



FAKULTAS  
MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI YOGYAKARTA

# SERTIFIKAT

NO : 2428/UN 34.13/PS/2012

Diberikan kepada:

*Nur Insani, M. Sc*

Atas partisipasinya sebagai:

**Pemakalah**

dengan judul:

*"PEMILIHAN ALGORITMA HEURISTIK TERBAIK UNTUK SUATU MASALAH GRAF  
BERDASARKAN SIFAT/KARAKTERISTIKNYA  
(INSTANCE FEATURES)"*

pada kegiatan:

**SEMINAR NASIONAL**  
**PENELITIAN, PENDIDIKAN DAN PENERAPAN MIPA**  
Pemantapan Profesionalisme Peneliti, Pendidik & Praktisi MIPA  
untuk Membangun Insan yang Kompetitif dan Berkarakter Ilmiah.

Diselenggarakan oleh FMIPA UNY dalam rangka  
DIES NATALIS UNIVERSITAS NEGERI YOGYAKARTA ke-48

Yogyakarta 02 Juni 2012

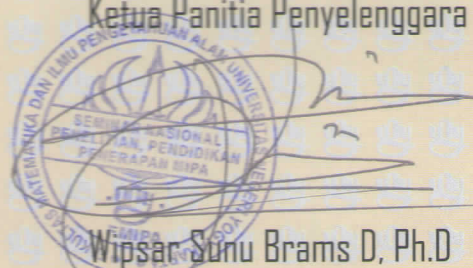
Mengetahui  
Dekan Fakultas MIPA UNY



Dr. Hartono

NIP. 196203291987021002

Ketua Panitia Penyelenggara



Wipar Sunu Brams D, Ph.D

NIP. 198001292005011003

Prosiding Seminar Nasional Penelitian, Pendidikan dan Penerapan MIPA,  
Fakultas MIPA, Universitas Negeri Yogyakarta, 02 Juni 2012



## PROSIDING SEMINAR NASIONAL

Penelitian, Pendidikan, dan Penerapan MIPA

Tanggal 02 Juni 2012, FMIPA UNIVERSITAS NEGERI YOGYAKARTA

ISBN: 978-979-99314-6-7

Bidang:

- Matematika dan Pendidikan Matematika
- Fisika dan Pendidikan Fisika
- Kimia dan Pendidikan Kimia
- Biologi dan Pendidikan Biologi
- Ilmu Pengetahuan Alam



Tema:

**Pemantapan Keprofesionalan Peneliti, Pendidik, dan Praktisi MIPA  
Untuk Membangun Insan yang Kompetitif dan Berkarakter Ilmiah**

Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Negeri Yogyakarta  
Tahun 2012

*Prosiding Seminar Nasional Penelitian, Pendidikan dan Penerapan MIPA,  
Fakultas MIPA, Universitas Negeri Yogyakarta, 02 Juni 2012*



## **PROSIDING SEMINAR NASIONAL**

**Penelitian, Pendidikan, dan Penerapan MIPA**

Tanggal 02 Juni 2012, FMIPA UNIVERSITAS NEGERI YOGYAKARTA

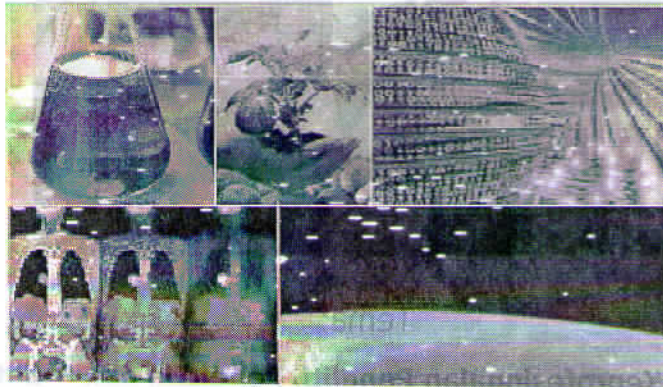
ISBN: 978-979-99314-6-7

### **Tim Editor:**

1. Kismiantini, M.Si
2. Denny Darmawan, M.Sc
3. Erfan Priyambodo, M.Si
4. Agung Wijaya, M.Pd
5. Sabar Nurohman, M.Pd

### **Tim Reviewer:**

1. Dr. Agus Maman Abadi
2. Wipar Sunu Brams Dwandaru, M.Sc, Ph.D
3. Dr. Endang Wijayanti
4. Dr. Heru Nurcahyo



Tema:

