

# Penerapan Algoritma Greedy Dalam Penentuan Prioritas Pengerjaan Tugas Kuliah Mahasiswa

Naufal Fakhri\*, Ary Sigit Julian, Darryl Sandi Augustin, Zhykwa Ceryl Mavanudin, Efrans Christian

Fakultas Teknik, Program Studi Teknik Informatika, Universitas Palangka Raya, Palangka Raya, Indonesia

Email: <sup>1,\*</sup>naufalfakhri@mhs.eng.upr.ac.id, <sup>2</sup>arysigitjulian@mhs.eng.upr.ac.id, <sup>3</sup>darrylsandiaugustin@mhs.eng.upr.ac.id,

<sup>4</sup>zhykwacerylmavanudin@mhs.eng.upr.ac.id, <sup>5</sup>efranschristian2@gmail.com

Email Penulis Korespondensi: naufalfakhri@mhs.eng.upr.ac.id

**Abstrak**—Tugas kuliah merupakan bagian penting bagi mahasiswa untuk memahami materi dan mencapai tujuan pembelajaran. Namun, tuntutan pengerjaan tugas dari berbagai mata kuliah seringkali membuat mahasiswa kebingungan dalam menentukan prioritas. Hal ini dapat berakibat pada pengerjaan tugas yang tidak optimal dan berpotensi menurunkan kualitas pembelajaran. Penelitian ini bertujuan untuk membantu mahasiswa dalam menentukan prioritas penyelesaian tugas kuliah dengan efektif menggunakan algoritma greedy. Algoritma ini akan membantu mahasiswa dalam membuat penjadwalan pengerjaan tugas yang optimal dengan mempertimbangkan berbagai faktor, seperti tenggat waktu dan tingkat kesulitan tugas. Metode penelitian ini menggunakan simulasi dengan berbagai mata kuliah sebagai variabel dan bobot yang berbeda, seperti tenggat waktu dan tingkat kesulitan tugas. Hasil sementara menunjukkan bahwa implementasi algoritma greedy terbukti lebih mengutamakan tenggat waktu tugas daripada tingkat kesulitan tugas, sehingga dapat meningkatkan efisiensi penyelesaian tugas, dengan lebih banyak tugas diselesaikan tepat waktu dan penurunan tingkat stres mahasiswa. Harapan dari penelitian ini yaitu untuk memberikan cara bagi mahasiswa untuk menyelesaikan tugas kuliah secara lebih efisien dan efektif. Hal ini dapat membantu meningkatkan kualitas belajar dan mencapai hasil belajar yang baik. Dengan pendekatan ini, mahasiswa dapat merasakan manfaat langsung dari penelitian ini dalam kehidupan akademik mereka, mengurangi stres dan meningkatkan produktivitas mereka.

**Kata Kunci:** Tugas Kuliah; Prioritas; Algoritma Greedy; Jadwal; Efektifitas

**Abstract**—College assignments play a crucial role in enabling students to grasp course material and achieve learning objectives. However, the demands of completing assignments from multiple courses often leave students confused about prioritizing tasks. This can lead to suboptimal assignment completion and potentially lower the quality of learning. This research aims to assist students in effectively prioritizing college assignment completion using the greedy algorithm. This algorithm will aid students in creating optimal assignment scheduling by considering various factors such as deadlines and assignment difficulty. The research methodology employs simulation with diverse courses as variables and different weights, such as deadlines and assignment difficulty. Preliminary results indicate that the implementation of the greedy algorithm prioritizes assignment deadlines over assignment difficulty, potentially enhancing assignment completion efficiency, leading to more timely task completion and a reduction in student stress levels. The anticipated outcome of this research is to provide students with a method for completing college assignments more efficiently and effectively. This could contribute to improving learning quality and achieving positive learning outcomes. By adopting this approach, students can directly benefit from the research in their academic lives, reducing stress and enhancing their productivity.

