidak tersedia, maka label dari simpul yang

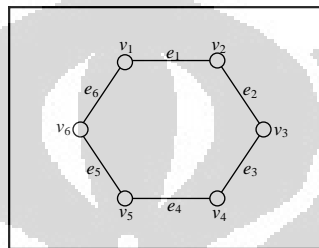
terakhir dilabel akan diganti labelnya dengan label lain yang tersedia, sampai label busur diperoleh. Jika label busur tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label simpul yang dilabel sebelumnya. Hal ini dilakukan hingga diperoleh label busur yang sesuai dengan label yang tersedia. Jika setelah dilakukan *backtracking* tetap tidak diperoleh label busur yang sesuai dengan label yang tersedia, maka graf tersebut untuk n dan k yang diberikan tidak memiliki PTBA. Agar diperoleh PTBA yang berbeda (tidak isomorfik) diberi syarat-syarat tertentu untuk menghindari pelabelan isomorfik. Selain mereduksi banyaknya pelabelan yang dicari, syarat ini juga membantu mengurangi komputasi yang dilakukan. Apabila sudah diperoleh satu PTBA maka label untuk setiap elemen graf yang diperoleh dicetak, kemudian proses akan berlanjut secara *backtracking* untuk memperoleh PTBA yang lain. Sehingga keluaran algoritma PTBA adalah semua PTBA yang berbeda pada suatu graf dengan nilai n dan k tertentu (jika ada).

Dalam Bab ini akan diberikan algoritma PTBA pada beberapa kelas graf. Algoritma PTBA dibangun dengan mengembangkan ide dari algoritma PTSA yang diberikan oleh Baker dan Sawada (2008). Pada Subbab 3.1 diberikan algoritma PTBA untuk graf lingkaran C_n , Subbab 3.2 untuk graf kipas f_n , dan Subbab 3.3 untuk graf roda W_n .

3.1 Algoritma PTBA pada Graf Lingkaran C_n

Baker dan Sawada (2008) telah membuat suatu algoritma untuk membangun semua pelabelan total simpul ajaib (PTSA) pada graf lingkaran C_n . Dari sembarang pelabelan total simpul ajaib untuk lingkaran selalu dapat dibangun suatu pelabelan total busur ajaib dengan menggeser label simpul menjadi label busur. Dengan pemahaman ini, dapat diperoleh algoritma PTBA pada graf lingkaran dari algoritma PTSA pada graf lingkaran yang diberikan oleh Baker dan Sawada. Namun, algoritma PTBA untuk graf lingkaran yang diberikan pada Subbab ini dibangun dengan pendekatan yang berbeda yang merupakan pengembangan ide dari apa yang dilakukan oleh Baker dan Sawada dalam membangun algoritma PTSA untuk graf lingkaran.

Pada Subbab 2.2 telah diberikan definisi graf lingkaran C_n , yaitu suatu graf yang diperoleh dari graf lintasan P_n dengan diberi tambahan busur antara simpul awal dan simpul akhir (busur v_nv_1). Banyaknya simpul dan busur yang dimiliki graf lingkaran C_n sama ($|V| = |E| = n$). Pada Gambar 3.1 diberikan graf lingkaran dengan 6 simpul (C_6). Untuk mempermudah pembuatan algoritma PTBA pada graf lingkaran C_n penamaan simpul dan busur diberikan seperti pada Gambar 3.1 tersebut.

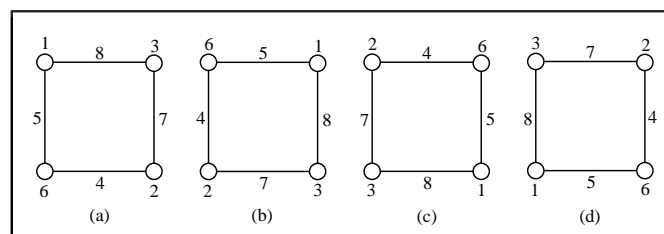


Gambar 3.1 Penamaan simpul dan busur pada graf C_6

Batas konstanta ajaib (k) yang mungkin untuk graf lingkaran C_n dapat diperoleh dengan perhitungan matematis. Seperti yang telah dijelaskan pada Subbab 2.3, batas k yang mungkin untuk C_n adalah

$$\frac{5n+3}{2} \leq k \leq \frac{7n+3}{2}$$

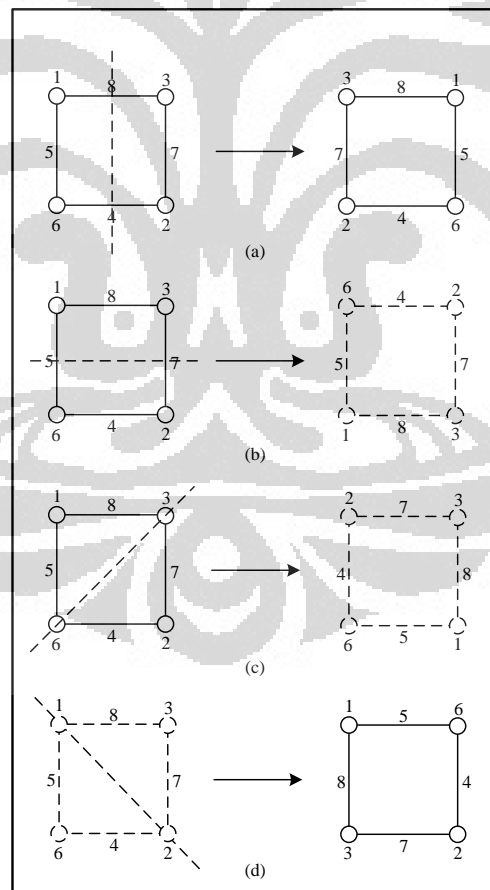
Sebagai contoh nilai k yang mungkin untuk PTBA dari C_4 adalah semua bilangan asli yang terdapat pada interval $11.5 \leq k \leq 15.5$. Banyaknya simpul pada graf lingkaran dan konstanta ajaib, yaitu n dan k merupakan masukan dari algoritma PTBA pada graf lingkaran sehingga harus ditentukan terlebih dahulu sebelum menjalankan algoritma tersebut.



Gambar 3.2 Kasus rotasi pada graf C_4 dengan $k = 12$

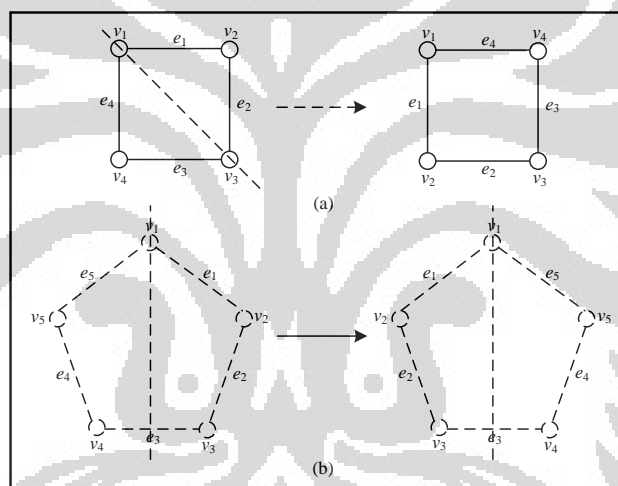
Untuk memperoleh PTBA yang berbeda pada graf lingkaran C_n perlu dihindari PTBA yang isomorfik. Seperti yang telah dijelaskan pada subbab 2.3, PTBA yang isomorfik pada graf lingkaran dapat diperoleh dengan rotasi dan refleksi terhadap pelabelan yang ada.

Pada Gambar 3.2 diberikan contoh kasus rotasi pada C_4 dengan $k = 12$ dimana simpul dan busur diberi label sebagai berikut: $\lambda(v_1) = 1, \lambda(v_2) = 3, \lambda(v_3) = 2, \lambda(v_4) = 6, \lambda(e_1) = 8, \lambda(e_2) = 7, \lambda(e_3) = 4, \text{ dan } \lambda(e_4) = 5$. Pelabelan pada Gambar 3.2 (b), (c), (d) diperoleh dengan merotasi pelabelan pada Gambar 3.2 (a). Jadi keempat pelabelan tersebut sama. Untuk menghindari kasus ini, dapat diberi syarat label salah satu simpul harus lebih kecil dari label simpul lainnya. Misalkan pilih simpul v_1 yang memiliki label terkecil ($\lambda(v_1) < \lambda(v_t), t = 2, 3, \dots, n$), maka label terbesar untuk v_1 adalah $n + 1$, yaitu ketika semua busur diberi label $1, 2, \dots, n$.



Gambar 3.3 Kasus refleksi pada graf C_4 dengan $k = 12$

Pada Gambar 3.3 diberikan contoh kasus refleksi pada C_4 dengan $k = 12$ dimana simpul dan busur diberi label sebagai berikut: $\lambda(v_1) = 1$, $\lambda(v_2) = 3$, $\lambda(v_3) = 2$, $\lambda(v_4) = 6$, $\lambda(e_1) = 8$, $\lambda(e_2) = 7$, $\lambda(e_3) = 4$, $\lambda(e_4) = 5$. Pada Gambar tersebut, pelabelan di sebelah kanan merupakan hasil refleksi pelabelan di sebelah kiri terhadap sumbu refleksi yang berupa garis putus-putus. Sumbu refleksi dari kasus refleksi pada Gambar 3.3 (a), (b), (c) tidak melalui simpul v_1 ,, sehingga adanya syarat bahwa label simpul v_1 lebih kecil dari label simpul lainnya yang diberikan untuk menghindari kasus rotasi, juga telah menghindari kasus ini. Sedangkan sumbu refleksi dari kasus refleksi pada Gambar 3.3 (d) melalui simpul v_1 sehingga perlu diberikan syarat yang baru. Kasus tersebut dapat dihindari dengan memberi syarat label simpul v_2 lebih kecil dari label simpul v_4 ($\lambda(v_2) < \lambda(v_4)$).

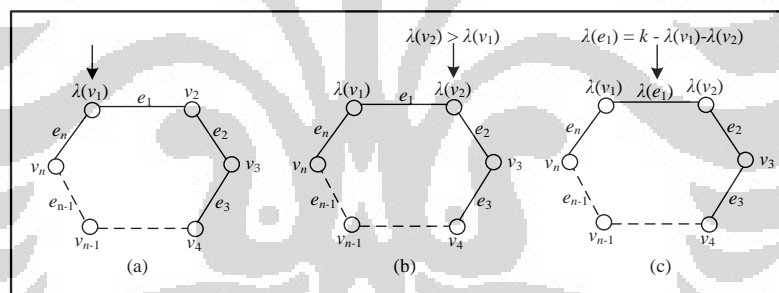


Gambar 3. 4 Kasus refleksi pada graf C_4 (a) dan C_5 yang dapat dihindari dengan syarat $\lambda(v_2) < \lambda(v_n)$

Untuk graf lingkaran dengan ukuran lainnya juga terjadi kasus refleksi seperti pada graf lingkaran dengan $n = 4$. Kasus refleksi dengan sumbu refleksi yang tidak melalui simpul v_1 dapat dihindari dengan diberinya syarat untuk kasus rotasi, yaitu label simpul v_1 lebih kecil dari label simpul lainnya. Sedangkan kasus refleksi dengan sumbu refleksi yang melalui simpul v_1 secara umum dapat dihindari dengan menyaratkan label salah satu simpul di salah satu sisi dari sumbu refleksi harus lebih kecil (atau lebih besar) dari label salah satu simpul pada sisi yang lain. Sehingga diberi syarat $\lambda(v_2) < \lambda(v_n)$. Pada Gambar 3.4 diberikan kasus refleksi pada graf lingkaran C_n dengan n genap yaitu C_4 (Gambar 3.4(a)) dan n ganjil yaitu

C_5 (Gambar 3.4(b)) terhadap sumbu refleksi yang berupa garis putus-putus yang melalui simpul v_1 dimana kasus ini dapat dihindari dengan syarat $\lambda(v_2) < \lambda(v_n)$.

Seperti yang telah dijelaskan sebelumnya, tujuan pembentukan algoritma PTBA pada graf lingkaran C_n adalah untuk mengetahui ada atau tidaknya PTBA pada suatu graf dengan nilai n dan k tertentu. Jika ada maka dapat dihasilkan semua PTBA yang berbeda pada suatu graf tersebut. Jadi masukan dari algoritma PTBA pada graf lingkaran adalah nilai n dan k . Label yang tersedia adalah 1 sampai $2n$. Kondisi yang harus dipenuhi oleh pelabelan yang dihasilkan adalah $0 < \lambda(x) \leq 2n$, $0 < \lambda(xy) \leq 2n$, $w_\lambda(xy) = \lambda(x) + \lambda(y) + \lambda(xy) = k$, dan label yang telah digunakan untuk melabel simpul atau busur tidak boleh digunakan lagi untuk melabel simpul atau busur yang lain. Selain itu syarat yang harus dipenuhi untuk menghindari pelabelan isomorfik juga harus dipenuhi, yaitu $\lambda(v_1) < \lambda(v_t)$, $t = 2, 3, \dots, n$ untuk menghindari kasus rotasi dan $\lambda(v_2) < \lambda(v_n)$ untuk menghindari kasus refleksi.

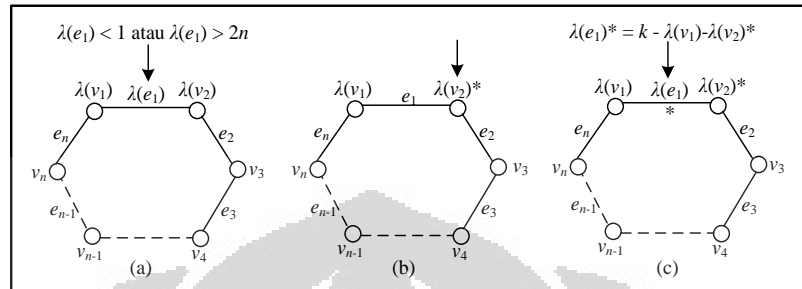


Gambar 3.5 Tahap pelabelan v_1 (a), v_2 (c) dan e_1 (c)

Label simpul akan diberikan sesuai dengan label yang tersedia. Sedangkan label busur ditentukan dengan pengurangan konstanta ajaib k dengan jumlah label dari simpul-simpul yang dihubungkan oleh busur tersebut. Oleh karena itu minimal 2 simpul yang diberi label terlebih dahulu. Ini sebagai dasar untuk pemberian label simpul dan busur lainnya. Pemberian label dimulai dengan melabel simpul v_1 dan v_2 . Seperti yang telah dijelaskan sebelumnya, untuk menghindari kasus rotasi diberikan syarat $\lambda(v_1) < \lambda(v_t)$, $t = 2, 3, \dots, n$ sehingga label yang terbesar untuk simpul v_1 adalah $n + 1$.

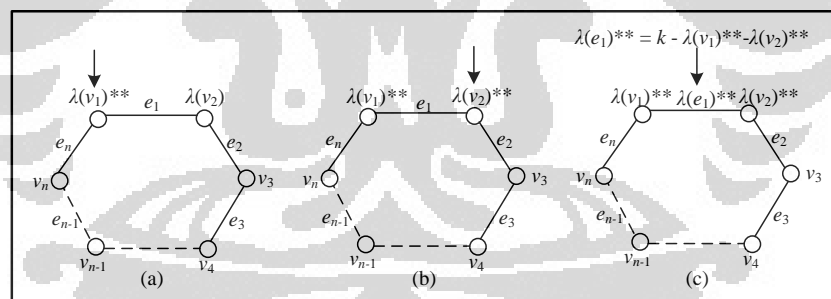
Selanjutnya simpul v_2 diberi label dimana $\lambda(v_2) > \lambda(v_1)$. Label busur e_1 ditentukan sebesar $\lambda(e_1) = k - \lambda(v_1) - \lambda(v_2)$ agar bobot busur e_1 sama dengan k . Tahap

ini diperlihatkan pada Gambar 3.5. Penandaan label telah digunakan adalah dengan fungsi avail, $avail[i] := \text{false}$ berarti label i telah digunakan sehingga label i tidak tersedia.



Gambar 3.6 Tahap mengganti label v_2 apabila label busur busur e_1 tidak tersedia ($\lambda(e_1) < 1$ atau $\lambda(e_1) > 2n$)

Apabila label busur e_1 tidak tersedia ($\lambda(e_1) < 1$ atau $\lambda(e_1) > 2n$) maka label simpul v_2 akan diganti. Label yang digunakan sebelumnya menjadi *available* ditandai dengan $avail[i] := \text{true}$. Ini dilakukan sampai diperoleh label busur e_1 yang sesuai dengan label yang tersedia (Gambar 3.6).

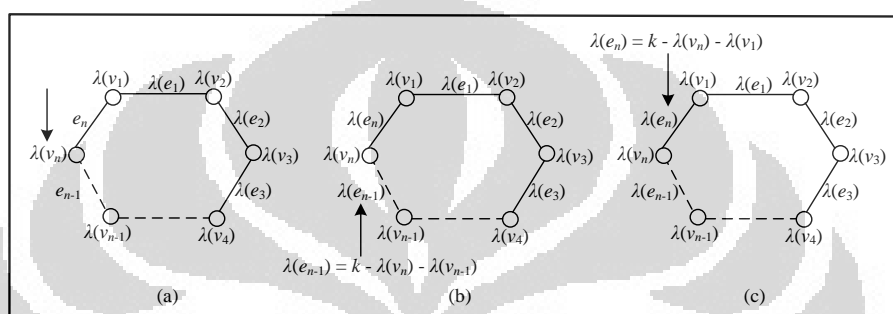


Gambar 3.7 Tahap *backtracking* ke v_1 apabila label e_1 tidak tersedia

Jika label busur e_1 tetap tidak diperoleh maka dilakukan *backtracking* untuk mengubah label simpul yang dilabel sebelumnya, yaitu v_1 . Hal ini dilakukan hingga diperoleh label busur e_1 yang sesuai dengan label yang tersedia (Gambar 3.7).

Selanjutnya melabel simpul v_3 dimana $\lambda(v_3) > \lambda(v_1)$. Busur e_2 diberi label sebesar $\lambda(e_2) = k - \lambda(v_1) - \lambda(v_2)$. Jika label busur e_2 tidak tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia. Dengan cara yang sama

diberi label v_3, e_2 , dst. Dikarenakan langkah melabel simpul dan busur setelah v_1, v_2 , dan e_1 sama maka bisa dijadikan sebuah fungsi. Sehingga pelabelan v_1, v_2 , dan e_1 di fungsi awal disebut fungsi **initializeCycle** yang akan memanggil fungsi selanjutnya yang disebut fungsi **extendCycle** dengan parameter $t = 2$ untuk melabel simpul v_3 dan busur e_2 . Selanjutnya untuk melabel simpul v_4 dan busur e_3 fungsi **extendCycle** akan memanggil dirinya sendiri dengan parameter $t = t + 1$, begitu seterusnya.



Gambar 3.8 Tahap pelabelan v_n (a), e_{n-1} (b), dan e_n (c)

Disaat $t = n - 1$ tersisa 1 simpul dan 2 busur yang belum dilabel yaitu v_n, e_{n-1}, e_n . Saat melabel simpul v_n perlu diberi syarat $\lambda(v_n) > \lambda(v_2)$ agar tidak terjadi kasus refleksi sehingga diperlukan kondisi if untuk $t = n - 1$. Setelah melabel simpul v_n , maka label e_{n-1} adalah $\lambda(e_{n-1}) = k - \lambda(v_{n-1}) - \lambda(v_n)$. Jika label busur e_{n-1} tidak tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia. Selanjutnya melabel e_n yang besarnya dengan $\lambda(e_n) = k - \lambda(v_n) - \lambda(v_1)$ (Gambar 3.8). Apabila nilai ini sama dengan label yang masih tersedia maka diperoleh satu PTBA untuk graf lingkaran. Kemudian label untuk setiap simpul dan busur yang diperoleh dicetak dengan memanggil fungsi **print()**.

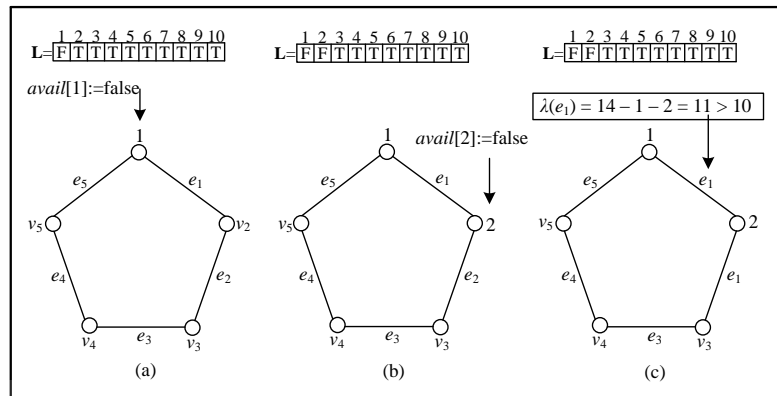
Walaupun telah ditemukan satu PTBA untuk graf lingkaran, algoritma ini akan berlanjut untuk mencari PTBA yang lainnya. Proses selanjutnya simpul v_n akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_{n-1} dan e_n diperoleh sehingga diperoleh PTBA yang lain. Jika label kedua busur tersebut tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_{n-1} dan seterusnya sesuai dengan urutan langkah-langkah yang telah dilakukan sebelumnya untuk memperoleh PTBA

yang lainnya. Algoritma ini akan berhenti apabila telah diperoleh semua PTBA yang mungkin dan berbeda pada graf lingkaran.

<p>Algoritma 1.i initializeCycle</p> <pre> 1 for each available label i where $i \leq n+1$ do 2 $\lambda(v_1) = i$ 3 $avail[i] := false$ 4 for each available label j where $j > i$ do 5 $\lambda(v_2) = j$ 6 $avail[j] := false$ 7 $\lambda(e_1) := k - \lambda(v_1) - \lambda(v_2)$ 8 if $0 < \lambda(e_1) \leq 2n$ and $avail[\lambda(e_1)]$ then 9 $avail[\lambda(e_1)] := false$ 10 extendCycle(2) 11 $avail[\lambda(e_1)] := true$ 12 end if 13 end for 14 $avail[j] := true$ 15 end for 16 $avail[i] := true$ 17 end for </pre>	<p>Algoritma 1.ii extendCycle(t)</p> <pre> 1 if $t = n-1$ then 2 for each available label i where 3 $i > \lambda(v_1)$ and $i > \lambda(v_2)$ do 4 $\lambda(v_n) := i$ 5 $avail[i] := false$ 6 $\lambda(e_{n-1}) := k - \lambda(v_{n-1}) - \lambda(v_n)$ 7 if $0 < \lambda(e_{n-1}) \leq 2n$ and $avail[\lambda(e_{n-1})]$ 8 then 9 $avail[\lambda(e_{n-1})] := false$ 10 $\lambda(e_n) := k - \lambda(v_n) - \lambda(v_1)$ 11 if $0 < \lambda(e_n) \leq 2n$ and $avail[\lambda(e_n)]$ then 12 print() 13 end if 14 end if 15 $avail[\lambda(e_{n-1})] := true$ 16 end if 17 end for 18 $avail[i] := true$ 19 end for 20 else 21 for each available label i where $i > \lambda(v_1)$ do 22 $\lambda(v_{t+1}) := i$ 23 $avail[i] := false$ 24 $\lambda(e_t) := k - \lambda(v_t) - \lambda(v_{t+1})$ 25 if $0 < \lambda(e_t) \leq 2n$ and $avail[\lambda(e_t)]$ then 26 $avail[\lambda(e_t)] := false$ 27 extendCycle($t + 1$) 28 $avail[\lambda(e_t)] := true$ 29 end if 30 end for 31 $avail[i] := true$ 32 end for 33 end if </pre>
--	--

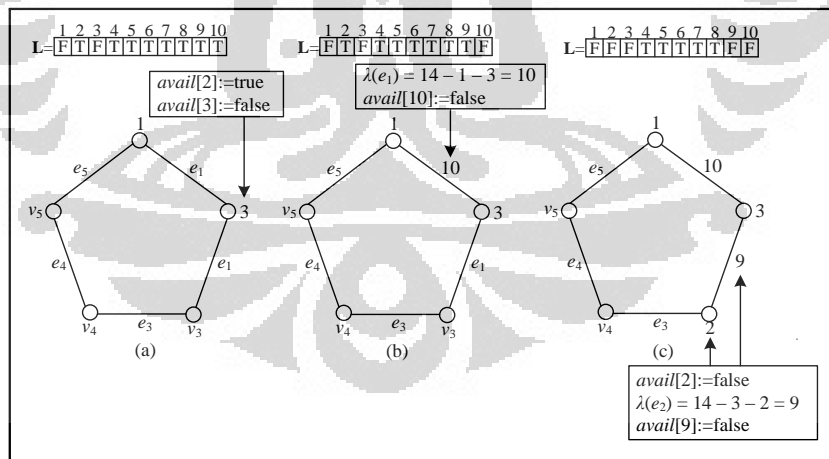
Jadi algoritma PTBA pada graf Lingkaran terdiri dari 2 fungsi yaitu **initializeCycle** (Algoritma 1.i) dan **extendCycle** (Algoritma 1.ii). Algoritma ini melabel simpul dan busur secara iteratif serta di dalamnya terdapat pemanggilan fungsi secara rekursif. Masukan dari algoritma ini adalah nilai n dan k dari graf lingkaran yang akan dilabel sedangkan keluaran yang diperoleh adalah semua PTBA yang mungkin dan berbeda untuk graf lingkaran tersebut.