**Pemantapan Keprofesionalan Peneliti, Pendidik, dan Praktisi MIPA  
Untuk Membangun Insan yang Kompetitif dan Berkarakter Ilmiah**

Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Negeri Yogyakarta  
Tahun 2012

PEMILIHAN ALGORITMA HEURISTIK TERBAIK  
UNTUK SUATU MASALAH GRAF  
BERDASARKAN SIFAT/KARAKTERISTIKNYA (INSTANCE FEATURES)

Nur Insani, M.Sc

Jurdik Matematika FMIPA UNY



**Abstrak**

Masalah pewarnaan simpul (*vertex coloring problem*) pada suatu graf merupakan suatu masalah mewarnai setiap simpul pada graf sehingga simpul-simpul yang bertetangga (berhubungan) tidak memiliki warna yang sama dan jumlah warna yang digunakan minimal. Telah banyak algoritma yang dikembangkan untuk menyelesaikan masalah ini, baik secara *exact* maupun secara heuristik (*heuristic*). Meskipun algoritma heuristik dapat memberikan solusi yang mendekati optimal (*near optimal solution*) lebih cepat, namun pemilihan algoritma heuristik yang terbaik untuk setiap masalah (*graf instance*) tertentu tetaplah penting karena tidak ada suatu algoritma yang dapat memberikan solusi optimal untuk semua masalah. Namun, akan sangat menyita waktu apabila kita mencoba satu per satu semua algoritma yang tersedia untuk memilih algoritma terbaik untuk suatu masalah (*instance*). Sifat-sifat atau karakteristik masalah (*instance features*) dapat membantu untuk menentukan algoritma heuristik yang terbaik untuk suatu kelas masalah tertentu. Penelitian ini mengeksplorasi sifat/karakteristik masalah (*instance*) untuk memilih algoritma heuristik: DSATUR dan Tabu, yang dapat memberikan solusi terbaik untuk suatu masalah tertentu.

**Kata kunci:** pewarnaan simpul, heuristik, *instance features*.

**PENDAHULUAN**

Pewarnaan simpul pada suatu graf  $G$  merupakan pemberian warna pada setiap simpul  $v \in V$  pada graf  $G$  dimana tidak ada dua simpul yang bertetangga mempunyai warna yang sama. Jumlah warna terkecil (optimal) yang dibutuhkan untuk mewarnai seluruh simpul tersebut dinamakan bilangan kromatik (*chromatics number*), dan dilambangkan dengan  $\chi(G)$ . Mencari jumlah warna yang optimal pada pewarnaan simpul merupakan suatu masalah yang menantang. Masalah ini termasuk suatu masalah *NP-hard*, dimana masalah ini tidak mempunyai solusi polinomial yang diketahui. Oleh sebab itu, telah banyak ilmuwan yang berusaha menyelesaikan masalah tersebut dengan beberapa algoritma, baik secara eksak maupun heuristik.

Meskipun heuristik mampu memberikan solusi yang mendekati optimal lebih cepat dibandingkan dengan algoritma eksak, namun masih ada variasi yang signifikan yang hanya dapat dijelaskan oleh fakta bahwa beberapa fitur atau struktur pada graf mungkin akan lebih cocok untuk beberapa heuristik dibanding dengan algoritma lainnya. Kebutuhan untuk memahami bagaimana algoritma heuristik yang terbaik untuk tipe/kelas graf yang bergantung pada fitur graf merupakan suatu masalah yang penting. Dengan demikian kita dapat meminimalkan waktu dan kompleksitas tanpa harus mencoba satu per satu algoritma yang ada (*trial and error*).

Penelitian ini mengeksplorasi beberapa fitur graf yang dapat membantu kita untuk memilih algoritma heuristik yang paling baik untuk suatu graf tertentu. Seluruh fitur yang digunakan pada penelitian ini mempunyai kompleksitas polinomial.