**Keywords:** College Assignments; Prioritization; Greedy Algorithm; Schedule; Effective

## 1. PENDAHULUAN

Seringkali, mahasiswa dihadapkan dengan beragam tugas dari berbagai mata kuliah yang harus dikerjakan dalam jangka waktu yang ditentukan. Dalam situasi ini, penentuan prioritas dalam menyelesaikan tugas menjadi hal yang sangat krusial. Dengan menentukan prioritas yang benar, mahasiswa dapat mengatur waktu dan tenaga mereka lebih efisien, sehingga dapat mencapai hasil yang lebih optimal. Tetapi, penentuan prioritas bukanlah hal yang sederhana bagi mahasiswa. Mereka harus memperhitungkan berbagai aspek, seperti batas waktu yang mendekati atau level kesulitan, untuk menentukan tugas apa yang harus diselesaikan terlebih dahulu. Penelitian ini dirancang untuk membantu mahasiswa dalam menentukan urutan penyelesaian tugas-tugas kuliah dengan cara yang paling efisien menggunakan algoritma greedy. Lebih lanjut, penelitian ini juga mengimplementasikan algoritma greedy dalam konteks penjadwalan tugas, berdasarkan penelitian-penelitian terdahulu yang relevan.

Studi pertama yang menjadi referensi dalam penelitian ini membicarakan tentang pembuatan jadwal peminjaman aset tetap di Desa Cipelah dengan memanfaatkan algoritma greedy. Studi ini dilaksanakan oleh Asep Ririh Riswaya, Jajat Sudrajat, dan Dheni Apriansani Budiman pada tahun 2023. Fokus utama dari studi ini adalah untuk mengatasi tantangan dalam proses pencatatan dan pengelolaan aset di desa, termasuk keterbatasan dalam pengetahuan teknologi dan variasi jenis aset. Tujuan dari penelitian ini adalah untuk meningkatkan efisiensi dan nilai tambah dalam pengelolaan aset melalui proses komputerisasi dan penjadwalan peminjaman aset. Namun, pencatatan aset desa masih dilakukan secara manual dan belum menggunakan website, yang berpotensi menyebabkan kesalahpahaman, penipuan, atau kehilangan data[1].

Studi kedua yang menjadi rujukan dalam analisis dan penyelesaian masalah dalam penelitian ini adalah mengenai perbandingan antara algoritma welch powell dan algoritma greedy dalam mengoptimalkan penjadwalan ruang kuliah di semester genap Fakultas Teknik. Penelitian ini, yang dilakukan oleh Yohanna Christela Oktaviani dan Yosefina Finsensia Riti pada tahun 2023, membahas secara detail tentang perbandingan kedua algoritma tersebut dalam konteks penjadwalan ruang kuliah di Fakultas Teknik, Universitas Katolik Darma Cendika. Kedua algoritma tersebut digunakan untuk menciptakan efisiensi waktu dan mencegah benturan dalam penjadwalan. Meskipun algoritma greedy membutuhkan waktu eksekusi yang lebih lama karena sifat heuristiknya, dari sudut pandang pemrograman, algoritma ini memiliki

kelebihan dibandingkan dengan algoritma welch powell. Namun, algoritma welch powell menghasilkan detail yang lebih baik[2].

Studi ketiga yang menjadi rujukan dalam penelitian ini adalah tentang optimalisasi implementasi algoritma greedy dalam proses konversi mata uang Rupiah. Penelitian yang dilakukan oleh Hery Sunandar dan Pristiwanto pada tahun 2019 ini berpusat pada upaya mencari metode terbaik untuk konversi mata uang Rupiah dengan menggunakan algoritma greedy. Studi ini menyoroti kebutuhan dalam perbankan untuk penukaran uang nasabah yang efisien. Aplikasi konversi mata uang Rupiah telah dirancang untuk memudahkan proses konversi yang biasanya memakan waktu dan kompleks, dengan algoritma greedy yang digunakan untuk mencari solusi terbaik dalam menentukan denominasi uang terkecil[3].