Berikut ini akan diberikan penjelasannya algoritma dengan menggunakan contoh. Misalkan akan dicari PTBA pada graf lingkaran C_5 . Dari (2.6) diperoleh batas k untuk C_5 adalah $14 \leq k \leq 19$. Misalkan dipilih $k = 14$. Jadi yang menjadi masukan pada algoritma adalah $n = 5$ dan $k = 14$. Label yang tersedia adalah 1, 2, 3, ..., 9, 10. Algoritma mulai bekerja pada tahap inialisasi dengan memanggil fungsi **initializeCycle** (Algoritma 1.i).



Gambar 3.9 Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ pada fungsi initializeCycle, melabel v_1 (a), v_2 (b), dan e_1 (c)

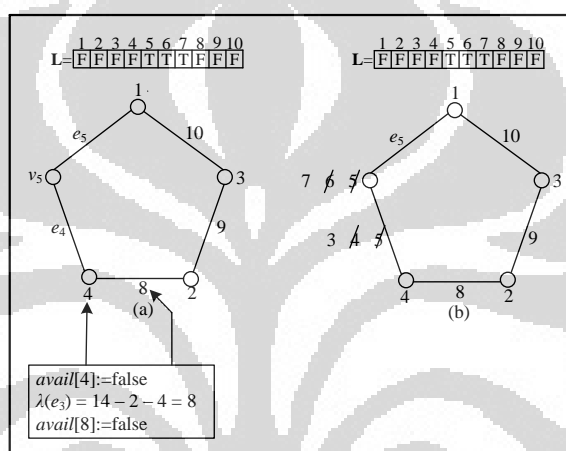
Pertama dilabel simpul v_1 . Misalkan simpul v_1 diberi label 1, maka $avail[1]:=false$ yang menandakan bahwa label 1 sudah tidak tersedia lagi dan tidak bisa digunakan lagi untuk melabel. Label yang tersedia sekarang adalah 2, 3, ..., 9, 10. Seperti yang diperlihatkan pada Gambar 3.9(a). Selanjutnya dilabel simpul v_2 dimana $\lambda(v_2) > \lambda(v_1)$ untuk menghindari kasus rotasi. Simpul v_2 diberi label 2 dimana $2 > \lambda(v_1) = 1$ dan maka $avail[2] := false$. Label yang tersedia sekarang adalah 3, 4, 5, 6, 7, 8, 9, 10 (Gambar 3.9(b)).



Gambar 3.10 Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi initializeCycle, mengganti label v_2 (a), melabel kembali e_1 (b), melabel v_3 dan e_2 (c)

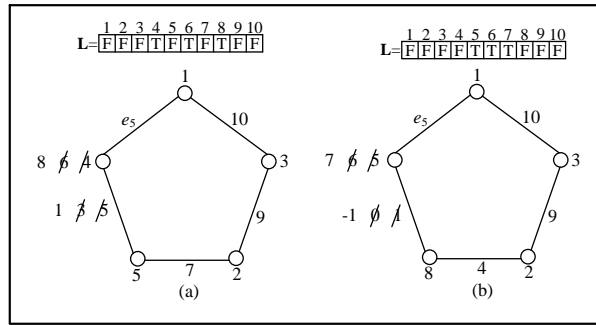
Selanjutnya label e_1 ditentukan agar bobot busur e_1 sama dengan k ($w_{\lambda}(e_1) = k$). Diperoleh label e_1 , yaitu $\lambda(e_1) = k - \lambda(v_1) - \lambda(v_2) = 14 - 1 - 2 = 11$ (Gambar 3.9(c)). Karena label busur e_1 yang diperoleh tidak tersedia maka $\lambda(e_1) = 11$ tidak

diterima. Elemen graf yang terakhir dilabel, yaitu simpul v_2 akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_1 diperoleh. Label yang digunakan sebelumnya disediakan kembali dengan tanda $avail[2] = true$. Berarti kembali ke baris 4 Algoritma 1.i. Simpul v_2 diberi label 3 dan ditandai $avail[3] := false$. Kemudian dihitung label e_1 , yaitu $\lambda(e_1) = k - \lambda(v_1) - \lambda(v_2) = 14 - 1 - 3 = 10$. Ternyata $\lambda(e_1) = 10$ tersedia sehingga $\lambda(e_1) = 10$ diterima dan ditandai $avail[10] := false$. Label yang tersedia sekarang adalah 2, 4, 5, 6, 7, 8, 9. Langkah-langkah ini diperlihatkan pada Gambar 3.10 (a) dan (b).



Gambar 3. 11 Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi `extendCycle`, melabel v_4, e_3 (a) dan melabel v_5, e_4

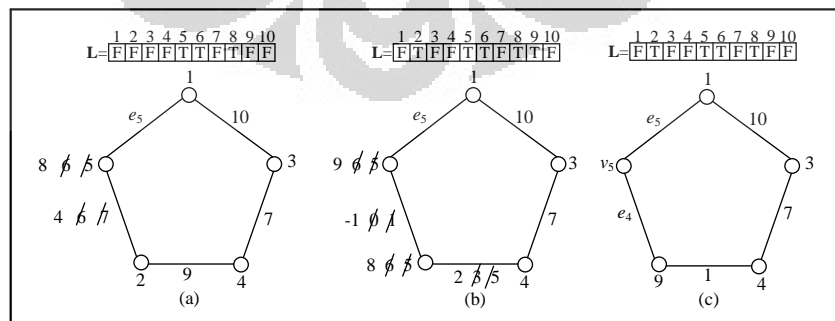
Kemudian fungsi `initializeCycle` memanggil fungsi `extendCycle` dengan parameter $t = 2$ (Algoritma 1.i baris 10) untuk melabel v_3 . Karena $t = 2$ maka masuk ke bagian `else` (baris 13 Algoritma 1.ii). Simpul v_3 dilabel 2, ditandai $avail[2] := false$. Kemudian dihitung label e_2 , yaitu sebesar $\lambda(e_2) = k - \lambda(v_2) - \lambda(v_3) = 14 - 3 - 2 = 9$. Ternyata $\lambda(e_2) = 9$ tersedia sehingga $\lambda(e_2) = 9$ diterima dan ditandai $avail[9] := false$. Label yang tersedia sekarang adalah 4, 5, 6, 7, 8 (Gambar 3.10(c)). Setelah itu fungsi `extendCycle` memanggil dirinya sendiri dengan parameter $t = 3$ (baris 20 Algoritma 1.ii). Dengan cara yang sama diperoleh $\lambda(v_4) = 4$ dan $\lambda(e_3) = 8$ sehingga label yang tersedia adalah 5, 6, 7 (Gambar 3.11(a)).



Gambar 3. 12 Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi `extendCycle`, semua pelabelan yang mungkin untuk $\lambda(v_4) = 5$ (a) dan $\lambda(v_4) = 8$ (b)

Fungsi `extendCycle` memanggil dirinya sendiri dengan parameter $t = 4$ sehingga memenuhi kondisi `if` (baris 1 Algoritma 1.ii). Untuk menghindari kasus rotasi $\lambda(v_5) > \lambda(v_1)$ dan refleksi $\lambda(v_5) > \lambda(v_2)$. Sehingga simpul v_5 dilabel 5 dan ditandai `avail[5] := false`. Kemudian dihitung label e_4 , yaitu $\lambda(e_4) = k - \lambda(v_4) - \lambda(v_5) = 14 - 4 - 5 = 5$. Ternyata $\lambda(e_4) = 5$ tidak tersedia di list label karena telah digunakan untuk melabel simpul v_5 . Sehingga simpul v_5 akan diganti labelnya dengan label lain yang tersedia, sampai diperoleh label busur e_4 yang sesuai dengan label yang tersedia.

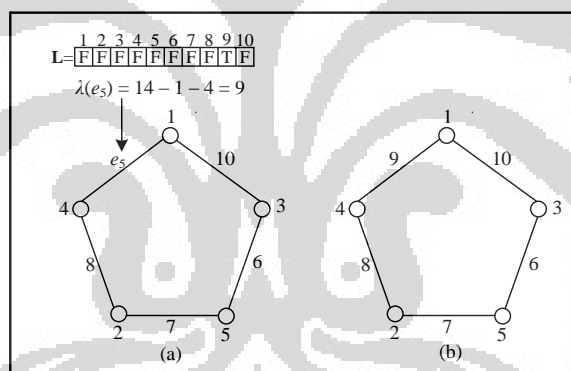
Karena label busur e_4 tetap tidak diperoleh (Gambar 3.11(b)), maka dilakukan `backtracking` untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_4 . Ternyata label busur e_4 tetap tidak diperoleh (Gambar 3.12) sehingga dilakukan `backtracking` ke simpul v_3 untuk mengganti label simpul tersebut. Simpul v_3 dilabel 4.



Gambar 3. 13 Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi `extendCycle`, semua pelabelan yang mungkin untuk $\lambda(v_3) = 4$

Kemudian dengan cara yang sama dilabel simpul dan busur selanjutnya (Gambar 3.13). Pada Gambar 3.13(a) diperlihatkan bahwa label busur e_4 tetap tidak diperoleh walaupun label simpul v_5 sudah diganti dengan semua label yang tersedia. Sehingga dilakukan *backtracking* ke simpul v_4 untuk mengganti label simpul tersebut. Ternyata label busur e_4 tetap tidak diperoleh (Gambar 3.13(b) dan (c)).

Dikarenakan label busur e_4 tetap tidak diperoleh, dilakukan *backtracking* ke simpul v_3 seperti yang dilakukan sebelumnya. Label busur e_2 dan simpul v_3 dinyatakan tersedia kembali dengan $avail[7]:=true$ dan $avail[4]:=true$. Label yang tersedia sekarang adalah 2, 4, 5, 6, 7, 8, 9. Simpul v_3 dilabel 5 dan diberi tanda $avail[5] := false$. Kemudian dihitung label e_2 , yaitu $\lambda(e_2) = k - \lambda(v_2) - \lambda(v_3) = 14 - 3 - 5 = 6$. Ternyata $\lambda(e_2) = 6$ tersedia sehingga $\lambda(e_2) = 6$ diterima dan ditandai $avail[6] := false$.



Gambar 3. 14 Contoh penerapan algoritma PTBA pada graf C_5 dengan $k = 14$ fungsi `extendCycle(4)`, melabel e_5 (a) dan PTBA yang diperoleh untuk C_5 dengan $k = 14$ (b)

Setelah itu fungsi `extendCycle` memanggil dirinya sendiri dengan parameter $t = 3$. Dengan cara yang sama diperoleh $\lambda(v_4) = 2$ dan $\lambda(e_3) = 7$. Kemudian fungsi `extendCycle` memanggil dirinya sendiri dengan parameter $t = 4$. Simpul v_5 dilabel 4 dan ditandai $avail[4] := false$. Label yang tersedia sekarang adalah 8 dan 9. Kemudian dihitung label e_4 , yaitu $\lambda(e_4) = k - \lambda(v_4) - \lambda(v_5) = 14 - 2 - 4 = 8$. Label $\lambda(e_4) = 8$ tersedia sehingga $\lambda(e_4) = 8$ diterima. Selanjutnya dihitung label e_5 , yaitu $\lambda(e_5) = k - \lambda(v_1) - \lambda(v_5) = 14 - 1 - 4 = 9$. Ternyata $\lambda(e_5) = 9$ sama dengan label

yang masih tersedia sehingga diperoleh PTBA pada graf C_5 dengan $k = 14$. Langkah-langkah ini diberikan pada Gambar 3.14 (a).

Kemudian dipanggil fungsi **print()** (baris 10 Algoritma 1.ii) untuk mencetak label-label yang diperoleh yaitu $\lambda(v_1) = 1, \lambda(v_2) = 3, \lambda(v_3) = 5, \lambda(v_4) = 2, \lambda(v_5) = 4, \lambda(e_1) = 10, \lambda(e_2) = 6, \lambda(e_3) = 7, \lambda(e_4) = 8, \lambda(e_5) = 9$. Seperti yang telah dijelaskan sebelumnya, algoritma ini akan berlanjut untuk mencari PTBA yang lainnya. Proses selanjutnya simpul v_n akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_{n-1} dan e_n diperoleh yang berarti diperoleh PTBA yang lain. Jika label kedua busur tersebut tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_{n-1} dan seterusnya sesuai dengan urutan langkah-langkah yang telah dilakukan sebelumnya untuk memperoleh PTBA yang lainnya. Algoritma ini akan berhenti sampai diperoleh semua PTBA yang mungkin dan berbeda pada graf lingkaran. Untuk graf lingkaran C_5 dengan $k = 14$ tidak ditemukan lagi PTBA yang lainnya. Sehingga PTBA untuk graf lingkaran C_5 dengan $k = 14$ hanya satu (Gambar 3.14(b)).

Pada Subbab 3.2 akan dijelaskan mengenai pembentukan algoritma PTBA pada graf kipas, syarat-syarat yang harus dipenuhi untuk menghindari pelabelan isomorfik, beserta contohnya.

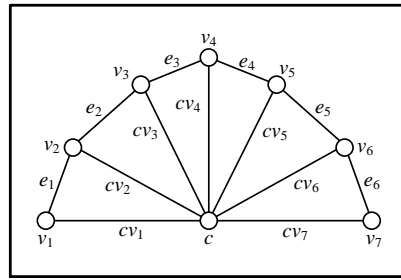
3.2 Algoritma PTBA pada Graf Kipas f_n

Graf kipas (f_n) adalah suatu graf yang diperoleh dengan menghubungkan semua simpul dari suatu graf lintasan P_n ($n \geq 2$) ke satu simpul lain yang disebut simpul pusat (*center*). Sehingga graf kipas memiliki $n + 1$ simpul dan $2n - 1$ busur. Apabila $n = 2$ maka graf kipas f_2 sama dengan graf lingkaran C_3 . Sehingga dalam bab ini akan dibahas graf kipas dengan ukuran $n > 2$. Pada gambar 3.15 diberikan graf kipas dengan 8 simpul (f_7) beserta penamaan simpul dan busurnya.

Pada Subbab 2.3 telah diperoleh batasan konstanta ajaib (k) untuk graf kipas, yaitu

$$\frac{11n^2+7n-4}{4n-2} \leq k \leq \frac{25n^2-13n-2}{4n-2}.$$

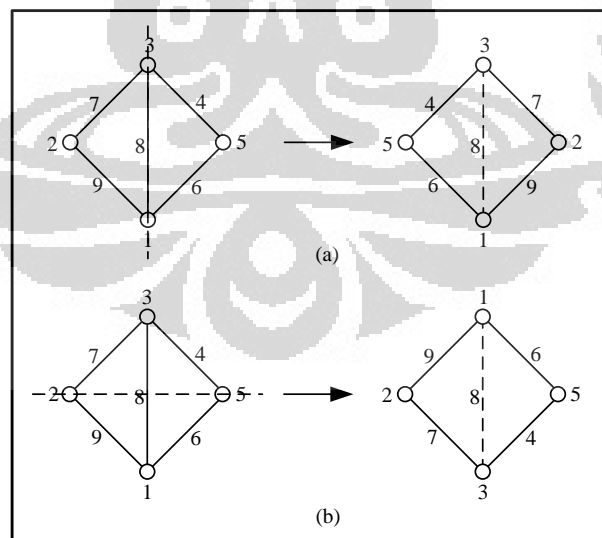
Sebagai contoh batas k yang mungkin untuk graf f_5 adalah $17 \leq k \leq 31$. Konstanta ajaib (k) untuk graf f_5 adalah 17, 18, ..., 31.



Gambar 3. 15 Graf kipas dengan 8 simpul (f_7)

Secara garis besar algoritma PTBA pada graf kipas sama dengan algoritma PTBA pada graf lingkaran. Algoritma ini melabel simpul dan busur graf kipas dengan label yang tersedia tepat satu kali secara iteratif. Sehingga setiap busur memiliki bobot yang sama dan menghasilkan semua PTBA yang mungkin (jika ada) yang tidak isomorfik.

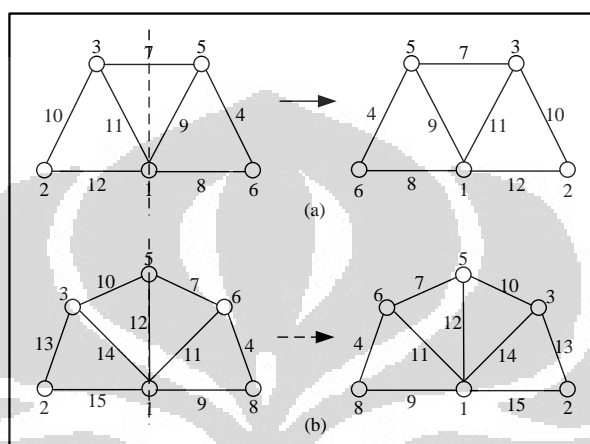
Seperti yang telah dijelaskan sebelumnya, agar diperoleh PTBA yang tidak isomorfik perlu dihindari kasus refleksi dan kasus rotasi. Kasus refleksi terjadi pada graf kipas untuk setiap $n \geq 2$. Untuk $n = 2$ sudah dibahas di Subbab 3.1 sehingga akan dibahas untuk $n > 2$.



Gambar 3. 16 Kasus refleksi pada graf f_3

Saat $n = 3$ kasus refleksi yang terjadi diperlihatkan pada Gambar 3.16. Pada Gambar 3.16(a) pelabelan di sebelah kiri direfleksikan terhadap sumbu refleksi

yang melalui simpul pusat sehingga diperoleh pelabelan di sebelah kanan. Untuk menghindari kasus ini diberi syarat $\lambda(v_1) < \lambda(v_3)$. Sedangkan pada Gambar 3.16(b) pelabelan di sebelah kiri direfleksikan terhadap sumbu refleksi yang melalui simpul v_1 dan v_3 sehingga diperoleh pelabelan di sebelah kanan. Kasus ini dapat dihindari dengan memberi syarat $\lambda(c) < \lambda(v_2)$.

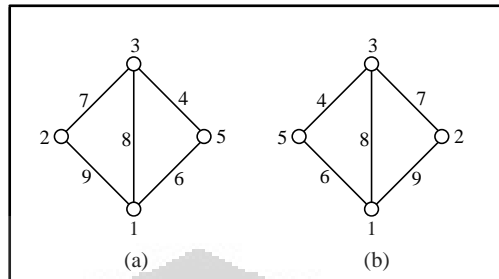


Gambar 3. 17 Kasus refleksi pada graf f_4 (a) dan f_5 (b)

Pada saat $n > 3$ kasus refleksi yang terjadi diperlihatkan pada Gambar 3.17. Pada Gambar 3.17 diberikan contoh kasus refleksi pada graf f_4 (a) dan f_5 (b). Pelabelan di sebelah kiri direfleksikan terhadap sumbu refleksi yang melalui simpul pusat sehingga diperoleh pelabelan di sebelah kanan. Kasus refleksi ini sama seperti kasus refleksi pada graf f_3 yang diperlihatkan pada Gambar 3.16(a). Secara umum kasus refleksi ini dapat dihindari dengan memberi syarat label salah satu simpul di salah satu sisi dari sumbu refleksi harus lebih kecil (atau lebih besar) dari label salah satu simpul pada sisi yang lain. Sehingga diberi syarat $\lambda(v_1) < \lambda(v_n)$ untuk $n > 2$. Sedangkan kasus refleksi pada Gambar 3.16(b) hanya terjadi saat $n = 3$. Syarat untuk menghindari kasus ini adalah $\lambda(c) < \lambda(v_t)$ dimana $t = 2$.

