

MEMBATASI k-KETENGGAAAN SIMPUL DALAM PEMBANGKITAN RANDOM GRAPH METODE ERDOS ROYI UNTUK MENINGKATKAN KINERJA KOMPUTASI

Zainal Abidin¹ dan Agus Zainal Arifin²

¹Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

²Jurusan Informatika Fakultas Teknologi Informatika Institut Teknologi Sepuluh Nopember

Keputih, Sukolilo, Surabaya, Indonesia

e-mail : ¹ br52s@cs.its.ac.id, ²agusza@cs.its.ac.id

ABSTRACT

Edges generation by random graph erdos-royi methods was needed high computation, it's caused low performance. In fact, edge generation was used frequently with many nodes. this paper is described a node restriction by k-nearest neighbour on edge generation of random graph erdos royi method. Result of node restriction by k-nearest neighbour can be reduced computation time.

Keywords: random Graph, erdos royi, k-nearest neighbour, computation time.

PENDAHULUAN

Graph adalah himpunan simpul dan busur yang menghubungkan semua atau beberapa simpul (Diestel, 2000). *Graph* dengan anggota simpul-simpul tidak saling terhubung antar satu dan simpul lain disebut *Graph* terisolasi. Pembangkitan busur dari *Graph* terisolasi sering digunakan sebagai sarana untuk simulasi.

Simulasi *Graph* sering digunakan untuk mengetahui hubungan antar penulis dan pembimbing dalam melakukan aktifitas penelitian (Newman, 2001a). *Graph* yang terbentuk bisa diketahui seberapa banyak seorang peneliti melakukan pembimbingan. Banyaknya bimbingan dapat diketahui dari jumlah busur yang terhubung keluar dari simpul peneliti dan bobotnya. *Graph* dipakai untuk melihat karya peneliti terbaik (Newman, 2001b). Karya tulis dianggap sebuah simpul kemudian dibangkitkan busur kearahnya jika ada karya tulis yang mengacu pada karya tulis tersebut.

Graph digunakan untuk mengetahui sebaran dari sel tumor otak (Gunduz, 2004; Demir 2005). Sel-sel dari otak dianggap sebagai suatu simpul. Simpul-simpul yang diperoleh dibangkitkan busur-busur menggunakan suatu probabilitas. Probabilitas digunakan untuk menentukan batas apakah suatu simpul terhubung atau tidak terhubung dengan simpul yang lain. *Graph* dipakai mengukur dan menganalisa kerapatan (Abidin dan Arifin, 2008; Abidin dan Arifin, 2009).

Simpul-simpul yang digunakan untuk membangkitkan busur berjumlah ribuan. Simpul digunakan dalam deteksi sebaran sel tumor

sejumlah sekitar 4000 buah (Gunduz, 2004; Demir 2005). Simpul digunakan dalam analisa kerapatan berjumlah antara 2000 sampai dengan sekitar 7000 buah (Abidin dan Arifin, 2008; Abidin dan Arifin, 2009). Random *Graph* dengan metode erdos royi adalah membangkitkan *Graph* dari *Graph* terisolasi dengan membangkitkan busur dari setiap simpul dengan semua simpul dengan suatu batasan sebuah probabilitas (Watts dan Strogatz, 1998). Menghubungkan setiap simpul ke semua simpul yang lain memerlukan komputasi yang besar. Persamaan 1 merupakan jumlah busur B yang mungkin terbentuk dengan jumlah simpul n.

$$B = \frac{1}{2}n(n-1). \quad (1)$$

Komputasi besar membuat kinerja komputer jadi rendah. Prosesor dari komputer jadi sibuk. Pembangkitan random *Graph* metode erdos royi masih dapat ditingkatkan kinerjanya. Pembatasan dalam bentuk probabilitas antar simpul dapat digunakan sebagai dasar peningkatan kinerja. Jika pembangkitan dibatasi dengan probabilitas, maka tidak perlu setiap simpul dicoba dibangkitkan busur dan dihitung probabilitasnya, tetapi mungkin hanya perlu dicoba pada simpul-simpul terdekatnya saja. Sejumlah n simpul terdekat atau ketetanggaan diberi notasi n-ketetanggaan. Dalam penelitian ini menjelaskan tentang pembatasan simpul sejumlah n-ketetanggaan untuk peningkatan kinerja pada pembangkitan *Graph* dengan metode erdos dan royi.

KAJIAN GRAPH

A. Dasar-dasar Graph

Graph (G) adalah himpunan simpul (*vertex*, V) dan busur (*edge*, E), ditulis dengan $G=(V, E)$ (Reinhard Diestel, 2000). *Graph* ditampilkan dalam titik dan garis. Titik adalah lambang dari simpul. Garis merupakan lambang dari busur yang menghubungkan antara dua simpul. Ukuran (*size*, *order*) *Graph* G , $|G|$ adalah jumlah simpul yang menjadi anggota himpunan dari *Graph*, walaupun simpul tersebut tidak dihubungkan oleh suatu busur. Jumlah busur dituliskan dengan $||G||$. **Gambar 1**, contoh *Graph* dengan ukuran 7. Simpul nomor enam tidak terhubung ke simpul yang lain.

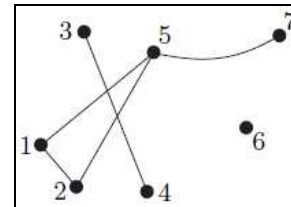
Graph pada **Gambar 1** dapat ditulis dalam bentuk himpunan simpul V dan busur E , misal, $V = \{1, 2, 3, 4, 5, 6, 7\}$, $E = \{\{1,2\}, \{1,5\}, \{2,5\}, \{3,4\}, \{5,7\}\}$. Busur $\{x,y\}$ dapat tulis dengan busur xy atau yx . Simpul x dan y dari *Graph* G dikatakan saling berketetanggaan, jika xy adalah busur *Graph* G . Jika semua simpul dalam *Graph* G saling berpasangan satu sama lain, maka *Graph* G disebut komplet (*complete*) dituliskan dengan K^n , dimana n adalah jumlah dari simpul.

k -ketetanggaan terdekat (k -NN) dari simpul i bisa diperoleh dengan menarik sebuah lingkaran dengan berpusat pada simpul i sampai diperoleh k simpul lain yang berada dalam lingkaran. **Gambar 2**, 3-ketetanggaan terdekat dari simpul A adalah tiga simpul, yaitu simpul B, C, dan D. 7-ketetanggaan terdekat dari simpul A diperoleh dengan memperpanjang jari-jari lingkaran sampai diperoleh 7 simpul yang berada dalam lingkaran, yaitu simpul B, C, D, E, F, G, dan H. Dua simpul (I dan J) bukan anggota dari 7-ketetanggaan terdekat dari simpul A, karena berada diluar lingkaran.