Studi keempat yang menjadi referensi dalam penelitian ini adalah mengenai implementasi algoritma greedy dalam menemukan rute terpendek ke berbagai lembaga penting di Kota Argamakmur, Kabupaten Bengkulu Utara. Penelitian yang dilakukan oleh Yulia Darnita dan Rozali Toyib pada tahun 2019 ini berpusat pada upaya mencari rute terpendek ke instansi-instansi di Kota Argamakmur dengan menggunakan algoritma greedy. Studi ini menyoroti pentingnya mendapatkan akses jarak dan waktu yang tepat untuk menentukan posisi instansi yang akurat bagi masyarakat, khususnya mereka yang berasal dari luar daerah. Hasil studi mencakup beberapa jalur pendek seperti Pemda dan Polres, dengan perhitungan berdasarkan bobot arah yang dilewati, namun tidak memperhitungkan aspek lain seperti keramaian penduduk[4].

Studi kelima yang menjadi referensi dalam penelitian ini adalah tentang solusi heuristik untuk isu-isu terkait rute inventori berkala untuk produk-produk segar. Penelitian yang dilakukan oleh Wahyu Satria, Mohamad Sofitra, dan Noveicalistus H. Djangu pada tahun 2022 ini berpusat pada pengembangan metode heuristik untuk optimasi jalur inventori dalam manajemen produk segar. Tujuannya adalah untuk mencegah kerusakan produk dan memaksimalkan pendapatan. Produk-produk ini termasuk produk pertanian yang mudah rusak dan mengalami penurunan kualitas hingga masa konsumsi. Dalam penelitian ini, metode Vendor Managed Inventory (VMI) dan VOI digabungkan. Algoritma greedy diterapkan untuk mengoptimalkan distribusi produk ke retailer yang memiliki riwayat permintaan yang tinggi dengan segera, meskipun beberapa aspek seperti alokasi kendaraan dan jadwal distribusi belum dipertimbangkan[5].

Berbagai penelitian telah menunjukkan penerapan algoritma greedy dalam beragam situasi. Misalnya, penelitian tentang pembuatan jadwal peminjaman aset desa menggunakan algoritma greedy sebagai metode utama. Dalam konteks lain, penelitian yang membandingkan algoritma Welch-Powell dan algoritma greedy dalam penjadwalan ruang kuliah menunjukkan bahwa algoritma greedy dapat meningkatkan efisiensi waktu dan mencegah konflik jadwal. Selain itu, penelitian tentang optimalisasi penggunaan algoritma greedy dalam konversi mata uang Rupiah menunjukkan bahwa algoritma greedy dapat menentukan denominasi terkecil sebagai solusi optimal. Akhirnya, penelitian tentang pencarian rute terpendek ke berbagai institusi penting di Kota Argamakmur menggunakan algoritma greedy untuk menemukan jalur tercepat.

Studi ini menggunakan algoritma greedy untuk menentukan urutan penyelesaian tugas berdasarkan prioritas. Ini berbeda dari penelitian-penelitian sebelumnya yang menerapkan algoritma greedy dalam berbagai konteks. Sebagai contoh, dalam studi tentang pembuatan jadwal pinjaman aset tetap di desa, algoritma greedy digunakan untuk merancang jadwal, bukan untuk menetapkan prioritas tugas berdasarkan faktor-faktor seperti tingkat kesulitan dan tenggat waktu. Demikian juga, dalam studi tentang optimalisasi penjadwalan kelas, fokus utamanya adalah efisiensi waktu dan pencegahan benturan jadwal, bukan pada bobot tugas. Selanjutnya, dalam studi tentang optimalisasi proses konversi mata uang Rupiah, algoritma greedy digunakan untuk mencari denominasi uang terendah, bukan untuk menentukan urutan penyelesaian tugas. Akhirnya, dalam studi tentang mencari rute terpendek ke berbagai institusi penting, algoritma greedy digunakan untuk menemukan rute terpendek, bukan untuk menentukan urutan penyelesaian tugas berdasarkan bobot tertentu.

Dalam penelitian ini, algoritma greedy menerapkan prinsip “take what you can get now”[6]. Sehingga, solusi yang ditawarkan adalah solusi optimal yang sudah tersedia di depan mata. Dengan menetapkan prioritas tugas berdasarkan bobot atau kesulitan tugas dan tenggat waktu, salah satu dari dua parameter tersebut dipilih sebagai pertimbangan dalam menentukan solusi optimal untuk prioritas penyelesaian tugas.

