Analisis Komparatif Heuristik Greedy untuk Aproksimasi Bilangan Kromatik Graf pada Masalah Penjadwalan Perkuliahan

Ghaisan Zaki Pratama - 10122078
Program Studi Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: ghaisanzakip@gmail.com , 10122078@mahasiswa.itb.ac.id

Abstrak*—Penjadwalan perkuliahan di institusi akademik merupakan masalah optimasi fundamental yang kompleks karena banyaknya batasan yang harus dipenuhi. Makalah ini mengaplikasikan pendekatan Teori Graf, dengan masalah penjadwalan dimodelkan sebagai masalah Pewarnaan Graf. Algoritma Greedy dengan enam strategi heuristik yang berbeda—termasuk Largest First, Smallest Last, dan Random Sequential—diterapkan untuk mencari solusi aproksimasi bilangan kromatik. Kinerja setiap heuristik dievaluasi secara komparatif melalui serangkaian percobaan pada berbagai skenario skala masalah, dengan metrik utama berupa kualitas solusi (jumlah periode) dan efisiensi komputasi (waktu eksekusi). Hasil eksperimen menunjukkan bahwa strategi Random Sequential dan Smallest Last secara konsisten memberikan kualitas solusi terbaik. Namun, dari segi efisiensi waktu, Random Sequential terbukti jauh lebih unggul dibandingkan Smallest Last dan secara signifikan lebih skalabel daripada strategi lainnya seperti Independent Set. Kesimpulannya bahwa cara acak menawarkan keseimbangan terbaik antara kualitas solusi yang optimal dan efisiensi komputasi, menjadikannya pilihan yang paling pragmatis untuk masalah penjadwalan ini.

***Kata Kunci*—Graf; Penjadwalan Perkuliahan; Algoritma Greedy; Heuristik; Analisis Komparatif; Bilangan Kromatik**

I. Pendahuluan

Penjadwalan perkuliahan merupakan salah satu masalah optimisasi kombinatorial klasik yang fundamental. Proses ini dihadapkan pada serangkaian batasan (*constraints*) yang kompleks, seperti ketersediaan sumber daya terbatas (dosen dan ruang kelas) dan aturan untuk menghindari konflik jadwal antar sesi perkuliahan. Kegagalan dalam menyusun jadwal yang optimal dapat menyebabkan inefisiensi penggunaan sumber daya dan mengganggu kegiatan akademik perkuliahan. Salah satu pendekatan yang paling efektif dan elegan untuk merepresentasikan masalah ini adalah dengan menggunakan model dari Teori Graf.

Dalam pemodelan ini, setiap sesi perkuliahan unik direpresentasikan sebagai simpul (*node*) dan setiap potensi konflik antar sesi direpresentasikan sebagai sisi (*edge*) yang menghubungkan dua simpul. Dengan demikian, masalah penjadwalan yang kompleks berhasil ditransformasikan menjadi masalah pewarnaan graf (*Graph Coloring*). Tujuan dari Pewarnaan graf adalah untuk menemukan jumlah minimum warna yang dibutuhkan untuk mewarnai semua simpul tanpa ada dua simpul bertetangga yang memiliki warna yang sama. Jumlah minimum warna ini dikenal sebagai bilangan kromatik graf.

Mengingat bahwa menemukan bilangan kromatik secara eksak adalah masalah yang sulit, maka algoritma heuristik sering digunakan untuk mencapai solusi aproksimasi yang baik dalam waktu yang efisien. Algoritma greedy adalah salah satu pendekatan heuristik yang populer untuk masalah ini. Implementasi algoritma greedy dapat menggunakan berbagai strategi heuristik yang berbeda dalam menentukan urutan pewarnaan simpul. Perbedaan heuristik ini dapat menghasilkan solusi dengan kualitas dan efisiensi komputasi yang bervariasi secara signifikan.

Oleh karena itu, penelitian dalam makalah ini bertujuan untuk melakukan analisis komparatif terhadap kinerja berbagai strategi heuristik pada algoritma greedy pewarnaan graf. Kinerja dievaluasi berdasarkan dua metrik utama: kualitas solusi, yang diukur dari jumlah periode yang dihasilkan (aproksimasi bilangan kromatik), dan efisiensi komputasi, yang diukur dari waktu eksekusi. Makalah ini akan menyajikan perbandingan kuantitatif dari strategi-strategi seperti *largest_first, smallest_first,* dan lainnya untuk memberikan rekomendasi mengenai heuristik yang paling efektif untuk masalah penjadwalan perkuliahan.

Makalah ini disusun dengan sistematika sebagai berikut. Bagian II akan membahas landasan teori mengenai masalah penjadwalan, pemodelan graf, dan berbagai heuristik pewarnaan graf. Bagian III menjelaskan metodologi eksperimen yang digunakan untuk pengujian. Bagian IV merangkum kesimpulan yang didapat dari analisis komparatif yang telah dilakukan. Selain itu, terdapat lampiran tabel hasil penjadwalan dengan berbagai heuristik yang diterapkan.