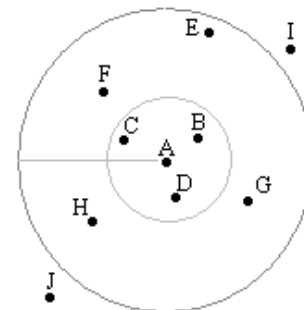
Graph terdiri dari berbagai jenis (Newman, 2003), yaitu *Graph* berarah, tak berarah, berbobot, dan tak berbobot. *Graph* tak berarah (*undirected Graph*) adalah pasangan busur xy sama dengan pasangan busur yx . Busur tersebut dikatakan sebagai busur tak berarah. Simpul x dan y disebut sebagai *titik akhir* (*endpoint*). Sebuah *Graph* G disebut *Graph* tak berarah jika setiap busurnya terhubung tak berarah (Levitin, 2005).

Jika pasangan busur xy tidak sama dengan busur yx , maka busur tersebut disebut sebagai busur berarah. Busur xy meninggalkan x menuju y disebut juga x sebagai *ekor*, dan y sebagai *kepala*. *Graph* G disebut sebagai *Graph* berarah jika semua busur terhubung secara berarah (Levitin, 2005). **Gambar 3a** adalah contoh penggambaran dari *Graph* tak berarah. Busur

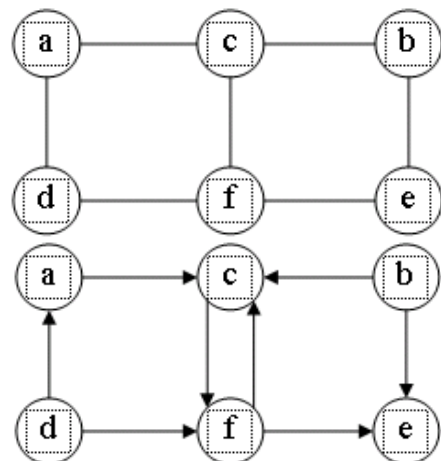
yang menghubungkan antar simpul tidak mempunyai tanda arah. **Gambar 3b** adalah contoh *Graph* berarah. Busur antar simpul pada **Gambar 3b** mempunyai simbol anak panah yang menunjukkan arah hubungan dari simpul *ekor* ke simpul *kepala*. Simpul c dan f mempunyai dua busur, yaitu cf dan fc .



Gambar 1. *Graph* dengan tujuh simpul dan empat busur (Diestel, 2000).



Gambar 2. Ilustrasi k -ketetanggaan terdekat



Gambar 3. *Graph* tidak berarah dan berarah.
(a) *Graph* tak berarah, (b) *Graph* berarah
(Levitin, 2005)

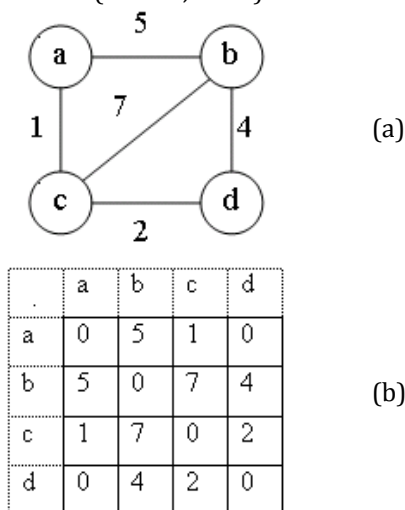
Untuk kepentingan suatu komputasi atau sebuah *algoritma*, secara umum *Graph* dapat digambarkan dengan bentuk matriks, disebut sebagai matriks *adjacency*. Matriks *adjacency* dari *Graph* dengan ukuran n adalah matriks $n \times n$. Setiap elemen dari matriks mewakili satu busur dari *Graph*. Elemen baris ke i dan kolom ke j bernilai satu jika simpul ke i terhubung dengan simpul ke j . Elemen baris ke i dan kolom ke j bernilai nol jika simpul ke i tidak terhubung dengan simpul ke j (Levitin, 2005). **Gambar 4**

merupakan *adjacency* matriks dari *Graph* tak berarah pada **Gambar 3a**. *Graph* tak berarah mempunyai matriks *adjacency* yang simetris.

Graph berbobot (*weighed graph*) merupakan *Graph* dengan suatu nilai pada simpul atau busur. *Graph* tak berbobot (*unweighed Graph*) adalah *Graph* dengan simpul dan busurnya tidak mempunyai nilai. Nilai bisa hanya dimiliki oleh salah satu elemen dari *Graph*, simpul saja atau hanya busur. Tetapi yang sering, nilai dimiliki oleh busur. Nilai pada simpul digunakan untuk mewakili jumlah keanggotaan, luasan, atau besaran suatu simpul. Nilai pada busur digunakan untuk mewakili jumlah busur yang terhubung dengan sepasang simpul, jarak dua simpul, atau biaya yang dibutuhkan untuk melewati busur. **Gambar 5** adalah contoh *Graph* berbobot dengan matriks *adjacency*.

	a	b	c	D	e	f
a	0	0	1	1	0	0
b	0	0	1	0	0	1
c	1	1	0	0	1	0
d	1	0	0	0	1	0
e	0	0	1	1	0	1
f	0	1	0	0	1	0

Gambar 4. *Graph* dalam adjacency matriks (Levitin, 2005)

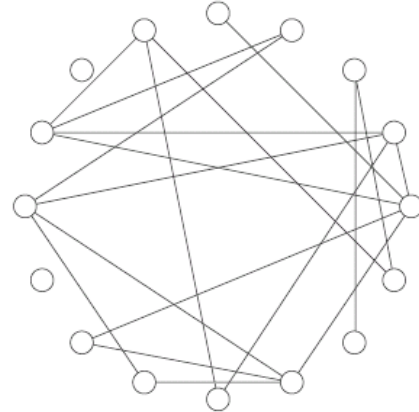


Gambar 5. (a) *Graph* berbobot, (b) matriks dari *Graph* berbobot.

Ditinjau dari jumlah penghubung dalam setiap simpul, terdapat pasangan simpul dengan busur jamak (*multi edge*) dan pasangan simpul dengan busur secara tunggal (*single edge*). *Graph* dengan busur jamak, simpul x dan y dihubungkan dengan lebih dari satu simpul. **Gambar 3b** merupakan *Graph* berbusur jamak. Penggambaran dalam matriks *adjacency* berupa matriks berbobot. Bobot dalam *Graph* busur jamak mewakili dari jumlah busur yang menghubungkan antara simpul x dan simpul y .