**Graf Bipartit**

Sebuah graf bipartit adalah sebuah graf yang simpul-simpulnya dapat dibagi ke dalam dua himpunan bagian yang saling asing,  $U$  dan  $V$ , dimana simpul-simpul di setiap himpunan tersebut *tidak bertetangga*. Setiap busur menghubungkan suatu simpul dari himpunan  $U$  ke himpunan  $V$ . Karena simpul-simpul pada  $U$  dan  $V$  saling bertetangga, maka pada setiap himpunan bagian  $U$  dan  $V$  dapat diwarnai secara berbeda. Dengan demikian, bilangan kromatik pada graf bipartit adalah 2.

### Penyeleeksian Algoritma

Untuk memahami hubungan antara fitur/karakteristik graf dengan kinerja suatu algoritma, penelitian ini menggunakan *algorithm performance framework* yang diusulkan oleh Rice [11] dan diinterpretasikan oleh Kate Smith-Miles [12], yaitu:

- Ruang masalah P: terdiri dari berbagai tipe input graf.
- Ruang fitur F: memuat seluruh perhitungan fitur dari input graf.
- Ruang algoritma A: memuat seluruh algoritma yang akan digunakan: DSATUR dan TABU
- Ruang kinerja Y: memuat perhitungan dari kinerja algoritma.

Koleksi dari seluruh data diatas {P, F, A, Y} dikenal sebagai meta-data.

### Fitur/Karakteristik (*Instance Features*) Graf

Fitur-fitur graf merupakan properti dari suatu graf yang bergantung pada struktur abstraknya. Didalam penelitian ini, fitur graf memainkan peranan penting dimana fitur tersebut mempunyai hubungan dengan kinerja pada algoritma pewarnaan simpul. Adapun fitur-fitur tersebut yaitu:

- **Diameter** graf: jarak terbesar diantara sepasang simpul. Untuk mencari diameter suatu graf, kita harus mencari lintasan terpendek diantara setiap pasang simpul dan ambil jarak terjauh dari lintasan-lintasan tersebut.
- **Kepadatan** graf: Rasio jumlah busur terhadap jumlah busur yang mungkin.
- **Rata-rata panjang lintasan** (*average path length*): rata-rata jumlah step pada lintasan terpendek untuk seluruh pasangan simpul. Fitur ini digunakan untuk mengukur efisiensi perjalanan diantara simpul.
- **Girth**: pajang dari siklus (cycle) terpendek dari suatu graf. Jika suatu graf mempunyai *girth* sama dengan 3, maka graf tersebut tidak mungkin suatu graf bipartite.
- **Derajat simpul**: jumlah keterikatan suatu simpul dengan simpul yang lain. Fitur ini dapat menjelaskan bagaimana suatu graf terhubung.
- **Koefisien kluster** (*clustering coefficient*): merupakan ukuran sejauh mana node dalam grafik cenderung mengelompok bersama.
- **Sentralitas keantaraan** (*betweenness centrality*): ukuran seberapa pusat simpul-simpul di suatu graf. Graf yang mempunyai simpul-simpul yang terkumpul / mengelompok bersama, akan mempunyai sentralitas keantaraan yang tinggi, sementara graf yang mempunyai simpul-simpul yang berpencah akan mempunyai sentralitas keantaraan yang rendah.
- **Sentralitas vektor eigen** (*eigenvector centrality*): ukuran seberapa penting suatu simpul didalam suatu jaringan. Google menggunakan metode ini pada algoritma perankingan halaman mereka.
- **Konektivitas aljabar** (*algebraic connectivity*): nilai eigen yang terkecil kedua dari matrik Laplacian [9]. Fitur ini mencerminkan seberapa baik suatu graf terhubung.
- **Spektrum**: himpunan nilai-nilai eigen dari matrik ikatan. Nilai ini bersifat simetrik, yaitu untuk setiap  $\lambda$  terdapat  $-\lambda$ , untuk graf bipartit.