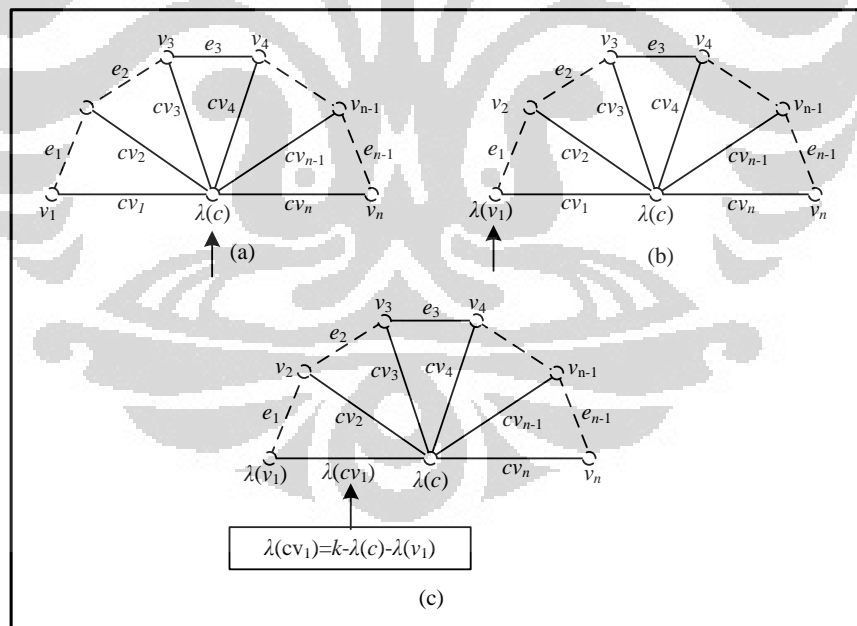
Kasus rotasi pada graf kipas f_n hanya terjadi pada saat $n = 2, 3$. Untuk $n = 2$ sudah dibahas di Subbab 3.1 sehingga akan dibahas untuk $n = 3$. Pada Gambar 3.16 diberikan contoh kasus rotasi pada graf f_3 . Pelabelan pada Gambar 3.18(b) dapat diperoleh dengan merotasi pelabelan pada Gambar 3.18(a) atau sebaliknya. Kasus ini bisa dihindari dengan memberikan syarat yang telah diberikan

untuk menghindari kasus refleksi pada graf f_3 . Syarat yang diberikan adalah $\lambda(v_1) < \lambda(v_n)$ dan $\lambda(c) < \lambda(v_t)$ dimana $t = 2$



Gambar 3. 18 Kasus rotasi pada graf f_3

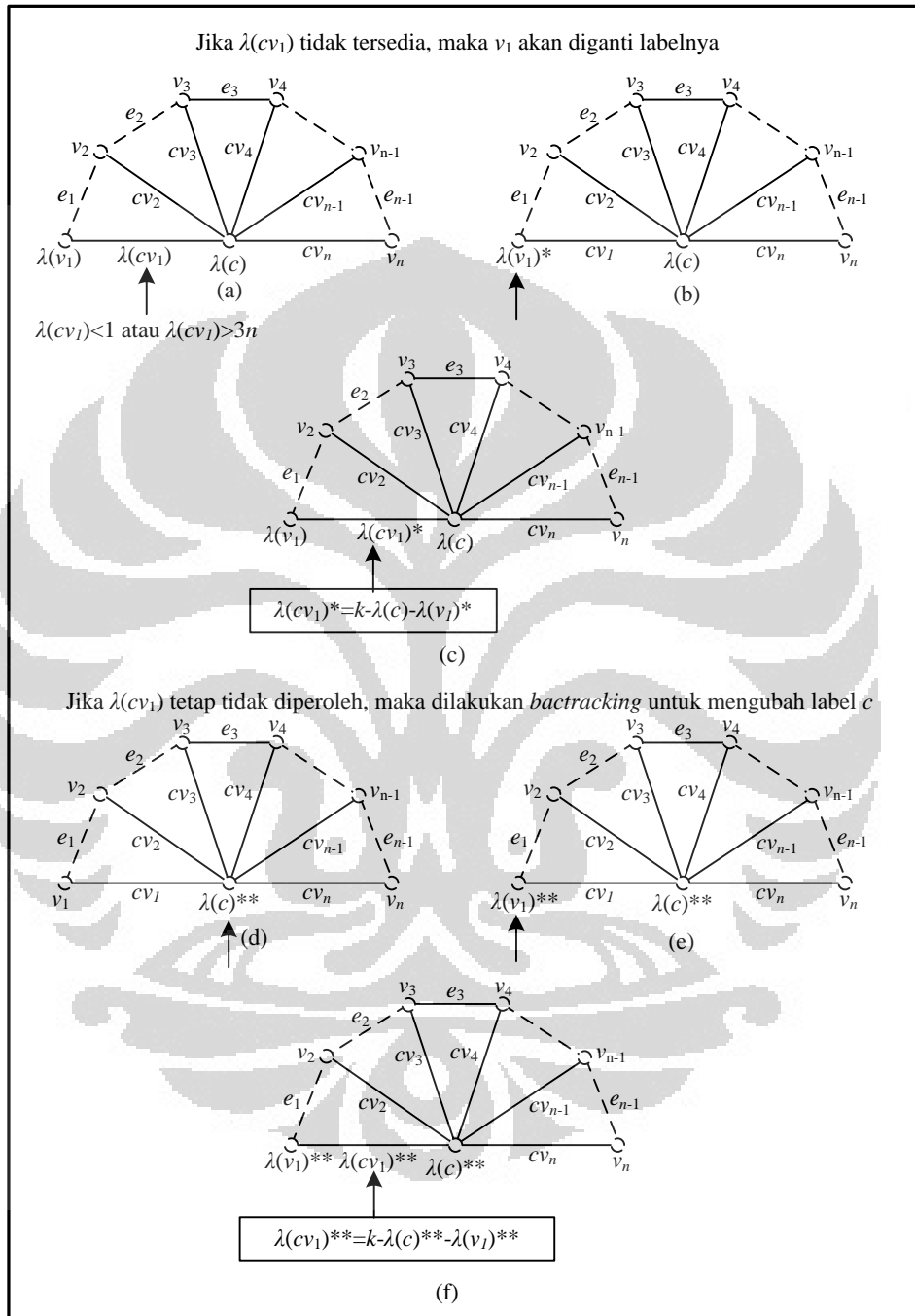
Seperti yang telah dijelaskan sebelumnya, secara garis besar algoritma PTBA pada graf kipas sama dengan algoritma PTBA pada graf lingkaran. Masukan dari algoritma ini, yaitu banyaknya simpul selain simpul pusat n dan konstanta ajaib k . Label yang tersedia adalah 1 sampai $3n$.



Gambar 3. 19 Tahap pelabelan c (a), v_1 (b), dan cv_1 (c)

Sama seperti yang telah dijelaskan pada Subbab 3.1, minimal 2 simpul yang diberi label terlebih dahulu. Ini sebagai dasar untuk pemberian label simpul dan busur

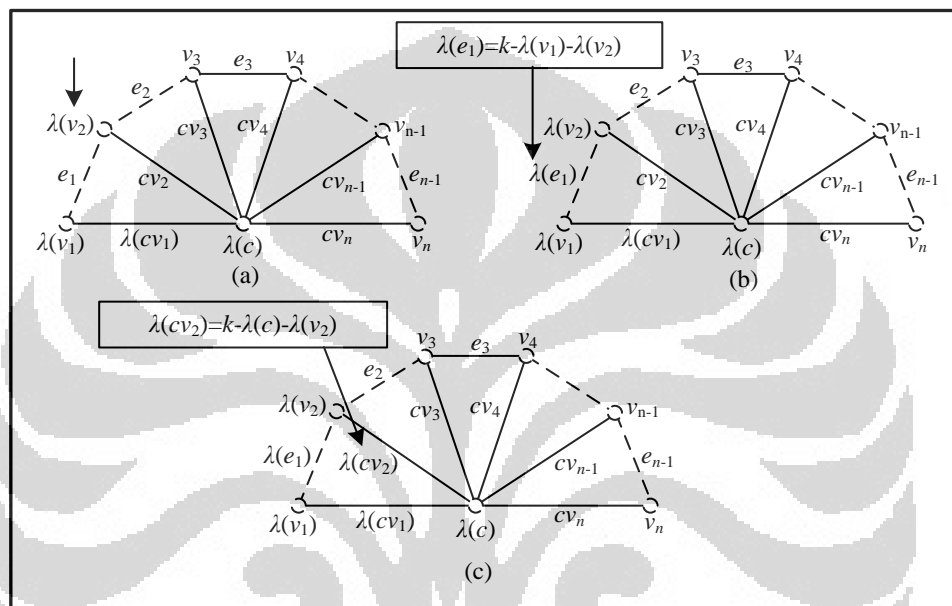
lainnya. Setelah melabel c , dilabel v_1 . Kemudian ditentukan besar label cv_1 , yaitu $\lambda(cv_1) = k - \lambda(c) - \lambda(v_1)$. Tahap ini diperlihatkan pada Gambar 3.19.



Gambar 3. 20 Tahap apabila label cv_1 tidak tersedia

Jika label busur cv_1 yang diperoleh tidak tersedia ($\lambda(cv_1) < 1$ atau $\lambda(cv_1) < 3n$), maka elemen graf yang terakhir dilabel, yaitu simpul v_1 akan diganti labelnya

dengan label lain yang tersedia, sampai label busur cv_1 diperoleh. Jika label busur cv_1 tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul c . Hal ini dilakukan sampai diperoleh label busur cv_1 yang sesuai dengan label yang tersedia (Gambar 3.20). Label i yang telah digunakan ditandai dengan $avail[i] := \text{false}$, jika label i yang telah digunakan disediakan kembali ditandai dengan $avail[i] := \text{true}$.

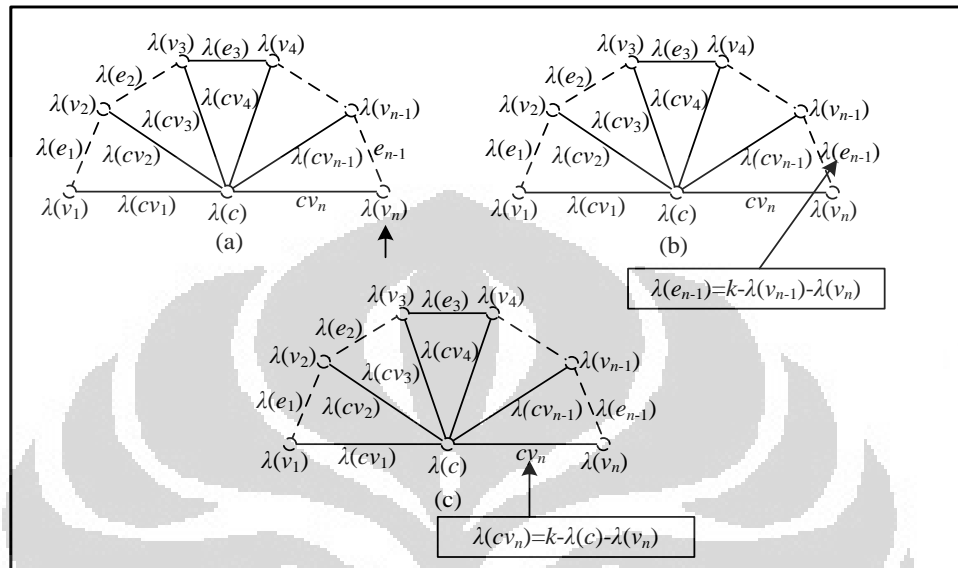


Gambar 3. 21 Tahap pelabelan v_2 (a), e_1 (b), dan cv_2 (c)

Setelah diperoleh label busur cv_1 , selanjutnya melabel v_2 . Dikarenakan terdapat kasus refleksi yang hanya terjadi pada saat $n = 3$ dimana syarat untuk menghindarinya melibatkan simpul v_2 maka diperlukan kondisi **if** $n = 3$. Syarat yang ditambahkan untuk menghindari kasus refleksi tersebut adalah $\lambda(c) < \lambda(v_t)$ dimana $t = 2$.

Selanjutnya ditentukan label dari busur e_1 , yaitu $\lambda(e_1) = k - \lambda(v_1) - \lambda(v_2)$. Jika label busur e_1 tidak tersedia maka dilakukan hal yang sama seperti saat label busur cv_1 tidak tersedia, yaitu simpul v_2 akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_1 diperoleh. Jika label busur e_1 tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_1 . Hal ini dilakukan sampai diperoleh label busur e_1 yang sesuai dengan label yang tersedia. Setelah label busur e_1 tersedia, di-

tentukan label busur cv_2 sebesar $\lambda(cv_2) = k - \lambda(c) - \lambda(v_2)$. Jika label busur cv_2 tidak tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia (Gambar 3.21).



Gambar 3. 22 Tahap pelabelan v_n (a), e_{n-1} (b), dan cv_n (c)

Kemudian dengan cara yang sama dilabel v_3, e_2, cv_3 , dst. Dikarenakan langkah melabel simpul dan busur setelah c, v_1 , dan cv_1 sama, maka dapat dijadikan sebuah fungsi. Sehingga pelabelan c, v_1 , dan cv_1 di fungsi awal disebut fungsi **initializeFan**. Fungsi **initializeFan** akan memanggil fungsi selanjutnya yang disebut fungsi **extendFan** dengan parameter $t = 2$ untuk melabel simpul v_2 , busur e_1 dan cv_2 . Selanjutnya untuk melabel simpul v_3 , busur e_2 dan cv_3 fungsi **extendFan** akan memanggil dirinya sendiri dengan parameter $t = t + 1$, begitu seterusnya.

Disaat $t = n$ tersisa 1 simpul dan 2 busur yang belum dilabel yaitu v_n, e_{n-1} , dan cv_n . Saat melabel simpul v_n perlu diberi syarat $\lambda(v_n) > \lambda(v_1)$ agar tidak terjadi kasus refleksi, sehingga diperlukan kondisi **if** untuk $t = n$. Setelah simpul v_n dilabel, ditentukan label e_{n-1} sebesar $\lambda(e_{n-1}) = k - \lambda(v_{n-1}) - \lambda(v_n)$. Apabila label busur e_{n-1} tidak tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia. Hal ini dilakukan sampai diperoleh label busur e_{n-1} yang sesuai dengan label yang tersedia. Lalu ditentukan label cv_n , yaitu $\lambda(cv_n) = k - \lambda(c) - \lambda(v_n)$ (Gambar 3.22).

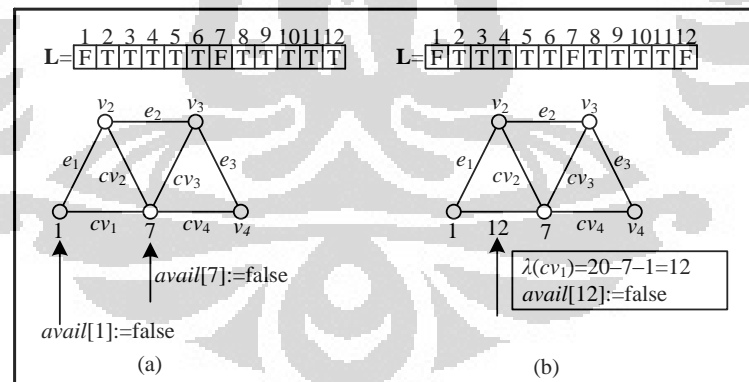
<p>Algoritma 2.i initializeFan</p> <pre> 1 for each available label i do 2 $\lambda(c) := i$ 3 $avail[i] := false$ 4 for each available label j do 5 $\lambda(v_1) := j$ 6 $avail[j] := false$ 7 $\lambda(cv_1) := k - \lambda(c) - \lambda(v_1)$ 8 if $0 < \lambda(cv_1) \leq 3n$ and $avail[\lambda(cv_1)]$ then 9 $avail[\lambda(cv_1)] := false$ 10 extendFan(2) 11 $avail[\lambda(cv_1)] := true$ 12 end if 13 end for 14 $avail[i] := true$ 15 end for </pre>	<p>Algoritma 2.ii extendFan(t)</p> <pre> 1 if $t = n$ 2 for each available label i where $i > \lambda(v_1)$ do 3 $\lambda(v_n) := i$ 4 $avail[i] := false$ 5 $\lambda(e_{n-1}) := k - \lambda(v_{n-1}) - \lambda(v_n)$ 6 if $0 < \lambda(e_{n-1}) \leq 3n$ and $avail[\lambda(e_{n-1})]$ then 7 $avail[\lambda(e_{n-1})] := false$ 8 $\lambda(cv_n) := k - \lambda(c) - \lambda(v_n)$ 9 if $0 < \lambda(cv_n) \leq 3n$ and $avail[\lambda(cv_n)]$ 10 then 11 print() 12 end if 13 end if 14 $avail[\lambda(e_{n-1})] := true$ 15 end for 16 else 17 if $n = 3$ 18 for each available label i where $i > \lambda(c)$ do 19 $\lambda(v_t) := i$ 20 $avail[i] := false$ 21 $\lambda(e_{t-1}) := k - \lambda(v_{t-1}) - \lambda(v_t)$ 22 if $0 < \lambda(e_{t-1}) \leq 3n$ and $avail[\lambda(e_{t-1})]$ 23 then 24 $avail[\lambda(e_{t-1})] := false$ 25 $\lambda(cv_t) := k - \lambda(c) - \lambda(v_t)$ 26 if $0 < \lambda(cv_t) \leq 3n$ and $avail[\lambda(cv_t)]$ 27 then 28 $avail[\lambda(cv_t)] := false$ 29 extendFan($t + 1$) 30 $avail[\lambda(cv_t)] := true$ 31 end if 32 end if 33 $avail[\lambda(e_{t-1})] := true$ 34 end for 35 else 36 for each available label i do 37 $\lambda(v_t) := i$ 38 $avail[i] := false$ 39 $\lambda(e_{t-1}) := k - \lambda(v_{t-1}) - \lambda(v_t)$ 40 if $0 < \lambda(e_{t-1}) \leq 3n$ and $avail[\lambda(e_{t-1})]$ 41 then 42 $avail[\lambda(e_{t-1})] := false$ 43 $\lambda(cv_t) := k - \lambda(c) - \lambda(v_t)$ 44 if $0 < \lambda(cv_t) \leq 3n$ and $avail[\lambda(cv_t)]$ 45 then 46 $avail[\lambda(cv_t)] := false$ 47 extendFan($t + 1$) 48 $avail[\lambda(cv_t)] := true$ 49 end if 50 end if 51 $avail[\lambda(e_{t-1})] := true$ 52 end for 53 end if 54 end if </pre>
---	--

Jika nilai $\lambda(cv_n)$ tidak sama dengan label yang masih tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia. Hal ini dilakukan sam-

pai diperoleh label busur cv_n yang sesuai dengan label yang tersedia. Jika label busur cv_n yang diperoleh sama dengan label yang masih tersedia maka diperoleh sebuah PTBA untuk graf kipas. Label simpul dan busur yang diperoleh dicetak dengan memanggil fungsi **print()**.

Walaupun telah diperoleh satu PTBA untuk graf kipas, algoritma ini akan berlanjut untuk mencari PTBA yang lainnya. Proses selanjutnya simpul v_n akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_{n-1} dan cv_n diperoleh sehingga diperoleh PTBA yang lain. Jika label kedua busur tersebut tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_{n-1} dan seterusnya sesuai dengan urutan langkah-langkah yang telah dilakukan sebelumnya untuk memperoleh PTBA yang lainnya. Algoritma ini akan berhenti apabila telah diperoleh semua PTBA yang mungkin dan berbeda pada graf kipas.

Jadi algoritma PTBA pada graf kipas terdiri dari 2 fungsi yaitu fungsi **initializeFan** (Algoritma 2.i) dan fungsi **extendFan** (Algoritma 2.ii). Keluaran yang diperoleh adalah semua PTBA yang mungkin dan berbeda untuk graf kipas tersebut.



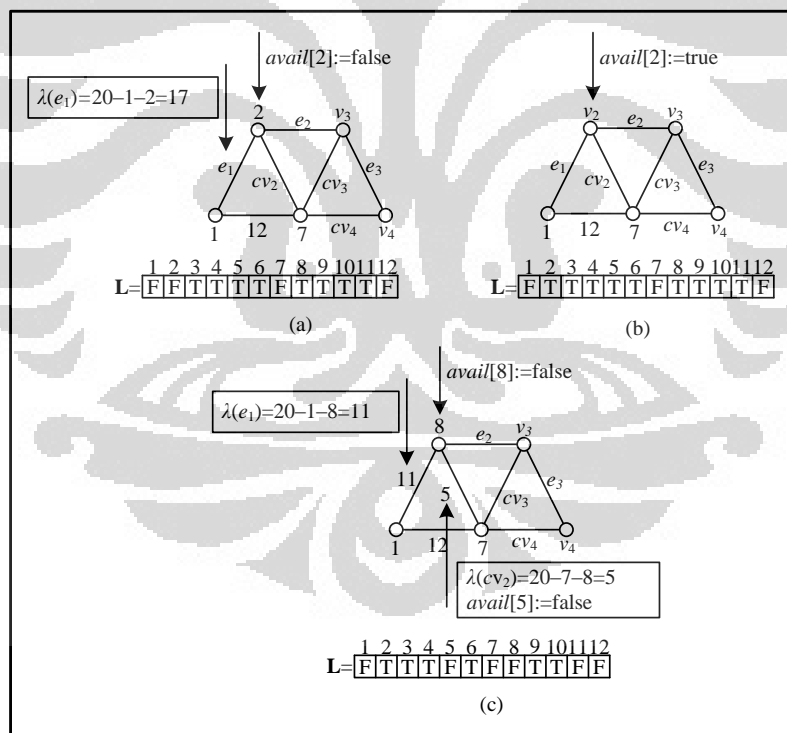
Gambar 3. 23 Contoh penerapan algoritma PTBA pada graf f_4 dengan $k = 20$ pada fungsi **initializeFan**, melabel c, v_1 (a) dan cv_1 (b)

Berikut ini akan diberikan penjelasan jalannya algoritma dengan menggunakan contoh. Misalkan akan dicari PTBA pada graf kipas f_4 . Graf f_4 memiliki 5 simpul dan 7 busur. Dari (2.10) diperoleh batas k untuk f_4 adalah $14.3 \leq k \leq 24.7$. Misalkan dipilih $k = 20$. Jadi yang menjadi masukan pada algoritma adalah $n = 4$ dan $k = 20$. Label yang tersedia adalah $1, 2, 3, \dots, 11, 12$. Algoritma bekerja di-

mulai pada tahap inisialisasi dengan memanggil fungsi **initializeFan** (Algoritma 2.i).

Pertama dilabel simpul c . Misalkan simpul c dilabel 7 sehingga ditandai $avail[7] := false$. Label yang tersedia sekarang adalah 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12. Selanjutnya simpul v_1 dilabel 1, $avail[1] := false$. Setelah simpul c dan v_1 dilabel maka ditentukan besar label busur cv_1 , yaitu sebesar $\lambda(cv_1) = k - \lambda(c) - \lambda(v_1) = 20 - 7 - 1 = 12$. Karena label 12 tersedia maka $\lambda(cv_1) = 12$ diterima, $avail[12] := false$ (Gambar 3.23).

Selanjutnya fungsi **initializeFan** memanggil fungsi **extendFan** dengan parameter $t = 2$ (algoritma 2.i baris 10) untuk melabel v_2 . Karena $t = 2$ maka masuk ke bagian **else** (algoritma 2.ii baris 13) dan karena $n \neq 3$ maka masuk ke bagian **else** (algoritma 2.ii baris 28). Simpul v_2 dilabel 2, $avail[2] := false$. Label yang tersedia adalah 3, 4, 5, 6, 8, 9, 10, 11.

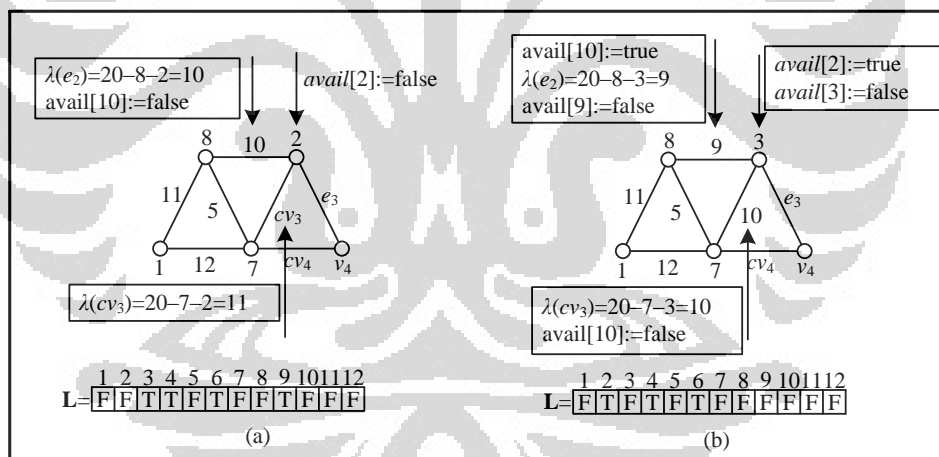


Gambar 3. 24 Contoh penerapan algoritma PTBA pada graf f_4 dengan $k = 20$ pada fungsi **extendFan**, melabel v_2 (a), e_1 (b) dan cv_2 (c)

Lalu ditentukan label busur e_1 sebesar $\lambda(e_1) = k - \lambda(v_1) - \lambda(v_2) = 20 - 1 - 2 = 17$. Karena label busur e_1 yang diperoleh tidak tersedia, maka elemen graf yang

terakhir dilabel, yaitu simpul v_2 akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_1 diperoleh. Sehingga diperoleh $\lambda(v_2) = 8$ dan $\lambda(e_1) = 11$. Kemudian ditentukan label busur cv_2 sebesar $\lambda(cv_2) = k - \lambda(c) - \lambda(v_2) = 20 - 7 - 8 = 5$. Karena besar label cv_2 yang telah ditentukan tersedia maka $\lambda(cv_2) = 5$ diterima (Gambar 3.24). Sekarang label yang tersedia adalah 2, 3, 4, 6, 9, 10.

