VISUALISASI POHON RENTANG MINIMUM MENGGUNAKAN ALGORITMA KRUSKAL DAN PRIM

Imam Husni Al Amin

Program Studi Teknik Informatika
Universitas Stikubank, Semarang, Jawa Tengah, Indonesia
Pakimam.husni@gmail.com

Abstrak

Graf memuat obyek titik dan obyek garis yang menghubungkan titik. Properti penting yang dimiliki oleh graf adalah arah dan bobot pada garis. Graf berbobot adalah graf yang setiap garis atau sisinya diberi sebuah harga (bobot). Bobot ini dapat menyatakan jarak antara dua buah kota, biaya perjalanan, waktu tempuh yang dibutuhkan, dan sebagainya.

Penelitian ini melakukan analisa pada salah satu bentuk graf yaitu pohon, khususnya pada proses penyusunan dan pembentukan pohon rentang minimum (minimum spanning tree) menggunakan algoritma Kruskal dan Prim. Kedua algoritma ini menghasilkan struktur pohon rentang minimum yang sama, meskipun proses penyusunannya berbeda.

Proses penyusunan melalui dua buah contoh graf akan divisualisasikan menggunakan perangkat lunak pengolah dokumen LaTeX. Graf A disusun oleh 7 buah titik dan 11 garis sedangkan graf B memiliki 7 buah titik dan 12 garis. Hasil visualisasi disimpan ke dalam berkas PDF (portable document format). Berkas ini dapat digunakan sebagai modul ajar yang menarik, khususnya untuk pokok bahasan pohon rentang minimum.

Kata Kunci: Pohon rentang minimum, algoritma Kruskal, algoritma Prim

I. PENDAHULUAN

Graf adalah suatu diagram yang dapat menggambarkan berbagai macam struktur yang bertujuan untuk memvisualisasi obyek-obyek agar lebih mudah dimengerti. Aplikasi graf dalam dunia nyata antara lain menggambarkan atau merepresentasikan struktur hubungan antar obyek, struktur hirarki organisasi, urutan langkah kerja, dan lain sebagainya. Dari sekian banyak bentuk representasi graf, bentuk pohon (*tree*) adalah salah satu yang memiliki terapan cukup banyak [10].

Diagram graf memuat obyek titik dan obyek garis yang menghubungkan titik. Properti penting yang dimiliki oleh graf adalah arah dan bobot pada garis. Garis dapat berarah atau tidak berarah. Garis berarah biasanya digunakan untuk menyatakan hubungan antar obyek dengan memperhatikan urutan antara obyek-obyek tersebut. Sebaliknya, pada garis tidak berarah digunakan untuk menyatakan hubungan antar obyek yang tidak memerlukan adanya urutan [10].

Graf berbobot adalah graf yang setiap garis atau sisinya diberi sebuah harga atau bobot. Bobot ini dapat menyatakan jarak antara dua buah kota, biaya perjalanan, waktu tempuh yang dibutuhkan, dan lain sebagainya. Istilah lain dari graf berbobot adalah graf berlabel [6]. Label disini dapat berarti memberi nilai pada garis juga memberi nama pada titik.

Penelitian ini akan melakukan analisa pada salah satu bentuk graf yang memiliki banyak penerapan di dunia nyata yaitu pohon (*tree*), khususnya pada proses penyusunan atau pembentukan pohon rentang minimum (*minimum spanning tree*) menggunakan algoritma Kruskal dan Prim. Konsep pohon rentang minimum secara lengkap dalam

perspektif matematika diskrit dan struktur data pernah ditulis oleh Graham dan Hell pada tahun 1985, dan Jayawant dan Glavin tahun 2009.

Dua buah contoh graf akan dicari pohon rentang minimumnya. Proses penyusunan pohon dikerjakan secara manual mengikuti langkah-langkah algoritma Kruskal dan Prim. Perangkat lunak pengolah dokumen LaTeX digunakan untuk menggambarkan atau memvisualisasikan hasil dari setiap langkah pada saat proses penyusunan pohon ini. Hasil visualisasi disimpan ke dalam berkas PDF.

II. METODE PENELITIAN

Metode penelitian yang digunakan yaitu studi pustaka terutama untuk konsep graf, pohon, algoritma Kruskal dan Prim. Menyusun perintah-perintah (*source code*) yang dikenali oleh LaTeX untuk menggambarkan proses pembentukan pohon rentang minimum.