### Algoritma Heuristik

Dalam penelitian ini, peneliti menggunakan dua sampel algoritma heuristik yang telah dikenal secara umum, yaitu DSATUR dan Tabu.

#### 1. DSATUR

Algoritma heuristik "degree of saturation" (DSATUR) merupakan suatu algoritma pewarnaan sekuensial yang secara dinamik merubah urutan simpul-simpul berdasarkan derajat saturasi (*saturation*) saat itu. Derajat saturasi suatu simpul didefinisikan sebagai jumlah warna yang berbeda dimana simpul tersebut terhubung dengan simpul yang lain [2]. Algoritma DSATUR menjadi algoritma eksak untuk graf bipartit. Spektrum dari graf bersifat simetrik jika dan hanya jika graf tersebut merupakan graf bipartit. Graf tidak berarah mempunyai matrik ikatan (*adjacency matrix*) yang simetrik dan karena itu mempunyai nilai eigen yang riil.

#### 2. TABU

Algoritma TABU [7] membagi simpul-simpul graf menjadi beberapa partisi warna awal. Pewarnaan awal mungkin akan mempunyai konflik atau menjadi suatu solusi layak yang tidak

optimal. Algoritma ini kemudian mencoba untuk mengurangi jumlah warna yang ilegal dan mengoptimalkan solusi dengan cara memindahkan simpul-simpul dari satu partisi ke partisi yang lainnya. Setelah suatu simpul dipindahkan, maka akan ditambahkan pada daftar "tabu". Setiap simpul pada daftar ini tidak dapat dipindahkan ke warna sebelumnya untuk iterasi  $n$ .

#### Input Graf

Untuk mempelajari fitur/karakteristik suatu graf, penelitian ini menggunakan graf-graf sampel acak yang dibangkitkan menggunakan perangkat lunak yang disediakan oleh Joseph Culberson [4]. Himpunan meta-data yang dihasilkan diambil dari Brandon Wreford [15]. Himpunan tersebut terdiri dari 2400 graf sampel yang terdiri atas 100, 500, dan 1000 simpul. Dengan cara ini, peneliti dapat mengambil jumlah simpul-simpul ini sebagai salah satu fitur graf, seperti yang telah dijelaskan sebelumnya. Hal sangatlah penting mengingat bahwa untuk jaringan yang besar (graf besar), algoritma Tabu lebih unggul (memberikan warna lebih sedikit) dibandingkan algoritma DSATUR. Adapun struktur dari graf-graf tersebut dijelaskan sebagai berikut [15]:

- **IID Graphs** - After choosing the number of vertices in the graph, each pair of vertices are assigned an edge with an independent identical probability  $p$ .
- **Girth and Degree Inhibited** - Each graph is assigned a probability  $p$ , girth limit  $g$ ; and a degree limit. The girth limit indicates that no cycle will be created with girth less than  $g$ . Hence if an edge  $(v;w)$  is being considered as a new edge, every pair of vertices  $(x; y)$  which will have a distance of less than  $g$  after the addition is blocked, and will never be selected as a possible new edge.  $p$  is the probability that a possible edge will be used. is a hard limit on the difference between the average node degree and the maximum degree of any vertex.
- **Geometric** [8] - These graphs are generated by choosing a radius  $r$  and uniformly distributing  $n$  pairs of numbers  $(x; y)$  in the range of  $0 < x; y < 1$ . Vertices in the graph  $U(n; r)$  correspond to the points  $(x; y)$  in the plane. The vertices are joined by an edge when the distance between a pair of vertices is less than the radius  $r$ .
- **Clique Driven Graphs** - This generator creates cliques of a given size. Each clique is generated by randomly creating  $h$  color partitions, then randomly selecting one of the vertices in each partition and joining every pair by an edge. Each clique is generated independently.
- **Cycle Driven Graphs** - Similar to the clique driven graphs, this generator creates cycles of a specified length.  $h$  color partitions are created, the algorithm then generates a cycle by randomly generating a path with each vertex from a different color partition than the last.

