

Visualisasi Pohon Perentang Minimum Dalam Segmentasi Citra Digital

Efron Manik¹

¹ Program Studi Pendidikan Matematika,
Universitas HKBP Nommensen
efmanik@gmail.com

Abstrak :

Citra digital dapat disajikan dalam bentuk graf atau graf grid. Segmentasi secara luas digunakan dalam visi komputer. Segmentasi citra dilakukan dalam berbagai pendekatan, antara lain: Pendekatan daerah berkembang, pendekatan batas, pendekatan graf. Masalah yang akan dipelajari dalam tulisan ini adalah apakah ada perbedaan bentuk segmen-segmen jika citra yang diubah dalam graf grid, graf triangular grid, atau graf super grid? Salah satu pendekatan segmentasi citra adalah penggunaan pohon perentang minimum. Caranya adalah sisi-sisi pohon perentang minimum suatu graf yang dibentuk dari citra yang bobot sisinya lebih besar dari ambang batas dihapus. Sehingga terbentuk sub-sub pohon yang merupakan titik-titiknya pembentuk segmen-segmen dari citra. Dari hasil penelitian diperoleh pertama, segmentasi dengan pohon perentang minimum dengan ambang batas $(\sigma + \bar{w})$ cocok untuk citra digital yang diubah ke graf grid. Kedua, segmentasi dengan pohon perentang minimum dengan ambang batas $(\sigma + \bar{w})$ cocok untuk citra digital yang diubah ke graf triangular grid. Ketiga, segmentasi dengan pohon perentang minimum dengan ambang batas $(\sigma + \bar{w})$ tidak cocok untuk citra digital yang diubah ke graf super grid.

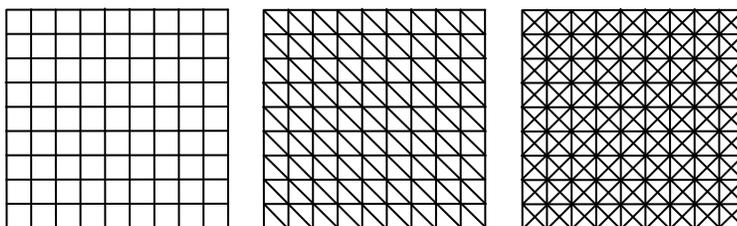
Kata kunci : Graf, Grid, Pohon, Segmentasi

1. Pendahuluan

Citra digital banyak dipelajari dalam ilmu komputer, terutama dalam komputer visi. Dalam tulisan ini, segmentasi citra digital dilakukan dengan memanfaatkan pohon perentang minimum. Bab ini

akan membahas latar belakang masalah, perumusan masalah, dan tujuan dan manfaat penelitian.

1.1. Latar Belakang



(a) graf *grid* (b) graf *triangular grid* (c) graf *super grid*

Gambar 1. Bentuk Graf dari Citra Digital

Citra digital dapat disajikan dalam bentuk graf atau graf *grid*. Citra digital juga dapat dipandang sebagai graf kisi berbobot tak berarah, dimana pixel dipandang sebagai titik dan selisih intensitas warna dua pixel yang berdekatan merupakan bobot dari sisi yang menghubungkan dua pixel tersebut (Zhang, Dai, Xiang, & Zhang, 2015). Beberapa cara penyajian Citra dalam graf antara lain: graf *grid*, graf *triangular grid*, dan graf *super grid* seperti pada Gambar 1. Graf *grid* adalah graf yang sisi-sisinya menghubungkan setiap pixel citra digital dengan empat pixel disekitarnya, yaitu: pixel di atas, bawah, kiri, dan kanannya. Sedangkan graf *triangular grid* adalah graf yang sisi-sisinya menghubungkan setiap pixel citra digital dengan enam pixel disekitarnya, yaitu: pixel kiri atas, kanan bawah, atas, bawah, kiri, dan kanannya. Selanjutnya graf *super grid* adalah graf yang sisi-sisinya menghubungkan setiap pixel citra digital dengan delapan atau semua pixel disekitarnya. Algoritma *multiscale graph-based segmentation* (MGS) menggunakan pendekatan populer *graph-cut* yang menyajikan citra digital sebagai graf *grid* (Kallasi, Rizzini, Oleari, & Aleotti, 2015).