II. Landasan Teori

A. Masalah Penjadwalan Perkuliahan

Penjadwalan merupakan salah satu bidang aplikasi klasik dari teori graf. Salah satu contohnya adalah *The Timetabling Problem*, yang berupaya menyusun jadwal untuk serangkaian kegiatan agar tidak terjadi tumpah tindih.[2] Dalam konteks akademik, masalah ini muncul sebagai kebutuhan untuk

menjadwalkan sejumlah sesi perkuliahan ke dalam slot-slot waktu yang tersedia, dengan memastikan tidak ada dosen atau kelas yang dijadwalkan secara bersamaan untuk lebih dari satu kegiatan. Tujuan utamanya adalah menggunakan jumlah total slot waktu seminimum mungkin, yang secara langsung berkaitan dengan efisiensi penggunaan sumber daya institusi.

B. Pemodelan Graf dan Pewarnaan Graf

Banyak situasi di dunia nyata dapat dideskripsikan secara mudah menggunakan diagram yang terdiri dari himpunan titik dan garis yang menghubungkan pasangan titik tertentu. Pendekatan ini disebut pemodelan graf. Sebuah graf G secara formal merupakan pasangan terurut $(V(G), E(G))$ yang terdiri dari himpunan tak kosong simpul (*vertices*) $V(G)$ dan himpunan sisi (*edges*) $E(G)$.[2] Dalam makalah ini, masalah penjadwalan perkuliahan dimodelkan sebagai graf dengan cara sebagai berikut:

- Setiap sesi perkuliahan unik direpresentasikan sebagai sebuah simpul dalam graf.
- Sebuah sisi ditarik untuk menghubungkan dua simpul jika terdapat konflik di antara kedua sesi yang direpresentasikannya. Dua simpul yang terhubung oleh sebuah sisi disebut bertetangga (*adjacent*).[2]

Dengan pemodelan ini, masalah penjadwalan ditransformasikan menjadi masalah pewarnaan simpul (*Vertex Coloring*). Tujuan dari pewarnaan simpul adalah memberikan warna pada setiap simpul di graf sedemikian sehingga tidak ada dua simpul yang bertetangga memiliki warna yang sama. Jumlah minimum warna yang diperlukan untuk mewarnai sebuah graf G disebut bilangan kromatik (*chromatic number*) dari G, dinotasikan dengan $\chi(G)$.[2] Dalam konteks penjadwalan, bilangan kromatik merepresentasikan jumlah minimum slot waktu yang dibutuhkan.

C. Algortima Greedy

Algoritma greedy merupakan salah satu metode paling populer dan sederhana untuk memecahkan persoalan optimasi, yaitu persoalan untuk mencari solusi optimal (maksimal atau minimal).[1] Prinsip utama dari algoritma greedy adalah "*take what you can get now!*".[1] Algoritma ini membentuk solusi langkah per langkah, pada setiap langkahnya sebuah keputusan terbaik dibuat berdasarkan informasi yang tersedia saat itu tanpa memperhatikan konsekuensi di masa depan, dengan harapan akan berakhir pada solusi optimum global.

Elemen-elemen dasar dari sebuah algoritma greedy adalah:[1]

- Himpunan Kandidat

  Berisi elemen-elemen yang akan membentuk solusi.

- Himpunan Solusi

  Berisi kandidat-kandidat yang telah terpilih.

- Fungsi Seleksi

  Memilih kandidat terbaik pada setiap langkah.

- Fungsi Kelayakan

  Memeriksa apakah seorang kandidat dapat dimasukkan ke dalam himpunan solusi.

- Fungsi Solusi

  Menentukan apakah himpunan solusi sudah lengkap.

- Fungsi Objektif

  Fungsi yang nilainya akan dioptimalkan.

Meskipun populer, algoritma greedy tidak selalu menjamin solusi optimal untuk semua persoalan dan terkadang hanya memberikan solusi suboptimal atau hampiran.[1] Hal ini disebabkan algoritma ini tidak beroperasi secara menyeluruh terhadap semua kemungkinan solusi yang ada.

D. Heuristik

Pada masalah pewarnaan graf, algoritma greedy bekerja dengan mengunjungi simpul-simpul secara sekuensial dan memberikan warna dengan indeks terkecil yang belum digunakan oleh tetangganya.[1] Perbedaan kualitas solusi bergantung pada strategi heuristik yang digunakan untuk menentukan urutan kunjungan simpul. Berikut adalah heuristik yang dianalisis dalam makalah ini:[3]

- *Largest First* (*LF*):

  Strategi ini mengurutkan simpul berdasarkan derajatnya, dari yang tertinggi ke terendah, sebelum pewarnaan dimulai. Intuisinya adalah untuk menangani simpul yang paling terbatas terlebih dahulu. Simpul paling terbatas yang dimaksud merupakan terbatas dalam pemilihan warna, karena untuk mendapatkan jumlah warna yang optimal simpul graf harus diwarnai dengan jumlah yang seminimal mungkin, sehingga simpul dengan derajat tertinggi merupakan simpul dengan pemberian warna yang paling terbatas.