B. Random Graph

Random Graph pertama kali dikenalkan oleh Erdős dan Rényi tahun 1959 (Diestel, 2000). *Random Graph* adalah *Graph* dengan simpul sejumlah n dan setiap pasang simpulnya terhubung atau tidak terhubung dengan suatu probabilitas p atau $(1-p)$ (Newman, 2003). *Random Graph* di atas dinotasikan sebagai $G_{n,p}$. Secara teknis, *Graph* G dengan m adalah jumlah busur yang muncul, maka probabilitas kemunculan busur dinyatakan dalam persamaan 8.



Gambar 6. *Graph* dibangun dengan model Erdos dan Renyi, $N = 16$ dan $p = 1/7$ (Newman, 2001c)

$$p^m (1-p)^{M-m}, \quad (8)$$

dimana

$$M = \frac{1}{2} n(n-1). \quad (9)$$

dengan M adalah maksimum jumlah busur yang mungkin terjadi, persamaan 9. Dari persamaan 8 dan 9 di atas, sering muncul *random Graph* yang dinotasikan dengan $G_{n,m}$. **Gambar 6** merupakan contoh *random Graph* yang dibangun dengan model Erdos dan Renyi dengan probabilitas, $p = 1/7$.

BAHAN DAN METODE

A. Bahan

Bahan untuk uji coba pembatasan k -ketetanggaan simpul menggunakan citra tiruan berwarna hitam. Citra tiruan hitam penuh dikenakan pengotoran dengan derau putih. Ukuran citra tiruan untuk uji coba adalah 200×500 piksel. Pengotoran citra hitam penuh dengan derau menggunakan teknik pengotoran *salt and paper* (Gonzales, 2002). Tingkat kepadatan derau pada citra tiruan mulai 0,003 dan 0,201. Citra tiruan yang dipakai untuk bahan uji coba sejumlah 100 buah.

Citra tiruan yang telah dikotori dengan derau putih digunakan sebagai bahan model dari

Graph. Satu piksel putih pada citra tiruan dianggap sebagai sebuah simpul pada *Graph*. Model menghasilkan *Graph* dengan simpul-simpul yang tersebar secara acak. *Graph* yang dihasilkan berupa *Graph* dengan simpul-simpul yang tidak saling terhubung atau disebut sebagai *Graph* terisolasi. Simpul-simpul dalam *Graph* terisolasi digunakan sebagai bahan untuk membangkitkan busur-busur dengan metode erdos royi. **Gambar 7** potongan dari citra dengan ukuran 20x50 piksel yang telah diperbesar 400%.

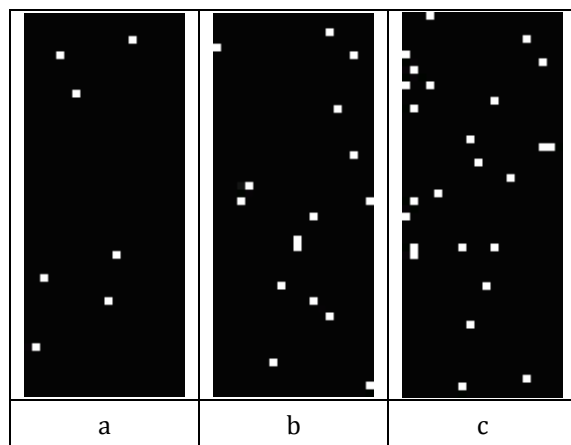
Data jumlah simpul pada 100 citra tiruan hasil pengkotoran citra hitam penuh dengan menggunakan teknik pengkotoran *salt and paper* terdapat di dalam **Tabel 1**. Pada **Tabel 1** tercantum jumlah simpul atau piksel putih dan tingkat kepadatan simpul dalam citra tiruan. Jumlah simpul terkecil 61 buah dan terbesar adalah 9951. Variasi jumlah simpul untuk mengetahui kelebihan dan kelemahan *random Graph* dengan metode erdos royi dengan *k*-NN.

B. Metode

Metode untuk menurunkan komputasi pembentukan *Graph* dengan metode *random Graph* erdos dan royi terdiri dari tiga tahapan. Tiga tahapan itu adalah : inialisasi *Graph*, mencari *k* ketetanggaan simpul, dan menghubungkan setiap simpul dengan *k* tetangga dengan probabilitas *P*. **Gambar 8** diagram alir metode *random Graph* erdos royi dengan *k*-NN.

1) Inialisasi *Graph*

Pada tahapan inialisasi *Graph* digunakan untuk menentukan nilai-nilai parameter awal yang dipakai untuk membangun *random Graph* metode erdos royi yang diintegrasikan dengan *k*-NN. Dua parameter awal adalah jumlah ketetanggaan dari setiap simpul, *k* dan jarak terjauh dari semua pasangan simpul, *L*.



Gambar 7. Citra sampel yang telah dikotori dengan *salt and paper*. (a) potongan dari citra dengan jumlah simpul 937. (b) potongan dari citra yang telah terkotori dengan jumlah simpul

2041. (c) potongan dari citra yang telah terkotori dengan jumlah simpul 2933.

$$P(v,u) = \alpha \cdot e^{-d(u,v)/\beta \cdot L} \quad (2)$$

Nilai *L* digunakan untuk menentukan nilai probabilitas antar dua simpul, $P(u,v)$, dengan metode waxman (Gunduz, 2004), persamaan 2, dimana α dan β adalah bilangan konstan dengan besar antara nol sampai dengan satu. $d(u,v)$ adalah jarak *euclidean* antara simpul *u* dan *v*. Dalam *Graph*, *L* bisa diperoleh dengan persamaan 3, dimana *skala* adalah lebar dimensi dari *Graph*. Dalam penelitian ini, *Graph* dibangun berdasarkan citra, maka *L* diperoleh dari panjang diagonal citra, persamaan 4, dimana *p* dan *l* merupakan panjang dan lebar citra sampel.

$$L = \sqrt{2} \cdot skala, \quad (3)$$

$$L = \sqrt{p^2 + l^2}, \quad (4)$$

Random Graph metode erdos royi menghubungkan setiap simpul (*n*) ke semua simpul lain (*n*-1). Keterhubungan pasangan simpul memperhatikan probabilitas waxman, seperti dalam persamaan 2. Dengan kata lain, *random Graph* metode erdos dan royi mencoba memeriksa keterhubungan semua kemungkinan pasangan simpul, $n(n-1)$.