Segmentasi secara luas digunakan dalam visi komputer. Segmentasi citra dilakukan dalam berbagai pendekatan, antara lain: Pendekatan daerah berkembang, pendekatan batas, pendekatan graf. Salah satu pendekatan adalah metode pergeseran (*meanshift*) daerah berkembang. Metode ini menggunakan serangkaian vektor menuju titik kekonvergenan (Freedman, & Kisilev, 2010). Fungsi tepi dalam Matlab biasanya digunakan untuk mengembangkan pendekatan batas daerah (Uemura, Koutaki, & Uchimura, 2011).

Semua langkah segmentasi masih mempunyai kelemahan. Pemilihan bandwidth sangat mempengaruhi hasil dari segmentasi ini. Contohnya adalah metode pergeseran. Metode ini masih memiliki kelemahan, yaitu: batas segmen luas yang terbentuk dengan penambahan nilai bandwidth tidak selalu merupakan kombinasi unsur batas segmen untuk bandwidth yang lebih kecil (Manik, 2010). Kelemahan ini terjadi karena metode yang diajukan tidak dapat mengecualikan data pencilan. Untuk memperbaikinya, Peter [6] menggunakan pendekatan pohon perentang minimum dalam graf untuk menggantikan persegi dalam pencocokan secara lokal.

Pohon perentang suatu graf tidak tunggal. Dengan algoritma Kruskal, kita dapat melihat bahwa pohon perentang minimum juga tidak selalu tunggal. Pengaruh pemilihan pohon perentang minimum yang berbeda terhadap hasil segmentasi menjadi hal yang harus dipikirkan.

1.2. Permasalahan

Peter (2010) (Manik, 2018) menggunakan pohon perentang untuk mensegmentasi citra digital dengan hanya untuk graf *grid*. Permasalahan yang akan diselesaikan dalam penelitian ini adalah bagaimana pengaruh segmentasi dengan pohon perentang terhadap bentuk citra untuk graf *grid*, graf *triangular grid*, dan graf *super grid*.

1.3. Tujuan dan Manfaat Penelitian

Tujuan penelitian ini adalah bentuk segmentasi citra yang dihasilkan dengan mengubah citra dalam bentuk graf *grid*, graf *triangular grid*, dan graf super *grid*. Manfaat penelitian ini adalah sebagai rujukan bagi peneliti dalam bidang komputer visi dalam memilih cara mengubah citra menjadi graf yang sesuai.

1.4. Graf

Suatu **graf tidak berarah** adalah suatu triple (V, E, Ψ) , dimana V dan E adalah himpunan hingga dan $\Psi: E \rightarrow \{X \subseteq V : |X|=2\}$. Suatu **graf berarah** atau **digraf** adalah suatu triple (V, E, Ψ) , dimana V dan E adalah himpunan hingga dan $\Psi: E \rightarrow \{(v, w) \in V \times V : v \neq w\}$. Jika ditulis **graf** maka kita mengartikannya sebagai digraf. Anggota dari V disebut titik, dan anggota dari E adalah sisi (Korte and Vygen, 2006). Dua sisi e, e' dengan $\Psi(e) = \Psi(e')$ disebut sisi paralel. Graf tanpa sisi paralel disebut graf sederhana.

Penulisan graf di atas dapat disederhanakan, misalnya disimbolkan dengan $G = (V, E)$. Kadang-kadang untuk memperjelas pembicaraan, himpunan titik-titik pada sebuah graf G dinotasikan dengan $V(G)$ dan himpunan sisi-sisi pada sebuah graf G dinotasikan dengan $E(G)$.