- *Smallest Last* (*SL*):

  Strategi ini mengurutkan urutan eliminasi terbalik. Secara iteratif, simpul dengan derajat terkecil pada sisa graf saat itu akan diidentifikasi dan ditempatkan di akhir daftar urutan. Proses ini diulangi hingga semua simpul terurut, lalu pewarnaan dilakukan mengikuti urutan yang telah terbentuk tersebut. Heuristik ini dikenal efektif karena memprioritaskan pewarnaan pada bagian graf yang paling terhubung.

- *Independent Set* (*IS*):

  Strategi ini bekerja dengan cara menemukan sebuah himpunan independen maksimal (kumpulan simpul yang saling tidak bertetangga), memberi mereka satu warna yang sama, menghapusnya dari graf, dan mengulangi proses pada sisa graf dengan warna baru. Heuristik ini sangat cocok pada graf yang tidak lengkap (tidak semua simpul bertetangga dengan semua simpul lainnya). Meskipun intuitif, kelemahan dari strategi ini adalah proses pencarian himpunan independen yang maksimal tidak selalu menghasilkan pilihan yang optimal untuk pewarnaan keseluruhan

graf, yang dapat menyebabkan penggunaan warna yang lebih banyak dibandingkan heuristik lain.

- *Random Sequential* (*RS*):

  Merupakan strategi dasar dengan simpul-simpul diwarnai dalam urutan sepenuhnya acak dan sering dijadikan sebagai pembanding dasar (*baseline*).

- *Connected Sequential* (*DFS/BFS*):

  Urutan pewarnaan simpul ditentukan oleh urutan penemuan melalui algoritma penjelajahan graf standar, yaitu *Depth-First Search* (*DFS*) atau *Breadth-First Search* (*BFS*).

III. Implementasi dan Pembahasan

A. Metodologi dan Implementasi

Bagian ini menjelaskan langkah-langkah konkret yang dilakukan, mulai dari persiapan hingga eksekusi.

1. Lingkungan Implementasi

Percobaan diimplementasikan menggunakan bahasa Python. Beberapa library eksternal dimanfaatkan untuk mendukung implementasi, antara lain:

- NetworkX

  Digunakan sebagai fondasi utama untuk membuat, memanipulasi, dan menganalisis struktur data graf.

- NumPy

  Digunakan untuk operasi numerik, terutama dalam generasi data masukan berupa matriks kebutuhan sesi secara efisien.

- Pandas

  Digunakan untuk menyajikan data hasil eksperimen dalam format tabel yang terstruktur dan mudah dibaca.

- Tabulate

  Digunakan untuk menyajikan data hasil eksperimen dalam format tabel yang terstruktur dan mudah dibaca.

- Matplotlib

  Digunakan untuk membuat visualisasi data dalam bentuk grafik guna mempermudah analisis perbandingan.

- Seaborn

  Digunakan untuk membuat visualisasi data dalam bentuk grafik guna mempermudah analisis perbandingan.

2. Proses Pemodelan Graf dari Data

Langkah awal adalah mengubah masalah penjadwalan perkuliahan menjadi sebuah model graf yang dapat dianalisis secara komputasional. Proses ini terdiri dari beberapa tahapan:

- Pembuatan Data Masukan

  Masukan untuk masalah ini adalah kebutuhan mengajar dari m dosen untuk n kelas. Ini direpresentasikan sebagai sebuah matriks integer P berukuran m x n, dengan setiap elemen P_{ij} menyatakan banyak sesi yang harus diajarkan oleh Dosen i untuk kelas j. Dalam percobaan ini, matriks P dibuat secara acak menggunakan fungsi numpy.random.randint untuk mensimulasikan berbagai beban kerja yang berbeda.

- Pembentukan Daftar Sesi

  Dari matriks P, sebuah daftar tunggal berisi semua sesi perkuliahan (daftar_sesi) dibuat. Jika Dosen i harus mengajar Kelas j sebanyak k kali, maka sebanyak k buah sesi (Dosen-i, Kelas-j) akan dimasukkan ke dalam daftar ini.

- Konstruksi Graf Konflik

  Sebuah graf tak berarah, graf_konflik, kemudian dikonstruksi. Setiap elemen dalam daftar_sesi direpresentasikan sebagai simpul (*node*) unik dalam graf. Untuk membedakan sesi yang identik (dosen dan kelas yang sama), sebuah indeks unik ditambahkan pada setiap simpul. Selanjutnya, sebuah sisi (*edge*) dibuat untuk menghubungkan dua simpul jika dan hanya jika kedua sesi tersebut tidak dapat dijadwalkan pada waktu yang bersamaan. Berdasarkan definisi masalah. Konflik terjadi jika:

  - Kedua sesi melibatkan dosen yang sama.
  - Kedua sesi melibatkan kelas yang sama.

