

PEMBUATAN SISTEM PENJADWALAN KEDATANGAN KAPAL DI PELABUHAN BAKAUHENI DENGAN ALGORITMA PEWARNAAN GRAF BERBASIS GUI DENGAN MENGGUNAKAN BAHASA PYTHON

Rangga Wahyu Pratama, Paskah Abadi Simanullang, Felix John P. Hutabarat, Putri Harliana

Ilmu Komputer, Universitas Negeri Medan
Jalan William Iskandar Ps. V Medan, Indonesia
ranggawahyupratama386@mhs.unimed.ac.id

ABSTRAK

Penjadwalan kapal di Pelabuhan Bakauheni merupakan tantangan besar mengingat tingginya volume kedatangan kapal yang seringkali menyebabkan penumpukan dan keterlambatan. Sistem ini dikembangkan untuk mengoptimalkan jadwal kedatangan kapal menggunakan algoritma pewarnaan graf berbasis GUI pada Python. Dengan menerapkan algoritma pewarnaan graf, kapal diidentifikasi melalui “warna” yang berbeda sehingga menghindari bentrokan waktu penggunaan dermaga. Data terkait kedatangan dan keberangkatan kapal akan dikumpulkan melalui antarmuka yang mudah digunakan berkat integrasi pustaka GUI seperti Tkinter. Proses pengumpulan dan pengelolaan data ini memungkinkan setiap kapal untuk dialokasikan dermaga secara efisien, mengurangi waktu tunggu, dan memaksimalkan rotasi dermaga. Sistem ini kemudian diuji untuk melihat efektivitas pengurangan penumpukan dan peningkatan kelancaran logistik. Hasil pengujian menunjukkan bahwa sistem ini mampu mengurangi waktu tunggu dan penumpukan kapal secara signifikan, serta mengoptimalkan pemanfaatan dermaga dengan menambah jeda waktu 30 menit untuk memastikan dermaga siap untuk kapal berikutnya. Sistem juga mampu mengatur jadwal dengan optimal, menunjukkan potensi signifikan sebagai solusi penjadwalan yang fleksibel dan efisien di pelabuhan padat. Temuan ini menegaskan bahwa pendekatan pewarnaan graf dapat diterapkan dalam skenario pelabuhan lainnya untuk meningkatkan efisiensi manajemen fasilitas pelabuhan secara keseluruhan.

Kata kunci : Algoritma Pewarnaan Graf, Penjadwalan Kapal, Python, Pelabuhan

1. PENDAHULUAN

Pelabuhan Bakauheni merupakan salah satu pelabuhan penyeberangan di Indonesia yang terletak di Lampung Selatan. Pelabuhan ini menghubungkan Pulau Sumatra dengan Pulau Jawa melalui jalur laut, dengan Pelabuhan Merak sebagai mitra penyeberangan. Setiap harinya, ribuan penumpang dan kendaraan melintasi jalur ini, menjadikan Pelabuhan Bakauheni sebagai pusat aktivitas transportasi laut yang padat. Dalam beberapa dekade terakhir, peningkatan volume lalu lintas penumpang dan barang, terutama melalui jalur penyeberangan ini, sangat penting adanya sistem manajemen yang lebih efisien.

Salah satu tantangan utama dalam pengelolaan pelabuhan adalah penjadwalan kedatangan dan keberangkatan kapal yang tepat waktu. Penumpukan kapal di dermaga akibat penjadwalan yang kurang efisien dapat menyebabkan keterlambatan, mengganggu arus logistik, dan merugikan perekonomian. Oleh karena itu, diperlukan sistem penjadwalan yang mampu mengatur kedatangan kapal secara optimal, memastikan bahwa dermaga digunakan secara maksimal tanpa mengganggu alur penyeberangan.

Pelabuhan merupakan pusat aktivitas ekonomi penting, dengan lalu lintas kapal yang padat dan dinamis. Selain itu, pelabuhan berfungsi sebagai pusat logistik dan transportasi yang membutuhkan solusi penjadwalan yang fleksibel untuk menghadapi

tantangan operasional yang semakin kompleks akibat kemajuan teknologi. Hal ini memberikan alasan yang kuat untuk memilih pelabuhan sebagai fokus penelitian, mengingat bahwa perbaikan efisiensi dalam penjadwalan di pelabuhan dapat memberikan dampak signifikan terhadap keseluruhan rantai pasokan global. Untuk mengelola kedatangan kapal secara efisien, diperlukan sistem penjadwalan yang mampu mengurangi waktu tunggu dan meningkatkan efisiensi operasional. Algoritma pewarnaan graf, yang dikenal efektif dalam mengatasi masalah penjadwalan, dapat diadaptasi untuk mengelola kedatangan kapal di pelabuhan.