1.5. Pohon

Pohon adalah graf terhubung yang tidak memuat lingkaran. Selanjutnya kita akan membahas graf khusus (Manik, Suwilo, Tulus, & Sitompul, 2018) yang disebut pohon. Kita akan memperhatikan sifat-sifat khusus dari pohon yang berguna dalam tulisan ini.

Kruskal memberikan algoritma berikut untuk Masalah Pohon perentang Minimum yang diberikan oleh Kruskal. Misalkan $n := |V(G)|$ and $m := |E(G)|$.

Algoritma Kruskal

Input : Graf tak berarah G , dengan bobot $c: E(G) \rightarrow \mathbf{R}$.

Output : Menemukan pohon perentang dengan bobot minimum dari G

1. Urutkan sisi sedemikian sehingga $c(e_1) \leq c(e_2) \leq \dots \leq c(e_n)$.
2. Set $T := (V(G), \emptyset)$.
3. **For** $i := 1$ **to** m **do**:
 If $T + e_i$ tidak memuat lingkaran **then** set $T := T + e_i$.

Algoritma Kruskal menghasilkan pohon minimum yang merentang graf tersambung sederhana. Pernyataan ini dapat dibuktikan dengan dua bagian (Korte, & Vygen, 2006) (Rosen, 2007) (Winston, 2004). Tetapi kita akan menggunakan bukti yang sedikit berbeda untuk teorema berikut ini.

Teorema 1 Algoritma Kruskal menghasilkan pohon perentang minimum dari graf berbobot terhubung.

Menurut Mayr (1992), setiap pohon perentang minimum dapat dihasilkan dengan menggunakan Algoritma Kruskal. Jadi semua pohon perentang minimum dapat dipandang sebagai hasil dari Algoritma Kruskal. Selanjutnya kita akan menyajikan teorema tentang ketunggalan segmentasi yang dihasilkan oleh pohon perentang minimum (Manik, Suwilo, Tulus, & Sitompul, 2017).

Teorema 2. Misalkan $G(V, E)$ adalah graf berbobot. Misalkan $\alpha \in \mathbf{R}$, dan S, T adalah pohon perentang minimum dari graf G . Anggap bahwa semua sisi-sisi yang bobotnya lebih besar atau sama dengan α dihapus dari pohon S , dan T . Maka komponen terhubung S_1, S_2, \dots, S_p dari pohon S dan T_1, T_2, \dots, T_q dari pohon T akan terbentuk. Maka $p = q$ dan jika $V(S_i) \cap V(T_j) \neq \emptyset$ maka $V(S_i) = V(T_j)$.

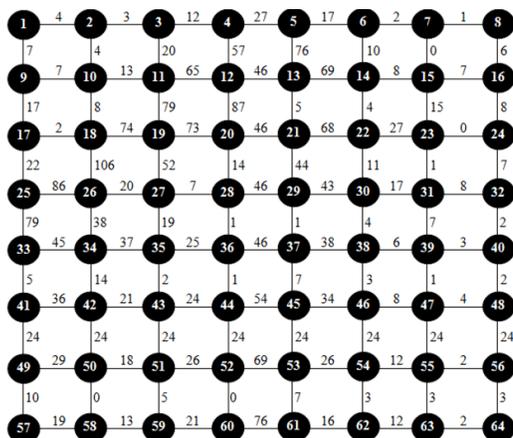
Misalkan graf G adalah graf yang dibangun dari suatu citra digital, dan komponen terhubung dari graf G di pandang sebagai segmen dari citra digital tersebut. Teorema 2 mengatakan bahwa walaupun

walaupun S , and T pohon perentang minimum dari graf G berbeda, tetapi setiap komponen terhubung dari S mempunyai pasangan yang sama komponen terhubung dari T yang memuat titik-titik yang sama dari graf G . Misalkan graf G adalah graf yang dibangun dari suatu citra digital, dan komponen terhubung dari graf G dipandang sebagai segmen dari citra digital tersebut. Kita dapat menyimpulkan bahwa dua himpunan segmen-segmen citra yang dibangun oleh pohon perentang minimum yang berbeda adalah sama.