  Struktur graf yang dihasilkan ini merepresentasikan semua batasan penjadwalan.

3. Analisis Elemen Greedy pada Setiap Heuristik

Meskipun semua heuristik yang diuji bekerja di bawah paradigma pewarnaan greedy, heuristik-heuristik tersebut dibedakan oleh fungsi seleksi yang digunakan untuk menentukan urutan pewarnaan simpul. Berikut adalah analisis enam elemen algoritma greedy untuk setiap strategi:

Untuk semua heuristik yang diuji:

- Himpunan Kandidat

  Himpunan seluruh simpul (sesi) dalam graf_konflik yang belum diwarnai.

- Himpunan Solusi

  Sebuah pemetaan (dictionary) yang berisi pasangan (simpul, warna), dengan setiap simpul telah diberi sebuah warna (integer).

- Fungsi Kelayakan

  Sebuah simpul v dianggap layak untuk diberi warna c jika tidak ada satupun tetangga dari v yang sudah memiliki warna c.

- Fungsi Solusi

Solusi dianggap tercapai jika semua simpul dalam graf_konflik telah berhasil diberi warna.

- Fungsi Objektif

  Meminimalkan jumlah warna unik yang digunakan dalam himpunan solusi, yang ekuivalen dengan mencari aproksimasi bilangan kromatik $\chi(G)$.

Perbedaan terletak pada fungsi seleksi:

- *Largest First* (*LF*)

  Sebelum pewarnaan, seluruh simpul diurutkan terlebih dahulu berdasarkan derajatnya secara menurun. Fungsi seleksi kemudian memilih simpul berikutnya berdasarkan urutan yang telah ditetapkan ini.

- *Smallest First* (*SL*)

  Fungsi seleksi pada *SL* membuat sebuah urutan melalui proses eliminasi terbalik. Pada setiap tahap, ia menyeleksi simpul dengan derajat terkecil pada sisa graf untuk ditempatkan di akhir urutan. Pewarnaan dilakukan dengan mengikuti urutan yang sudah jadi ini.

- *Independent Set* (*IS*)

  Fungsi seleksi bekerja setiap putaran warna. Pada setiap putaran untuk warna c, ia menyeleksi sebuah himpunan simpul independen maksimal dari sisa graf yang belum diwarnai. Semua simpul dalam himpunan ini akan diberi warna c.

- *Random Sequential (RS*)

  Merupakan fungsi seleksi paling sederhana. Ia memilih simpul berikutnya dari himpunan kandidat dalam urutan yang acak.

- *Connected Sequential DFS*

  Fungsi seleksi memilih simpul berikutnya berdasarkan urutan kunjungan dari algoritma penjelajahan graf tanpa informasi, yaitu *Depth-First Search* (pencarian mendalam), yang dimulai dari sebuah simpul acak.

- *Connected Sequential BFS*

  Fungsi seleksi memilih simpul berikutnya berdasarkan urutan kunjungan dari algoritma penjelajahan graf tanpa informasi, yaitu *Breadth-First Search* (pencarian melebar), yang dimulai dari sebuah simpul acak.

B. Desain dan Prosedur Percobaan

Untuk melakukan evaluasi yang komprehensif terhadap kinerja berbagai strategi heuristik, sebuah desain percobaan yang sistematis dan terkontrol perlu dirancang. Tujuan utama dari percobaan ini adalah untuk mengukur dan membandingkan kinerja heuristik tidak hanya pada satu kasus tunggal, tetapi pada serangkaian skenario dengan skala masalah yang berbeda. Hal ini memungkinkan analisis yang lebih mendalam terkait aspek kualitas solusi (optimalitas), efisiensi komputasi (kecepatan), dan robustisitas (ketahanan) dari setiap strategi.

1. Definisi Skenario Pengujian

Untuk memastikan analisis yang mencakup berbagai tingkat kompleksitas, empat skenario pengujian utama telah didefinisikan. Skenario-skenario ini merepresentasikan kombinasi beban kerja yang berbeda antara jumlah dosen (m) dan jumlah kelas (n). Pemilihan skenario ini bertujuan untuk menyimulasikan kondisi yang beragam.

Berikut adalah definisi dari keempat skenario tersebut:

- Skenario 1: Kecil-Kecil

  Pada skenario ini, parameter yang digunakan adalah $m = 5$ dosen dan $n = 10$ kelas. Skenario ini dirancang untuk menyimulasikan kondisi pada sebuah program studi atau departemen berskala kecil, dengan beban komputasi yang diharapkan relatif rendah.