Penggunaan algoritma pewarnaan graf dapat menjadi solusi untuk mengatasi masalah penjadwalan kapal. Algoritma ini bertujuan untuk menghindari bentrokan waktu kedatangan kapal dengan memetakan setiap kapal ke dalam “warna” sehingga tidak ada kapal yang tiba pada waktu yang sama dan menggunakan dermaga yang sama. Dengan penerapan algoritma ini yang berbasis antarmuka grafis (GUI) menggunakan bahasa pemrograman Python, sistem penjadwalan dapat dibuat lebih intuitif dan mudah digunakan oleh pengelola pelabuhan.

Dengan menggunakan bahasa pemrograman Python, sistem ini dapat diimplementasikan secara komputasional untuk menyusun jadwal kedatangan yang optimal, sehingga dapat meminimalkan konflik jadwal dan memaksimalkan penggunaan fasilitas Pelabuhan.

Salah satu konsep dalam teori graf yang digunakan dalam memodelkan permasalahan adalah pewarnaan graf (graph colouring). Banyak persoalan kehidupan sehari-hari yang memiliki karakteristik seperti mewarnai graf (graph colouring), yaitu pewarnaan titik (vertex), pewarnaan sisi (edge), dan pewarnaan wilayah (region). Salah satu permasalahan yang banyak memanfaatkan teori graf adalah masalah penjadwalan. Penjadwalan merupakan kegiatan untuk mengalokasikan sejumlah sumber daya yang tersedia untuk memastikan bahwa perencanaan dapat berjalan dengan baik dengan waktu dan tenaga yang digunakan secara efisien. Salah satu bidang yang memanfaatkan penjadwalan adalah kedatangan kapal di pelabuhan. [1]

Algoritma pewarnaan graf telah terbukti efektif dalam penjadwalan kedatangan kapal di pelabuhan, dengan cara menghindari konflik waktu kedatangan kapal yang tiba di dermaga yang sama melalui pewarnaan simpul yang mewakili kapal dan dermaga. Pendekatan ini mengoptimalkan penggunaan dermaga dan meningkatkan efisiensi operasional, serupa dengan penerapan pada pengaturan lalu lintas yang mampu mengurangi durasi lampu hijau dan merah secara signifikan. Selain itu, algoritma ini juga diterapkan dalam pengelolaan perpustakaan, membantu menyusun buku secara teratur dan efisien dengan menggunakan pewarnaan simpul untuk mengelompokkan kategori buku. [2]

2. TINJAUAN PUSTAKA

2.1. Python

Python adalah bahasa pemrograman yang sangat populer di kalangan perusahaan besar dan para pengembang untuk menciptakan berbagai aplikasi berbasis desktop, web, dan mobile. Bahasa ini pertama kali dibuat oleh Guido van Rossum di Belanda pada tahun 1990. Nama Python sendiri terinspirasi dari acara televisi favorit Van Rossum, *Monty Python's Flying Circus*. Awalnya, ia mengembangkan Python sebagai hobi, tetapi bahasa ini berkembang menjadi salah satu pilihan utama dalam industri dan pendidikan. Hal ini karena Python dikenal sederhana, ringkas, memiliki sintaks intuitif, serta dilengkapi dengan pustaka yang sangat luas. [3]

Python memiliki keunggulan sebagai bahasa pemrograman untuk aplikasi penjadwalan berkat pustaka GUI, seperti *Tkinter*, yang mendukung pengembangan antarmuka yang ramah pengguna. Python mempermudah proses pembuatan algoritma kompleks, seperti pewarnaan graf, untuk mengelola konflik waktu dalam penjadwalan. Dengan fleksibilitas pustaka dan struktur yang sederhana, Python memungkinkan pengembangan algoritma secara cepat dan praktis, yang sangat dibutuhkan dalam aplikasi penjadwalan di sektor logistik. [4]