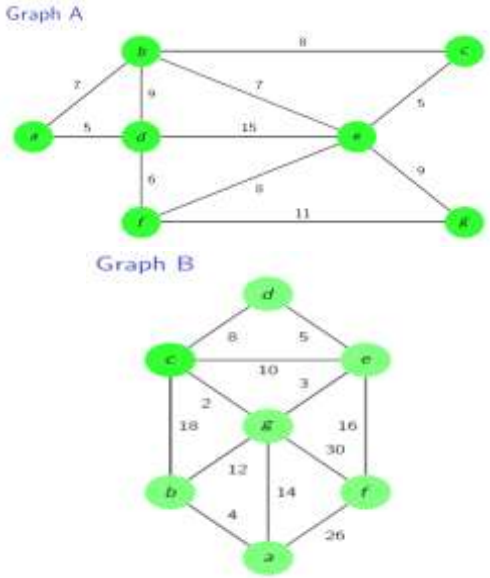
Bahan penelitian menggunakan dua buah contoh graf, dimana matriks bobot dan bentuk kedua graf ini dapat dilihat pada Gambar 1 dan Gambar 2. Graf A memiliki 7 buah titik dan 11 garis atau sisi, sedangkan graf B memiliki 7 buah titik dan 12 garis. Perangkat lunak yang digunakan adalah TexMaker (*universal LaTex editor*) versi 4.0.4, Beamer (*document class*) versi 3.24 untuk menghasilkan berkas presentasi PDF, dan Tikz (*package library*) versi 2.10.

Matriks bobot graph A (kiri) dan graph B (kanan)

	a	b	c	d	e	f	g
a	x	7		5			
b	7	x	8	9	7		
c		8	x		5		
d	5	9		x	15	6	
e		7	5	15	x	8	9
f				6	8	x	11
g					9	11	x

	a	b	c	d	e	f	g
a	x	4				26	14
b	4	x	18				12
c		18	x	8	10		2
d			8	x	5		
e			10	5	x	16	3
f	26				16	x	30
g	14	12	2		3	30	x

Gambar 1 Matriks Bobot dan Hubungan Antar Titik



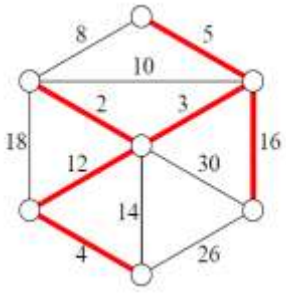
Gambar 2 Gambar graf A dan Graf B

A. Pohon Rentang Minimum (minimum spanning tree)

Pohon adalah sebuah graf terhubung yang tidak mengandung sirkuit. Definisi lain pohon yaitu graf tidak berarah yang bersifat hanya terdapat sebuah lintasan unik antara setiap pasang simpul. Beberapa pohon dapat membentuk hutan (*forest*) yaitu kumpulan pohon yang saling lepas. Konsep pohon ini memiliki penerapan yang luas, baik di dalam ilmu komputer maupun di luar bidang ilmu komputer [9].

Pohon membantu untuk mempelajari struktur data. Para ahli bahasa menggunakan konsep pohon parsing (*parse tree*) untuk menguraikan kalimat. Dalam kehidupan sehari-hari kita telah menggunakan konsep pohon untuk menggambarkan diagram sistem gugur pertandingan olahraga, silsilah keluarga, struktur organisasi, dan lain-lain [6]. Contoh aplikasi pohon rentang yaitu pada pemeliharaan jalan raya dengan dana terbatas. Alokasi dana yang terbatas akan menyebabkan pertimbangan hanya merawat ruas jalan sesedikit mungkin, namun semua daerah masih tetap dapat terhubung satu sama lain.

Jika G adalah graf berbobot, maka bobot pohon rentang T dari G didefinisikan sebagai jumlah bobot sisi di T. Pohon rentang yang berbeda mempunyai bobot yang berbeda pula. Di antara semua pohon rentang di G, pohon rentang yang berbobot minimum disebut pohon rentang minimum (*minimum spanning tree*) [2]. Algoritma pohon rentang minimum secara lengkap pernah ditulis oleh Nesetril dan Nesetrilova pada tahun 2012. Sementara itu, Tapia dan Rojas pada tahun 2004 berhasil menerapkan konstruksi pohon rentang minimum untuk mengenali tulisan tangan secara *on-line*. Contoh sederhana pohon rentang minimum dapat dilihat pada Gambar 3 [2].