- Skenario 2: Kecil-Besar

  Skenario ini menggunakan $m = 5$ dosen dan $n = 50$ kelas. Ini merepresentasikan kondisi dengan sejumlah kecil staf pengajar memiliki tanggung jawab yang sangat luas untuk mengampu banyak sekali mata kuliah atau kelas yang berbeda, sehingga menciptakan potensi konflik yang tinggi meskipun jumlah dosennya sedikit.

- Skenario 3: Besar-Kecil

  Skenario ini menggunakan $m = 20$ dosen dan $n = 10$ kelas. Ini menyimulasikan kondisi pada sebuah departemen besar dengan banyak staf pengajar, tetapi dengan penawaran kelas yang lebih terkonsentrasi atau terspesialisasi, sehingga jumlah dosen jauh melebihi jumlah kelas.

- Skenario 4: Besar-Besar

  Skenario ini menggunakan $m = 20$ dosen dan $n = 50$ kelas. Ini adalah skenario dengan beban komputasi paling tinggi, merepresentasikan kondisi pada sebuah fakultas atau universitas berskala besar. Skenario ini dirancang khusus untuk menguji batas skalabilitas dari setiap heuristik yang dianalisis.

2. Prosedur Eksperimen Berulang

Dari pembuatan data masukan (matriks kebutuhan sesi) yang bersifat acak dapat menimbulkan bias pada satu kali eksekusi, maka setiap skenario diuji secara berulang untuk mendapatkan hasil statistik yang lebih valid dan reliabel. Prosedur eksperimen yang dijalankan untuk setiap skenario adalah sebagai berikut:

- Iterasi Skenario

  Sebuah loop utama diatur untuk berjalan sebanyak 10 kali ($N = 10$) untuk setiap dari empat skenario yang telah didefinisikan.

- Pembuatan Graf Konflik Unik

Pada setiap iterasi, sebuah masalah penjadwalan baru dibuat dengan membuat matriks kebutuhan sesi secara acak sesuai parameter m dan n yang sesuai. Dari matriks ini, sebuah graf konflik yang unik kemudian dibangun. Proses ini memastikan bahwa setiap dari 10 pengujian dalam satu skenario dilakukan pada contoh masalah yang berbeda, sehingga hasilnya lebih dapat digeneralisasi.

- Aplikasi Seluruh Heuristik

  Untuk setiap graf konflik yang terbentuk, keenam strategi heuristik diterapkan untuk melakukan proses pewarnaan graf.

- Pencatatan Metrik Kinerja

  Dua metrik kinerja utama dicatat secara presisi untuk setiap penerapan heuristik:

  - Kualitas solusi, diukur dari jumlah periode atau jumlah warna yang digunakan (aproksimasi bilangan kromatik).
  - Efisiensi Komputasi, diukur dari waktu eksekusi yang dibutuhkan dalam satuan detik.

- Agregasi Data

  Setelah 10 iterasi untuk satu skenario selesai, seluruh data yang terkumpul (total 60 data point per skenario) kemudian diagregasi untuk menghitung nilai statistik, yaitu dengan rata-rata, yang akan digunakan dalam tahap analisis dan pembahasan.

Dengan menjalankan prosedur ini, percobaan ini berhasil mengumpulkan data kinerja yang kaya, memungkinkan dilakukannya analisis yang tidak hanya membandingkan heuristik satu sama lain, tetapi juga menganalisis bagaimana kinerja semua heuristik berubah seiring dengan perubahan skala dan dimensi masalah.

C. Hasil dan Analisis Pembahasan

Bagian ini menyajikan dan menganalisis secara mendalam data yang diperoleh dari percobaan multi-skenario.

1. Penyajian Hasil Eksperimen

TABLE I. TABEL RATA-RATA JUMLAH PERIODE PER SKENARIO

Strategi	Besar-Besar	Besar-Kecil	Kecil-Besar	Kecil-Kecil
Connected Sequential BFS	153.5	65.3	138	33.1
Connected Sequential DFS	153.7	65	137.7	33.2
Independent Set	158.6	64.3	137.7	33.2
Largest First	150.8	63	137.5	32.7
Random Sequential	150.2	62.4	137.5	32
Smallest First	150.2	62.4	137.5	32.4

TABLE II. TABEL RATA-RATA WAKTU EKSEKUSI (DETIK) PER SKENARIO

Strategi	Besar-Besar	Besar-Kecil	Kecil-Besar	Kecil-Kecil
Connected Sequential BFS	0.2545	0.0134	0.0342	0.0033
Connected Sequential DFS	0.46	0.0314	0.1008	0.0055
Independent Set	164.856	2.754	8.52	0.1512
Largest First	0.1027	0.0077	0.0173	0.0013
Random Sequential	0.1232	0.0072	0.0168	0.0014
Smallest First	1.2558	0.0995	0.1971	0.0139