2.2. Pelabuhan

Penjadwalan kapal yang efektif sangat krusial bagi pelabuhan untuk mengurangi waktu tunggu dan

menghindari penumpukan kapal di dermaga. Sistem penjadwalan yang efisien penting dalam menjaga kelancaran operasi pelabuhan dengan mengelola kedatangan kapal. Algoritma berbasis graf menjadi solusi yang dapat diterapkan untuk mendukung alokasi dermaga yang tepat waktu, mengoptimalkan penggunaan fasilitas dermaga, dan menjaga agar aliran logistik tetap berjalan tanpa hambatan yang merugikan. [5]

2.3. Algoritma Pewarnaan Graf

Pewarnaan graf adalah teknik penting dalam penjadwalan yang menghindari konflik waktu. Teknik ini membagi interval waktu yang berbeda untuk setiap kapal menggunakan “warna” sebagai penanda agar tidak ada dua kapal yang dijadwalkan bersamaan di dermaga yang sama. Implementasi ini mengurangi potensi tabrakan waktu dan memaksimalkan efektivitas dermaga. Pendekatan ini menawarkan solusi efektif dalam mengatur jadwal pada sistem yang melibatkan sumber daya yang terbatas dan waktu yang sangat terstruktur, cocok untuk konteks pelabuhan dengan kepadatan tinggi. [6]

2.4. Implementasi GUI

Graphical User Interface (GUI) adalah sebuah program yang memungkinkan pengguna untuk berinteraksi dengan sistem melalui tampilan visual dan elemen grafis, seperti ikon, tombol, dan jendela, tanpa harus menggunakan perintah teks. Untuk menciptakan UI yang efektif, diperlukan beberapa tahapan, salah satunya adalah pembuatan *design guideline*. Design guideline berfungsi sebagai acuan dalam menciptakan UI yang serasi dengan mempertimbangkan berbagai elemen visual. Beberapa aspek visual yang harus diperhatikan dalam design guideline mencakup warna, tipografi, dan elemen grafis lainnya agar tampilan keseluruhan terasa konsisten dan mudah dipahami. [7]

GUI yang mudah digunakan memainkan peran penting dalam penjadwalan real-time, khususnya bagi operator pelabuhan yang tidak memiliki latar belakang teknis. Antarmuka yang intuitif memungkinkan pengguna untuk memahami dan mengatur jadwal tanpa kendala teknis. Dengan menggunakan pustaka GUI Python, sistem dapat menyajikan data secara visual yang memudahkan penjadwalan interaktif. Fitur GUI juga membuat sistem dapat diakses dengan lebih luas, baik oleh operator pelabuhan maupun pihak lain yang terlibat dalam pengelolaan jadwal kedatangan kapal. [8]

2.5. Penjadwalan

Penjadwalan adalah proses pengaturan urutan dan alokasi sumber daya, seperti waktu, fasilitas, peralatan, dan tenaga kerja, yang diperlukan untuk menyelesaikan setiap operasi. Proses ini bertujuan untuk mencapai optimasi terhadap satu atau lebih target yang ingin dicapai, seperti efisiensi waktu, pemanfaatan sumber daya, atau hasil produksi. Dalam berbagai konteks, penjadwalan juga

mempertimbangkan prioritas dan urutan pelaksanaan tugas untuk mencapai tujuan secara efektif. Dalam hirarki pengambilan keputusan, penjadwalan menjadi langkah akhir yang disusun sebelum memulai operasi, memastikan semua sumber daya siap digunakan secara efisien agar kegiatan berjalan sesuai rencana. [9]

Implementasi pewarnaan graf di berbagai sektor menunjukkan efektivitas metode ini dalam menangani konflik waktu secara optimal. Contoh penerapan pewarnaan graf adalah dalam penjadwalan kelas di perguruan tinggi dan sekolah untuk menghindari tumpang tindih jadwal. Aplikasi serupa dalam konteks pelabuhan membantu memastikan bahwa waktu kedatangan kapal terdistribusi dengan baik, menjaga kelancaran aliran operasional dan memastikan pemanfaatan fasilitas secara maksimal. Metode ini terbukti efektif dalam mengelola sistem dengan kebutuhan sumber daya dan waktu yang saling bersaing. [10]

3. METODE PENELITIAN

3.1. Pendekatan Deskriptif Kualitatif

Metode yang digunakan pada penelitian ini adalah metode deskriptif kualitatif, metode penelitian deskriptif kualitatif adalah metode yang dapat membantu peneliti mendeskripsikan atau menggambarkan fenomena atau peristiwa yang sedang terjadi secara mendalam dan rinci, menggunakan data-data yang biasanya berbentuk kata-kata, teks, gambar, atau cerita, bukan angka ataupun statistik. serta metode ini digunakan juga untuk memahami makna, persepsi, atau pandangan subjek penelitian terhadap suatu fenomena.