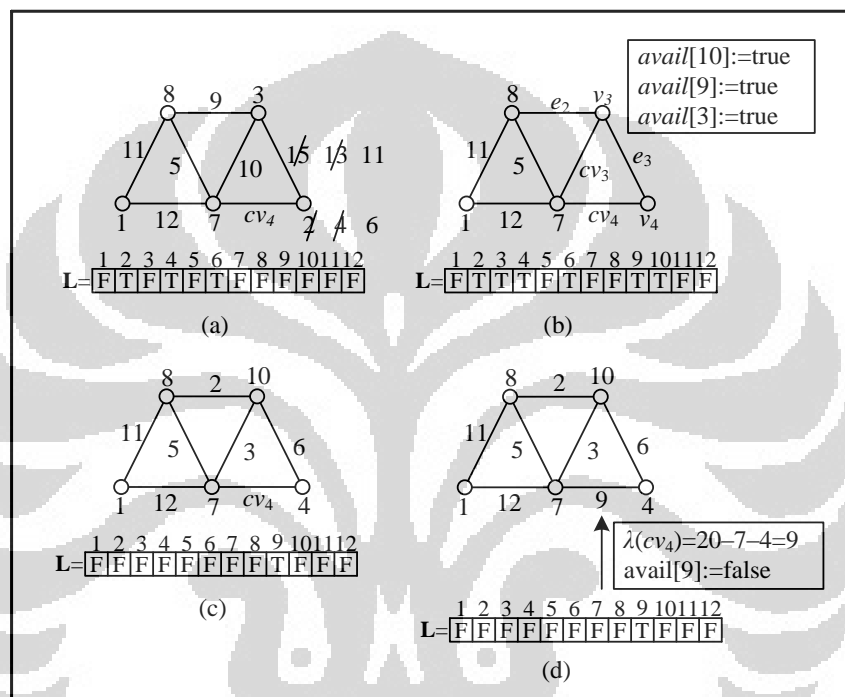
Kemudian fungsi **extendFan** memanggil dirinya sendiri dengan parameter $t = 3$ sehingga masuk ke algoritma 2.ii baris 38. Simpul v_3 dilabel 2 lalu ditandai dengan $avail[2] := false$. Label busur e_2 ditentukan sebesar $\lambda(e_2) = k - \lambda(v_2) - \lambda(v_3) = 20 - 8 - 2 = 10$. Karena label yang diperoleh tersedia maka $\lambda(e_2) = 10$ diterima. Lalu label cv_3 ditentukan sebesar $\lambda(cv_3) = k - \lambda(c) - \lambda(v_3) = 20 - 7 - 2 = 11$. Label 11 tidak tersedia sehingga simpul v_3 akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_2 dan cv_3 diperoleh. Sehingga $avail[10] := true$, $avail[2] := true$ dan diperoleh $\lambda(v_3) = 3$, $\lambda(e_2) = 9$, $\lambda(cv_3) = 10$ (Gambar 3.25). Label yang tersedia adalah 2, 4, 6.



Gambar 3. 25 Contoh penerapan algoritma PTBA pada graf f_4 dengan $k = 20$ pada fungsi **extendFan**, melabel v_3 , e_2 dan cv_3

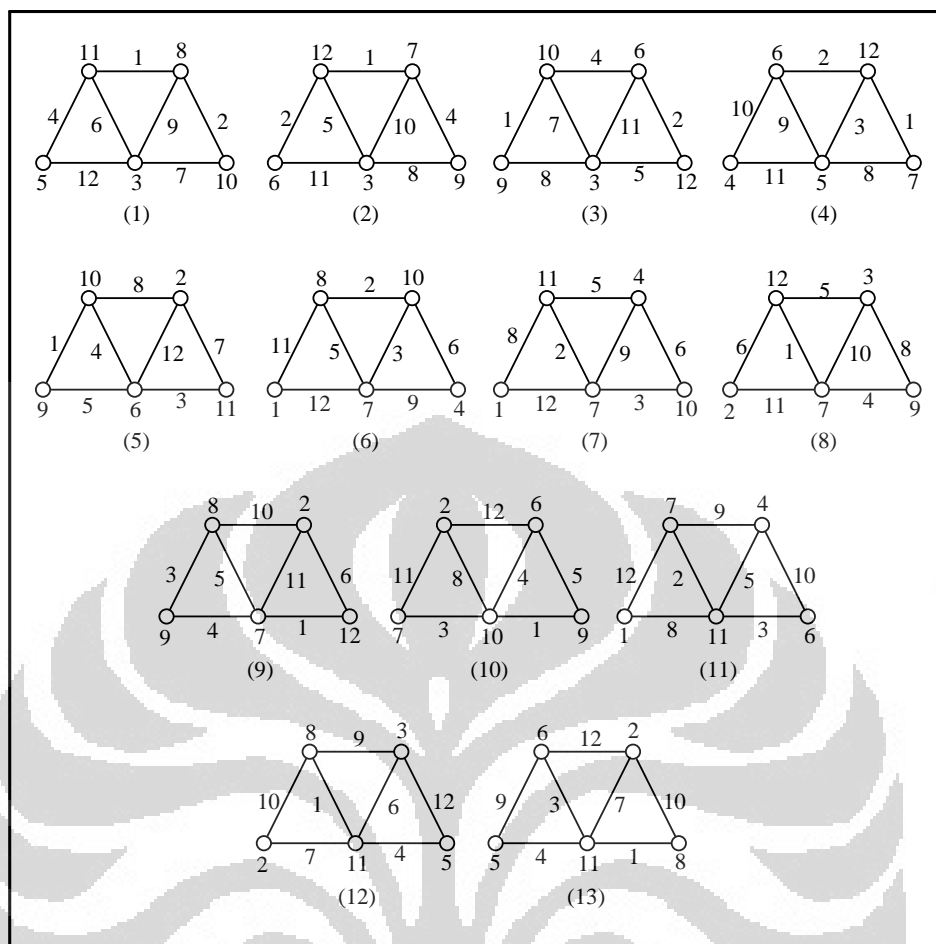
Selanjutnya fungsi **extendFan** memanggil dirinya sendiri dengan parameter $t = 4$ sehingga memenuhi kondisi **if** (algoritma 2.ii baris 1). Simpul v_4 dilabel 2, $avail[2] := false$. Label busur e_3 ditentukan sebesar $\lambda(e_3) = k - \lambda(v_3) - \lambda(v_4) = 20 - 3 - 2 = 15$. Label busur e_3 yang diperoleh tidak tersedia, maka simpul v_4 akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_3 diperoleh.

Karena label busur e_3 tetap tidak diperoleh (Gambar 3.26(a)), maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_3 . Berarti label yang tersedia sekarang adalah 2, 3, 4, 6, 9, 10 (Gambar 3.26(b)). Hal ini dilakukan sampai diperoleh label busur e_3 yang sesuai dengan label yang tersedia. Sehingga diperoleh $\lambda(v_3) = 10$, $\lambda(e_2) = 2$, $\lambda(cv_3) = 3$, $\lambda(v_4) = 4$, dan $\lambda(e_3) = 6$ (Gambar 3.26(c)).



Gambar 3. 26 Contoh penerapan algoritma PTBA pada graf f_4 dengan $k = 20$ pada fungsi `extendFan` melabel v_4, e_3 (a), *backtracking* ke v_3 (b), melabel v_3, e_2, cv_3, v_4, e_3 (c), dan melabel cv_4 (d)

Lalu ditentukan label busur terakhir, yaitu busur cv_4 sebesar $\lambda(cv_4) = k - \lambda(c) - \lambda(v_4) = 20 - 7 - 4 = 9$. Ternyata nilai ini sama dengan label yang masih tersedia maka diperoleh sebuah PTBA untuk graf kipas (Gambar 3.26(d)). Kemudian dipanggil fungsi `print()` (algoritma 2.ii baris 10) untuk mencetak label-label yang diperoleh, yaitu $\lambda(c) = 7$, $\lambda(v_1) = 1$, $\lambda(v_2) = 8$, $\lambda(v_3) = 10$, $\lambda(v_4) = 4$, $\lambda(e_1) = 11$, $\lambda(e_2) = 2$, $\lambda(e_3) = 6$, $\lambda(cv_1) = 12$, $\lambda(cv_2) = 5$, $\lambda(cv_3) = 3$, $\lambda(cv_4) = 9$. Algoritma ini akan berlanjut untuk mencari PTBA yang lainnya.



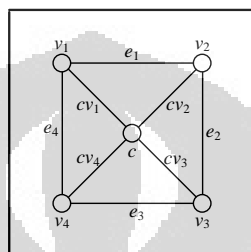
Gambar 3. 27 PTBA yang berbeda pada graf f_4 dengan $k = 20$

Proses selanjutnya simpul v_n akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_{n-1} dan cv_n diperoleh. Ini berarti diperoleh PTBA yang lain. Jika label kedua busur tersebut tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_{n-1} dan seterusnya sesuai dengan urutan langkah-langkah yang telah dilakukan sebelumnya. Hal ini dilakukan sampai diperoleh semua PTBA yang mungkin dan berbeda. Untuk graf f_4 dengan $k = 20$ diperoleh 13 PTBA yang berbeda (Gambar 3.27).

Pada Subbab 3.3 akan dijelaskan mengenai pembentukan algoritma PTBA pada graf roda, syarat-syarat yang harus dipenuhi untuk menghindari pelabelan isomorfik, beserta contohnya.

3.3 Algoritma PTBA pada Graf Roda W_n

Graf roda, W_n ($n \geq 3$) adalah suatu graf yang diperoleh dari suatu graf lingkaran C_n dengan menghubungkan semua simpul ke satu simpul lain yang disebut simpul pusat (*center*). Jadi W_n terdiri dari $n + 1$ simpul dan $2n$ busur. Graf roda W_4 yang diperoleh dengan menghubungkan semua simpul dari C_4 ke simpul pusat beserta penamaan simpul dan busurnya terdapat pada Gambar 3.28.



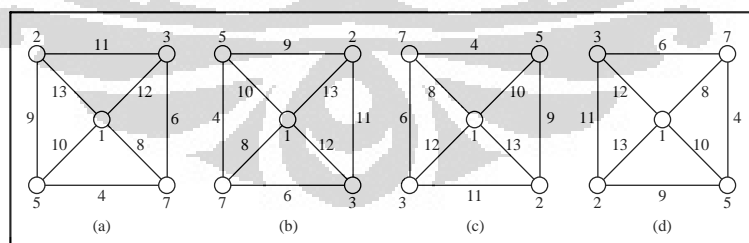
Gambar 3. 28 Graf roda dengan 5 simpul (W_4)

Pada Subbab 2.3 telah diperoleh batasan konstanta ajaib k untuk graf roda, yaitu

$$\frac{11n+17}{4} \leq k \leq \frac{25n+7}{4}.$$

Sebagai contoh batas k yang mungkin untuk graf W_4 adalah $15.25 \leq k \leq 26.75$.

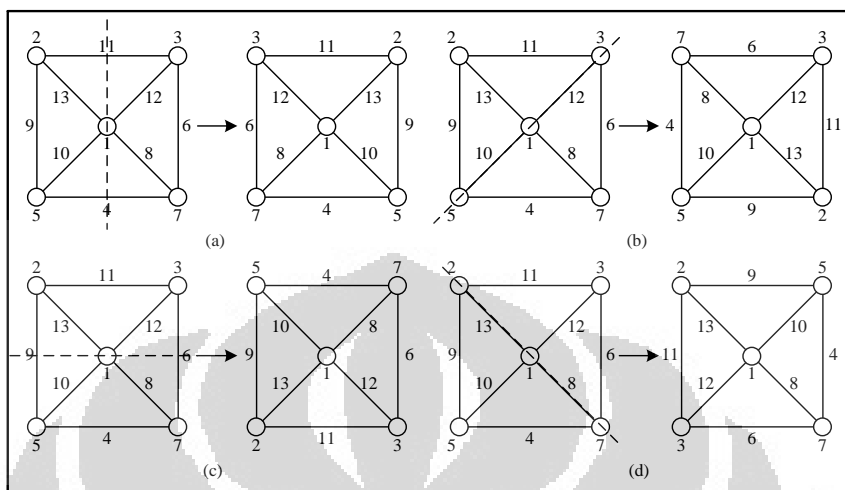
Konstanta ajaib (k) untuk graf W_4 adalah semua bilangan asli yang terdapat pada batasan tersebut.



Gambar 3. 29 Kasus rotasi pada graf W_4

Algoritma PTBA pada graf roda secara garis besar sama dengan algoritma PTBA pada graf kipas. Algoritma ini melabel simpul dan busur graf roda dengan label yang tersedia tepat satu kali secara iteratif. Sehingga setiap busur dari graf

roda memiliki bobot yang sama dan menghasilkan semua PTBA yang mungkin (jika ada) yang tidak isomorfik.

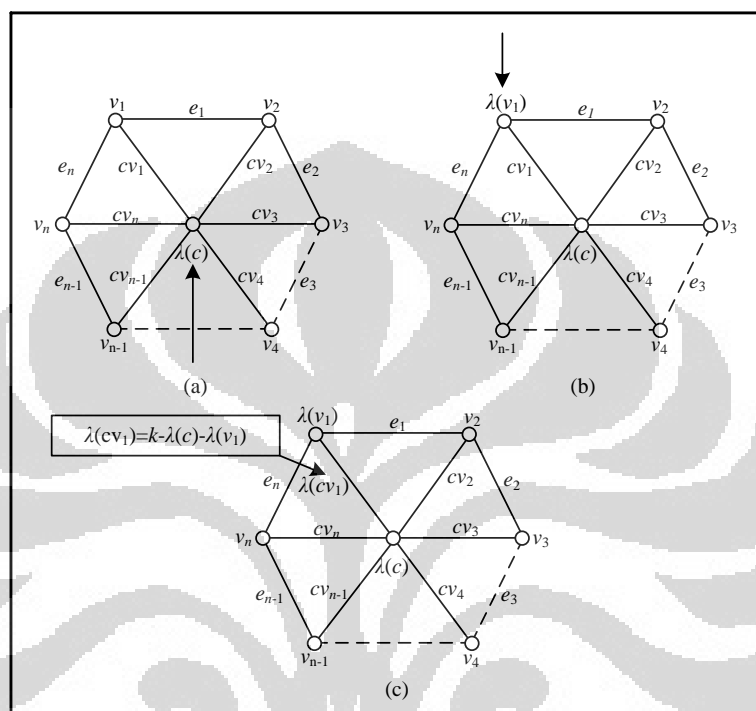


Gambar 3. 30 Kasus refleksi pada graf W_4

Seperti yang dijelaskan sebelumnya, agar diperoleh PTBA yang tidak isomorfik perlu dihindari kasus rotasi dan kasus refleksi. Kasus rotasi terjadi pada graf roda. Pada Gambar 3.29 diberikan contoh kasus rotasi pada graf W_4 . Pelabelan pada Gambar 3.29 (b), (c), (d) diperoleh dengan merotasi pelabelan pada Gambar 3.29 (a). Jadi keempat pelabelan tersebut sama. Secara umum kasus rotasi pada graf roda dapat dihindari dengan menyaratkan untuk semua simpul selain simpul pusat, salah satu simpulnya dilabel lebih kecil dari simpul yang lainnya. Misalkan dipilih simpul v_1 yang memiliki label terkecil ($\lambda(v_1) < \lambda(v_t), t = 2, 3, \dots, n$), maka label terbesar untuk v_1 adalah $2n + 2$, yaitu ketika busur-busur diberi label $1, 2, \dots, 2n$ dan simpul pusat dilabel $2n + 1$.

Kasus refleksi juga terjadi pada graf roda. Pada Gambar 3.30 diberikan contoh kasus refleksi pada graf W_4 dimana garis putus-putus merupakan sumbu refleksi. Pada Gambar tersebut, pelabelan di sebelah kanan merupakan hasil refleksi pelabelan di sebelah kiri terhadap sumbu refleksi. Kasus refleksi pada Gambar 3.30 (a), (b), (c) sumbu refleksinya tidak melalui simpul v_1 . Sehingga adanya syarat bahwa label simpul v_1 lebih kecil dari label simpul lainnya selain simpul pusat yang diberikan untuk menghindari kasus rotasi, juga telah menghindari kasus ini. Sedangkan sumbu refleksi dari kasus refleksi pada Gambar

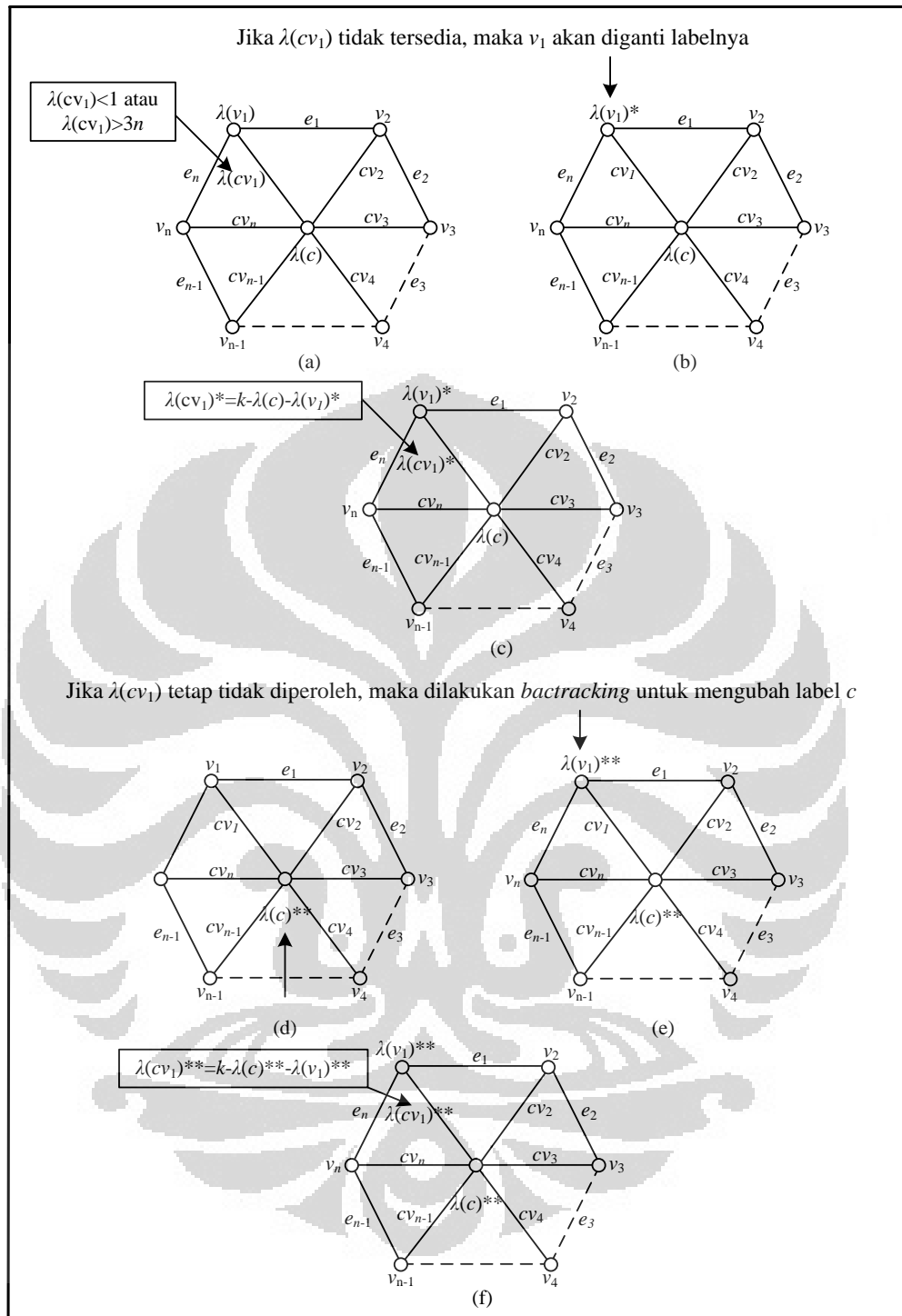
3.30(d) melalui simpul v_1 sehingga perlu diberikan syarat yang baru. Secara umum kasus ini dapat dihindari dengan menyaratkan label salah satu simpul di salah satu sisi dari sumbu refleksi harus lebih kecil (atau lebih besar) dari label salah satu simpul pada sisi yang lain. Sehingga diberi syarat $\lambda(v_2) < \lambda(v_n)$.



Gambar 3. 31 Tahap pelabelan c (a), v_1 (b), dan cv_1 (c)

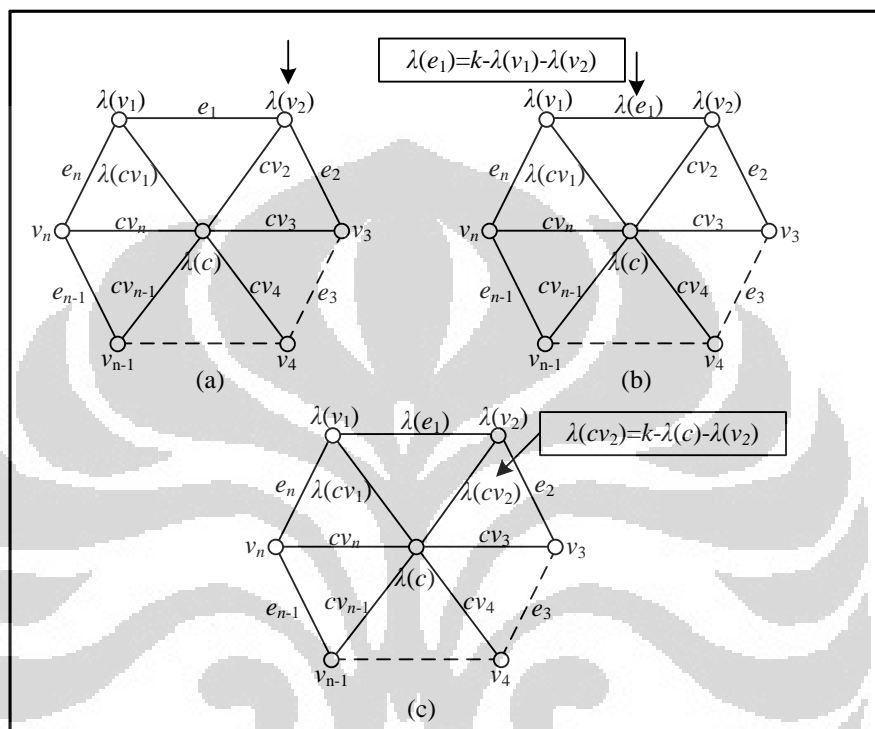
Seperti yang telah dijelaskan sebelumnya, secara garis besar algoritma PTBA pada graf roda sama dengan algoritma PTBA pada graf kipas. Ide dari algoritma ini sama dengan algoritma PTBA pada graf kipas, pada bagian akhir ditambahkan satu langkah untuk melabel busur e_n . Masukan dari algoritma ini adalah banyaknya simpul selain simpul pusat pada graf roda n dan konstanta ajaib k . Label yang tersedia adalah 1 sampai $3n + 1$.

Sama seperti ketika memberikan PTBA pada graf lingkaran dan kipas, dalam melabel graf roda minimal 2 simpul yang diberi label terlebih dahulu. Pemberian label dimulai dengan melabel simpul c , lalu simpul v_1 . Label terbesar untuk simpul v_1 adalah $2n + 2$ seperti yang telah dijelaskan sebelumnya. Kemudian ditentukan besar label cv_1 , yaitu $\lambda(cv_1) = k - \lambda(c) - \lambda(v_1)$. Tahap ini diperlihatkan pada Gambar 3.31.