Gambar 3 Pohon rentang minimum

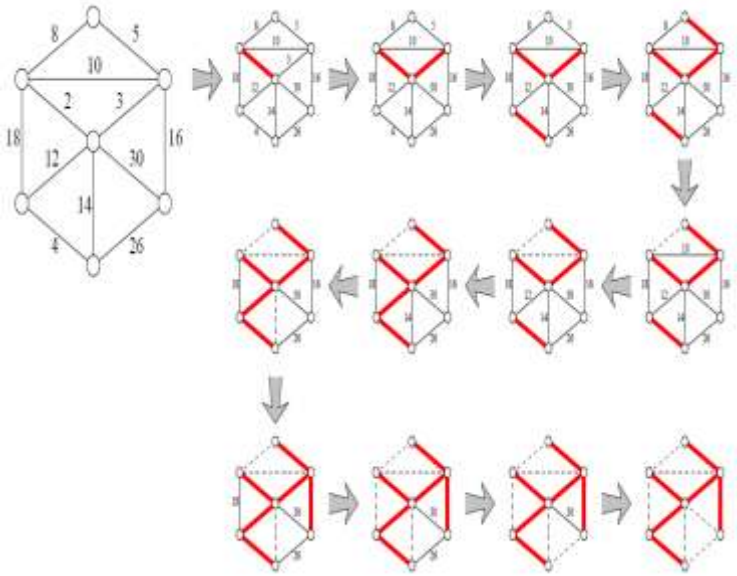
Di sini akan digunakan dua buah algoritma untuk menyusun pohon rentang minimum yaitu algoritma Kruskal dan Prim. Proses atau langkah-langkah penyusunan pohon menggunakan kedua algoritma ini berbeda, namun bentuk akhir pohon yang dihasilkan adalah sama.

B. Algoritma Kruskal

Langkah awal algoritma Kruskal adalah sisi-sisi di dalam graf diurutkan terlebih dulu berdasarkan bobot dari yang kecil ke besar. Sisi-sisi yang sudah urut berdasarkan bobot akan membentuk hutan dengan masing-masing pohon di hutan hanya berupa satu buah simpul, disebut dengan hutan rentang (*spanning forest*). Misalkan T adalah pohon rentang yang sisi-sisinya diambil dari graf G. Sisi yang dimasukkan ke dalam himpunan T adalah sisi graf G sedemikian sehingga T adalah pohon. Sisi dari graf G ditambahkan ke T jika ia tidak membentuk sirkuit atau siklus di T [6]. Berikut ini adalah langkah-langkah algoritma Kruskal yaitu:

1. Urutkan secara menaik berdasarkan bobotnya untuk semua sisi-sisi graf. T masih kosong.
2. Pilih sisi e (u,v) dengan bobot minimum yang tidak membentuk sirkuit di T. Masukkan e (u,v) ke dalam T.
3. Ulangi langkah nomor 2 sebanyak n-1 kali.

Proses memperoleh pohon rentang minimum menggunakan algoritma Kruskal pada graf B dapat dilihat pada Gambar 4 [2].



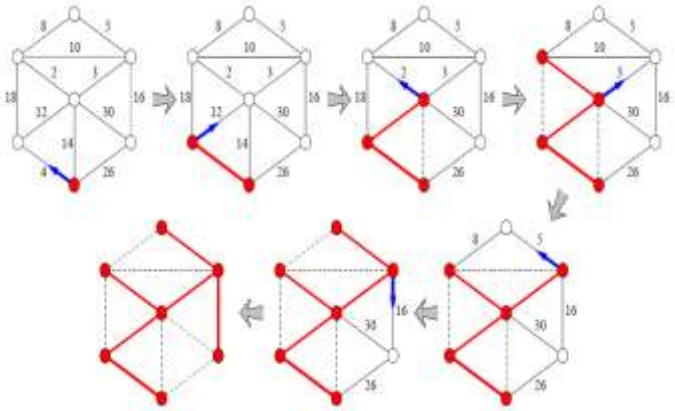
Gambar 4 Pohon rentang minimum algoritma Kruskal