3.2. Pengembangan Algoritma Pewarnaan Graf

Algoritma ini dirancang untuk mengalokasikan waktu kedatangan kapal sehingga tidak terjadi konflik di dermaga. Konsep pewarnaan graf sangat efektif dalam mengelola sumber daya yang terbatas dan penjadwalan yang kompleks, yang menekankan pentingnya algoritma ini dalam berbagai aplikasi penjadwalan.

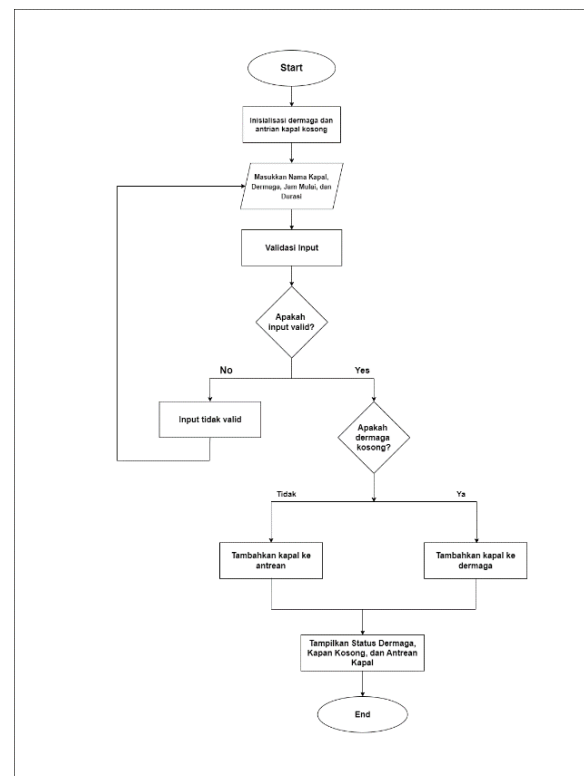
3.3. Implementasi Sistem Berbasis GUI

Penelitian ini juga mencakup pembuatan antarmuka grafis (GUI) menggunakan bahasa pemrograman Python, yang memberikan kemudahan bagi pengguna dalam mengoperasikan sistem. Penggunaan pustaka seperti Tkinter memungkinkan pengembangan antarmuka yang intuitif dan interaktif. GUI (Graphical User Interface) dirancang agar pengguna dapat dengan mudah memasukkan data dan memantau status dermaga secara real-time, meningkatkan pengalaman pengguna secara keseluruhan.

4. HASIL DAN PEMBAHASAN

4.1. Pemaparan Algoritma

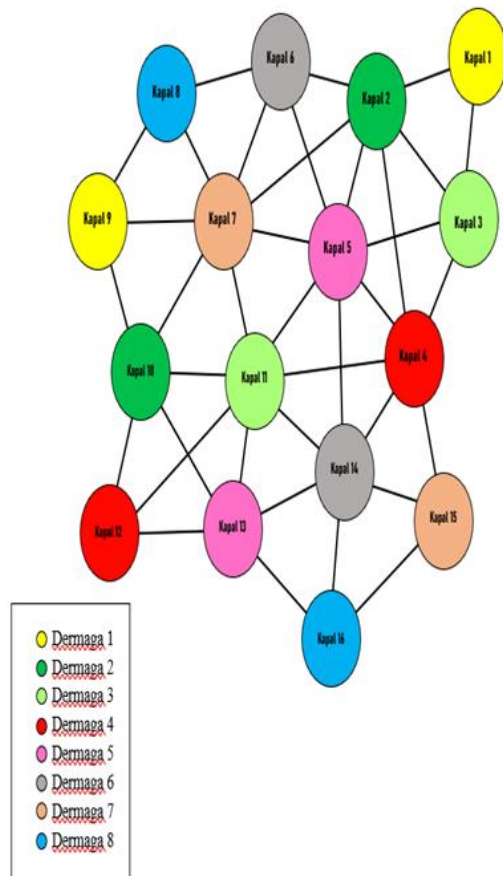
Sebuah sistem yang terprogram menggunakan bahasa pemrograman pasti memiliki sebuah algoritma, algoritma dapat dipaparkan (representasikan) dengan berbagai cara yaitu, *language structure*, pseudocode, flowchart, dan block diagram. Tujuan dari memrepresentasikan algoritma adalah agar algoritma dapat lebih mudah dilihat cara kerjanya dari awal hingga akhir dan jika ada keperluan lain seperti penambahan fungsi algoritma dapat dengan mudah diubah sesuai keinginan. Disini kami akan memaparkan algoritma programnya menggunakan flowchart.