Tujuan dari penelitian ini adalah untuk memahami bagaimana variasi bobot, seperti tingkat kesulitan dan tenggat waktu, mempengaruhi solusi optimal yang dihasilkan oleh algoritma greedy. Penelitian ini juga mencari metode yang efisien dalam menerapkan algoritma greedy untuk menentukan prioritas dalam penyelesaian tugas kuliah. Selain itu, penelitian ini bertujuan untuk memahami lebih lanjut tentang bagaimana algoritma greedy dapat digunakan dalam menentukan prioritas tugas kuliah. Dengan menjawab pertanyaan-pertanyaan ini, penelitian ini berharap dapat memberikan pemahaman baru tentang bagaimana algoritma greedy dapat digunakan dalam konteks pendidikan.

## **2. METODOLOGI PENELITIAN**

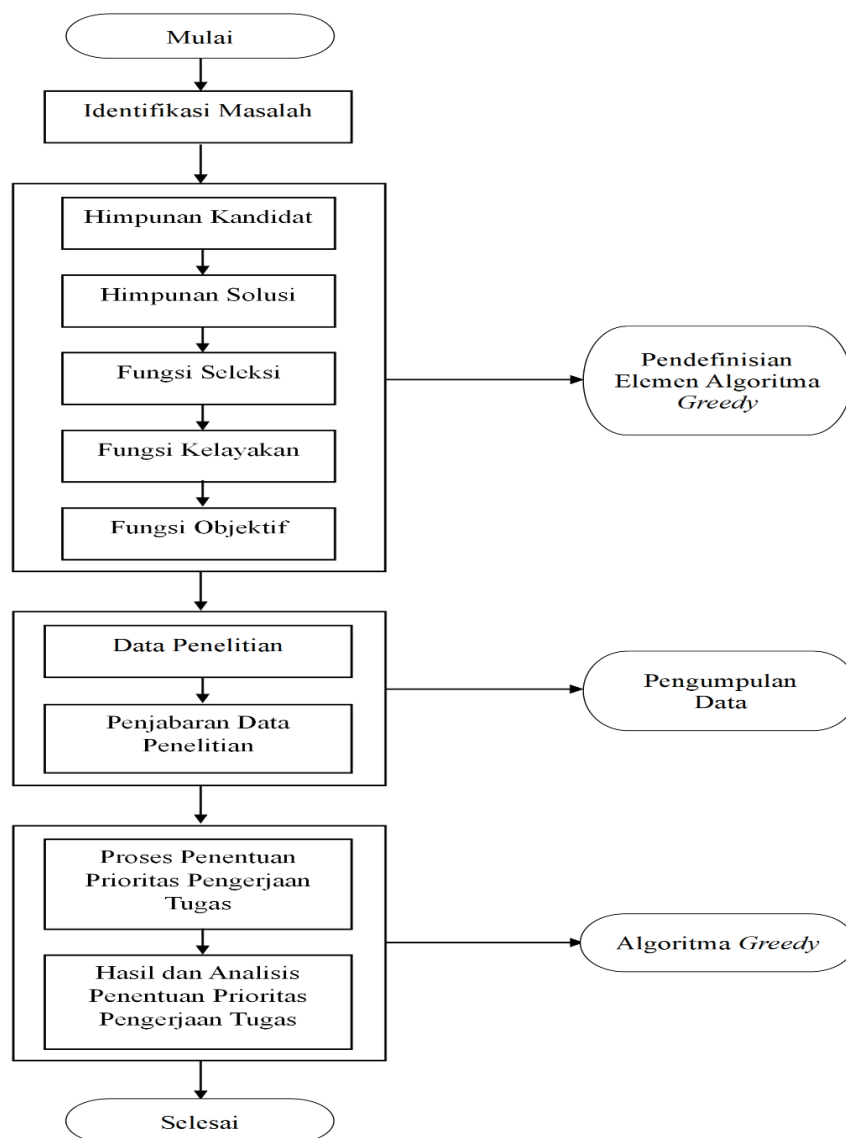
### **2.1 Definisi Algoritma Greedy**

Algoritma greedy adalah algoritma yang biasa digunakan dalam menyelesaikan masalah optimasi[7]. Terdapat dua variasi dari masalah optimasi, yaitu yang mencari nilai maksimum dan minimum[8]. Algoritma ini bekerja dengan cara memilih solusi terbaik dari sejumlah opsi yang ada untuk menyelesaikan masalah yang dihadapi[9]. Algoritma greedy mirip dengan algoritma pencarian lainnya, bekerja dengan menetapkan titik awal dan memilih titik-titik sepanjang jalur hingga mencapai tujuan atau solusi[10]. Algoritma greedy menciptakan solusi secara bertahap, di mana setiap langkah pasti