Di sisi lain, jika *Graph G* dengan setiap simpul dihubungkan dengan *k* ketetanggaan dan *k* jauh lebih besar dari $\ln(n)$, maka *Graph G* dijamin menjadi *Graph* terhubung (Watts dan Strogatz, 1998, Distel, 2000), sesuai persamaan 5. Syarat agar *random Graph* menjadi terhubung, seperti pada persamaan 6 (Watts dan Strogatz, 1998, Distel, 2000).

$$k \gg \ln(n). \quad (5)$$

$$n \gg k \gg \ln(n) \gg 1. \quad (6)$$

Dalam penelitian ini, untuk mendapatkan nilai *k* jauh lebih besar dari $\ln(n)$, *k* diperoleh dengan dua pangkat pembulatan ke bawah $\ln(n)$, seperti persamaan 7.

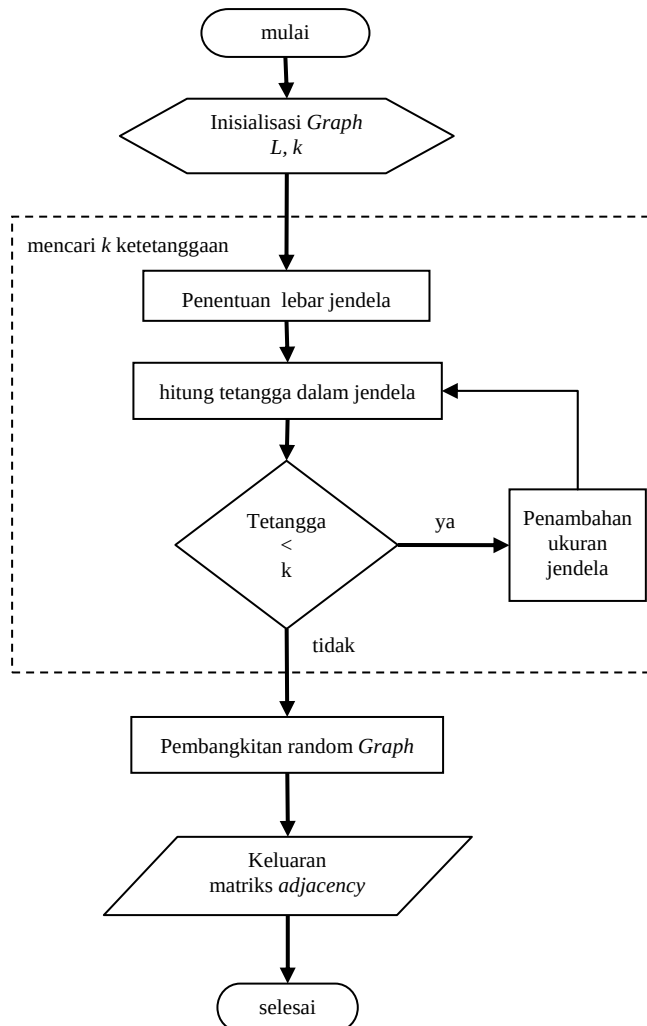
$$k = 2^{\lfloor \ln(n) \rfloor}. \quad (7)$$

Random Graph metode erdos dan royi dengan *k*-NN menghubungkan setiap simpul (*n*) dengan *k* ketetanggaannya. Sehingga *Graph* dapat dihasilkan dengan *random Graph* metode erdos dan royi dengan *k*-NN. Komputasi dari *random Graph* metode erdos dan royi dengan *k*-NN adalah $n(k+k^2)$.

2) Mencari *k* Ketetanggaan Simpul.

Setelah diperoleh jumlah tetangga dari setiap simpul adalah *k*, tahapan selanjutnya

adalah mencari simpul yang menjadi tetangga dari setiap simpul dengan jumlah k . k tetangga terdekat dapat diperoleh dengan membuat jendela persegi panjang. Ukuran panjang sisi adalah pembulatan ke atas akar k , seperti persamaan 8.



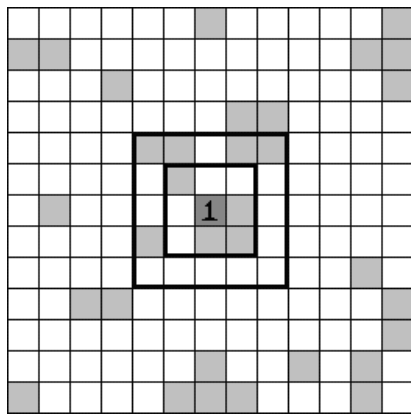
Gambar 8. Diagram alir metode *random Graph* erdos dan royi dengan k -NN

Tabel 1. Data jumlah simpul dalam citra tiruan untuk uji coba

Nama File	Kerapatan	Jumlah simpul
UG001.TIF	0.003	61
UG002.TIF	0.005	153
UG003.TIF	0.007	257
UG004.TIF	0.009	351
UG005.TIF	0.011	459
UG006.TIF	0.013	559
UG007.TIF	0.015	610
UG008.TIF	0.017	789
UG009.TIF	0.019	919
UG010.TIF	0.021	937
UG011.TIF	0.023	1080
UG012.TIF	0.025	1101
UG013.TIF	0.027	1258
UG014.TIF	0.029	1338

UG015.TIF	0.031	1482
UG016.TIF	0.033	1627
UG017.TIF	0.035	1695
UG018.TIF	0.037	1781
UG019.TIF	0.039	1861
UG020.TIF	0.041	2041
UG021.TIF	0.043	2156
UG022.TIF	0.045	2192
UG023.TIF	0.047	2237
UG024.TIF	0.049	2289
UG025.TIF	0.051	2482
UG026.TIF	0.053	2481
UG027.TIF	0.055	2713
UG028.TIF	0.057	2836
UG029.TIF	0.059	2889
UG030.TIF	0.061	2933
UG031.TIF	0.063	3046
UG032.TIF	0.065	3167
UG033.TIF	0.067	3384
UG034.TIF	0.069	3397
UG035.TIF	0.071	3504
UG036.TIF	0.073	3540
UG037.TIF	0.075	3572
UG038.TIF	0.077	3673
UG039.TIF	0.079	3837
UG040.TIF	0.081	4004
UG041.TIF	0.083	3970
UG042.TIF	0.085	4082
UG043.TIF	0.087	4221
UG044.TIF	0.089	4424
UG045.TIF	0.091	4417
UG046.TIF	0.093	4491
UG047.TIF	0.095	4631
UG048.TIF	0.097	4778
UG049.TIF	0.099	4696
UG050.TIF	0.101	4897
UG051.TIF	0.103	5036
UG052.TIF	0.105	5150
UG053.TIF	0.107	5249
UG054.TIF	0.109	5336
UG055.TIF	0.111	5428
UG056.TIF	0.113	5546
UG057.TIF	0.115	5685
UG058.TIF	0.117	5773
UG059.TIF	0.119	5925
UG060.TIF	0.121	5952
UG061.TIF	0.123	6041
UG062.TIF	0.125	6088
UG063.TIF	0.127	6072
UG064.TIF	0.129	6255
UG065.TIF	0.131	6515
UG066.TIF	0.133	6409
UG067.TIF	0.135	6542
UG068.TIF	0.137	6736
UG069.TIF	0.139	6794
UG070.TIF	0.141	6991
UG071.TIF	0.143	7017
UG072.TIF	0.145	7217
UG073.TIF	0.147	7297
UG074.TIF	0.149	7281
UG075.TIF	0.151	7502
UG076.TIF	0.153	7543
UG077.TIF	0.155	7611
UG078.TIF	0.157	7843
UG079.TIF	0.159	7906