Jika label busur cv_1 yang diperoleh tidak tersedia, maka elemen graf yang terakhir dilabel, yaitu simpul v_1 akan diganti labelnya dengan label lain yang tersedia, sampai label busur cv_1 diperoleh. Jika label busur cv_1 tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel

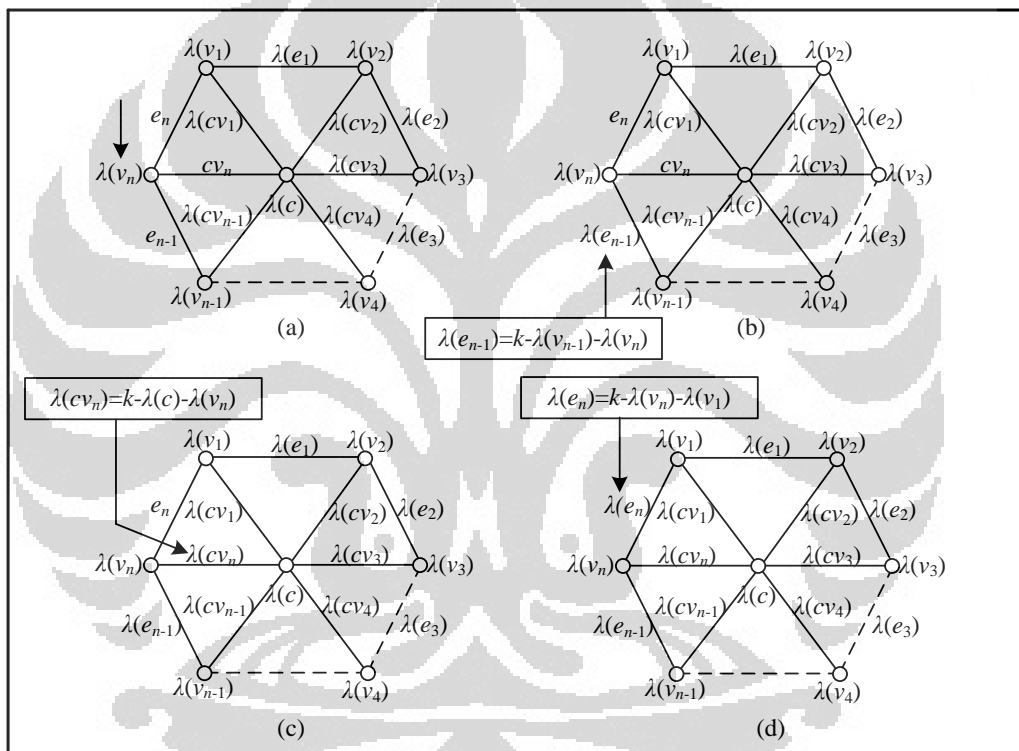
sebelumnya, yaitu simpul c . Hal ini dilakukan sampai diperoleh label busur cv_1 yang sesuai dengan label yang tersedia. Tahap ini diperlihatkan pada Gambar 3.32. Label i yang telah digunakan ditandai dengan $avail[i] := \text{false}$, jika label i yang telah digunakan disediakan kembali ditandai dengan $avail[i] := \text{true}$.



Gambar 3.33 Tahap pelabelan v_2 (a), e_1 (b), dan cv_2 (c)

Setelah diperoleh label busur cv_1 , selanjutnya melabel v_2 . Kemudian ditentukan label dari e_1 , yaitu $\lambda(e_1) = k - \lambda(v_1) - \lambda(v_2)$. Jika label busur e_1 tidak tersedia maka dilakukan hal yang sama seperti saat label busur cv_1 tidak tersedia, yaitu simpul v_2 akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_1 diperoleh. Jika label busur e_1 tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_1 . Hal ini dilakukan sampai diperoleh label busur e_1 yang sesuai dengan label yang tersedia. Setelah label busur e_1 tersedia, ditentukan label busur cv_2 sebesar $\lambda(cv_2) = k - \lambda(c) - \lambda(v_2)$. Jika label busur cv_2 tidak tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia (Gambar 3.33).

Kemudian dengan cara yang sama dilanjutkan melabel v_3, e_2, cv_3 , dst. Di-karenakan langkah melabel simpul dan busur setelah c, v_1 , dan cv_1 sama maka bisa dijadikan sebuah fungsi. Sehingga pelabelan c, v_1 , dan cv_1 di fungsi awal disebut fungsi **initializeWheel** yang akan memanggil fungsi selanjutnya yang disebut fungsi **extendWheel** dengan parameter $t = 2$ untuk melabel simpul v_2 , busur e_1 dan cv_2 . Selanjutnya untuk melabel simpul v_3 , busur e_2 dan cv_3 fungsi **extendWheel** akan memanggil dirinya sendiri dengan parameter $t = t + 1$, begitu seterusnya.



Gambar 3. 34 Tahap pelabelan v_n (a), e_{n-1} (b), cv_n (c), dan e_n (d)

Disaat $t = n$ tersisa 1 simpul dan 3 busur yang belum dilabel yaitu v_n, e_{n-1}, cv_n, e_n . Saat melabel simpul v_n (Gambar 3.34(a)) perlu diberi syarat $\lambda(v_n) > \lambda(v_2)$ agar tidak terjadi kasus refleksi. Oleh karena itu diperlukan kondisi **if** untuk $t = n$. Setelah simpul v_n dilabel, ditentukan label e_{n-1} sebesar $\lambda(e_{n-1}) = k - \lambda(v_{n-1}) - \lambda(v_n)$ (Gambar 3.34(b)). Apabila label busur e_{n-1} tidak tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia. Hal ini dilakukan sampai diperoleh label busur e_{n-1} yang sesuai dengan label yang tersedia.

Lalu ditentukan label cv_n , yaitu $\lambda(cv_n) = k - \lambda(c) - \lambda(v_n)$ (Gambar 3.34(c)). Jika label busur cv_n tidak tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia. Hal ini dilakukan sampai diperoleh label busur cv_n yang sesuai dengan label yang tersedia. Kemudian ditentukan label e_n , yaitu $\lambda(e_n) = k - \lambda(v_n) - \lambda(v_1)$ (Gambar 3.34(d)). Jika nilai ini tidak sama dengan label yang masih tersedia maka dilakukan hal yang sama seperti saat label busur e_1 tidak tersedia. Hal ini dilakukan sampai diperoleh label busur e_n yang sesuai dengan label yang tersedia. Jika label busur e_n yang diperoleh sama dengan label yang masih tersedia maka diperoleh satu PTBA untuk graf roda.

PTBA yang diperoleh dicetak dengan memanggil fungsi **print()**. Walaupun PTBA untuk graf roda sudah ditemukan, algoritma ini akan berlanjut untuk mencari PTBA yang lainnya. Proses selanjutnya simpul v_n akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_{n-1} , cv_n , dan e_n diperoleh sehingga diperoleh PTBA yang lain. Jika label ketiga busur tersebut tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_{n-1} dan seterusnya sesuai dengan urutan langkah-langkah yang telah dilakukan sebelumnya untuk memperoleh PTBA yang lainnya. Algoritma ini akan berhenti apabila telah diperoleh semua PTBA yang mungkin dan berbeda pada graf roda.

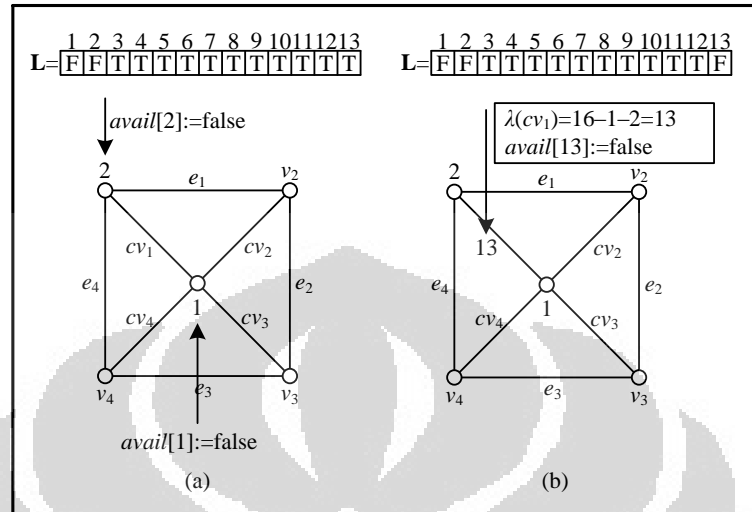
Jadi algoritma PTBA pada graf roda terdiri dari 2 fungsi yaitu fungsi **initializeWheel** (Algoritma 3.i) dan fungsi **extendWheel** (Algoritma 3.ii). Keluaran yang diperoleh adalah semua PTBA yang mungkin dan berbeda untuk graf roda tersebut.

Berikut ini akan diberikan penjelasan jalannya algoritma dengan menggunakan contoh. Misalkan akan dicari PTBA pada graf roda W_4 yang memiliki 5 simpul dan 8 busur. Dari (2.14) yang telah dijelaskan sebelumnya, diperoleh batas k untuk W_4 adalah $15.25 \leq k \leq 26.75$. Misalkan dipilih $k = 16$. Jadi yang menjadi masukan pada algoritma adalah $n = 4$ dan $k = 16$. Label yang tersedia adalah 1, 2, 3, ..., 12, 13. Algoritma mulai bekerja pada tahap inisialisasi dengan memanggil fungsi **initializeWheel** (Algoritma 3.i).

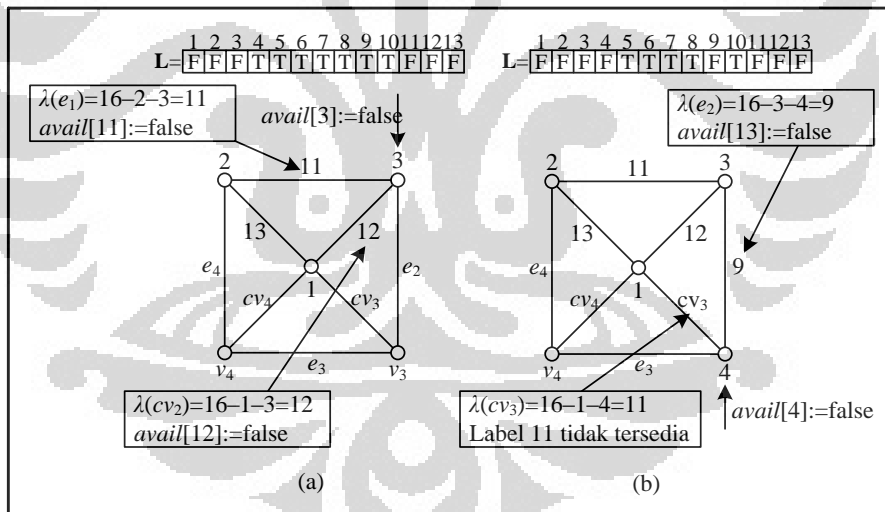
Algorithm 3.i	Algorithm 3.ii
<pre> initializeWheel 1 for each available label i do 2 $\lambda(c) := i$ 3 $avail[i] := \text{false}$ 4 for each available label j where $j \leq 2n+2$ do 5 $\lambda(v_1) := j$ 6 $avail[j] := \text{false}$ 7 $\lambda(cv_1) := k - \lambda(c) - \lambda(v_1)$ 8 if $0 < \lambda(cv_1) \leq 3n$ and $avail[\lambda(cv_1)]$ then 9 $avail[\lambda(cv_1)] := \text{false}$ 10 extendWheel(2) 11 $avail[\lambda(cv_1)] := \text{true}$ 12 end if 13 $avail[j] := \text{true}$ 14 end for 15 $avail[i] := \text{true}$ 16 end for </pre>	<pre> extendWheel(t) 1 if $t = n$ 2 for each available label i where $i > \lambda(v_1)$ and 3 $i > \lambda(v_2)$ do 4 $\lambda(v_n) := i$ 5 $avail[i] := \text{false}$ 6 $\lambda(e_{n-1}) := k - \lambda(v_{n-1}) - \lambda(v_n)$ 7 if $0 < \lambda(e_{n-1}) \leq 3n$ and $avail[\lambda(e_{n-1})]$ 8 then 9 $avail[\lambda(e_{n-1})] := \text{false}$ 10 $\lambda(cv_n) := k - \lambda(c) - \lambda(v_n)$ 11 if $0 < \lambda(cv_n) \leq 3n$ and $avail[\lambda(cv_n)]$ 12 then 13 $avail[\lambda(cv_n)] := \text{false}$ 14 $\lambda(e_n) := k - \lambda(v_n) - \lambda(v_1)$ 15 if $0 < \lambda(e_n) \leq 3n$ and $avail[\lambda(e_n)]$ 16 then 17 print() 18 $avail[\lambda(cv_n)] := \text{true}$ 19 end if 20 $avail[\lambda(e_{n-1})] := \text{true}$ 21 end if 22 $avail[i] := \text{true}$ 23 end for 24 else 25 for each available label i where $i > \lambda(v_1)$ do 26 $\lambda(v_t) := i$ 27 $avail[i] := \text{false}$ 28 $\lambda(e_{t-1}) := k - \lambda(v_{t-1}) - \lambda(v_t)$ 29 if $0 < \lambda(e_{t-1}) \leq 3n$ and $avail[\lambda(e_{t-1})]$ then 30 $avail[\lambda(e_{t-1})] := \text{false}$ 31 $\lambda(cv_t) := k - \lambda(c) - \lambda(v_t)$ 32 if $0 < \lambda(cv_t) \leq 3n$ and $avail[\lambda(cv_t)]$ 33 then 34 $avail[\lambda(cv_t)] := \text{false}$ 35 extendWheel(t + 1) 36 $avail[\lambda(cv_t)] := \text{true}$ 37 end if 38 $avail[\lambda(e_{t-1})] := \text{true}$ 39 end if 40 $avail[i] := \text{true}$ 41 end for 42 end if </pre>

Dimulai dengan melabel simpul c . Misalkan simpul c dilabel 1 sehingga ditandai $avail[1] := \text{false}$. Label yang tersedia sekarang adalah 2, 3, ..., 12, 13. Selanjutnya dilabel simpul v_1 dimana $\lambda(v_1) \leq 2n + 2 = 10$. Misalkan simpul v_1 dilabel 2 sehingga $avail[2] := \text{false}$. Setelah simpul c dan v_1 dilabel maka ditentukan besar label busur cv_1 , yaitu sebesar $\lambda(cv_1) = k - \lambda(c) - \lambda(v_1) = 16 - 1 - 2 = 13$.

Karena label 13 tersedia maka $\lambda(cv_1) = 13$ diterima, $avail[13] := false$ (Gambar 3.35).



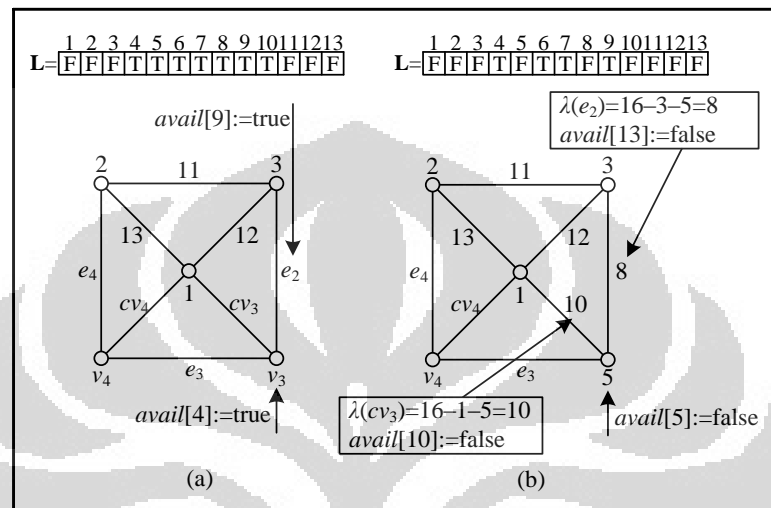
Gambar 3. 35 Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi initializeWheel, melabel c, v_1 , (a) dan cv_1 (b)



Gambar 3. 36 Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi extendWheel, melabel v_2, e_1, cv_2 (a) dan v_3, e_2, cv_3 (b)

Selanjutnya fungsi **initializeWheel** memanggil fungsi **extendWheel** dengan parameter $t = 2$ (algoritma 3.i baris 10) untuk melabel v_2 . Karena $t = 2$ maka masuk ke bagian **else** (algoritma 3.ii baris 17). Simpul v_2 dilabel dimana $\lambda(v_2) > \lambda(v_1) = 2$ untuk menghindari kasus rotasi. Sehingga simpul v_2 dilabel 3, $avail[3] := false$. Label yang tersedia adalah 4, 5, ..., 11, 12. Lalu ditentukan label

busur e_1 sebesar $\lambda(e_1) = k - \lambda(v_1) - \lambda(v_2) = 16 - 2 - 3 = 11$. Karena label busur e_1 yang diperoleh tersedia maka $\lambda(e_1)=11$ diterima. Kemudian ditentukan label busur cv_2 sebesar $\lambda(cv_2) = k - \lambda(c) - \lambda(v_2) = 16 - 1 - 3 = 12$. Dikarenakan label busur cv_2 yang diperoleh tersedia maka $\lambda(cv_2) = 12$ diterima (Gambar 3.36(a)).

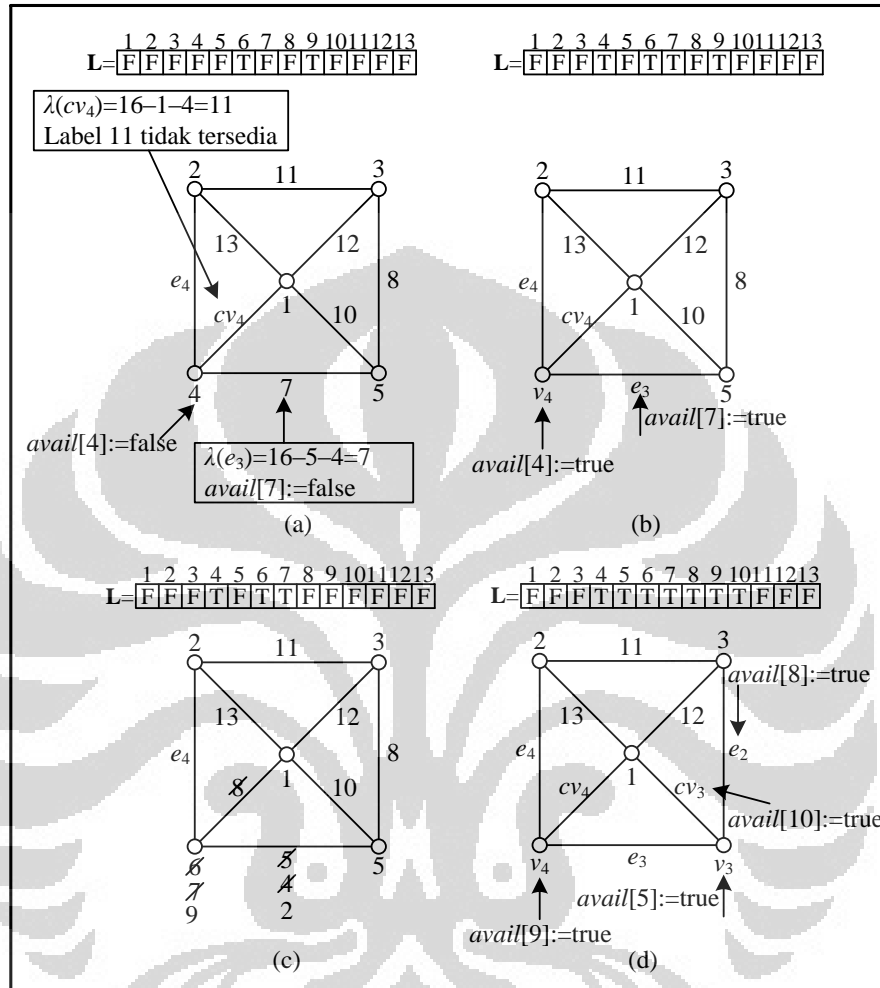


Gambar 3. 37 Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi `extendWheel`, penggantian label v_3 (a) dan melabel v_3, e_2, cv_3 dengan label yang lainnya (b)

Setelah itu fungsi `extendWheel` memanggil dirinya sendiri dengan parameter $t = 3$ (algoritma 3.ii baris 27) sehingga masuk ke algoritma 3.ii baris 17. Dengan cara yang sama diperoleh $\lambda(v_3) = 4$ dan $\lambda(e_2) = 9$. Lalu ditentukan label busur cv_3 sebesar $\lambda(cv_3) = k - \lambda(c) - \lambda(v_3) = 16 - 1 - 4 = 11$ (Gambar 3.36(b)). Karena label busur cv_3 yang diperoleh tidak tersedia, maka elemen graf yang terakhir dilabel, yaitu simpul v_3 akan diganti labelnya dengan label lain yang tersedia, sampai label busur cv_3 diperoleh. Sehingga diperoleh $\lambda(v_3) = 5, \lambda(e_2) = 8$, dan $\lambda(cv_3) = 10$ (Gambar 3.37(b)). Label yang tersedia sekarang adalah 4, 6, 7, 9.

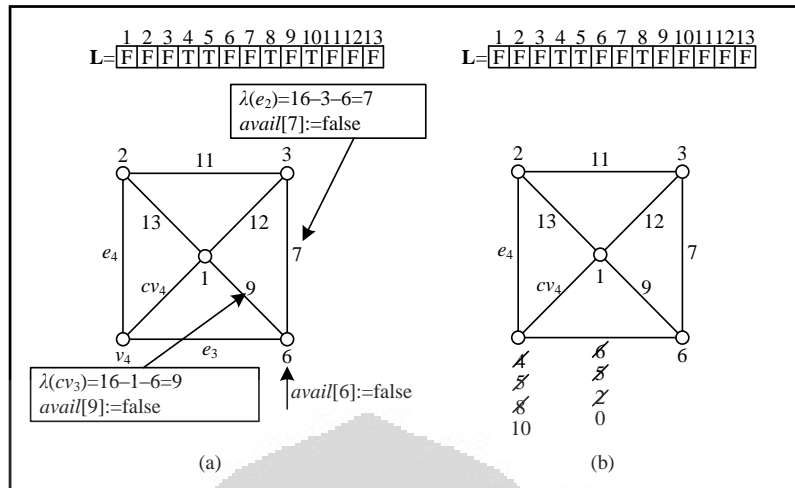
Selanjutnya fungsi `extendWheel` memanggil dirinya sendiri dengan parameter $t = 4$ sehingga memenuhi kondisi `if` (algoritma 3.b baris 1). Dilabel simpul v_4 dimana $\lambda(v_4) > \lambda(v_1) = 2$ untuk menghindari kasus rotasi dan $\lambda(v_4) > \lambda(v_2) = 3$ untuk menghindari kasus refleksi. Sehingga simpul v_4 dilabel 4 dimana $4 > 2$ dan $4 > 3, avail[4] := false$. Label busur e_3 ditentukan sebesar $\lambda(e_3) = k - \lambda(v_3) - \lambda(v_4) = 16 - 5 - 4 = 7$. Label busur e_3 yang diperoleh tersedia maka $\lambda(e_3) = 7$ diterima. La-

lu ditentukan label busur cv_4 sebesar $\lambda(cv_4) = k - \lambda(c) - \lambda(v_4) = 16 - 1 - 4 = 11$ (Gambar 3.38(a)).



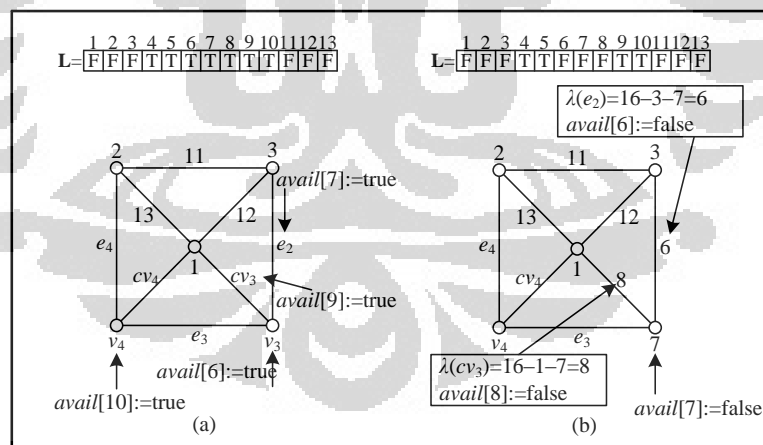
Gambar 3. 38 Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi `extendWheel`, melabel v_4 , e_3, cv_4 (a), penggantian label v_4 (b), melabel v_4 , e_3, cv_4 (c), dan `backtracking` ke v_3 untuk mengganti label simpul tersebut (d)

Label busur cv_4 yang diperoleh tidak tersedia maka elemen graf yang terakhir dilabel, yaitu simpul v_4 akan diganti labelnya dengan label lain yang tersedia, sampai label busur cv_4 diperoleh. Karena label busur cv_4 tetap tidak diperoleh (Gambar 3.38(c)), maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_3 (algoritma 3.ii baris 17). Label yang tersedia sekarang adalah 4, 5, ..., 9, 10 (Gambar 3.38(d)).