1.7. Segmentasi Citra

172	168	165	177	150	133	135	134
165	172	185	120	74	143	135	128
182	180	106	33	79	147	120	120
160	74	54	47	93	136	119	127
81	36	73	48	94	132	126	129
86	50	71	47	101	135	127	131
81	52	70	44	113	139	127	129
71	52	65	44	120	136	124	126

a. Nilai Intensitas Pixel



b. Graf dari Gambar a

Gambar 2. Graf yang Dibentuk dari Citra Digital

Graf digunakan untuk menyajikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Citra digital merupakan salah satu contoh graf *grid* berbobot, dimana gambar dapat dipandang sebagai kumpulan titik-titik pixel, dan antara pixel-pixel yang berdekatan dihubungkan. Bobot dari hubungan tersebut juga dapat diberikan dengan selisih intensitas pixel-pixel yang berdekatan. Salah satu cara pembentukan graf dari citra digital dapat dilihat pada Gambar 2.

Algoritma Kruskal dikerjakan untuk graf pada Gambar 2(b), sehingga diperoleh pohon perentang minimum dari garaf. Selanjutnya rata-rata, \bar{w} , dari bobot semua sisi pada pohon perentang minimum dihitung, sehingga diperoleh $\bar{w} = 8.5$. Demikian juga simpangan baku, σ , dari semua bobot sisi pohon perentang minimum dihitung sehingga diperoleh $\sigma = 10.9$. Jadi $\bar{w} + \sigma = 19.4$ digunakan sebagai nilai ambang batas. Jika semua sisi yang bobotnya lebih besar dari ambang batas dihapus dari pohon perentang minimum maka diperoleh sub-sub pohon pada Gambar 3.

Segmen yang hanya mempunyai 1 titik merupakan outlier yang mengganggu segmentasi citra digital. Kumpulan titik-titik outlier ini merupakan batas dari segmen-segmen yang lebih luas. Untuk setiap titik/pixel pada suatu sub pohon diberikan warna sama. Hasilnya [12] dapat dilihat pada Gambar 4.

2. Metode Penelitian

Penelitian ini dimulai dengan studi literatur tentang cara mengubah citra digital menjadi graf. Selanjutnya sifat-sifat graf dan pohon perentang minimum dipelajari. Akhirnya code program dalam bahasa Matlab dibuat untuk komputasi dan visualisasi pohon dalam citra digital.

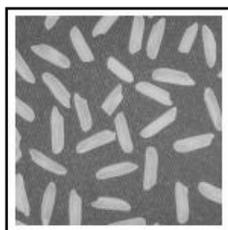
Citra digital yang digunakan dalam penelitian ini adalah citra digital *rice.png* yang ada dalam software Matlab. Karena keterbatasan kemampuan perintah *imshow* dalam Matlab maka citra digital ini dipotong hanya pixel-pixel pada 137×137 yang pertama seperti pada Gambar 5. Citra digital ini dipilih karena nilai intensitas warna butir berasnya hampir sama dan demikian juga nilai intensitas warnanya latar belakangnya hampir sama.

Citra digital ini diubah dalam graf dengan cara setiap pixel pada citra digital dipandang sebagai titik dalam graf. Visualisasi setiap titik diberikan sebagai satu pixel dengan intensitas warnanya sama dengan nol. Selanjutnya sisi diantara setiap dua pixel/titik digambarkan sebagai tiga titik yang nilai intensitas warnanya sebesar selisih nilai intensitas warna dari pixel yang berdekatan tersebut. Contoh citra digital pada Gambar 5 dirubah menjadi graf *super grid* seperti pada Gambar 6. Setiap pixel dihubungkan dengan delapan titik disekitarnya. Dengan cara yang sama dilakukan untuk graf *grid* yang setiap titik dihubungkan dengan empat titik disekitarnya (di sebelah atas, bawah, kiri, dan kanan dari titik) dan graf *triangular grid* yang

setiap titik dihubungkan dengan enam titik disekitarnya (di sebelah atas, bawah, kiri, kanan, kiri atas, dan kanan bawah dari titik).