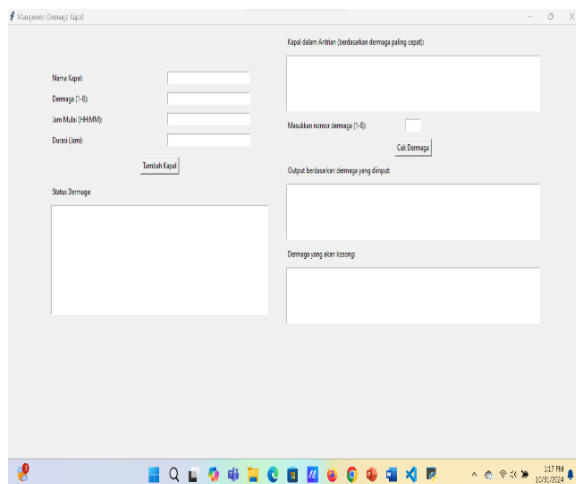
Gambar 1. Flowchart Sistem

4.2. Implementasi dan Pembahasan

Sistem ini di buat berdasarkan teori graf lebih tepatnya algoritma pewarnaan simpul (*vertex coloring*) pada sebuah graf, algoritma *vertex coloring* adalah algoritma dimana setiap simpul dalam graf diberi warna dengan aturan bahwa dua simpul yang terhubung dengan sisi yang sama tidak boleh memiliki warna yang sama. *Vertex coloring* dapat membantu permasalahan seperti penjadwalan kapal di Pelabuhan agar kapal-kapal yang datang tidak bertabrakan waktu dalam menggunakan dermaga yang sama. Di mana setiap simpul itu mewakili setiap dermaga di Pelabuhan dan saat ada kapal yang bersandar maka dermaga itu akan diberi warna sesuai yang ditentukan.



Gambar 2. Graf Berwarna



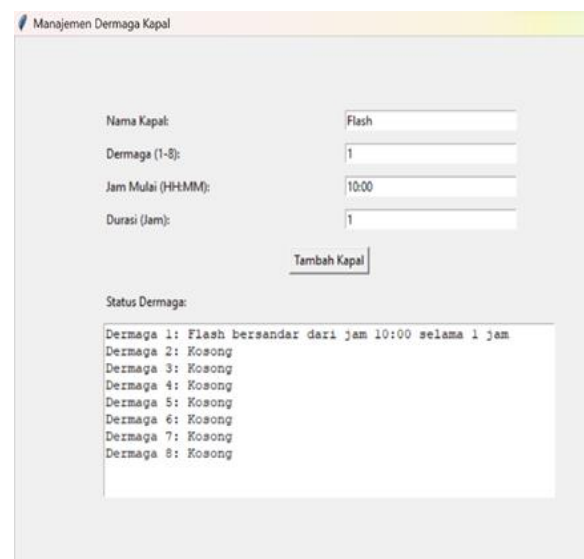
Gambar 3. Tampilan awal

Seperti yang terlihat di atas sistem ini dibuat berbasis GUI, GUI adalah sebuah tampilan antarmuka antara sistem dengan pengguna serta pengguna juga dapat berinteraksi dengan program dengan lebih mudah dipahami melalui elemen visual seperti ketikan, tombol maupun elemen lainnya yang dapat ditambahkan melalui bahasa pemrograman yang digunakan.