C. Algoritma Prim

Misalkan T adalah pohon rentang yang sisi-sisinya diambil dari graf G. Algoritma Prim membentuk pohon rentang minimum langkah demi langkah. Pada setiap langkah kita mengambil sisi e dari graf G yang mempunyai bobot minimum dan bersisian dengan simpul-simpul di dalam T tetapi e tidak membentuk sirkuit di dalam T [6]. Berikut ini adalah langkah-langkah algoritma Prim yaitu:

1. Ambil sisi dari graf G yang berbobot minimum, masukkan ke dalam T.
2. Pilih sisi e (u,v) yang mempunyai bobot minimum dan bersisian dengan simpul di T, tetapi e (u,v) tidak membentuk sirkuit di T. Masukkan e (u,v) ke dalam T.
3. Ulangi langkah 2 sebanyak n-2 kali.

Proses memperoleh pohon rentang minimum menggunakan algoritma Prim pada graf B dapat dilihat pada Gambar 5 [2].



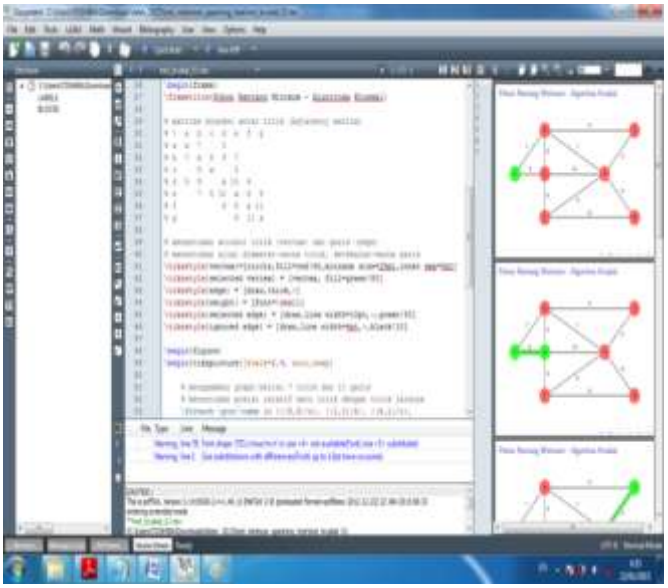
Gambar 5 Pohon rentang minimum algoritma Prim

III. HASIL DAN PEMBAHASAN

Percobaan dilakukan menggunakan dua buah graf yaitu graf A dan graf B. Struktur kedua graf ini dapat dilihat pada Gambar 1 dan Gambar 2. Graf A memiliki 7 buah titik dan 11 garis sedangkan graf B disusun oleh 7 buah titik dan 12 garis. Secara umum, penulisan perintah (*source code*) agar dimengerti oleh pustaka Beamer dan Tikz untuk proses visualisasi pohon rentang minimum mengikuti aturan berikut ini (perintah lengkap dapat dilihat pada Tabel 1):

1. Menentukan posisi relatif antara satu titik dengan titik lainnya. Format penulisan perintah menggambar sebuah titik adalah`{(posisi_kolom,posisi_baris)/nama_titik}`. Posisi kolom paling kiri diberi nilai 0 dan akan bertambah 1 untuk kolom yang di sebelah kanannya. Pada posisi baris, maka baris tengah bernilai 0, akan bertambah 1 untuk baris di atasnya dan berkurang 1 untuk baris di bawahnya. Contoh penulisan perintah ada pada kolom ketiga yang diberi catatan A.
2. Menghubungkan titik dengan garis dan menuliskan bobot setiap garis. Format penulisan perintah menggambar sebuah garis atau sisi adalah `{titik_pertama/titik_kedua/bobot}`. Pada graf tidak berarah maka penulisan titik pertama dan titik kedua bisa dipertukarkan. Contoh penulisan perintah ada pada kolom ketiga yang diberi catatan B.
3. Menggambar urutan titik yang membentuk pohon rentang minimum. Format penulisan perintahnya adalah `{nama_titik/nomor_slide}`. Contoh penulisan perintah ada pada kolom ketiga yang diberi catatan C.
4. Menggambar urutan garis yang membentuk pohon rentang minimum. Format penulisan perintahnya adalah `{titik_pertama/titik_kedua}`. Contoh penulisan perintah ada pada kolom ketiga yang diberi catatan D.
5. Menentukan garis yang boleh dihilangkan karena akan membentuk sirkuit. Format penulisan perintahnya adalah `{titik_pertama/titik_kedua/nomor_slide}`. Contoh penulisan perintah ada pada kolom ketiga yang diberi catatan E.