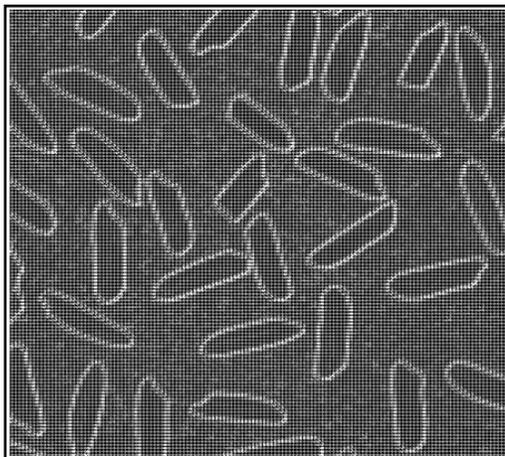
Setelah graf terbentuk, algoritma Kruskal digunakan untuk menentukan pohon perentang minimum dari graf yang sudah dibentuk dari suatu citra digital. Selanjutnya dihitung rata-rata, \bar{w} , dan simpangan baku, σ , dari semua sisi pada pohon tersebut. Nilai $\bar{w} + \sigma$ digunakan sebagai nilai ambang batas. Semua sisi yang lebih besar dari ambang batas dihapus dari pohon dan akan membentuk sub-sub pohon. Satu sub pohon dipandang sebagai satu segmen. Pembentukan graf dari suatu citra pada Gambar 5 yang terbaik adalah setiap butir beras merupakan satu sub pohon dari pemotongan pohon perentang minimum dari citra.

3. Hasil Penelitian dan Pembahasan



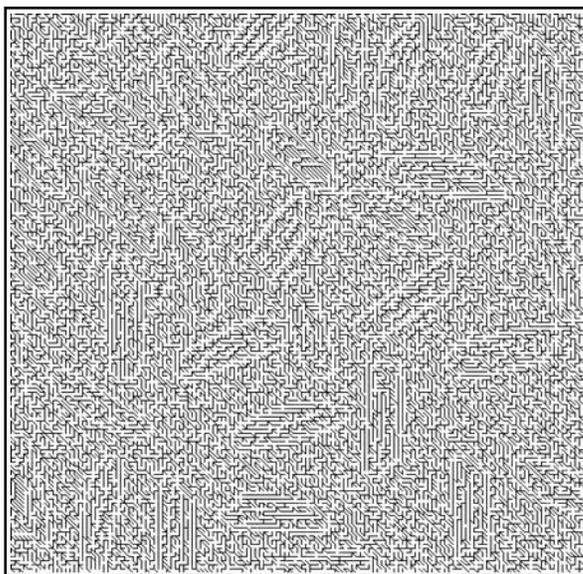
Gambar 5. Rice.png dari Matlab

Citra yang digunakan dalam sub Bab ini adalah citra yang ada di MatLab, yaitu: rice.png. Karena keterbatasan perintah *imshow* dalam MatLab untuk matriks yang berukuran besar maka citra ini hanya diambil sebagian, yaitu: 137×137 pixel, seperti Gambar 5. Gambar ini berwarna abu-abu dengan nilai intensitas warna di antara nilai 0 dengan 225.