Pada sistem ini tampilan GUI diprogram menggunakan Bahasa python, python dipilih karena beberapa alasan yaitu python memiliki banyak library yang memudahkan pengembangan GUI (*Graphical User*

Interface), seperti Tkinter yang menyediakan komponen-komponen antarmuka visual seperti tombol, kotak input, dan table yang dapat membuat tampilan menjadi lebih interaktif dan mudah digunakan. Selain itu python juga memiliki *library* seperti *datetime* yang dapat yang sangat cocok untuk memproses jadwal kapal, menghitung durasi bersandar, serta memastikan dermaga digunakan secara efisien.

Pada sistem ini pertama pengguna harus memasukkan nama kapal, dermaga yang akan dimasuki, jam datang dan lama bersandar kapal kemudian akan ditampilkan pada status dermaga. Selain ditampilkan status dermaga pada saat inputan pertama juga ditampilkan kapan dermaga akan kosong pada kotak kanan bawah.



Gambar 4. Penginputan kapal yang masuk



Gambar 5. Output dermaga yang akan kosong

Dapat terlihat pula bahwa waktu keluar dilebihkan 30 menit dari waktu seharusnya ini bertujuan agar dermaga benar-benar siap digunakan untuk persandaran kapal selanjutnya.

Pada saat dermaga penuh dan pengguna menginput kapal lagi maka sistem akan memberi 2 output yaitu berdasarkan kapal paling cepat bersandarnya dan dermaga mana yang ingin dimasuki sesuai dengan input dari pengguna.

Manajemen Dermaga Kapal

Nama Kapal: Lapice

Dermaga (1-8): 3

Jam Mulai (HH:MM): 10:00

Durasi (Jam): 4

Tambah Kapal

Status Dermaga:

Dermaga 1: Flash bersandar dari jam 10:00 selama 1 jam
 Dermaga 2: Grove bersandar dari jam 10:00 selama 2 jam
 Dermaga 3: Arden bersandar dari jam 10:00 selama 3 jam
 Dermaga 4: Boston bersandar dari jam 10:00 selama 4 jam
 Dermaga 5: Calder bersandar dari jam 10:00 selama 5 jam
 Dermaga 6: Diana bersandar dari jam 10:00 selama 6 jam
 Dermaga 7: Eugene bersandar dari jam 10:00 selama 7 jam
 Dermaga 8: Max bersandar dari jam 10:00 selama 8 jam

Gambar 6. Dermaga penuh

Kapal dalam Antrian (berdasarkan dermaga paling cepat):

Kapal dalam antrian (berdasarkan dermaga paling cepat):
 Kapal 'Lapice' dapat bersandar di dermaga 1 setelah kapal 'Flash' keluar pada jam 11:30

Gambar 7. Output berdasarkan kapal paling cepat bersandarnya

Pada output dermaga mana yang ingin dimasukan oleh kapal yang akan datang sesuai inputan, pengguna harus memilih dermaga mana yang akan dimasukan dapat terlihat pada gambar di bawah ini

Masukkan nomor dermaga (1-7): 3

Cek Dermaga

Output berdasarkan dermaga yang diinput:

Kapal 'Arden' di dermaga 3 akan selesai pada jam 13:30

Gambar 8. Output dermaga sesuai input

4.3. Hasil Penelitian

Penelitian ini menghasilkan sistem penjadwalan kapal di Pelabuhan Bakauheni yang memanfaatkan algoritma pewarnaan graf dan antarmuka GUI berbasis Python. Sistem ini mengatur alokasi dermaga berdasarkan input waktu kedatangan dan durasi bersandar dari pengguna, menambahkan jeda 30 menit agar tidak terjadi bentrokan penggunaan dermaga. Setiap dermaga direpresentasikan sebagai simpul (*vertex*) dengan warna berbeda, sehingga ketika dermaga digunakan oleh satu kapal, kapal berikutnya dialokasikan ke dermaga dengan warna lain, memastikan

penggunaan dermaga yang efisien dan bebas konflik waktu.

GUI yang dirancang memungkinkan pengguna memonitor status dermaga dan jadwal kapal secara *real-time*, memberikan informasi kapan dermaga kosong dan siap digunakan kembali. Hasil pengujian menunjukkan bahwa sistem ini secara signifikan mengurangi potensi penumpukan kapal, memastikan pemanfaatan dermaga yang optimal, dan cocok diterapkan di pelabuhan dengan lalu lintas padat.