#### PEMBAHASAN

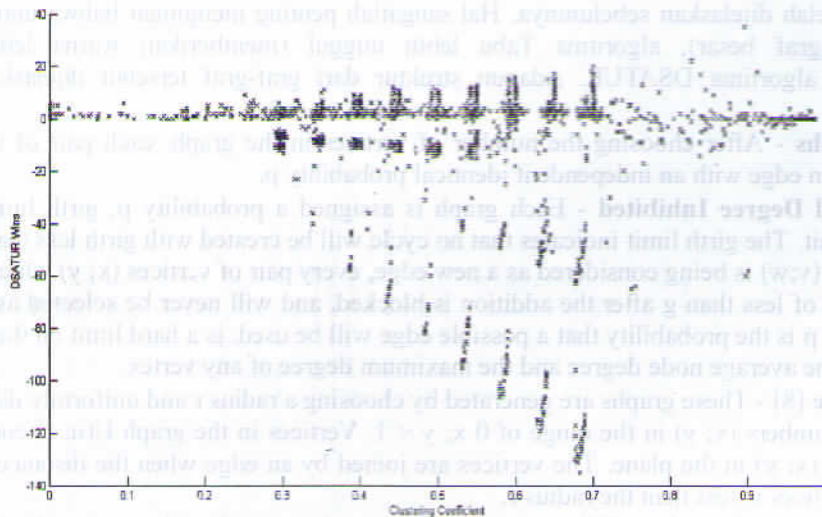
Untuk menghitung fitur-fitur yang telah dijelaskan sebelumnya, penelitian ini memanfaatkan data yang telah diambil pada penelitian terakhir [14], dimana perhitungan fitur diformulasi dengan menggunakan interface Python yang dikenal dengan *igraph* dan *scientific tools* yang dikenal dengan *SciPy* [6]. Perangkat lunak ini menyediakan metode yang fleksibel untuk mengevaluasi ribuan graf sampel dengan mudah dan cepat. Adapun output yang diberikan juga bersifat fleksibel dimana data tersebut dapat diimpor ke dalam bentuk excel.

Adapun pencarian solusi layak untuk pewarnaan input graf, penelitian ini menggunakan dua algoritma, DSATUR dan TABU, seperti yang telah dibahas sebelumnya. Adapun kedua algoritma tersebut juga dijalankan menggunakan interface Python dan output yang dihasilkan diimpor ke dalam bentuk excel.

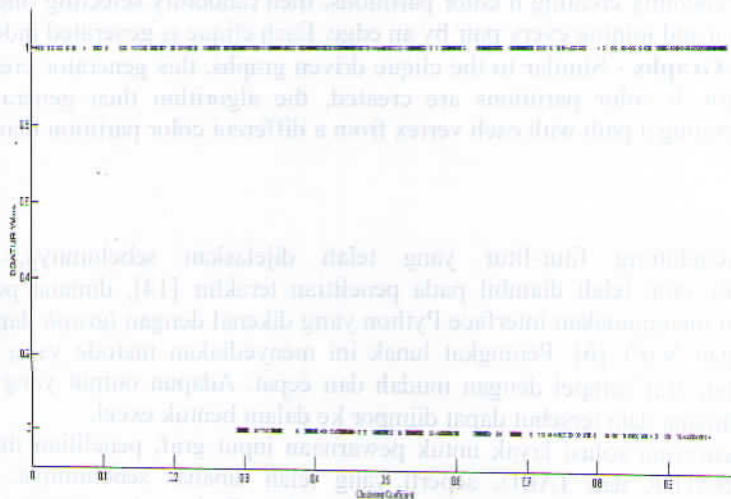
Seperti yang diketahui, disetiap kasus pewarnaan graf menggunakan kedua algoritma diatas, khususnya untuk graf dengan derajat besar, algoritma Tabu memberikan solusi yang lebih baik (warna lebih kecil) dibanding dengan algoritma DSATUR. Untuk tujuan penelitian, peneliti menformulasi  $DSATURWins = T - D$ , dimana  $T$  adalah jumlah warna yang digunakan oleh algoritma Tabu dan  $D$  adalah jumlah warna yang digunakan oleh algoritma DSATUR.