Gambar 6. Graf *Super Grid* dari Citra *rice.png*

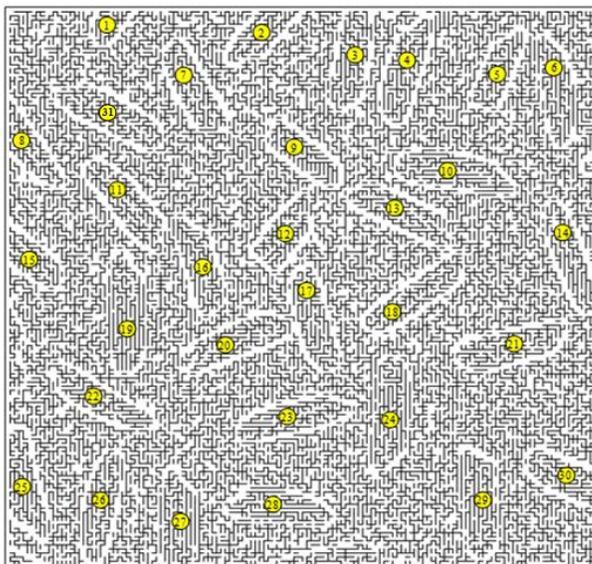
Citra *rice.png* dapat disajikan dalam bentuk graf, misalnya: graf *super grid*. Pixel-pixel citra pada Gambar 5 dipandang sebagai titik dari graf. Untuk graf *super grid*, setiap titik dihubungkan dengan delapan titik/ pixel disekitarnya. Bobot suatu sisi sama dengan selisih nilai intensitas dua titik/ pixel yang dihubungkannya. Semakin besar nilai bobot sisi maka warna sisinya semakin putih dalam gambar. Untuk dapat diamati, panjang sisi digambarkan dalam tiga pixel, sedangkan besar bobot dari sisi tersebut dapat dilihat dari warna sisi tersebut. Dengan menggunakan cara seperti ini, penyajian Gambar 5 dapat dilihat dalam Gambar 6



Gambar 7. Pohon dari Graf *Triangular Grid* untuk Citra *rice.png*

Selanjutnya, jika pohon perentang minimum dari graf *triangular grid* dibangun maka gambarnya akan tampil seperti pada Gambar 7. Kebanyakan sisi-sisi antara beras dan latar belakang tidak diikuti dalam pohon perentang minimum karena nilai bobotnya yang sangat besar.

Citra digital pada Gambar 5 merupakan gambar dari 31 beras yang nilai intensitas warnanya hampir sama. Warna beras sangat kontras dengan warna latar belakangnya. Hasil segmentasi citra digital beras tersebut dengan menggunakan pohon perentang minimum diharapkan bahwa setiap beras terdiri dari satu pohon yang titik-titiknya merupakan pixel dari satu beras pada Gambar 5.



Gambar 8. Segmentasi dengan Pohon Perentang Minimum untuk Citra *rice.png*

Setelah Gambar 5 diubah menjadi graf *grid*, segmentasi dengan menggunakan pohon perentang minimum diperlakukan kepada graf *grid*. Hasilnya dapat dilihat pada Gambar 8. Setelah Gambar 8 diperiksa, setiap butir beras pada citra digital Gambar 5 merupakan 1 sub pohon pada Gambar 8. Karena setiap sub pohon mewakili satu segmen maka segmentasi dengan pohon perentang minimum dengan ambang batas $(\sigma + \bar{w})$ cocok untuk citra digital yang diubah ke graf *grid*. Dengan cara yang sama, segmentasi dengan pohon perentang minimum dengan ambang batas $(\sigma + \bar{w})$ cocok untuk citra digital yang diubah ke graf *triangular grid*.

Kemudian citra digital beras Gambar 5 diubah menjadi graf *super grid*. Pohon perentang minimum untuk graf *super grid* tersebut ditentukan. Selanjutnya sisi pohon yang bobotnya lebih besar dari ambang batas $(\sigma + \bar{w})$ dihapus dari pohon sehingga sub-sub pohon akan terbentuk. Tetapi tidak semua citra digital dari suatu beras terdiri dari satu sub pohon. Hanya 8 citra

digital beras yang setiap butir berasnya terdiri dari satu sup pohon, yaitu butir beras nomor 1, 4, 7, 8, 11,15, 26, dan 35. Sedangkan butir beras nomor 2, 5, 9, 10, 12, 14, 18, 20, 22, 23, 24, 28, dan 30 terdiri dari beberapa sup pohon. Berarti satu butir beras dibuat dalam beberapa segmen. Hal ini tidak seperti yang kita harapkan. Akhirnya butir beras dengan nomor lain diperoleh bukan merupakan sub pohon dan masih terhubung dengan sub pohon latar belakang. Kasus ini juga tidak diharapkan. Jadi segmentasi dengan pohon perentang minimum dengan ambang batas ($\sigma + \bar{w}$) tidak cocok untuk citra digital yang diubah ke graf *super grid*.