Pada Gambar 6 dapat dilihat tampilan antar muka TexMaker yang digunakan untuk menulis kode sumber, yang memiliki fasilitas untuk melihat hasilnya (*output*) dari proses *quick build* dan *view PDF*. TexMaker dapat memberitahu terjadi kesalahan (*error* atau *warning*) apabila penulisan perintah masih belum benar.



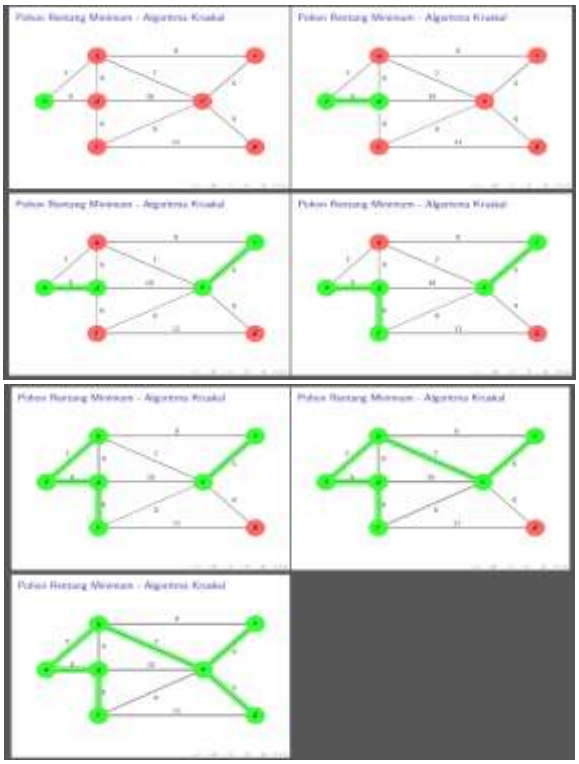
Gambar 6 Tampilan antarmuka TexMaker

Hasil penelitian dapat dilihat pada Gambar 7, Gambar 8, Gambar 9, dan Gambar 10. Pada graf A dan graf B masing-masing membutuhkan 7 tahap untuk menghasilkan pohon rentang minimum. Disini dapat dilihat melalui proses penyusunan yang berbeda namun menghasilkan struktur pohon rentang yang sama.