UG080.TIF	0.161	7995
UG081.TIF	0.163	7986
UG082.TIF	0.165	8135
UG083.TIF	0.167	8362
UG084.TIF	0.169	8276
UG085.TIF	0.171	8564
UG086.TIF	0.173	8592
UG087.TIF	0.175	8787
UG088.TIF	0.177	8709
UG089.TIF	0.179	8724
UG090.TIF	0.181	9087
UG091.TIF	0.183	9044
UG092.TIF	0.185	9092
UG093.TIF	0.187	9170
UG094.TIF	0.189	9537
UG095.TIF	0.191	9380
UG096.TIF	0.193	9504
UG097.TIF	0.195	9693
UG098.TIF	0.197	9667
UG099.TIF	0.199	9886
UG100.TIF	0.201	9951



Gambar 9 : Ilustrasi pembuatan jendela k -NN dengan $n=34$ dan $k=8$,

$$\lceil \sqrt{k} \rceil. \quad (8)$$

Diasumsikan bahwa semua bagian dalam jendela terisi dengan simpul secara penuh. Jika semua bagian jendela terisi penuh, maka terdapat jumlah tetangga $\geq k$. Jika dalam jendela terdapat tetangga terdekat kurang dari k , maka lebar sisi jendela ditambah sebesar $k/2$. **Gambar 9** merupakan ilustrasi pembuatan jendela k -NN. Ukuran *Graph* 34, diperoleh $\ln(n) = 3,5$ sehingga $k = 8$.

Pada ilustrasi dicari 8-ketetanggaan dari titik berlabel angka satu. Proses inisialisasi diperoleh lebar jendela tiga dan titik yang menjadi tetangganya delapan. Pada jendela ukuran 3×3 hanya terdapat empat simpul tetangga, jumlah simpul masih kurang dari k . Lebar jendela ditambah untuk mendapatkan jumlah tetangga lebih besar atau sama dengan k . Hasil dari penambahan lebar jendela diperoleh simpul tetangga sejumlah sembilan.

3) Pembangunan Random Graph

Setiap memperoleh k tetangga terdekat, suatu simpul dibuat busur yang menghubungkan simpul dengan semua k tetangganya. Dalam pembentukan *Graph* ditetapkan suatu nilai probabilitas setiap pasang simpul, P , dengan persamaan 2. Dengan kondisi di atas bisa dihasilkan *random Graph* $G_{n,p}$.

Tabel 2. Simulasi perhitungan probabilitas *waxman*

D	P
1	0.2309609
2	0.0561505
3	0.0136511
4	0.0033188
5	0.0008069
6	0.0001962
7	0.0000477
8	0.0000116
9	0.0000028
10	0.0000007

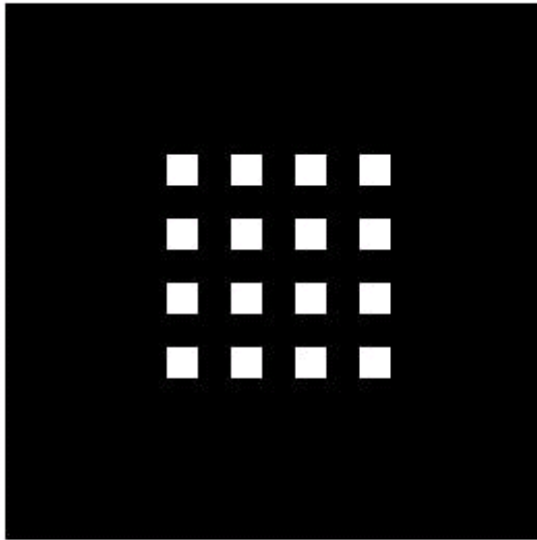
Persamaan 2 menghasilkan probabilitas yang mendekati satu sampai mendekati nol. Jika jarak *euclidean* antara dua buah titik semakin dekat maka probabilitasnya semakin besar. Demikian pula sebaliknya, jika jaraknya jauh maka probabilitasnya semakin kecil, cenderung nol. **Tabel 2** uji coba rumus 21 dalam simulasi sepuluh angka dengan lebar *skala* 10×10 .

4) Pembangunan Graph

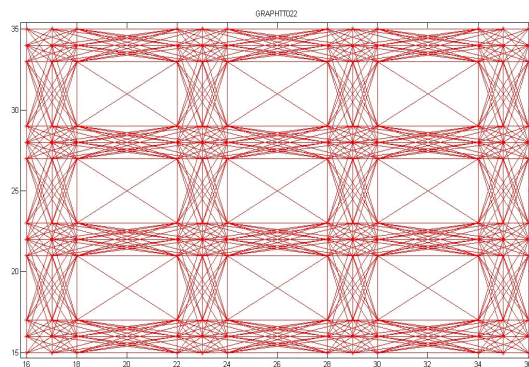
Hasil dari proses penentuan obyek berupa citra *biner*, nilai nol dan satu di citra *biner* dijadikan acuan pembentukan *Graph*. Piksel bernilai satu dianggap sebagai sebuah simpul. Asumsi bahwa simpul-simpul dalam sampel sebagai simpul terasing yang tidak terhubung dengan busur. **Gambar 10** merupakan citra tiruan simpul-simpul yang tersimpan dalam citra *biner*. Dalam citra tiruan terdapat 144 piksel putih yang berarti terdapat 144 simpul terasing. Satu kotak kecil berisikan 9 piksel putih.