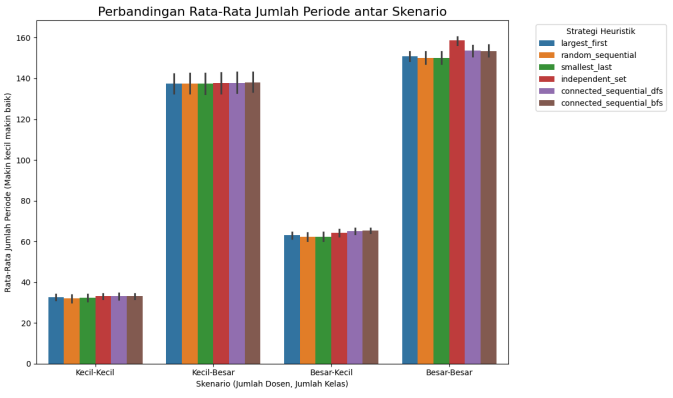

Fig. 1. Perbandingan Rata-Rata Jumlah Periode antar Skenario

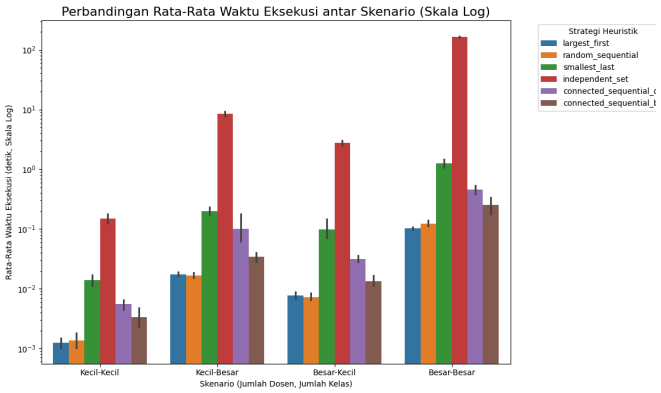

Fig. 2. Perbandingan Rata-Rata Waktu Eksekusi antar Skenario (Skala Log)

2. *Analisis Kuantitatif Kinerja*

Hasil eksperimen yang disajikan pada Tabel 1 dan Gambar 1 menunjukkan sebuah pola kinerja yang konsisten terkait kualitas solusi (jumlah periode). Di semua empat skenario, mulai dari skala "Kecil-Kecil" hingga "Besar-Besar", strategi *Random Sequential* dan *Smallest Last* secara konsisten menjadi yang terbaik dengan menghasilkan rata-rata jumlah periode paling rendah (aproksimasi bilangan kromatik paling kecil). Keduanya secara signifikan lebih baik daripada tiga strategi lainnya, yaitu *Independent Set, Connected Sequential BFS,* dan *Connected Sequential DFS*, yang performanya cenderung lebih buruk dan berdekatan satu sama lain. Strategi *Largest First* menempati posisi menengah dengan kinerja baik tetapi tidak menjadi yang terbaik.

Selanjutnya, analisis efisiensi komputasi (waktu eksekusi) pada Tabel 2 dan Gambar 2 mengungkap perbedaan skalabilitas yang drastis. Grafik dengan skala logaritmik memperjelas bahwa strategi *Independent Set* memiliki masalah skalabilitas yang ekstrim. Waktu eksekusinya melonjak dari 0.15 detik pada skenario "Kecil-Kecil" menjadi lebih dari 164 detik pada skenario "Besar-Besar", sebuah peningkatan lebih dari 1000 kali lipat. Hal ini membuatnya tidak praktis untuk masalah berskala besar. Sebaliknya *Largest First* dan *Random Sequential* menunjukkan efisiensi tertinggi dengan waktu eksekusi yang sangat rendah dan stabil.

Hasil yang didapat cukup menarik. Strategi *Smallest Last*, yang kualitas solusinya setara dengan *Random Sequential*, ternyata memerlukan waktu komputasi yang jauh lebih tinggi (sekitar 5-10 kali lebih lambat). Ini mengindikasikan bahwa meskipun mekanisme pengurutan eliminasinya sangat efektif untuk menemukan solusi yang baik, biaya komputasi yang tidak sepadan jika dibandingkan dengan kecepatan strategi lain yang hasilnya sama baik.

3. *Rekomendasi Heuristik dan Implikasi*

Berdasarkan analisis komparatif ini, dapat disimpulkan bahwa tidak ada satu pun strategi yang unggul mutlak di semua metrik. Namun, jika tujuan utamanya adalah mencari solusi yang paling seimbang antara kualitas solusi dan efisiensi komputasi, maka strategi *Random Sequential* adalah pemenangnya. Strategi ini secara konsisten memberikan hasil yang mendekati optimal (bahkan seringkali yang paling optimal) dengan waktu eksekusi yang sangat cepat. Untuk aplikasi dunia nyata dengan kecepatan pengambilan keputusan sama pentingnya dengan kualitas keputusan itu sendiri, *Random Sequential* menawarkan solusi yang paling andal dan efisien di antara semua heuristik yang diuji.