5. KESIMPULAN DAN SARAN

Sistem penjadwalan kapal di Pelabuhan Bakauheni yang dikembangkan menggunakan algoritma pewarnaan graf berbasis GUI Python mampu mengoptimalkan alokasi dermaga dengan signifikan. Sistem ini berhasil mengurangi konflik waktu penggunaan dermaga melalui penerapan pewarnaan graf, di mana setiap kapal dialokasikan ke dermaga yang berbeda dengan penanda warna, sehingga mencegah terjadinya bentrokan jadwal. Penambahan jeda waktu 30 menit untuk setiap kapal setelah bersandar terbukti efektif dalam meminimalkan potensi penumpukan, memastikan dermaga siap digunakan kembali tepat waktu, dan meningkatkan efisiensi operasional pelabuhan. Saran untuk pengembangan lebih lanjut adalah menambahkan integrasi teknologi IoT untuk pemantauan dermaga secara *real-time* dan fitur prediksi kedatangan kapal berdasarkan kondisi cuaca serta metode optimasi tambahan untuk menghadapi fluktuasi lalu lintas kapal yang lebih kompleks.

DAFTAR PUSTAKA

- [1] L. Maro and L. K. S. Purab, "1," *Jurnal Ilmiah Wahana Pendidikan*, vol. 7, no. 6, pp. 193–197, 2021.
- [2] A. Yaqin *et al.*, "4.+125-136," *Jurnal Ilmiah Matematika dan Pendidikan Matematika*, vol. 13, no. 2, pp. 125–136, 2023.
- [3] M. Romzi and B. Kurniawan, "PEMBELAJARAN PEMROGRAMAN PYTHON DENGAN PENDEKATAN LOGIKA ALGORITMA," *Jurnal Teknik Informatika Mahakarya*, vol. 3, no. 2, pp. 37–44, 2020.
- [4] T. Bonald, N. de Lara, Q. Lutz, T. Paris, and B. Charpentier BERTRANDCHARPENTIER, "Scikit-network: Graph Analysis in Python," *Journal of Machine Learning Research*, vol. 21, pp. 1–6, 2020, [Online]. Available: <https://scikit-network.readthedocs.io/en/latest/>.
- [5] E. Deogratias, "Available online at ijci.wcci-international.org International Journal of Curriculum and Instruction 14(2) Using graph coloring for effective timetable scheduling at ordinary secondary level conditions of the Creative Commons Attribution license (CC BY-NC-ND) (<http://creativecommons.org/licenses/by-nc->

- nd/4.0/),” *International Journal of Curriculum and Instruction*, pp. 1166–1188, 2022.
- [6] A. Malviya, B. Agrawal, S. Kumar, and A. Mansuri, “Study of algorithm for coloring in various graph,” *International Journal of Statistics and Applied Mathematics*, vol. 7, no. 2, pp. 88–91, Mar. 2022, doi: 10.22271/math.2022.v7.i2b.798.
- [7] D. L. Santoso, I. Mufidah, and M. Iqbal, “Perancangan User Interface Website Laboratorium Analisis Perancangan Kerja dan Ergonomi Universitas Telkom dengan Metode Double Diamond,” *INNOVATIVE: Journal Of Social Science Research*, vol. 4, no. 5, pp. 1004–1016, 2024, [Online]. Available: <https://j-innovative.org/index.php/Innovative>
- [8] S. Abolhassani Khajeh, M. Saberikamarposhti, and A. M. Rahmani, “Real-Time Scheduling in IoT Applications: A Systematic Review,” *Sensors*, vol. 23, no. 1, pp. 1–35, Jan. 2023, doi: 10.3390/s23010232.
- [9] S. N. Sari, Roberto Kaban, Abdul Khaliq, and Ayu Andari, “Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Metode Hybrid Artificial Bee Colony (HABC),” *Jurnal Nasional Teknologi Komputer*20, vol. 2, no. 1, pp. 20–32, 2022.
- [10] J. Ojala, A. Minard, G. Madraki, and S. Mousavian, “Graph-based modeling in shop scheduling problems: Review and extensions,” *Applied Sciences (Switzerland)*, vol. 11, no. 11, pp. 1–48, 2021, doi: 10.3390/app11114741.