Pembentukan *Graph* diawali dengan penghubungan setiap simpul dari citra sampel dengan busur. Pembangkitan busur-busur dalam pembentukan *graph* menggunakan *random graph* metode *erdos* dan *royi* dengan k -NN. Simpul yang saling terhubung dihitung probabilitasnya menggunakan metode *waxman*, persamaan 2. Nilai probabilitas rendah menunjukkan bahwa dua simpul mempunyai jarak yang jauh (panjang). Sebaliknya probabilitas tinggi menunjukkan bahwa dua simpul mempunyai jarak yang dekat (pendek), lihat **Tabel 2**.

Keterhubungan antar dua simpul dibatasi dengan nilai ambang. Jika nilai probabilitas dua simpul lebih kecil dari nilai ambang, maka busur yang menghubungkan dua simpul tersebut dihapus. Tujuan dari pemotongan garis penghubung yang mempunyai probabilitas rendah adalah untuk menghapus hubungan dua simpul yang jauh. Manfaatnya, simpul hanya terhubung dengan simpul-simpul yang dekat, sehingga dapat diketahui nilai-nilai karakter dari setiap simpul terhadap simpul tetangga terdekatnya.



Gambar 10. Citra tiruan simpul



Gambar 11. Graph dari citra tiruan.

Gambar 10 merupakan *Graph* hasil dari citra tiruan. Setiap simpul pada citra tiruan dihubungkan dengan nilai ambang probabilitas 0.2 dari citra berukuran 50x50. Tampak *graph* dengan simpul-simpul yang saling berdekatan mempunyai busur lebih banyak dibandingkan dengan yang letaknya berjauhan. Jumlah busur dari sebuah simpul dipengaruhi jumlah simpul tetangga yang dekat dan jarak antar simpul tetangga.

Tampak pada **gambar 11**, simpul-simpul yang mempunyai jarak dekat dengan tetangganya mempunyai jumlah busur lebih banyak dibandingkan dengan simpul yang jarak antar

tetangganya jauh. Jumlah busur pada setiap simpul merupakan *degree* dari simpul tersebut. Simpul yang mempunyai *degree* tinggi adalah simpul yang dikelilingi oleh simpul lain dengan jarak yang dekat. Sebaliknya simpul dengan *degree* rendah adalah simpul yang di sekitarnya terdapat simpul-simpul dengan jarak yang cenderung jauh. Simpul dengan jumlah tetangga sedikit cenderung mempunyai *degree* rendah. Simpul dengan tetangga sedikit biasanya berada dipinggir suatu area.

APLIKASI DAN PEMBAHASAN

1) Lingkungan Uji Coba

Uji coba dilakukan di komputer Acer Aspire 3610. Perangkat keras pendukungnya adalah prosesor 1.8 GHz, Kapasitas memori 2 Gbyte, Kapasitas Harddisk 40 Gbyte. Perangkat lunak pendukung adalah i windows xp service pack 1, Bahasa pemrograman menggunakan matlab versi 7.1

2) Skenario Uji Coba

Uji coba *random Graph* metode erdos royi dengan k -NN. Uji Coba akan membandingkan *random Graph* metode erdos royi murni dengan *random Graph* metode erdos royi dengan k -NN. Uji Coba ini untuk melihat kebutuhan waktu komputasi dari masing-masing metode.

Uji coba dilaksanakan guna mengetahui keberhasilan *random Graph* metode erdos dan royi dengan k -NN dalam menurunkan waktu komputasi pada saat membangkitkan suatu *graph*. Metode diujicobakan dengan 100 data (**tabel 4.1**). Data berupa citra yang dikotori dengan derau menggunakan teknik pengkotoran *salt and paper* (Gonzalez, 2002). Semua piksel putih dalam citra dianggap sebagai simpul-simpul terasing pada *Graph*. Semua data digunakan untuk masukan dalam pembangkitan *random Graph* menggunakan metode erdos dan royi. Kemudian hasil coba pertama dibandingkan dengan hasil uji coba pembangkitan *random Graph* yang dilaksanakan dengan metode erdos dan royi dengan k -NN.

3) Pelaksanaan dan Evaluasi Uji Coba

Uji coba dilaksanakan sesuai dengan skenario yang telah dibahas pada subbab Skenario Uji Coba. Uji coba dilaksanakan untuk mengevaluasi hasil-hasil uji coba. Hasil metode erdos royi murni dibandingkan dengan erdos royi dengan k -NN. Kedua metode diujicobakan pada citra tiruan diperoleh dari citra hitam penuh yang dikotori dengan derau. Pengkotoran menggunakan metode pengkotoran *salt and*

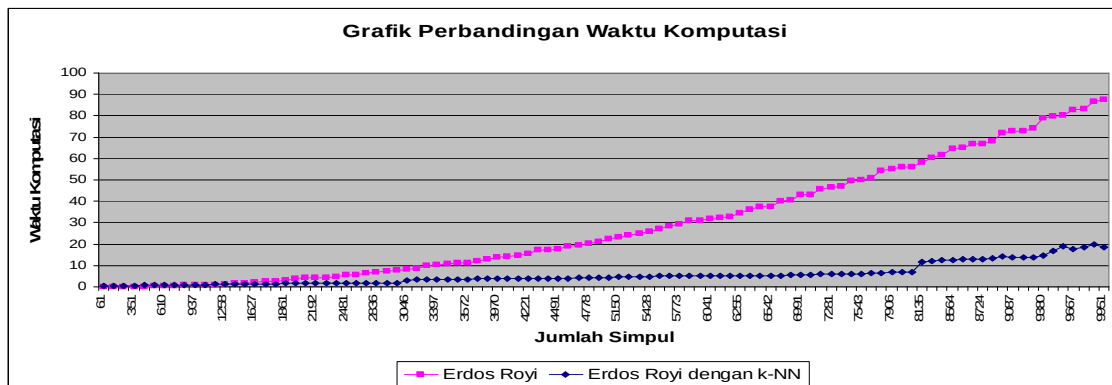
paper (Gonzalez, 2002). Jumlah citra 100 dengan kerapatan mulai 0.002 sampai dengan 0.2. Jumlah piksel putih lebih lengkap dapat dilihat pada **Tabel 1**. Perlakuan uji coba kedua metode dikenakan pada lingkungan uji coba tertera pada **Tabel 3**.

Gambar 12 perbandingan waktu komputasi dari dua metode. Pada **Gambar 12** waktu komputasi erdos royi lebih besar jika banding dengan erdos royi dengan k -NN. Data lebih lengkap tercantum dalam **Tabel 4**. Pada jumlah simpul kecil di bawah 1258, waktu komputasi *random Graph* metode erdos dan royi

lebih cepat. Tetapi waktu komputasi *random Graph* metode erdos dan royi dengan k -NN tidak terlalu lambat dibanding dengan komputasi *random Graph* metode erdos dan royi murni.