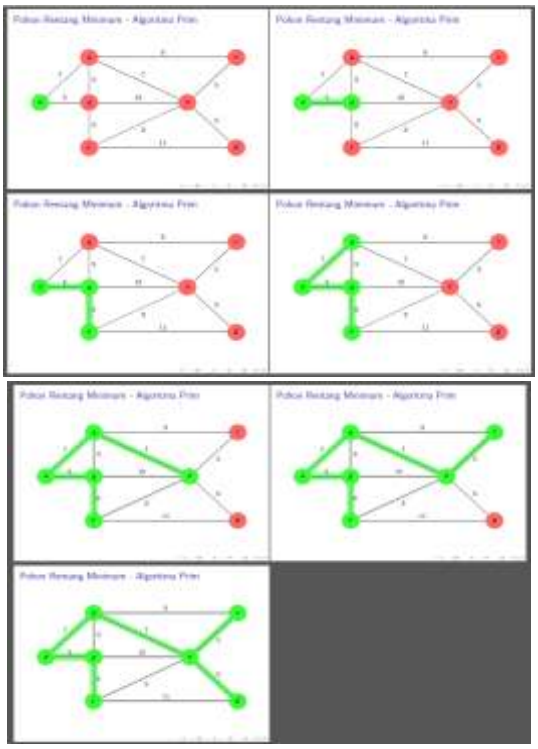
TABEL 1
PERINTAH VISUALISASI PENYUSUNAN POHON RENTANG MINIMUM

Nama graph	Algoritma	Proses visualisasi langkah demi langkah penyusunan pohon rentang minimum	Catatan
graph A (7 titik 11 garis)	Kruskal	({[0,0)/a}, {(1,1)/b}, {(4,1)/c},{(1,0)/d}, {(3,0)/e}, {(1,-1)/f}, {(4,-1)/g})	A
		(b/a/7, c/b/8, d/a/5, d/b/9,e/b/7, e/c/5, e/d/15, f/d/6, f/e/8, g/e/9, g/f/11)	B
		(a/1, d/2, c/3, e/3, f/4, b/5, g/7)	C
		(a/d, c/e, d/f, a/b, b/e, e/g)	D
		(b/d/5, d/e/6, e/f/6,b/c/6, f/g/7)	E
	Prim	({[0,0)/a}, {(1,1)/b}, {(4,1)/c},{(1,0)/d}, {(3,0)/e}, {(1,-1)/f}, {(4,-1)/g})	A
		(b/a/7, c/b/8, d/a/5, d/b/9,e/b/7, e/c/5, e/d/15, f/d/6, f/e/8, g/e/9, g/f/11)	B
		(a/1, d/2, f/3, b/4, e/5, c/6, g/7)	C
		(a/d, d/f, a/b, b/e, e/c, e/g)	D
		(b/d/4,d/e/5,e/f/5,b/c/6,f/g/7)	E
graph B (7 titik 12 garis)	Kruskal	({[1,-2)/a}, {(0,-1)/b}, {(0,1)/c},{(1,2)/d}, {(2,1)/e}, {(2,-1)/f}, {(1,0)/g})	A
		(a/b/4, b/c/18, c/d/8, d/e/5,e/f/16, a/f/26, a/g/14, b/g/12, c/g/2, e/g/3, c/e/10, f/g/30)	B
		(c/1, g/2, e/3, a/4, b/4, d/5, f/7)	C
		(c/g, e/g, a/b, d/e, b/g, e/f)	D
		(c/e/3,c/d/5,b/c/6,a/g/6,a/f/7,f/g/7)	E
	Prim	({[1,-2)/a}, {(0,-1)/b}, {(0,1)/c},{(1,2)/d}, {(2,1)/e}, {(2,-1)/f}, {(1,0)/g})	A
		(a/b/4, b/c/18, c/d/8, d/e/5,e/f/16, a/f/26, a/g/14, b/g/12, c/g/2, e/g/3, c/e/10, f/g/30)	B
		(a/1, b/2, g/3, c/4, e/5, d/6, f/7)	C
		(a/b, b/g, g/c, g/e, e/d, e/f)	D
		(a/g/3,b/c/4,c/e/5,c/d/6,a/f/7,f/g/7)	E

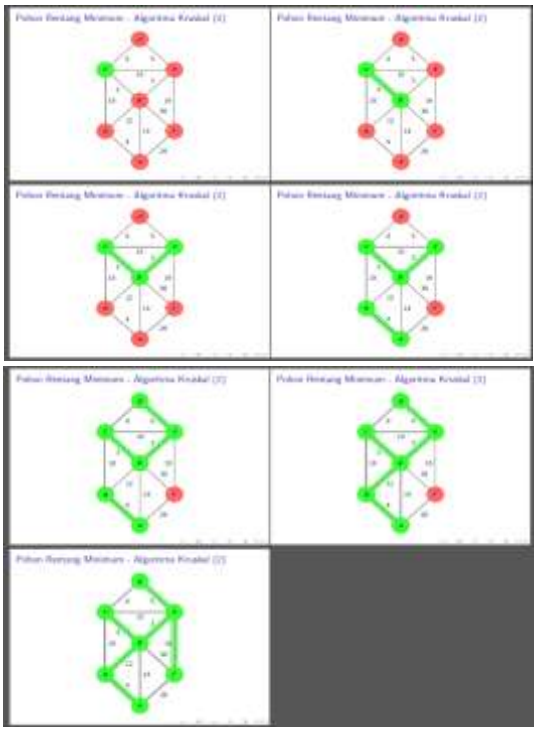
Catatan:
A menentukan posisi relatif satu titik dengan titik lainnya
B menghubungkan titik dengan garis (sisi) dan menuliskan bobot setiap garis
C animasi memilih titik dan garis, menggambar urutan titik yang membentuk pohon
D menggambar urutan garis yang membentuk pohon
E menentukan garis yang boleh dihilangkan karena akan membentuk sirkuit