IV. KESIMPULAN

Percobaan dalam makalah ini berangkat dari masalah optimasi klasik penjadwalan perkuliahan, sebuah tantangan fundamental di berbagai universitas. Dengan merepresentasikan masalah ini sebagai model pewarnaan graf, percobaan ini bertujuan untuk melakukan analisis komparatif terhadap kinerja enam strategi heuristik yang berbeda dari algoritma greedy. Melalui serangkaian percobaan pada empat skenario dengan skala masalah yang bervariasi, kinerja setiap heuristik dievaluasi berdasarkan dua metrik utama, yaitu kualitas solusi (jumlah periode yang dibutuhkan atau bilangan kromatik) dan efisiensi komputasi (waktu eksekusi).

Berdasarkan hasil analisis kuantitatif yang telah dilakukan, beberapa hasil penting didapat. Pertama, dari segi kualitas solusi, strategi *Random Sequential* dan *Smallest Last* secara konsisten terbukti paling unggul, menghasilkan aproksimasi bilangan kromatik (jumlah periode) terendah di semua skenario yang diuji. Kedua, dari segi efisiensi waktu, *Random Sequential* dan *Largest First* menunjukkan kecepatan komputasi tertinggi, sementara strategi *Independent Set* terbukti tidak efektif dan sangat tidak efisien seiring dengan meningkatnya kompleksitas masalah. Ketiga, terdapat *trade-off* yang jelas pada strategi *Smallest Last* yang meskipun menghasilkan solusi berkualitas tinggi, tetapi memerlukan waktu eksekusi yang jauh lebih lama dibandingkan strategi cepat lainnya.

Kesimpulan utama yang dapat ditarik dari percobaan ini adalah sebuah hasil yang menarik. Fakta bahwa *Random Sequential* yang merupakan strategi non deterministik yang tidak memanfaatkan informasi struktural graf seperti derajat simpul mampu secara konsisten menyamai atau bahkan melampaui heuristik-heuristik yang lebih kompleks, mengimplikasikan sebuah poin penting yaitu untuk struktur graf yang dihasilkan dari masalah penjadwalan, belum tentu ada satu urutan pewarnaan deterministik yang superior secara general. Keberhasilan strategi acak ini menunjukkan bahwa heuristik-heuristik yang lebih kaku mungkin dapat terjebak dalam urutan pemrosesan yang sub optimal pada graf tertentu. Dengan kata lain, percobaan ini menunjukkan bahwa di antara heuristik-heuristik greedy yang diuji, belum ditemukan satupun yang terbukti paling optimal secara mutlak, karena strategi acak yang paling sederhana justru mampu memberikan keseimbangan kinerja terbaik secara keseluruhan.

Meskipun begitu, percobaan yang dilakukan pada makalah ini memiliki beberapa keterbatasan. Pertama, data masukan yang digunakan dibuat secara acak dan mungkin tidak sepenuhnya merepresentasikan semua kasus dan struktur dari data penjadwalan dunia nyata. Kedua, analisis terbatas pada enam heuristik yang tersedia di library NetworkX dan tidak membandingkannya dengan paradigma algoritma lain di luar greedy.

Untuk pengembangan selanjutnya, beberapa arah percobaan dapat dieksplorasi. Pertama, melakukan pengujian serupa menggunakan dataset penjadwalan riil dari sebuah universitas untuk validasi lebih lanjut. Kedua, melakukan analisis yang lebih mendalam untuk mengidentifikasi properti-properti graf tertentu (seperti densitas atau distribusi derajat) yang membuat suatu heuristik lebih unggul dari yang lain. Terakhir, membandingkan pendekatan greedy ini dengan paradigma algoritma lain seperti backtracking atau algoritma metaheuristik untuk melihat apakah solusi yang lebih baik dapat dicapai.

Daftar Pustaka

[1] R. Munir, "Algoritma Greedy (Bagian 1,2,3)" dalam Bahan Kuliah IF2211 Strategi Algoritma. Program Studi Teknik Informatika, Institut Teknologi Bandung, 2025.

[2] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*. New York: North-Holland, 1976.