Tabel 3 : Parameter uji coba *random Graph*.

Parameter	Nilai
Probabilitas	0,5
L	538,516
Alfa	0,95
Beta	0,05



Gambar 12 : Grafik komputasi *random Graph* Erdos Royi dan Erdos Royi dengan k -NN.

Tabel 4 : Perbandingan waktu komputasi *random Graph* erdos dan royi dengan *random Graph* erdos royi k -NN.

Jumlah simpul	Kerapatan	Waktu Komputasi	
		ER k-NN	ER
61	0.003	0.57294	0.00697
153	0.005	0.62569	0.02100
257	0.007	0.54349	0.05890
351	0.009	0.47827	0.10926
459	0.011	0.88367	0.18505
559	0.013	0.86169	0.27676
610	0.015	0.86304	0.32878
789	0.017	0.82398	0.55010
919	0.019	0.83146	0.74452
937	0.021	0.80041	0.77365
1080	0.023	0.82323	1.02799
1101	0.025	1.45573	1.06906
1258	0.027	1.46713	1.39193
1338	0.029	1.44778	1.57528
1482	0.031	1.44725	1.93332
1627	0.033	1.49927	2.32916
1695	0.035	1.49051	2.52808
1781	0.037	1.50575	2.79336
1861	0.039	1.53051	3.04980
2041	0.041	1.54944	3.66728
2156	0.043	1.60874	4.09628
2192	0.045	1.62677	4.23201
2237	0.047	1.64248	4.40775

2289	0.049	1.63693	4.61314
2481	0.051	1.67972	5.42537
2482	0.053	1.67597	5.42627
2713	0.055	1.74183	6.47728
2836	0.057	1.81065	7.08220
2889	0.059	1.83860	7.34796
2933	0.061	1.84444	7.57942
3046	0.063	3.15187	8.16770
3167	0.065	3.23873	8.83156
3384	0.067	3.54660	10.08787
3397	0.069	3.46173	10.20142
3504	0.071	3.50742	10.81216
3540	0.073	3.65244	11.03016
3572	0.075	3.64004	11.22606
3673	0.077	3.67181	11.87704
3837	0.079	3.76098	12.96722
3970	0.081	3.73449	13.87264
4004	0.083	3.73913	14.11632
4082	0.085	3.75886	14.67194
4221	0.087	3.70121	15.68447
4417	0.089	3.80541	17.17473
4424	0.091	3.83085	17.21913
4491	0.093	3.91290	17.75805
4631	0.095	4.01942	18.87249
4696	0.097	4.10087	19.40843
4778	0.099	4.19061	20.10800
4897	0.101	4.29645	21.10801
5036	0.103	4.47701	22.30385
5150	0.105	4.63355	23.35821

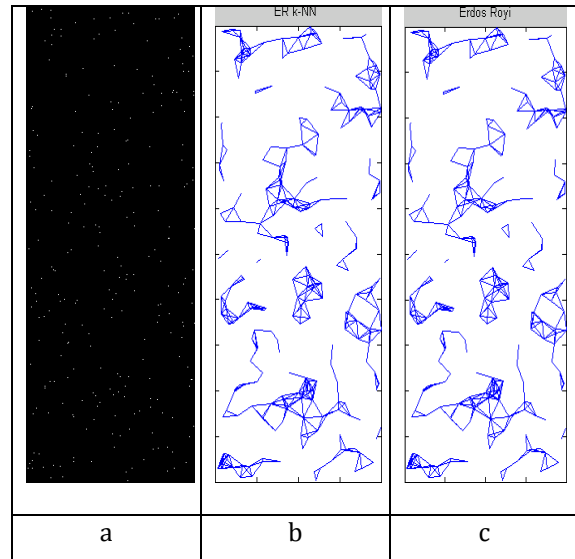
5249	0.107	4.73978	24.23739
5336	0.109	4.80595	25.05935
5428	0.111	4.92789	25.94123
5546	0.113	5.00948	27.06993
5685	0.115	5.16376	28.43455
5773	0.117	5.19992	29.34578
5925	0.119	5.17961	30.91552
5952	0.121	5.20977	31.16242
6041	0.123	5.20647	32.10460
6072	0.125	5.29950	32.45810
6088	0.127	5.24221	32.61672
6255	0.129	5.20859	34.42880
6409	0.131	5.20513	36.13983
6515	0.133	5.22762	37.36209
6542	0.135	5.26608	37.65784
6736	0.137	5.32041	39.93466
6794	0.139	5.40414	40.61880
6991	0.141	5.62366	43.02693
7017	0.143	5.63420	43.31324
7217	0.145	5.84581	45.84301
7281	0.147	5.90240	46.63678
7297	0.149	5.93010	46.84052
7502	0.151	6.24529	49.53723
7543	0.153	6.22858	50.08511
7611	0.155	6.30604	50.96423
7843	0.157	6.65871	54.11894
7906	0.159	6.72723	54.98794
7986	0.161	6.76583	56.11708
7995	0.163	6.84309	56.23221
8135	0.165	11.69431	58.21586
8276	0.167	12.07412	60.26962
8362	0.169	12.30286	61.56285
8564	0.171	12.70018	64.59212
8592	0.173	12.79342	64.96993
8709	0.175	13.00610	66.77934
8724	0.177	13.08824	66.93141
8787	0.179	13.16771	67.93681
9044	0.181	14.28052	71.92699
9087	0.183	13.72087	72.75217
9092	0.185	13.75461	72.71987
9170	0.187	13.69513	74.00391
9380	0.189	14.59682	79.08805
9504	0.191	16.60406	79.52803
9537	0.193	18.77744	80.03217
9667	0.195	17.57210	82.66109
9693	0.197	18.70974	83.11653
9886	0.199	19.80950	86.60223
9951	0.201	18.38680	87.59296

Terlihat pada **Tabel 4** bahwa *random Graph* metode erdos royi dengan k -NN lebih unggul pada posisi jumlah simpul lebih dari 1258. Penambahan jumlah simpul tidak menambah waktu komputasi secara signifikan. Penambahan waktu komputasi pada *random Graph* metode erdos dan royi dengan k -NN bertambah secara linier terhadap jumlah simpul. Waktu komputasi

random Graph metode erdos dan royi bertambah secara eksponensial terhadap jumlah simpul.