Gambar 3. 39 Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi `extendWheel`, melabel v_3, e_2, cv_3 (a) dan v_4, e_3 (b)

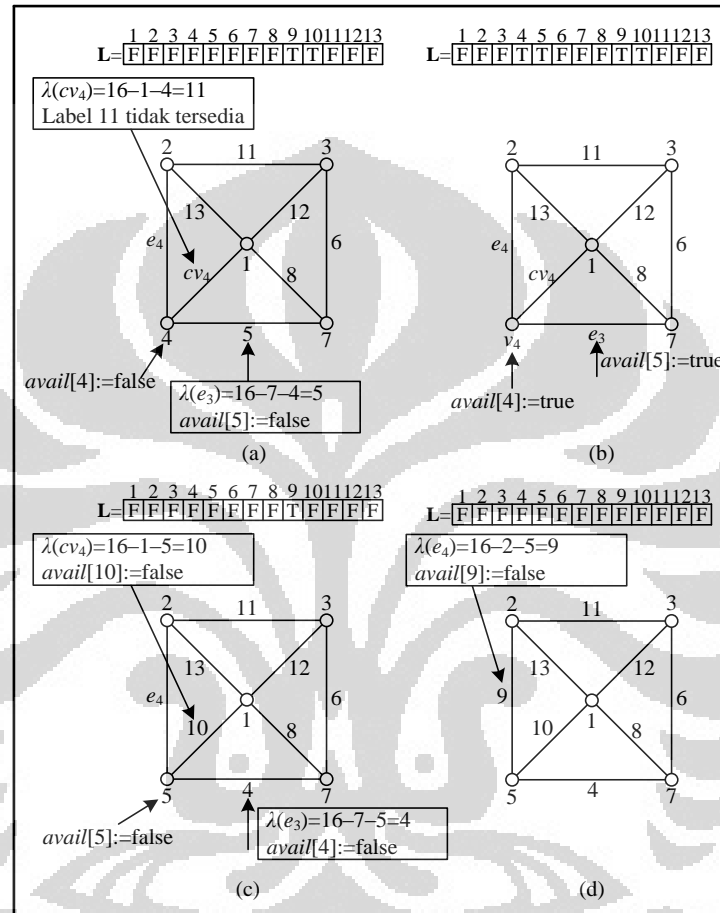
Kemudian dengan cara yang sama diperoleh $\lambda(v_3) = 6$, $\lambda(e_2) = 7$ dan $\lambda(cv_3) = 9$ (Gambar 3.39(a)). Lalu fungsi `extendWheel` memanggil dirinya sendiri dengan parameter $t = 4$ sehingga memenuhi kondisi `if` (algoritma 3.ii baris 1). Simpul v_4 dilabel 4 dimana $4 > 2$ dan $4 > 3$, $avail[4] := false$. Label busur e_3 ditentukan sebesar $\lambda(e_3) = k - \lambda(v_3) - \lambda(v_4) = 16 - 6 - 4 = 6$.



Gambar 3. 40 Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi `extendWheel`, backtracking ke v_3 untuk mengganti label simpul tersebut (a) dan melabel v_3, e_2, cv_3 (b)

Label busur e_3 yang diperoleh tidak tersedia, maka elemen graf yang terakhir dilabel, yaitu simpul v_4 akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_3 diperoleh. Karena label busur e_3 tetap tidak

diperoleh (Gambar 3.39(b)), maka dilakukan *backtracking* untuk mengubah label simpul v_3 (algoritma 3.ii baris 17). Label yang tersedia sekarang adalah 4, 5, ..., 9, 10 (Gambar 3.40(a)). Selanjutnya dengan cara yang sama diperoleh $\lambda(v_3) = 7$, $\lambda(e_2) = 6$ dan $\lambda(cv_3) = 8$ (Gambar 3.40(b)).

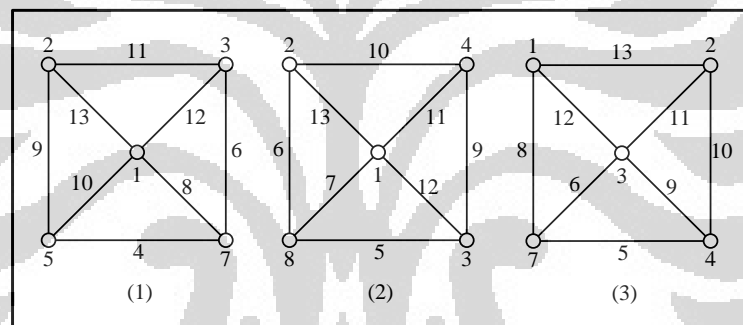


Gambar 3. 41 Contoh penerapan algoritma PTBA pada graf W_4 dengan $k = 16$ pada fungsi *extendWheel*, melabel v_4 , e_3, cv_4 (a), penggantian label v_4 (b), melabel v_4 , e_3, cv_4 (c), dan melabel e_4 (d)

Lalu fungsi **extendWheel** memanggil dirinya sendiri dengan parameter $t = 4$ sehingga memenuhi kondisi **if** (algoritma 3.ii baris 1). Simpul v_4 dilabel 4. Label busur e_3 ditentukan sebesar $\lambda(e_3) = k - \lambda(v_3) - \lambda(v_4) = 16 - 7 - 4 = 5$. Label busur e_3 yang diperoleh tersedia maka $\lambda(e_3) = 5$ diterima. Lalu ditentukan label busur cv_4 sebesar $\lambda(cv_4) = k - \lambda(c) - \lambda(v_4) = 16 - 1 - 4 = 11$ (Gambar 3.41(a)). Label busur cv_4 yang diperoleh tidak tersedia, maka simpul v_4 akan diganti labelnya dengan label lain yang tersedia, sampai label busur cv_4 diperoleh.

Label simpul v_4 kita ganti menjadi 5 dimana $5 > 2$ dan $5 > 3$. Label simpul tersebut sebelumnya disediakan kembali, $avail[4]:=true$. Label busur e_3 ditentukan sebesar $\lambda(e_3) = k - \lambda(v_3) - \lambda(v_4) = 16 - 7 - 5 = 4$. Label busur e_3 yang diperoleh tersedia maka $\lambda(e_3) = 4$ diterima. Lalu ditentukan label busur cv_4 sebesar $\lambda(cv_4) = k - \lambda(c) - \lambda(v_4) = 16 - 1 - 5 = 10$. Label busur cv_4 yang diperoleh tersedia maka $\lambda(cv_4) = 10$ diterima (Gambar 3.41(c)).

Selanjutnya tersisa 1 busur. Label busur e_4 ditentukan sebesar $\lambda(e_4) = k - \lambda(v_4) - \lambda(v_1) = 16 - 5 - 2 = 9$. Ternyata nilai ini sama dengan label yang masih tersedia maka diperoleh sebuah PTBA untuk graf roda (Gambar 3.41(d)). Fungsi **print()** (algoritma 3.b baris 13) dipanggil untuk mencetak label simpul dan busur yang diperoleh, yaitu $\lambda(c) = 1$, $\lambda(v_1) = 2$, $\lambda(v_2) = 3$, $\lambda(v_3) = 7$, $\lambda(v_4) = 5$, $\lambda(e_1) = 11$, $\lambda(e_2) = 6$, $\lambda(e_3) = 4$, $\lambda(e_4) = 9$, $\lambda(cv_1) = 13$, $\lambda(cv_2) = 12$, $\lambda(cv_3) = 8$, $\lambda(cv_4) = 10$.



Gambar 3. 42 PTBA yang berbeda pada graf W_4 dengan $k = 16$

Algoritma ini akan berlanjut untuk mencari PTBA yang lainnya. Proses selanjutnya simpul v_n akan diganti labelnya dengan label lain yang tersedia, sampai label busur e_{n-1} , cv_n , dan e_n diperoleh. Ini berarti diperoleh PTBA yang lain. Jika label kedua busur tersebut tetap tidak diperoleh, maka dilakukan *backtracking* untuk mengubah label elemen graf yang dilabel sebelumnya, yaitu simpul v_{n-1} dan seterusnya sesuai dengan urutan langkah-langkah yang telah dilakukan sebelumnya untuk memperoleh PTBA yang lainnya. Untuk graf W_4 dengan $k = 16$ diperoleh 3 PTBA yang berbeda (Gambar 3.42).

Di bagian selanjutnya akan dijelaskan mengenai implementasi dan simulasi algoritma PTBA pada graf terkait yang dibangun pada Bab 3.

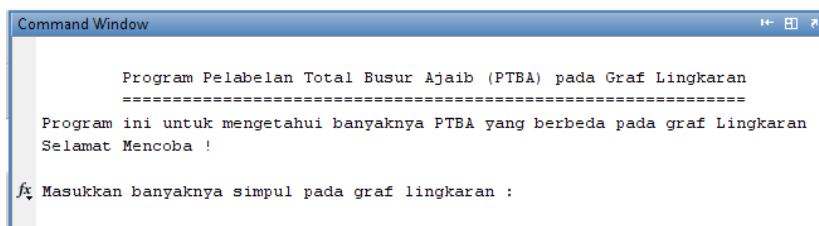
BAB 4 IMPLEMENTASI DAN SIMULASI

Pada Bab 3 telah dibentuk algoritma PTBA pada graf lingkaran, kipas, dan roda. Pada Bab ini akan diberikan implementasi algoritma-algoritma tersebut dalam bentuk program MATLAB. Program dijalankan pada *Personal Computer* dengan *processor Pentium(R)Dual-Core 2.00 GHz 2.00GHz*, *RAM 1.00 GB*, dan sistem operasi *Microsoft Windows 7 Professional*. Program akan menghasilkan semua PTBA yang mungkin yang berbeda pada graf terkait untuk nilai n dan konstanta ajaib (k) yang diberikan. Sebagai hasil implementasi dilakukan simulasi. Namun, simulasi yang dilakukan pada Bab ini adalah banyaknya PTBA yang tidak isomorfik (berbeda) pada graf terkait untuk n dan k yang diberikan. Secara teoritis, ini dapat dijalankan untuk sembarang n tetapi karena keterbatasan waktu simulasi baru dapat dilakukan sampai n tertentu untuk graf terkait. Pada Subbab 4.1 diberikan implementasi dan simulasi algoritma PTBA untuk graf lingkaran C_n , Subbab 4.2 untuk graf kipas f_n , dan Subbab 4.3 untuk graf roda W_n .

4.1 Implementasi dan Simulasi Algoritma PTBA pada Graf Lingkaran C_n

Pada Subbab 3.1 telah dibentuk algoritma PTBA pada graf lingkaran. Algoritma tersebut terdiri dari 2 fungsi yaitu fungsi **initializeCycle** (Algoritma 1.i) dan **extendCycle** (Algoritma 1.ii). Algoritma PTBA pada graf lingkaran akan dijalankan oleh 2 fungsi, yaitu fungsi `cycle` dan `extendcycle`. Fungsi `cycle` akan melakukan tahap-tahap pada Algoritma 1.i, yaitu melabel simpul v_1 , v_2 , dan busur e_1 . Kemudian fungsi `cycle` akan memanggil fungsi `extendcycle`. Fungsi `extendcycle` akan melakukan tahap-tahap pada Algoritma 1.ii. Fungsi `extendcycle` akan melabel 1 simpul dan 1 busur. Lalu fungsi `extendcycle` akan memanggil dirinya sendiri untuk melabel simpul dan busur selanjutnya. Untuk lebih jelas, program dapat dilihat pada Lampiran 3 (CD).

Pemanggilan program dilakukan dengan cara mengetik “`cycle`” pada *Command Window*. Tampilan yang pertama kali muncul pada *Command Window* adalah seperti Gambar 4.1.



```

Command Window

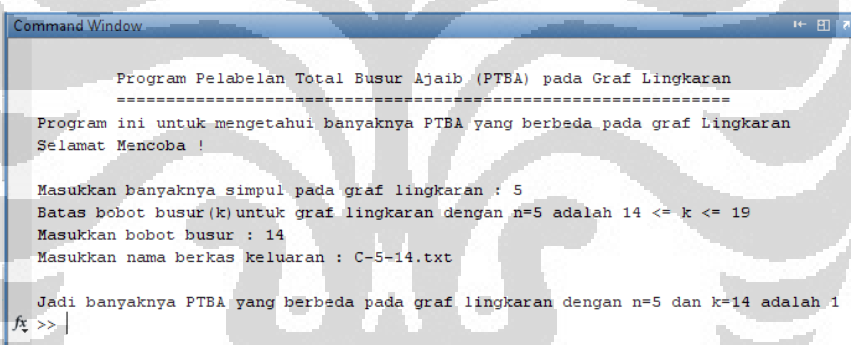
Program Pelabelan Total Busur Ajaib (PTBA) pada Graf Lingkaran
=====
Program ini untuk mengetahui banyaknya PTBA yang berbeda pada graf Lingkaran
Selamat Mencoba !

fx Masukkan banyaknya simpul pada graf lingkaran :

```

Gambar 4.1 Tampilan pada *Command Window* saat fungsi cycle dipanggil

Pengguna akan diminta memasukkan banyaknya simpul pada graf lingkaran (n), bobot busurnya (k), dan nama berkas keluaran (*file output*). Berkas keluaran digunakan untuk menyimpan keluaran dari program yang berupa label simpul dan busur yang membentuk PTBA pada graf lingkaran serta banyaknya PTBA yang berbeda yang diperoleh, dengan nilai n dan k yang telah dimasukkan oleh pengguna.



```

Command Window

Program Pelabelan Total Busur Ajaib (PTBA) pada Graf Lingkaran
=====
Program ini untuk mengetahui banyaknya PTBA yang berbeda pada graf Lingkaran
Selamat Mencoba !

Masukkan banyaknya simpul pada graf lingkaran : 5
Batas bobot busur (k) untuk graf lingkaran dengan n=5 adalah 14 <= k <= 19
Masukkan bobot busur : 14
Masukkan nama berkas keluaran : C-5-14.txt

Jadi banyaknya PTBA yang berbeda pada graf lingkaran dengan n=5 dan k=14 adalah 1
fx >> |

```

Gambar 4.2 Tampilan *Command Window* saat program dipanggil untuk mencari PTBA pada graf C_5 dengan $k = 14$

Pada Gambar 4.2 akan diberi contoh pemanggilan program untuk mencari PTBA pada graf lingkaran C_5 dengan $n = 5$ dan $k = 14$. Seperti yang dijelaskan sebelumnya, ketik “cycle” untuk memanggil program. Kemudian masukkan banyaknya simpul pada graf lingkaran $n = 5$ dan bobot busurnya $k = 14$. Setelah itu masukkan nama berkas keluaran, misalkan C-5-14.txt.

Pada Gambar 4.2 terlihat bahwa bobot busur untuk graf C_5 adalah bilangan asli dari 14 s.d 19. Banyaknya PTBA yang berbeda pada graf lingkaran dengan $n = 5$ dan $k = 14$ adalah 1. Untuk mengetahui PTBA yang diperoleh, dibuka berkas keluaran C-5-14.txt (Gambar 4.3).

```

=====
PTBA untuk graf lingkaran dengan n=5 dan k=14 adalah :

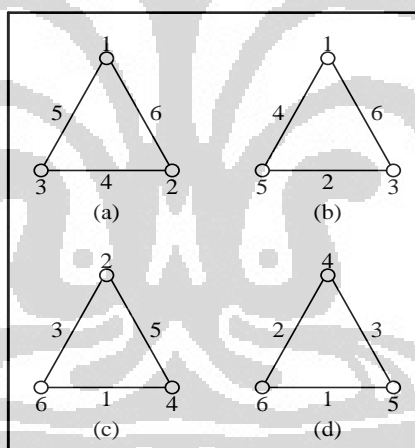
1.  V=[1 3 5 2 4]
    E=[10 6 7 8 9]

Jadi banyaknya PTBA yang berbeda pada graf lingkaran dengan n=5 dan
k=14 adalah 1

```

Gambar 4.3 PTBA yang diperoleh untuk graf C_5 dengan $k = 14$ yang disimpan di berkas keluaran C-5-14.txt

Pada Gambar 4.3 terlihat bahwa diperoleh PTBA untuk graf C_5 dengan $k = 14$. Label-label dinyatakan dalam bentuk matriks baris $V=[1\ 3\ 5\ 2\ 4]$ untuk simpul dan $E=[10\ 6\ 7\ 8\ 9]$ untuk busur. Kolom ke- i menyatakan label dari simpul v_i (busur e_i). Hasil ini sama dengan hasil yang telah diperoleh pada Subbab 3.1.



Gambar 4.4 PTBA pada C_3 dengan $k = 9$ (a), $k = 10$ (b), $k = 11$ (c), dan $k = 12$ (d)

Dengan cara yang sama, dicari PTBA pada C_3 untuk semua kemungkinan nilai k , yaitu $9 \leq k \leq 12$. Hasil yang diperoleh telah digambar dan diperlihatkan pada Gambar 4.4. Pada Gambar 4.4 diberikan PTBA pada C_3 dengan $k = 9$ (a), $k = 10$ (b), $k = 11$ (c), dan $k = 12$ (d). Seperti yang telah dijelaskan pada Bab 2, dalam PTBA terdapat pelabelan dual. Terlihat pada Gambar 4.4 bahwa PTBA pada C_3 , $k = 12$ (d) merupakan pelabelan dual dari PTBA pada C_3 , $k = 9$ (a) (begitu juga sebaliknya) dan PTBA pada C_3 , $k = 11$ (c) merupakan pelabelan dual dari

PTBA pada C_3 , $k = 10$ (b) (begitu juga sebaliknya). Pelabelan dual juga berlaku pada PTBA untuk graf kipas dan roda.

Simulasi program algoritma PTBA pada graf lingkaran dilakukan mulai dari $n = 3$, dimana n merupakan banyaknya simpul pada graf lingkaran. Simulasi baru dapat dilakukan sampai $n = 12$ karena waktu yang terbatas. Semakin besar nilai n maka semakin banyak waktu yang dibutuhkan untuk menjalankan program. Batas konstanta ajaib (k) untuk graf lingkaran dapat diperoleh dari (2.6) pada Subbab 2.3. Hasil dari simulasi yang dilakukan, diberikan pada Tabel 4.1.

Tabel 4.1 Banyaknya PTBA yang berbeda pada graf lingkaran C_3 sampai C_{12} berdasarkan konstanta ajaib k ($\frac{5n+3}{2} \leq k \leq \frac{7n+3}{2}$)

C_3		C_4		C_5		C_6		C_7	
k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA
9	1	12	1	14	1	17	3	19	9
10	1	13	2	15	0	18	1	20	10
11	1	14	2	16	2	19	6	21	11
12	1	15	1	17	2	20	6	22	29
				18	0	21	1	23	29
				19	1	22	3	24	11
								25	10
								26	9
Total=	4	Total=	6	Total=	6	Total=	20	Total=	118
C_8		C_9		C_{10}		C_{11}		C_{12}	
k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA
22	10	24	31	27	125	29	308	32	1602
23	19	25	43	28	236	30	711	33	4111
24	57	26	125	29	698	31	1781	34	10834
25	55	27	264	30	1138	32	3371	35	19183
26	55	28	307	31	1349	33	4945	36	30877
27	57	29	307	32	1349	34	6948	37	36817
28	19	30	264	33	1138	35	6948	38	36817
29	10	31	125	34	698	36	4945	39	30877
		32	43	35	236	37	3371	40	19183
		33	31	36	125	38	1781	41	10834
						39	711	42	4111
						40	308	43	1602
Total=	282	Total=	1540	Total=	7092	Total=	36128	Total=	206848

Tabel 4.1 memberikan banyaknya PTBA yang berbeda pada graf lingkaran C_3 sampai C_{10} berdasarkan konstanta ajaib k . Baris pertama di setiap kolom C_n , menyatakan graf lingkaran dengan ukuran n . Kolom k adalah konstanta ajaib yang

mungkin yang diperoleh dari (2.6). Kolom # PTBA berisi banyaknya PTBA yang berbeda pada C_n untuk n dan k yang diberikan. Baris terakhir di setiap kolom C_n , menyatakan total banyaknya PTBA yang berbeda pada C_n untuk nilai n yang diberikan.

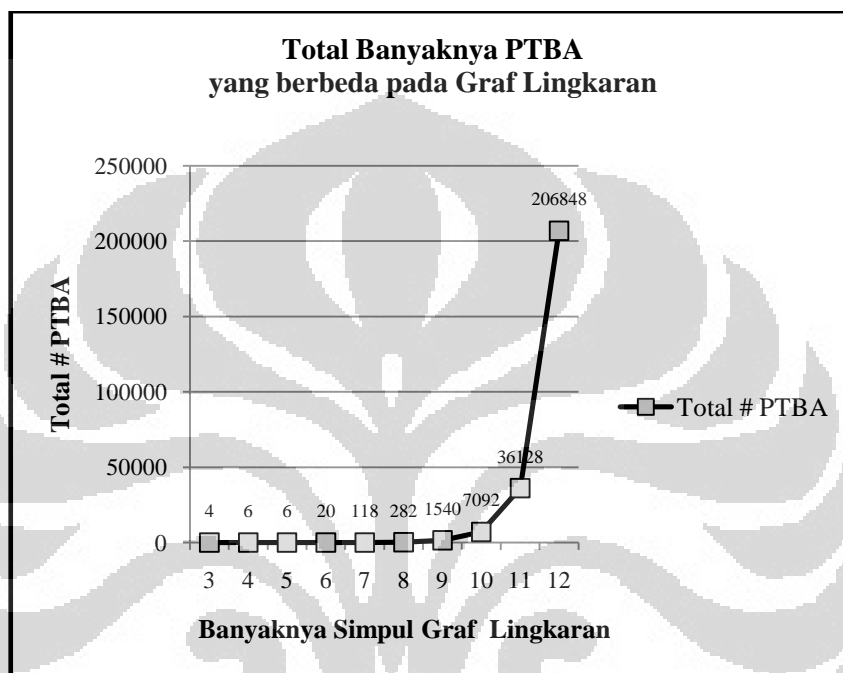
Terlihat pada Tabel 4.1 bahwa di setiap nilai n terdapat PTBA yang berbeda yang banyaknya sama untuk k tertentu. Sebagai contoh, banyaknya PTBA yang berbeda pada $C_4, k = 12$ sama dengan $C_4, k = 15$. Begitu juga pada $C_4, k = 13$ banyaknya PTBA yang berbeda sama dengan $C_4, k = 14$. Ini dikarenakan PTBA pada $C_4, k = 15$ merupakan pelabelan dual dari PTBA pada $C_4, k = 12$ (begitu juga sebaliknya). PTBA pada $C_4, k = 14$ merupakan pelabelan dual dari PTBA pada $C_4, k = 13$ (begitu juga sebaliknya). Hal ini sama seperti pelabelan dual yang telah dijelaskan sebelumnya.

Banyaknya PTBA yang berbeda untuk pelabelan dual diberi warna biru. Walaupun hasilnya sama, program tetap dijalankan untuk pelabelan dual. Ini berguna untuk memverifikasi kebenaran program. Pada graf lingkaran banyaknya konstanta ajaib genap. Dimana setengahnya adalah konstanta ajaib untuk pelabelan dual. Apabila hanya ingin mengetahui total banyaknya PTBA yang berbeda untuk satu nilai n maka program sebenarnya tidak perlu dijalankan untuk nilai k yang menyatakan dual dari k yang lain. Sehingga total banyaknya PTBA yang berbeda untuk satu nilai n dapat dihitung sebesar 2 kali banyaknya PTBA yang berbeda tanpa menjalankan program untuk k yang merupakan dual.