[3] NetworkX Developers, *NetworkX Documentation*: *networkx.algorithms.coloring.greedy_color*. Diakses pada 20 Juni 2025, dari greedy_color — NetworkX 3.5 documentation

Lampiran

Berikut merupakan satu contoh tabel jadwal yang dibuat untuk kasus berikut:

Jumlah Dosen (m) : 5

Jumlah Kelas (n) : 10

Matriks Sesi yang Harus Dijadwalkan :

```
[[5 4 4 3 5 0 3 3 3 2]
 [3 3 0 3 2 4 5 0 0 0]
 [4 1 3 1 1 3 2 4 3 3]
 [4 5 5 2 1 4 0 3 0 4]
 [3 5 3 0 0 4 3 3 4 0]]
```

```
--- Contoh Jadwal Perkuliahan Optimal ---

Jadwal Senin:
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen   | Periode 1   | Periode 2   | Periode 3   | Periode 4   | Periode 5   | Periode 6   | Periode 7   |
+=========+=============+=============+=============+=============+=============+=============+=============+
| Dosen-1 | Kelas-3     | Kelas-5     | Kelas-2     | Kelas-4     | Kelas-5     | Kelas-8     | Kelas-8     |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-2 | Kelas-5     | Kelas-1     | Kelas-4     | Kelas-5     | -           | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-3 | Kelas-6     | Kelas-2     | Kelas-7     | Kelas-1     | Kelas-3     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-4 | Kelas-4     | Kelas-8     | Kelas-3     | Kelas-2     | Kelas-1     | Kelas-3     | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-5 | Kelas-7     | Kelas-9     | Kelas-9     | Kelas-6     | Kelas-2     | Kelas-1     | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
```

Fig. 3. Contoh Hasil Jadwal Hari Senin

```
Jadwal Selasa:
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen   | Periode 1   | Periode 2   | Periode 3   | Periode 4   | Periode 5   | Periode 6   | Periode 7   |
+=========+=============+=============+=============+=============+=============+=============+=============+
| Dosen-1 | Kelas-5     | Kelas-5     | Kelas-2     | Kelas-5     | Kelas-1     | Kelas-1     | Kelas-1     |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-2 | Kelas-7     | Kelas-4     | Kelas-7     | Kelas-7     | -           | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-3 | Kelas-10    | Kelas-9     | Kelas-5     | Kelas-1     | Kelas-8     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-4 | Kelas-6     | Kelas-10    | Kelas-3     | Kelas-3     | Kelas-4     | Kelas-3     | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-5 | Kelas-1     | Kelas-6     | Kelas-6     | Kelas-6     | Kelas-2     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
```

Fig. 4. Contoh Hasil Jadwal Hari Selasa

```
Jadwal Rabu:
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen   | Periode 1   | Periode 2   | Periode 3   | Periode 4   | Periode 5   | Periode 6   | Periode 7   |
+=========+=============+=============+=============+=============+=============+=============+=============+
| Dosen-1 | Kelas-4     | Kelas-9     | Kelas-9     | Kelas-2     | Kelas-10    | Kelas-10    | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-2 | Kelas-2     | Kelas-2     | Kelas-6     | Kelas-6     | Kelas-1     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-3 | Kelas-6     | Kelas-1     | Kelas-1     | Kelas-8     | Kelas-3     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-4 | Kelas-10    | Kelas-8     | Kelas-5     | Kelas-1     | Kelas-6     | Kelas-1     | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-5 | Kelas-8     | Kelas-7     | Kelas-2     | Kelas-3     | Kelas-7     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
```

Fig. 5. Contoh Hasil Jadwal Hari Rabu

```
Jadwal Kamis:
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen   | Periode 1   | Periode 2   | Periode 3   | Periode 4   | Periode 5   | Periode 6   | Periode 7   |
+=========+=============+=============+=============+=============+=============+=============+=============+
| Dosen-1 | Kelas-9     | Kelas-3     | Kelas-3     | Kelas-7     | Kelas-7     | Kelas-2     | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-2 | Kelas-1     | Kelas-2     | Kelas-7     | Kelas-6     | -           | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-3 | Kelas-8     | Kelas-4     | Kelas-10    | Kelas-3     | Kelas-9     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-4 | Kelas-6     | Kelas-8     | Kelas-2     | Kelas-2     | Kelas-10    | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-5 | Kelas-2     | Kelas-9     | Kelas-8     | Kelas-1     | Kelas-2     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
```

Fig. 6. Contoh Hasil Jadwal Hari Kamis

```
Jadwal Jumat:
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen   | Periode 1   | Periode 2   | Periode 3   | Periode 4   | Periode 5   | Periode 6   | Periode 7   |
+=========+=============+=============+=============+=============+=============+=============+=============+
| Dosen-1 | Kelas-8     | Kelas-7     | Kelas-1     | Kelas-3     | Kelas-1     | Kelas-4     | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-2 | Kelas-7     | Kelas-4     | Kelas-6     | -           | -           | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-3 | Kelas-9     | Kelas-8     | Kelas-7     | Kelas-6     | Kelas-10    | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-4 | Kelas-2     | Kelas-2     | Kelas-10    | Kelas-1     | Kelas-6     | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
| Dosen-5 | Kelas-3     | Kelas-3     | Kelas-8     | Kelas-9     | -           | -           | -           |
+---------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
```

Fig. 7. Contoh Hasil Jadwal Hari Jumat

Link Google Colab : SourceCode

Video Link at Youtube

Video Makalah Stima

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Juni 2025

Ghaisan Zaki Pratama 10122078