4. Kesimpulan

Ada tiga hal yang dapat disimpulkan pada penelitian ini, yaitu: pertama, segmentasi dengan pohon perentang minimum dengan ambang batas ($\sigma + \bar{w}$) cocok untuk citra digital yang diubah ke graf *grid*. Kedua, segmentasi dengan pohon perentang minimum dengan ambang batas ($\sigma + \bar{w}$) cocok untuk citra digital yang diubah ke graf *triangular grid*. Ketiga, segmentasi dengan pohon perentang minimum dengan ambang batas ($\sigma + \bar{w}$) tidak cocok untuk citra digital yang diubah ke graf *super grid*. Sedangkan saran bagi orang berkecimpung dalam visi komputer adalah metode pohon perentang minimum dapat digunakan untuk mensegmentasi citra digital. Tetapi citra digital sebaiknya diubah ke dalam graf *grid*, atau graf *triangular grid*.

5. Referensi

Freedman, D., and Kisilev P.,(2010), "KDE Paring and a Faster Mean Shift Algorithm," SIAM J. Imaging Sci., 3(4), pp. 878–903.

- Kallasi, F., D.L. Rizzini, F. Oleari, and J. Aleotti. (2015). Computer Vision in Underwater Environments: a Multiscale Graph Segmentation Approach. *Proceeding of the OCEANS 2015-Genova*. IEEE. p. 1-6.
- Korte, B. and J. Vygen. (2006). *Combinatorial Optimization: Theory and Algorithms, Third Edition*. Springer, Germany.
- Mayr, E. W., and Plaxton, C. G., (1992), "On the spanning trees of weight graphs," *Combinatorica*, 12(4), pp. 433-44.
- Manik, E., (2010), "Pengaruh Bandwidth Terhadap Segmentasi Citra Digital Dengan Menggunakan Mean Shift," *VISI*, 18 (1), pp. 43 -49.
- Manik, E. (2018). The minimum spanning tree in stereo vision. *IOP Conf. Series: Materials Science and Engineering 420*
- Manik, E. , S. Suwilo, Tulus, and O.S. Sitompul. (2017). The Uniqueness of Image Segmentation Generated by Different Minimum Spanning Tree. *Global Journal of Pure and Applied Mathematics*.Vol. 13(7). pp. 2975–2982.
- Manik, E., S. Suwilo, Tulus, and O.S. Sitompul. (2017). The Uniqueness of Image Segmentation Generated by Different Minimum Spanning Tree. *Global Journal of Pure and Applied Mathematics*, 13(7), pp. 2975–2982.
- Manik, E., S. Suwilo, Tulus, and O. S. Sitompul. (2018). On the 5-Local Profiles of Trees. *IOP Conf. Series: Materials Science and Engineering 300*
- Peter, S.J. (2010). Minimum Spanning Tree-based Structural Similarity Clustering for Image Mining with Local Region Outliers. *IJCA* Vol. 8(6), p. 0975–8887.
- Rosen, K. H. (2007). *Discrete Mathematics and its applications, 6th ed.* McGraw-Hill.
- Uemura, T., Koutaki, G., and Uchimura, K., (2011), "Image Segmentation Based on Edge Detection Using Boundary Code," *Proc. ICIC International*.
- Winston, W.L. (2004). *Operations Research: Applications and Algorithms, Fourth Edition*. USA: Thomson.
- Zhang, F., L.D. S. Xiang, and X. Zhang . (2015). Segment Graph Based Image Filtering: Fast Structure-Preserving Smoothing. *Proceeding of the IEEE ICCV*. p. 361-369.