Setelah dilakukan seluruh perhitungan seluruh fitur pada seluruh graf pada meta-data, kemudian diperoleh beberapa hasil yang menarik. Salah satu fitur yang memberikan hasil yang menarik yaitu koefisien kluster (*clustering coefficient*). Apabila fitur ini diplot terhadap DSATUR Wins, maka diperoleh pola yang cukup menarik seperti yang terlihat pada Gambar 1 dibawah ini.

Untuk koefisien kluster yang bernilai kecil, maka algoritma DSATUR memberikan solusi yang lebih baik (warna yang lebih sedikit) dibandingkan dengan algoritma TABU. Seiring dengan naiknya nilai koefisien kluster, kedua algoritma memberikan performa yang seimbang. Akan tetapi jika kita perhatikan dengan lebih seksama, algoritma Tabu memberikan solusi yang lebih baik karena mempunyai margin yang lebih besar. Ketika nilai koefisien kluster membesar lagi, khususnya ketika nilai tersebut lebih dari 0,7, performa kedua algoritma masih tetap seimbang, walau dengan margin yang serupa. Dengan kata lain, peneliti tidak dapat menarik kesimpulan algoritma mana yang lebih menang atau kalah. Setelah fase ini, graf-graf yang tersisa terhubung (*clustered*) dengan baik, sehingga tidak ada dari kedua algoritma yang dapat memberikan performa luar biasa dibanding dengan yang lain.



Gambar 1. Plot antara Koefisien Kluster vs DSATURWins.

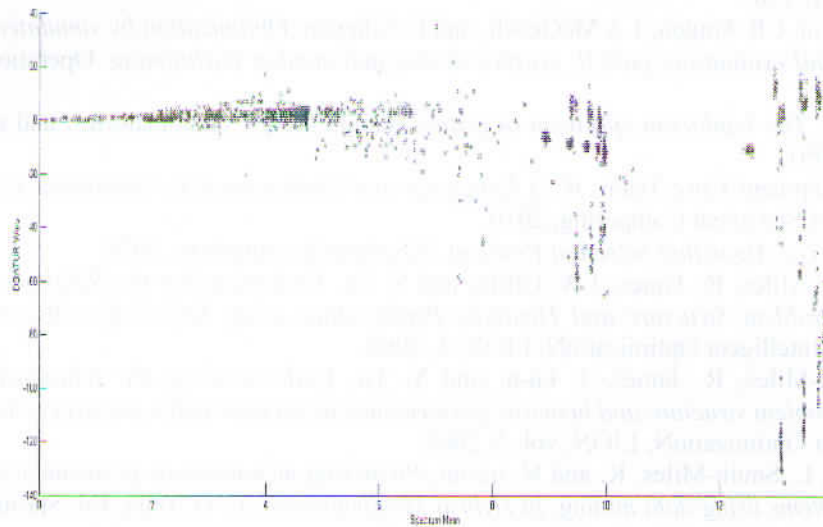


Gambar 2. Plot antara Koefisien Kluster vs DSATURWins (Kalah atau Menang).

Diketahui bahwa heuristik DSATUR menjadi algoritma eksak untuk graf bipartit. Dari definisi graf bipartit, graf tersebut tidak dapat mempunyai *triplets/triangles* yang tertutup. *Sedemikian sehingga nilai koefisien kluster graf tersebut menjadi 0. Seiring dengan meningkatnya* nilai koefisien kluster, algoritma heuristik DSATUR menjadi tidak eksak. Hal ini mengiring penelitian ini untuk mengambil kesimpulan bahwa untuk graf dengan nilai koefisien kluster yang kecil, maka graf tersebut dekat untuk menjadi graf bipartit dan algoritma DSATUR akan memberikan solusi yang terbaik.

Hasil lain yang menarik yaitu dari fitur spektrum dari graf. Jika fitur spektrum diplotkan dengan DSATURWins, maka akan dihasilkan pola yang menarik seperti yang terlihat pada Gambar

3. Spektrum dari setiap graf bipartit bersifat simetrik [5]. Dari pernyataan ini, peneliti dapat mengambil kesimpulan bahwa ada hubungan menarik antara nilai eigen dari graf dan DSATURWins.



Gambar 3. Plot antara Spektrum vs DSATURWins.