Menurut MacDougall dkk. (2002) graf C_5 tidak memiliki PTSA untuk $k = 15$ dan $k = 18$. Dikarenakan dari sembarang PTSA untuk graf lingkaran selalu dapat dibangun PTBA dengan menggeser label simpul menjadi label busur maka graf C_5 juga tidak memiliki PTBA untuk $k = 15$ dan $k = 18$. Ini sama dengan hasil yang diperoleh dengan menjalankan program. Terlihat pada Tabel 4.1 kolom C_5 .

Pada Gambar 4.5 diberikan grafik total banyaknya PTBA yang berbeda pada graf lingkaran untuk $n = 3, 4, \dots, 12$. Sumbu horizontal adalah banyaknya simpul pada graf lingkaran (n) sedangkan sumbu vertikal adalah total banyaknya PTBA yang berbeda pada graf lingkaran untuk suatu nilai n , yang merupakan hasil dari simulasi program. Terlihat bahwa semakin besar nilai n , total banyaknya

PTBA yang berbeda juga semakin besar. Untuk $n = 9, 10, 11, 12$ total banyaknya PTBA yang berbeda bertambah secara signifikan. Terlihat bahwa penambahan total banyaknya PTBA yang berbeda adalah eksponensial. Oleh karena itu total banyaknya PTBA yang berbeda untuk $n > 12$ bertambahnya juga akan signifikan. Sehingga memerlukan waktu yang lebih lama untuk menjalankan program.



Gambar 4.5 Grafik total banyaknya PTBA yang berbeda pada graf lingkaran untuk $n = 3, 4, \dots, 12$

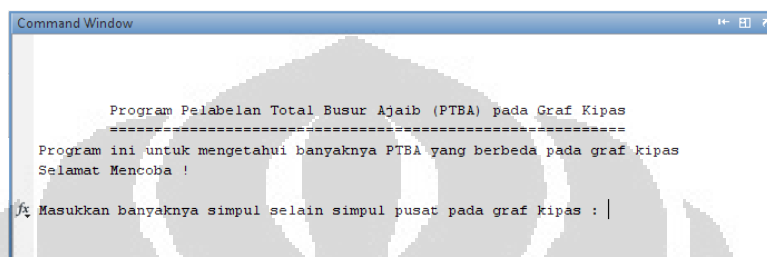
Selanjutnya akan dibahas implementasi dan simulasi algoritma PTBA pada graf kipas beserta contohnya.

4.2 Implementasi dan Simulasi Algoritma PTBA pada Graf Kipas f_n

Pada Subbab 3.2 telah dibentuk algoritma PTBA pada graf kipas yang terdiri dari 2 fungsi yaitu fungsi **initializeFan** (Algoritma 2.i) dan **extendFan** (Algoritma 2.i). Algoritma ini akan diimplementasikan dalam bentuk program MATLAB yang terdiri dari 2 fungsi, yaitu fungsi **fan** dan **extendfan**. Fungsi **fan** akan melakukan tahap-tahap pada Algoritma 2.i, yaitu melabel simpul c , v_1 , dan busur cv_1 . Kemudian fungsi **fan** akan memanggil fungsi **extendfan**. Fungsi **extendfan** akan melakukan tahap-tahap pada Algoritma 2.ii. Fungsi **extendfan**

akan melabel 1 simpul dan 2 busur. Lalu fungsi `extendfan` akan memanggil dirinya sendiri untuk melabel simpul dan busur selanjutnya. Untuk lebih jelas, program dapat dilihat pada Lampiran 4 (CD).

Pemanggilan program dilakukan dengan cara mengetik “fan” pada *Command Window*. Tampilan yang pertama kali muncul pada *Command window* adalah seperti Gambar 4.6.



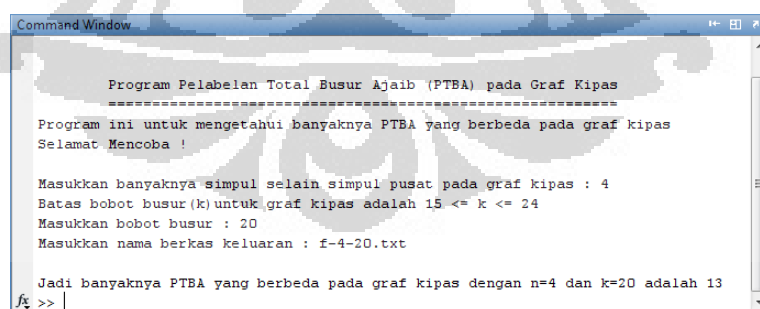
```

Command Window
-----
Program Pelabelan Total Busur Ajaib (PTBA) pada Graf Kipas
=====
Program ini untuk mengetahui banyaknya PTBA yang berbeda pada graf kipas
Selamat Mencoba !

f4 Masukkan banyaknya simpul selain simpul pusat pada graf kipas : |
  
```

Gambar 4.6 Tampilan pada *Command Window* saat fungsi `fan` dipanggil

Pengguna akan diminta memasukkan banyaknya simpul selain simpul pusat pada graf kipas (n), bobot busurnya (k), dan nama berkas keluaran. Sama seperti program sebelumnya, berkas keluaran digunakan untuk menyimpan hasil keluaran yang berupa label simpul dan busur yang membentuk PTBA pada graf kipas serta banyaknya PTBA yang berbeda yang diperoleh, dengan nilai n dan k yang telah dimasukkan oleh pengguna.



```

Command Window
-----
Program Pelabelan Total Busur Ajaib (PTBA) pada Graf Kipas
=====
Program ini untuk mengetahui banyaknya PTBA yang berbeda pada graf kipas
Selamat Mencoba !

Masukkan banyaknya simpul selain simpul pusat pada graf kipas : 4
Batas bobot busur (k) untuk graf kipas adalah 15 <= k <= 24
Masukkan bobot busur : 20
Masukkan nama berkas keluaran : f-4-20.txt

Jadi banyaknya PTBA yang berbeda pada graf kipas dengan n=4 dan k=20 adalah 13
f4 >> |
  
```

Gambar 4.7 Tampilan *Command Window* saat program dipanggil untuk masalah PTBA pada graf f_4 dengan $k = 20$

Berikut akan diberi contoh pemanggilan program untuk mencari PTBA pada graf kipas f_4 dengan $k = 20$. Seperti yang dijelaskan sebelumnya, ketik “fan” untuk memanggil program. Kemudian masukkan banyaknya simpul selain simpul

pusat pada graf kipas, yaitu $n = 4$ dan bobot busurnya $k = 20$. Kemudian masukkan nama berkas keluaran, misalkan f-4-20.txt. Pada Gambar 4.6 diberikan contoh pemanggilan program untuk mencari PTBA pada graf f_4 dengan $k = 20$. Pada Gambar 4.6 terlihat bahwa bobot busur untuk graf f_4 adalah bilangan asli dari 15 s.d. 24. Banyaknya PTBA yang berbeda pada graf kipas dengan $n = 4$ dan $k = 20$ adalah 13. Untuk mengetahui PTBA yang diperoleh, dibuka berkas keluaran f-4-20.txt. Isi dari berkas keluaran f-4-20.txt dapat dilihat pada Gambar 4.7.

```

PTBA untuk graf kipas dengan n=4 dan k=20 adalah :
1. c=3
V=[5 11 8 10]
E=[4 1 2]
CV=[12 6 9 7]

2. c=3
V=[6 12 7 9]
E=[2 1 4]
CV=[11 5 10 8]

3. c=3
V=[9 10 6 12]
E=[1 4 2]
CV=[8 7 11 5]

4. c=5
V=[4 6 12 7]
E=[10 2 1]
CV=[11 9 3 8]

5. c=6
V=[9 10 2 11]
E=[1 8 7]
CV=[5 4 12 3]

6. c=7
V=[1 8 10 4]
E=[11 2 6]
CV=[12 5 3 9]

7. c=7
V=[1 11 4 10]

8. c=7
V=[2 12 3 9]
E=[6 5 8]
CV=[11 1 10 4]

9. c=7
V=[9 8 2 12]
E=[3 10 6]
CV=[4 5 11 1]

10. c=10
V=[7 2 6 9]
E=[11 12 5]
CV=[3 8 4 1]

11. c=11
V=[1 7 4 6]
E=[12 9 10]
CV=[8 2 5 3]

12. c=11
V=[2 8 3 5]
E=[10 9 12]
CV=[7 1 6 4]

13. c=11
V=[5 6 2 8]
E=[9 12 10]
CV=[4 3 7 1]

Jadi banyaknya PTBA yang berbeda pada graf kipas dengan n=4 dan k=20 adalah 13

```

Gambar 4.8 PTBA yang diperoleh untuk graf f_4 dengan $k = 20$ yang disimpan di berkas keluaran f-4-20.txt

Pada Gambar 4.7 terlihat bahwa diperoleh 13 PTBA yang berbeda untuk graf f_4 dengan $k = 20$. Seperti yang terlihat pada Gambar 4.7 bahwa selain label simpul pusat c , label-label dinyatakan dalam bentuk matriks baris V untuk simpul, E untuk busur yang menghubungkan simpul v_i dan v_{i+1} untuk $i = 1, 2, \dots, n - 1$, dan CV untuk busur yang menghubungkan simpul c dan v_i untuk $i = 1, 2, \dots, n$. Kolom ke- i menandakan simpul v_i (busur e_i ataupun cv_i). Dengan cara yang sama akan dijalankan program untuk mencari semua PTBA yang mungkin yang berbeda pada graf f_3 untuk semua nilai k yang mungkin, yaitu $12 \leq k \leq 18$. Hasil yang diperoleh telah digambar dan diperlihatkan pada Lampiran 1. Hasil yang telah diperoleh pada Subbab 3.2 merupakan PTBA ke-6 pada Gambar 4.7. Hasil yang

diperoleh pada Subbab 3.2 sama dengan hasil yang telah diperoleh dengan menggunakan program, yaitu sebanyak 13 PTBA yang berbeda.

Tabel 4.2 Banyaknya PTBA yang berbeda pada graf kipas f_3 sampai f_{10} berdasarkan konstanta ajaib $k \left(\frac{11n^2+7n-4}{4n-2} \leq k \leq \frac{25n^2-13n-2}{4n-2} \right)$

f_3		f_4		f_5		f_6		f_7		f_8		f_9		f_{10}	
k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA
12	4	15	10	17	0	20	0	23	0	26	0	28	0	31	0
13	1	16	3	18	15	21	45	24	107	27	196	29	0	32	0
14	4	17	6	19	13	22	41	25	85	28	292	30	858	33	3098
15	6	18	16	20	14	23	49	26	169	29	488	31	828	34	3915
16	4	19	13	21	33	24	76	27	144	30	427	32	1588	35	7103
17	1	20	13	22	21	25	69	28	230	31	604	33	1988	36	8269
18	4	21	16	23	33	26	88	29	210	32	641	34	2333	37	10346
		22	6	24	28	27	113	30	314	33	779	35	2512	38	11679
		23	3	25	33	28	96	31	217	34	688	36	3148	39	14929
		24	10	26	21	29	96	32	277	35	775	37	2763	40	14230
				27	33	30	113	33	322	36	904	38	3047	41	15587
				28	14	31	88	34	277	37	764	39	3722	42	17681
				29	13	32	69	35	217	38	764	40	3083	43	16966
				30	15	33	76	36	314	39	904	41	3036	44	15679
				31	0	34	49	37	210	40	775	42	4398	45	19455
						35	41	38	230	41	688	43	3036	46	15405
						36	45	39	144	42	779	44	3083	47	15405
						37	0	40	169	43	641	45	3722	48	19455
								41	85	44	604	46	3047	49	15679
								42	107	45	427	47	2763	50	16966
								43	0	46	488	48	3148	51	17681
										47	292	49	2512	52	15587
										48	196	50	2333	53	14230
										49	0	51	1988	54	14929
												52	1588	55	11679
												53	828	56	10346
												54	858	57	8269
												55	0	58	7103
												56	0	59	3915
														60	3098
														61	0
														62	0
Total=	24	Total=	96	Total=	286	Total=	1154	Total=	3828	Total=	13116	Total=	62210	Total=	348684

Seperti halnya simulasi program algoritma pada graf lingkaran, karena keterbatasan waktu simulasi baru dapat dilakukan sampai $n = 10$. Dimana n merupakan banyaknya simpul selain simpul pusat pada graf kipas. Dikarenakan untuk $n = 2$ graf f_2 sama dengan graf C_3 maka simulasi dilakukan mulai dari $n = 3$. Hasil simulasi untuk f_2 dapat dilihat pada Tabel 4.1 kolom C_3 . Sedangkan hasil dari simulasi yang dilakukan untuk f_n ($3 \leq n \leq 10$) diberikan pada Tabel 4.2.

Tabel 4.2 memberikan banyaknya PTBA yang berbeda pada graf kipas f_3 sampai f_{10} berdasarkan konstanta ajaib k . Sama seperti pada Tabel 4.1 kolom k adalah konstanta ajaib yang diperoleh dari (2.10). Kolom # PTBA adalah banyaknya PTBA yang berbeda pada f_n untuk n dan k yang diberikan. Baris terakhir di

setiap kolom f_n , menyatakan total banyaknya PTBA yang berbeda pada f_n untuk nilai n yang diberikan.

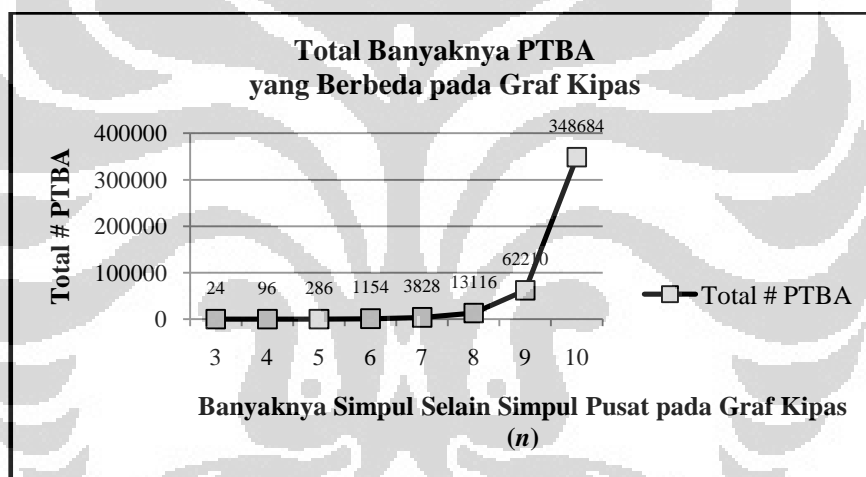
Sama seperti Tabel 4.1, pada Tabel 4.2 banyaknya PTBA yang berbeda untuk pelabelan dual diberi warna biru. Terlihat bahwa terdapat banyaknya PTBA yang berbeda yang sama. Ini merupakan pelabelan dualnya. Sebagai contoh, PTBA pada $f_3, k = 17$ merupakan pelabelan dual dari PTBA pada $f_3, k = 13$ (begitu juga sebaliknya). Pada graf kipas untuk n genap, banyaknya konstanta ajaibnya genap. Sedangkan untuk n ganjil, banyaknya konstanta ajaibnya ganjil. Sehingga untuk k yang di tengah, pelabelan dual untuk k tersebut adalah pelabelan dengan k yang sama. Secara teoritis dapat dibuktikan bahwa ini selalu berlaku untuk k yang ditengah. Contohnya, pelabelan dual dari PTBA untuk $f_3, k = 13$ adalah PTBA untuk $f_3, k = 13$. Pada Tabel 4.2, banyaknya PTBA yang berbeda untuk k yang di tengah diberi warna merah. Terlihat juga bahwa rentang konstanta ajaib untuk graf kipas lebih panjang dibandingkan dengan graf lingkaran. Semakin besar n , total banyaknya PTBA untuk graf kipas lebih besar dari graf lingkaran. Pada graf kipas f_n dengan $n > 4$, untuk k terkecil dan terbesar tidak memiliki PTBA. Sehingga perlu dicari cara untuk memperketat batas dari k . Apabila bisa ditemukan batas dari k yang lebih ketat maka tidak perlu menjalankan program untuk k tersebut. Menurut Slamini dkk. (2002), terdapat PTBA pada graf kipas untuk $k = 4n + 2$. Pada simulasi ini dijalankan untuk k tersebut dan juga untuk k yang lain. Dari Tabel 4.2 dan Tabel 4.1 (untuk f_2 sama dengan C_3) dapat disimpulkan bahwa graf kipas f_n dengan $2 \leq n \leq 10$ memiliki PTBA untuk semua nilai k kecuali $k =$

$$\left\lfloor \frac{11n^2+7n-4}{4n-2} \right\rfloor, \left\lfloor \frac{25n^2-13n-2}{4n-2} \right\rfloor \text{ untuk } 5 \leq n \leq 8 \text{ dan } k = \left\lfloor \frac{11n^2+7n-4}{4n-2} \right\rfloor, \left\lfloor \frac{11n^2+7n-4}{4n-2} \right\rfloor + 1, \\ \left\lfloor \frac{25n^2-13n-2}{4n-2} \right\rfloor - 1, \left\lfloor \frac{25n^2-13n-2}{4n-2} \right\rfloor \text{ untuk } n = 9, 10.$$

Pada Gambar 4.9 diberikan grafik total banyaknya PTBA yang berbeda pada graf kipas untuk $n = 3, 4, \dots, 10$. Sumbu horizontal adalah banyaknya simpul selain simpul pusat pada graf kipas (n) sedangkan sumbu vertikal adalah total banyaknya PTBA yang berbeda pada graf kipas untuk suatu nilai n , yang merupakan hasil dari simulasi program.

Sama seperti pada graf lingkaran, terlihat bahwa semakin besar nilai n , total banyaknya PTBA yang berbeda juga semakin besar. Terlihat juga bahwa per-

tambahan total banyaknya PTBA yang berbeda adalah eksponensial. Pada graf lingkaran untuk $n \geq 9$ total banyaknya PTBA yang berbeda bertambah secara signifikan. Sedangkan pada graf kipas pertambahan total banyaknya PTBA yang berbeda sudah signifikan mulai dari $n = 6$. Total banyaknya PTBA yang berbeda untuk graf f_6 mencapai empat kali lipat dari total banyaknya PTBA yang berbeda untuk graf f_5 . Tetapi untuk $n = 7$ dan $n = 8$ pertambahan total banyaknya PTBA yang berbeda hanya mencapai tiga kali lipat. Kemudian untuk $n = 9$ dan $n = 10$ pertambahan total banyaknya PTBA yang berbeda kembali mencapai lima kali lipat. Oleh karena itu total banyaknya PTBA yang berbeda untuk $n > 10$ bertambahnya juga akan signifikan. Sehingga diperlukan waktu yang lebih lama untuk menjalankan program.



Gambar 4.9 Grafik total banyaknya PTBA yang berbeda pada graf kipas untuk $n = 3, 4, \dots, 10$

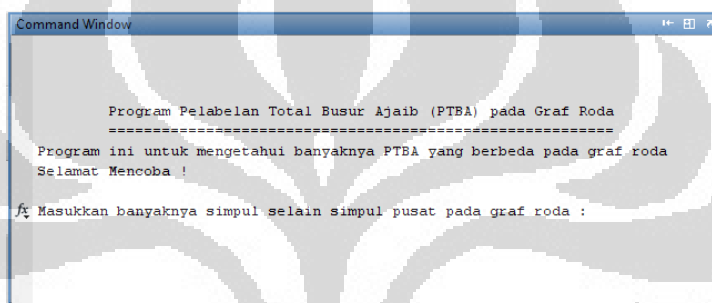
Selanjutnya akan dibahas implementasi dan simulasi algoritma PTBA pada graf roda beserta contohnya.

4.3 Implementasi Algoritma PTBA pada Graf Roda W_n

Pada Subbab 3.3 telah dibangun algoritma PTBA pada graf roda. Algoritma ini terdiri dari 2 fungsi yaitu fungsi **initializeWheel** (Algoritma 3.i) dan **extendWheel** (Algoritma 3.ii). Algoritma ini akan diimplementasikan dalam bentuk program MATLAB. Algoritma PTBA pada graf roda akan dijalankan oleh 2 fungsi, yaitu fungsi **wheel** dan **extendwheel**. Fungsi **wheel** akan melakukan tahap-

tahap pada Algoritma 3.i, yaitu melabel simpul c , v_1 , dan busur cv_1 . Kemudian fungsi wheel akan memanggil fungsi extendwheel. Fungsi extendwheel akan melakukan tahap-tahap pada Algoritma 3.ii. Fungsi extendwheel akan melabel 1 simpul dan 2 busur. Lalu fungsi extendwheel akan memanggil dirinya sendiri untuk melabel simpul dan busur selanjutnya. Untuk lebih jelas, program dapat dilihat pada Lampiran 5 (CD).