Kelemahan erdos royi dengan k -NN adalah pada waktu komputasi untuk mencari k ketetanggaan. Pada data dengan tingkat kepadatan simpul rendah, jarak antar simpul relatif berjauhan, sehingga pencarian simpul tetangga memerlukan waktu relatif lama. Data dengan tingkatan kepadatan simpul tinggi, waktu untuk pencarian simpul tetangga relatif cepat, karena dengan ukuran jendela yang tidak lebar tetangga simpul sejumlah k dapat ditemukan. Pencarian ketetanggaan pada area dengan kerapatan tinggi hanya memerlukan proses memperlebar jendela dengan waktu yang pendek. Pada area dengan jumlah simpul 1338 waktu komputasi dari *random Graph* metode erdos dan royi dengan k -NN.

Gambar 13b *Graph* hasil *random Graph* dari erdos royi. **Gambar 13c** *Graph* hasil *random Graph* dengan metode erdos royi dengan k -NN. Kedua *Graph* dibangkitkan dari citra biner (**Gambar 13a**) yang berisi 257 piksel putih dengan tingkat kepadatan 0.007. *Graph* pada **Gambar 13** dibangkitkan dengan probabilitas 0,4. Tampak dalam gambar bahwa dua *graph* yang dihasilkan tidak memiliki perbedaan, jumlah simpul yang terhubung oleh busur adalah sama, yaitu 252.



Gambar 13 : Contoh dari hasil uji coba *random Graph*, (a) Citra biner, (b) Erdos Royi, (c) Erdos Royi dengan k -NN.

Simpul-simpul pada **Gambar 13a** yang tidak terhubung oleh busur merupakan simpul yang mempunyai tetangga sangat jauh, atau probabilitasnya lebih kecil dari 0,5. Pasangan simpul yang probabilitasnya lebih kecil dari nilai ambang, maka busurnya tidak dihubungkan. Simpul-simpul dengan jarak yang saling berdekatan mempunyai busur yang banyak. Sim-

pul-simpul dengan jarak yang saling berjauhan mempunyai busur yang sedikit.

Dari *graph* dapat diketahui jumlah simpul yang saling berdekatan dengan melihat jumlah busur yang terhubung dengan simpul. Simpul dengan jumlah busur (*degree*) besar berarti simpul mempunyai tetangga yang banyak. Dengan kata lain, simpul berada di area yang rapat.

PENUTUP

A. Simpulan

Waktu komputasi pada *random Graph* metode *erdos* dan *royi* dapat dikurangi dengan mengintegrasikan metode *k*-NN. *k*-NN digunakan untuk menghubungkan busur dari setiap simpul dengan *k* tetangga terdekat. *Random Graph metode erdos* dan *royi* yang semula menghubungkan $n(n-1)$ simpul, setelah diintegrasikan dengan *k*-NN menjadi hanya menghubungkan $n(k)$ simpul. Waktu komputasi *random Graph metode erdos* dan *royi* $n(n-1)$, sedangkan *Random Graph metode erdos* dan *royi* dengan *k*-NN menjadi $n(k+k^2)$, dimana *k* adalah waktu untuk menghubungkan busur dari simpul ke *k* jumlah tetangga, dan k^2 adalah waktu untuk mencari tetangga dalam jendela.

B. Saran

Pengurangan waktu komputasi pada *random Graph erdos roy* dengan *k*-NN masih terdapat waktu komputasi untuk mencari *k* ketetanggaan yang relatif besar. Waktu komputasi mencari *k* ketetanggaan masih perlu dikurangi. Metode pembuatan jendela bisa ditingkatkan ketepatannya agar diperoleh *k* ketetanggaan lebih tepat dan waktu pencarian *k* ketetanggaan lebih cepat.

DAFTAR PUSTAKA

- [1] Abidin, Zainal dan Arifin Agus Z. 2008. Generation *Graph* with random *Graph* erdos roy method by medical image to help diagnoses of osteoporosis. International Conference Bio Medical Engenering 2008 di ITS Surabaya
- [2] Abidin, Zainal dan Arifin Agus Z. 2009. Analisa Kerapatan Trabecular Bone Berbasis *Graph* Berbobot pada Citra Panorama Gigi untuk Identifikasi Osteoporosis. Jurnal Teknik Informatika Volume 7 Nomor 3 Januari 2009
- [3] Albert, R., Barabasi, A. -L..(2002). "*Statistical Mechanics of Complex Networks*" Review of Modern Physics volume 75 halaman 47 – 98.
- [4] Andre, M., Ijaz, K., Tillinghast, J. D., Krebs, V. E., Diem, L. A., Metchock, B., Crisp, T., McElroy, P. D.(2006) "Transmission Network Analysis to Complement Routine Tuberculosis Contact Investigations" *American Journal of Public Healt.* volume 96 nomor 11.
- [5] Diestel, Reinhard (2000). *Graph Theory Electronic Edition*. Springer-Verlag: New York.
- [6] Demir, C., Yener, B., dan Gultekin, S. H. (2005). "Augmented Cell-*Graph* for Automated Cancer Diagnosis". *Bioinformatic Vol 21 suplemen 2*. ii7-ii12
- [7] Gonzalez, R. C., Woods, R. E., Eddins, S. L.. (2002). *Digital Image Processing*. Prentice Hall. New Jersey.
- [8] Guesebroek, Jan-Mark., Smeulders, A.W .M., dan Cornellisen F (1999). "Segmentation of Tissue by Distance *Graph* Matching". *Cytometry* 35:11-22
- [9] Gunduz, C., Yener, B., dan Gultekin, S. H. (2004). "The Cell *Graphs* of Cancer". *Bioinformatic Vol. 20 suplemen 1*. i145-i151.
- [10] Levitin, Anany. (2007). *Intoduction to The Design and Analysis of Algoritms second edition*. Pearson Education, Inc.
- [11] Newman, M. E. J.. (2001a). "Scientific Collaboration Network. II. Shostest Paths, Weighted Network, and Centrality". *Physical review Vol. E64* 016132
- [12] Newman, M. E. J.. (2001b), "Who is the Best Connected Scientist ? A Study of Scientific Coauthorship Network". *Physical review Vol. E64* 0011144.
- [13] Newman, M. E. J., (2001c), "Random *Graph* models of Sosial Network". *Proc. Natl. Acad. Sci. USA* 99. 2566-2572.
- [14] Newman, M. E. J., 2003, "The Structure and Function of Complex Networks". *SIAM review Vol. 45 Number 2* : 167-256
- [15] Watts, D. J., Strogatz, S. H.. (1998). "Collective dynamics of 'small-world' models" *Nature*. Volume 393 halaman 440-442.