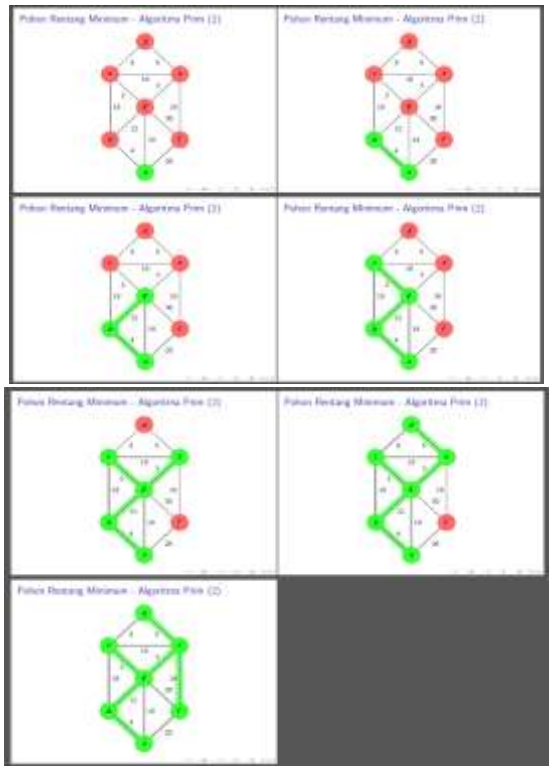
Gambar 7 Pohon Rentang Minimum Algoritma Kruskal01



Gambar 8 Pohon Rentang Minimum Algoritma Prim01



Gambar 9 Pohon Rentang Minimum Algoritma Kruskal02



Gambar 10 Pohon Rentang Minimum Algoritma Prim02

III. SIMPULAN

Melalui penelitian ini dapat disimpulkan:

1. Algoritma Kruskal dan algoritma Prim menghasilkan struktur pohon rentang minimum yang sama, meskipun melalui proses penyusunan yang berbeda.
2. Proses visualisasi pohon rentang minimum dapat disusun menggunakan perangkat lunak pengolah dokumen LaTex.
3. Berkas hasil visualisasi disimpan menggunakan format PDF yang dapat digunakan sebagai modul ajar yang menarik, khususnya untuk pokok bahasan pohon rentang minimum.

IV. SARAN

Untuk melengkapi apa yang telah diperoleh dari penelitian ini maka dapat ditambahkan algoritma lain yang bisa digunakan untuk menyusun pohon rentang minimum. Selanjutnya proses algoritma ini akan dibandingkan dengan algoritma Kruskal dan algoritma Prim. Perlu dipertimbangkan pemilihan perangkat lunak yang mampu membuat visualisasi sekaligus dapat menerima masukan dari pemakai agar lebih interaktif.

V. DAFTAR PUSTAKA

[1] van Dongen, M.R.C., 2012, Latex and Friends, Springer-Verlag Berlin Heidelberg.
[2] Erickson, J., 2013, Algorithms, Department of Computer Science University of Illinois Urbana-Champaign, USA.
[3] Goossens, M., dkk., 2008, The Latex Graphics Companion 2nd edition, Adisson-Wesley, USA.
[4] Graham, R.L., dan Hell, P, 1985, On the History of the Minimum Spanning Tree Problem, Annals of the History of Computing, Volume 7, Number 1, 43-57, January 1985.
[5] Jayawant, P., dan Glavin, K., 2009, Minimum spanning trees, Involve a journal of mathematics, Vol.2 No.4 pp.437-448, mathematical sciences publishers.
[6] Munir, R., 2010, Matematika Diskrit edisi ketiga revisi keempat, Informatika, Bandung.
[7] Nesetril, J., dan Nesetrilova, H, 2012, The Origins of Minimal Spanning Tree Algorithms - Boruvka and Jarnik, Documenta Mathematica Extra Volume ISMP (2012) 127–141.
[8] Oetiker, T., dkk., 2011, The Not So Short Introduction to LATEX2, Version 5.01, April 06, 2011.
[9] Sedgewick, R., and Wayne, K., 2011, Algorithms 4th edition, Adisson-Wesley USA.
[10] Siang, J.J., 2009, Matematika Diskrit dan Aplikasinya pada Ilmu Komputer edisi keempat, Penerbit Andi, Yogyakarta.
[11] Tapia, E., dan Rojas, R., 2004, Recognition of On-line Handwritten Mathematical Expressions Using a Minimum Spanning Tree Construction and Symbol Dominance, GREC 2003 LNCS 3088, pp.329-340, Springer-Verlag Berlin Heidelberg.