Pemanggilan program dilakukan dengan cara mengetik “wheel” pada *Command Window*. Tampilan yang pertama kali muncul pada *Command Window* adalah seperti Gambar 4.10.



```

Command Window

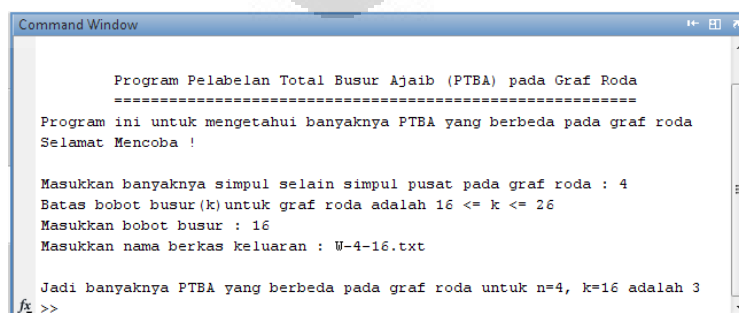
Program Pelabelan Total Busur Ajaib (PTBA) pada Graf Roda
=====
Program ini untuk mengetahui banyaknya PTBA yang berbeda pada graf roda
Selamat Mencoba !

fx Masukkan banyaknya simpul selain simpul pusat pada graf roda :

```

Gambar 4. 10 Tampilan pada *Command Window* saat fungsi wheel dipanggil

Pengguna akan diminta memasukkan banyaknya simpul selain simpul pusat pada graf roda (n), bobot busurnya (k), dan nama berkas keluaran. Berkas keluaran digunakan untuk menyimpan hasil keluaran yang berupa label simpul dan busur yang membentuk PTBA pada graf roda serta banyaknya PTBA yang berbeda yang diperoleh, dengan nilai n dan k yang telah dimasukkan oleh pengguna.



```

Command Window

Program Pelabelan Total Busur Ajaib (PTBA) pada Graf Roda
=====
Program ini untuk mengetahui banyaknya PTBA yang berbeda pada graf roda
Selamat Mencoba !

Masukkan banyaknya simpul selain simpul pusat pada graf roda : 4
Batas bobot busur(k) untuk graf roda adalah 16 <= k <= 26
Masukkan bobot busur : 16
Masukkan nama berkas keluaran : W-4-16.txt

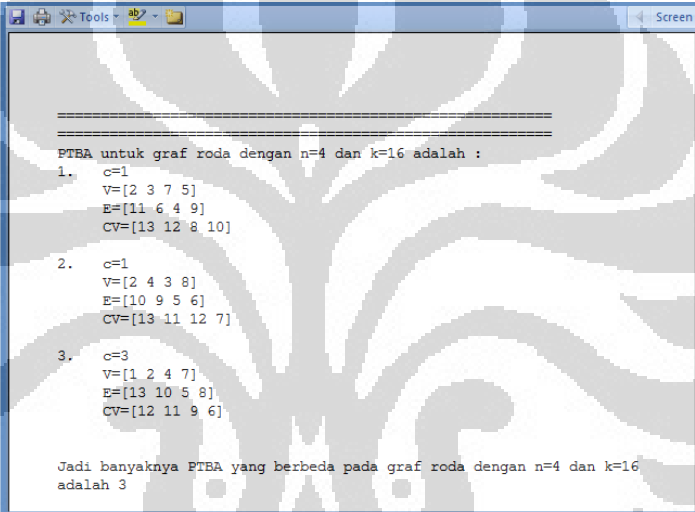
Jadi banyaknya PTBA yang berbeda pada graf roda untuk n=4, k=16 adalah 3
fx >>

```

Gambar 4. 11 Tampilan *Command Window* saat program dipanggil untuk masalah PTBA pada graf W_4 dengan $k = 16$

Pada Gambar 4.11 diberikan contoh pemanggilan program untuk mencari PTBA pada graf roda W_4 dengan $n = 4$ dan $k = 16$. Seperti yang dijelaskan sebelumnya, ketik “wheel” untuk memanggil program. Kemudian masukkan banyaknya simpul selain simpul center pada graf roda, yaitu $n = 4$ dan bobot busurnya $k=16$. Kemudian masukkan nama berkas keluaran, misalkan W-4-16.txt

Pada Gambar 4.9 terlihat bahwa bobot busur yang mungkin untuk graf W_4 adalah 16, 17, ..., 26. Banyaknya PTBA yang berbeda pada graf roda dengan $n = 4$ dan $k = 16$ adalah 3. Untuk mengetahui PTBA yang diperoleh, dibuka berkas keluaran f-4-20.txt (Gambar 4.12).



```

=====
PTBA untuk graf roda dengan n=4 dan k=16 adalah :
1.  c=1
    V=[2 3 7 5]
    E=[11 6 4 9]
    CV=[13 12 8 10]

2.  c=1
    V=[2 4 3 8]
    E=[10 9 5 6]
    CV=[13 11 12 7]

3.  c=3
    V=[1 2 4 7]
    E=[13 10 5 8]
    CV=[12 11 9 6]

Jadi banyaknya PTBA yang berbeda pada graf roda dengan n=4 dan k=16
adalah 3

```

Gambar 4. 12 PTBA yang diperoleh untuk graf W_4 dengan $k = 16$ yang disimpan di berkas keluaran W-4-16.txt

Pada Gambar 4.12 terlihat bahwa diperoleh 3 PTBA yang berbeda untuk graf f_4 dengan $k = 20$. Sama seperti keluaran program pada graf kipas, selain label simpul pusat c , label-label dinyatakan dalam bentuk matriks baris. V untuk simpul, E untuk busur yang menghubungkan simpul v_i dan v_{i+1} (dengan indeks diambil dalam mod n) untuk $i = 1, 2, \dots, n$, dan CV untuk busur yang menghubungkan simpul c dan v_i untuk $i = 1, 2, \dots, n$. Kolom ke- i menandakan simpul v_i (busur e_i ataupun cv_i). Hasil yang telah diperoleh pada Subbab 3.3 merupakan PTBA ke-1 pada Gambar 4.12. Hasil yang diperoleh pada Subbab 3.3 sama dengan hasil yang telah diperoleh dengan menggunakan program, yaitu sebanyak 3 PTBA yang berbeda. Dengan cara yang sama, akan dicari PTBA yang mungkin dan

berbeda pada graf W_4 untuk semua nilai k yang mungkin, yaitu $16 \leq k \leq 26$. Hasil yang diperoleh telah digambar, dapat dilihat pada Lampiran 2.

Tabel 4.3 Banyaknya PTBA yang berbeda pada graf roda W_3 sampai W_{10} berdasarkan konstanta ajaib k ($\frac{11n+17}{4} \leq k \leq \frac{25n+7}{4}$)

W_3		W_4		W_5		W_6		W_7		W_8		W_9		W_{10}	
k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA	k	# PTBA
13	0	16	3	18	0	21	0	24	0	27	0	29	0	32	0
14	0	17	0	19	3	22	4	25	0	28	37	30	0	33	0
15	0	18	2	20	2	23	2	26	0	29	16	31	62	34	83
16	0	19	1	21	4	24	6	27	0	30	16	32	84	35	43
17	0	20	4	22	1	25	6	28	0	31	14	33	145	36	253
18	0	21	6	23	3	26	4	29	0	32	33	34	149	37	372
19	0	22	4	24	6	27	11	30	0	33	29	35	187	38	279
20	0	23	1	25	4	28	5	31	0	34	41	36	246	39	448
		24	2	26	4	29	8	32	0	35	14	37	218	40	479
		25	0	27	6	30	0	33	0	36	46	38	164	41	598
		26	3	28	3	31	8	34	0	37	29	39	219	42	609
				29	1	32	5	35	0	38	46	40	246	43	698
				30	4	33	11	36	0	39	64	41	236	44	472
				31	2	34	4	37	0	40	46	42	292	45	664
				32	3	35	6	38	0	41	29	43	222	46	395
				33	0	36	6	39	0	42	46	44	222	47	631
						37	2	40	0	43	14	45	292	48	822
						38	4	41	0	44	41	46	236	49	631
						39	0	42	0	45	29	47	246	50	395
								43	0	46	33	48	219	51	664
								44	0	47	14	49	164	52	472
								45	0	48	16	50	218	53	698
										49	16	51	246	54	609
										50	37	52	187	55	598
										51	0	53	149	56	479
												54	145	57	448
												55	84	58	279
												56	62	59	372
												57	0	60	253
												58	0	61	43
														62	83
														63	0
														64	0
Total=	0	Total=	26	Total=	46	Total=	92	Total=	0	Total=	706	Total=	4940	Total=	12870

Seperti halnya simulasi program algoritma PTBA pada graf kipas, simulasi program algoritma PTBA pada graf roda juga baru dapat dilakukan sampai $n = 10$. Nilai n merupakan banyaknya simpul selain simpul pusat pada graf roda. Simulasi dilakukan mulai dari $n = 3$. Hasil dari simulasi yang dilakukan, diberikan pada Tabel 4.3.

Tabel 4.3 memberikan banyaknya PTBA yang berbeda pada graf roda W_3 sampai W_{10} berdasarkan konstanta ajaib k . Sama seperti pada Tabel 4.1 dan 4.2, kolom k adalah konstanta ajaib yang diperoleh dari (2.10). Kolom # PTBA adalah

banyaknya PTBA yang berbeda pada W_n untuk n dan k yang diberikan. Baris terakhir di setiap kolom W_n , menyatakan total banyaknya PTBA yang berbeda pada W_n untuk nilai n yang diberikan.

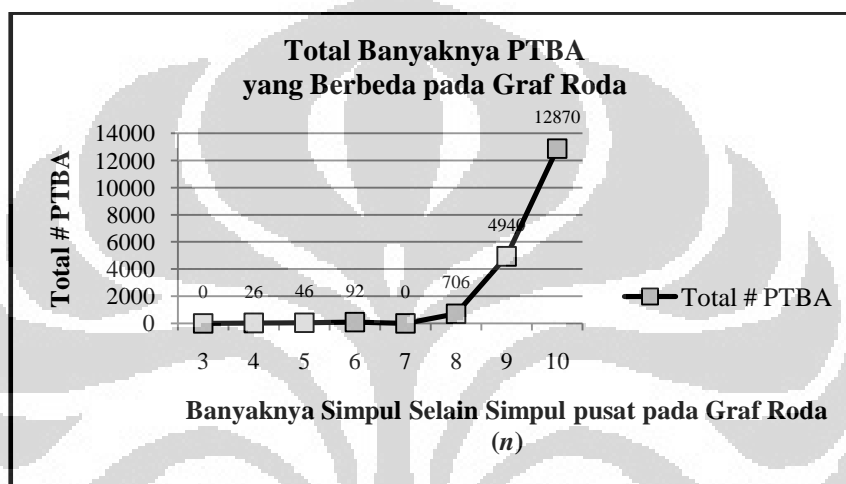
Menurut Galian (2009), Enomoto dkk. membuktikan bahwa W_n tidak memiliki PTBA untuk $n \equiv 3 \pmod{4}$. Sehingga dapat diketahui bahwa W_3 dan W_7 tidak memiliki PTBA. Namun, program tetap dijalankan untuk memverifikasi kebenaran program. Menurut Gallian (2009), Phillips dkk. membuktikan bahwa W_n memiliki PTBA untuk $n \equiv 0,1 \pmod{4}$. Program dijalankan untuk n tersebut dan n yang lainnya. Sama seperti graf kipas, graf roda juga memiliki rentang konstanta ajaib yang lebih panjang dibanding dengan graf lingkaran. Namun, total banyaknya PTBA yang berbeda untuk graf roda lebih sedikit dibanding dengan graf kipas.

Sama seperti Tabel 4.2, pada Tabel 4.3 banyaknya PTBA yang berbeda untuk pelabelan dual diberi warna biru dan untuk k yang ditengah diberi warna merah. Sebagai contoh, PTBA pada W_4 , $k = 22$ merupakan pelabelan dual dari PTBA pada W_4 , $k = 20$ (begitu juga sebaliknya). Pelabelan dual dari PTBA pada W_4 , $k = 20$ adalah PTBA pada W_4 , $k = 20$. Perbedaan dengan graf kipas adalah pada graf roda dengan n genap memiliki banyaknya konstanta ajaib ganjil. Sedangkan untuk n ganjil memiliki banyaknya konstanta ajaib yang genap. Sama seperti graf kipas, graf roda W_n dengan $n > 4$, untuk k terkecil dan terbesar tidak memiliki PTBA. Sehingga perlu dicari cara untuk memperketat batas dari k .

Menurut Slamain dkk. (2002) bahwa W_n memiliki PTBA dengan $k = 5n + 2$ untuk $n \equiv 6 \pmod{8}$. Pada simulasi ini dijalankan untuk k tersebut dan juga untuk k yang lain. Dari Tabel 4.3 dapat disimpulkan bahwa graf roda W_n dengan $3 \leq n \leq 10$ kecuali $n = 3, 7$ memiliki PTBA untuk semua nilai k kecuali $k = 17, 25$ untuk $n = 4$, $k = 30$ untuk $n = 6$, $k = \left\lfloor \frac{11n+17}{4} \right\rfloor, \left\lfloor \frac{25n+7}{4} \right\rfloor$ untuk $n = 5, 6, 8$, dan $k = \left\lfloor \frac{11n+17}{4} \right\rfloor, \left\lfloor \frac{11n+17}{4} \right\rfloor + 1, \left\lfloor \frac{25n+7}{4} \right\rfloor - 1, \left\lfloor \frac{25n+7}{4} \right\rfloor$ untuk $n = 9, 10$.

Pada Gambar 4.13 diberikan grafik total banyaknya PTBA yang berbeda pada graf roda untuk $n = 3, 4, \dots, 10$. Sumbu horizontal adalah banyaknya simpul selain simpul pusat pada graf roda (n) sedangkan sumbu vertikal adalah total banyaknya PTBA yang berbeda pada graf roda untuk suatu nilai n , yang merupakan hasil dari simulasi program. Sama seperti PTBA pada kelas-kelas graf sebelum-

nya, terlihat bahwa semakin besar nilai n maka semakin besar total banyaknya PTBA yang berbeda. Perbedaannya adalah untuk graf roda terdapat total banyaknya PTBA yang bernilai nol, yaitu untuk $n = 3, 4$. Pertambahan total banyaknya PTBA yang berbeda sudah signifikan mulai dari $n = 8$. Total banyaknya PTBA yang berbeda untuk graf W_9 hampir mencapai tujuh kali lipat dari total banyaknya PTBA yang berbeda untuk graf W_8 . Tetapi untuk W_{10} pertambahan total banyaknya PTBA yang berbeda hanya mencapai tiga kali lipat dari W_9 .



Gambar 4. 13 Grafik total banyaknya PTBA yang berbeda pada graf roda untuk $n = 3, 4, \dots, 10$

Jadi total banyaknya PTBA yang berbeda untuk setiap graf bertambah secara signifikan mulai dari n yang berbeda-beda. Untuk graf lingkaran mulai dari $n = 9$, graf kipas mulai dari $n = 6$, dan graf roda mulai dari $n = 8$.

BAB 5 KESIMPULAN

Dalam skripsi ini telah dibangun algoritma PTBA pada graf lingkaran, kipas, dan roda. Dengan algoritma-algoritma tersebut dapat diketahui ada atau tidak PTBA pada masing-masing graf terkait untuk sembarang nilai n dan k yang diberikan. Jika ada PTBA, maka PTBA yang mungkin dan tidak isomorfik dapat diketahui. Algoritma-algoritma tersebut telah diimplementasikan dalam bentuk program MATLAB. Secara teoritis program ini dapat dijalankan untuk sembarang n . Namun, simulasi yang dilakukan adalah untuk menghitung banyaknya PTBA yang berbeda (tidak isomorfik) pada graf terkait untuk n dan k yang diberikan. Pada Tabel 5.1 diberikan ringkasan hasil simulasi algoritma PTBA yang telah dilakukan pada Bab 4. Tabel tersebut menyatakan nilai k (konstanta ajaib) yang mungkin untuk graf lingkaran, kipas, dan roda serta nilai k dimana graf-graf tersebut tidak memiliki PTBA.

Tabel 5. 1 Ringkasan hasil simulasi algoritma PTBA pada graf lingkaran C_n , Kipas f_n , dan Roda W_n

Graf	Nilai k yang mungkin	Nilai k yang tidak ada PTBA	Graf	Nilai k yang mungkin	Nilai k yang tidak ada PTBA
C_3	9 s.d. 12	-	f_6	20 s.d. 37	20, 37
C_4	12 s.d. 15	-	f_7	23 s.d. 43	23, 43
C_5	14 s.d. 19	15, 18	f_8	26 s.d. 49	26, 49
C_6	17 s.d. 22	-	f_9	28 s.d. 56	28, 29, 55, 56
C_7	19 s.d. 26	-	f_{10}	31 s.d. 62	31, 32, 61, 62
C_8	22 s.d. 29	-	W_3	13 s.d. 20	13 s.d. 20
C_9	24 s.d. 33	-	W_4	16 s.d. 26	17, 25
C_{10}	27 s.d. 36	-	W_5	18 s.d. 33	18, 33
C_{11}	29 s.d. 40	-	W_6	21 s.d. 39	21, 30, 39
C_{12}	32 s.d. 43	-	W_7	24 s.d. 45	24 s.d. 45
f_2	9 s.d. 12	-	W_8	27 s.d. 51	27, 51
f_3	12 s.d. 18	-	W_9	29 s.d. 58	29, 30, 57, 58
f_4	15 s.d. 24	-	W_{10}	32 s.d. 64	32, 33, 63, 64
f_5	17 s.d. 31	17, 31			

Semakin besar nilai n maka pertambahan total banyaknya PTBA yang berbeda semakin signifikan. Grafik pertambahan total banyaknya PTBA yang berbeda memiliki tren eksponensial. Untuk graf dengan n dan k tertentu yang tidak

memiliki PTBA (banyaknya PTBA yang berbeda adalah nol) perlu diteliti lebih jauh untuk dapat dibuktikan bahwa untuk k tersebut tidak memiliki PTBA.



DAFTAR PUSTAKA

Baker, A., & Sawada, J. (2008). Magic Labelings on Cycles and Wheels. *COCOA LNCS* , 361-373.

Gallian, J. A. (2009). A Dynamic Survey of Graf Labeling. *The Electronic Journal of Combinatorics* 16 , #DS6.

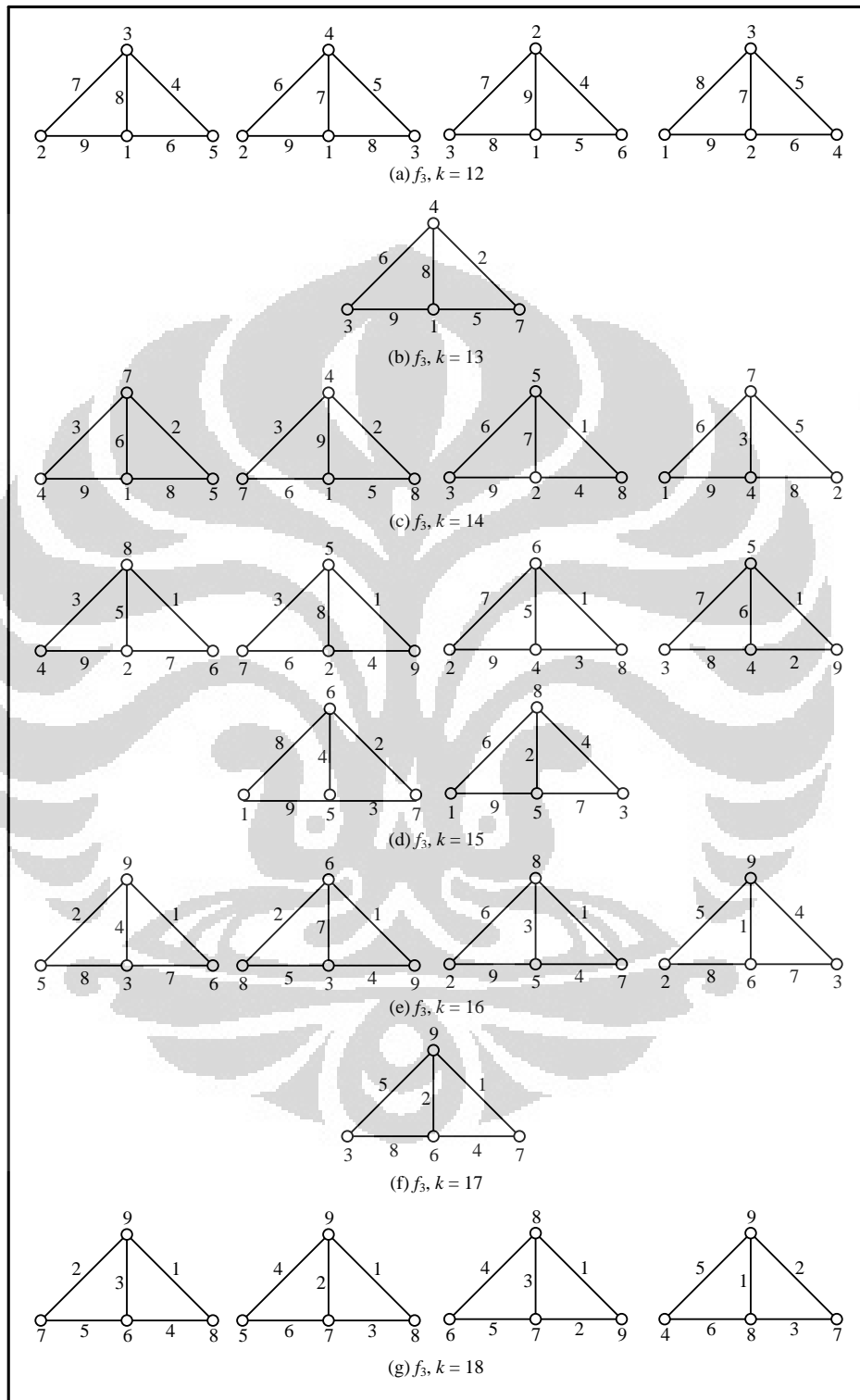
Graph Isomorphism. (t.thn.). Dipetik Februari 22, 2010, dari <http://www.cs.uu.nl/docs/vakken/an/>

MacDougall, J. A., Miller, M., Slamin, & Wallis, W. D. (2002). Vertex-Magic Total Labelings of Graphs. *Util. Math.* 61 , 3-21.

Slamin, Bača, M., Lin, Y., Miller, M., & Simanjuntak, R. (2002). Edge-Magic Total Labelings of Wheels, Fans and Friendship Graphs. *Bull. ICA* 35 , 89-98.

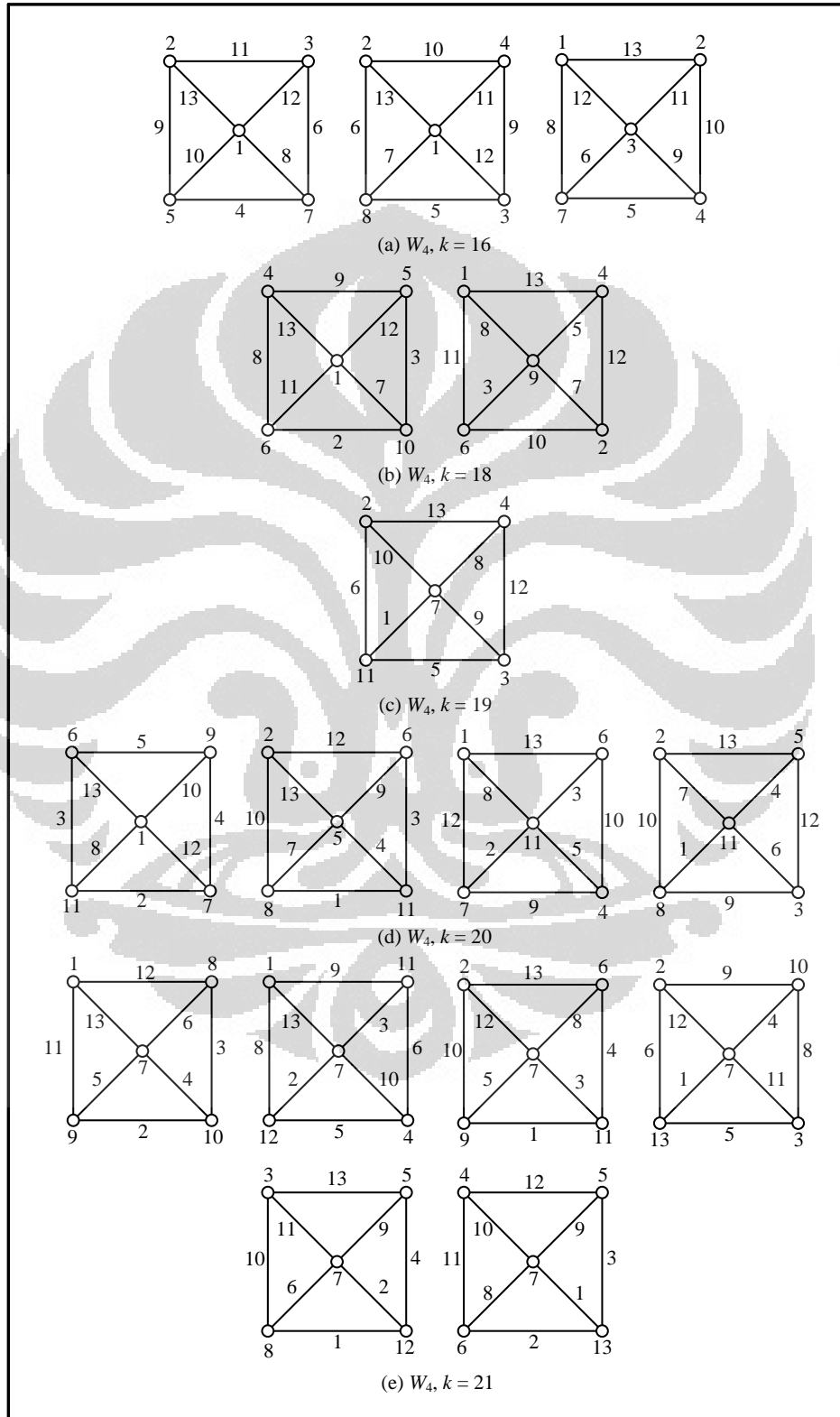
West, D. B. (2001). *Introduction to Graph Theory*. United States: Prentice-Hall, Inc.

Lampiran 1
PTBA yang berbeda pada graf f_3 dengan $12 \leq k \leq 18$



Lampiran 2
PTBA yang berbeda pada graf W_4 dengan $16 \leq k \leq 26$

Untuk graf W_4 dengan $k = 17, 25$ tidak memiliki PTBA



(lanjutan)