Dari Gambar 3 diatas terlihat bahwa algoritma DSATUR tampaknya memberikan solusi yang lebih baik untuk nilai spektrum yang kecil, akan tetapi ketika algoritma TABU menang (memberikan warna lebih sedikit), algoritma ini memberikan memberikan jumlah warna jauh lebih baik daripada DSATUR. Ketika algoritma DSATUR menang, algoritma ini hanya memberikan jumlah warna yang berbeda sedikit dengan algoritma Tabu. Semakin jauh suatu graf dari bipartit, menjadi suatu graf yang lebih besar, algoritma Tabu lebih mungkin untuk menang dari algoritma DSATUR. Karena diketahui algoritma DSATUR merupakan algoritma eksak untuk graf bipartit [2], hal ini membuat masuk akal untuk mengeksplorasi perilaku graf yang semakin menjauh dari bipartit.

Dari hasil diatas, dapat disimpulkan bahwa semakin besar nilai koefisien kuster dan nilai spektrum yang dimiliki oleh suatu graf, maka algoritma TABU akan lebih sering memberikan performa yang lebih baik dibandingkan dengan algoritma DSATUR.

## KESIMPULAN

Dari penelitian ini, peneliti telah mengeksplorasi beberapa macam fitur/karakteristik dari graf yang dapat membantu untuk menjelaskan tentang graf itu sendiri dan seberapa baik graf tersebut akan performa pada suatu algoritma heuristik, dalam hal ini DSATUR dan Tabu. Dari hasil penelitian juga menunjukkan bahwa koefisien kuster dan spektrum dapat merepresentasikan seberapa jauh graf menjadi graf bipartit, dan menentukan kapan algoritma DSATUR harus digunakan. Dari penelitian ini, peneliti juga telah mengeksplorasi bagaimana algoritma TABU dapat memberikan solusi yang lebih baik dengan beberapa kondisi tertentu.

## DAFTAR PUSTAKA

- [1] N. Biggs. *Algebraic graph theory*. Cambridge Univ Pr, 1993.
- [2] Daniel Brélaz. *New methods to color the vertices of a graph*. Communications of the ACM, 22(4):251–256, April 1979.
- [3] G Csárdi and T Nepusz. *The igraph software package for complex network research*. InterJournal Complex Systems, 1695:2006, 2006.
- [4] Joseph Culberson. Graph Coloring Programs.
- [5] D.M. Cvetkovic, M. Doob, and H. Sachs. *Spectra of graphs*. Deutscher Verlag der Wissenschaften, 1980.



- [6] Jones Eric, Travis Oliphant, Pearu Peterson, and Others. SciPy: Open source scientific tools for Python.
- [7] A. Hertz and D. Werra. *Using tabu search techniques for graph coloring*. Computing, 39(4):345–351, 1987.
- [8] DS Johnson, CR Aragon, LA McGeoch, and C Schevon. *Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning*. Operations research, 1991.
- [9] B. Mohar. *The Laplacian spectrum of graphs*. Graph theory, combinatorics, and applications, 2:871–898, 1991.
- [10] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2010.
- [11] JR Rice. *The Algorithm Selection Problem*. Advances in computers, 1976.
- [12] K. Smith-Miles, R. James, J.W. Giffin, and Y. Tu. *Understanding the Relationship between Scheduling Problem Structure and Heuristic Performance using Knowledge Discovery*, LNCS. Learning and Intelligent Optimization, LION, 3, 2009.
- [13] K. Smith-Miles, R. James, J. Gi-n, and Y. Tu, *Understanding the relationship between scheduling problem structure and heuristic performance using knowledge discovery*, lncs, Learning and Intelligent Optimization, LION, vol. 3, 2009.
- [14] W. B. L. L. Smith-Miles, K. and N. Insani, *Predicting metaheuristic performance on a graph coloring problems using data mining*, in *Hybrid Metaheuristics*, E.-G. Talbi, Ed. Springer, 2012, p. in press.
- [15] W. Brendan. *The Impact of Graph Features on the Performance of Graph Colouring Heuristics*. , unpublished project, 2010.