memiliki berbagai opsi dan potensi, akan tetapi langkah selanjutnya yang diambil adalah yang paling menguntungkan dalam kondisi saat ini[11]. Dengan demikian, algoritma greedy seringkali menjadi pendekatan yang dipilih dalam menangani masalah optimasi, walaupun solusi hasilkan bisa jadi tidak selalu yang terbaik[12]. Algoritma greedy mungkin tidak selalu memberikan solusi terbaik karena dua alasan, tidak melakukan peninjauan secara komprehensif pada semua opsi solusi yang tersedia seperti yang dilakukan oleh metode exhaustive search dan terdapat berbagai alternatif fungsi seleksi yang dapat digunakan[13]. Prinsip kunci dari algoritma ini adalah memanfaatkan apa yang bisa didapat saat ini[14]. Algoritma greedy ini sering digunakan, karena metode algoritma greedy terbilang cukup cepat untuk mendapatkan solusi dari permasalahan[15].

## 2.2 Tahapan Penelitian

Tahapan yang digunakan didalam penelitian ini dibagi menjadi 4 bagian, yaitu pertama ada identifikasi masalah, pendefinisian elemen algoritma greedy, pengumpulan data, dan algoritma greedy. Tahapan-tahapan ini dapat dilihat pada gambar 1 di bawah ini.



**Gambar 1.** Tahapan-Tahapan Penelitian

Berdasarkan gambar 1, penelitian yang dilakukan untuk penentuan urutan prioritas pengerjaan tugas kuliah mahasiswa dimulai dengan mengidentifikasi masalah. Pada tahap ini, identifikasi masalah dalam penelitian ini berfokus pada tantangan yang dihadapi oleh mahasiswa dalam menentukan prioritas pengerjaan tugas kuliah. Mahasiswa sering kali merasa kewalahan dengan berbagai tugas yang memiliki bobot dan tenggat waktu yang berbeda-beda. Hal ini dapat menyebabkan beberapa tugas terlambat dikerjakan atau bahkan tidak dikerjakan sama sekali. Tahap penelitian selanjutnya yaitu mendefinisikan masing-masing elemen dari algoritma greedy sebagai berikut.

### 1. Himpunan Kandidat

Himpunan kandidat mencakup elemen-elemen yang merupakan bagian dari solusi[16]. Pada penelitian yang dilakukan, himpunan kandidat berisi semua tugas-tugas yang diberikan dan dilambangkan dengan "F".

## 2. Himpunan Solusi

Himpunan solusi mencakup kandidat yang dipilih sebagai jawaban dari masalah yang sedang ditangani[17]. Pada penelitian ini, himpunan solusi berisi urutan penyelesaian tugas seperti  $\{F_1, F_2, \dots, F_n\}$  dengan  $F_i$  adalah tugas ke- $i$ .

## 3. Fungsi Seleksi

Fungsi seleksi digunakan untuk memilih kandidat yang memiliki potensi untuk mencapai solusi terbaik[18]. Pada penelitian ini, fungsi seleksi yang digunakan untuk menyeleksi urutan dari pengerjaan tugas berupa bobot tingkat kesulitan tugas dan tenggat waktu dari tugas yang ada.

## 4. Fungsi Kelayakan

Fungsi kelayakan berguna untuk mengevaluasi pilihan kandidat tertentu dapat menghasilkan solusi yang layak untuk diterima[19]. Pada penelitian ini, fungsi kelayakan berupa pengecekan setiap tugas pada mata kuliah dengan memperhatikan bobot tingkat kesulitan tugas dan tenggat waktu.

## 5. Fungsi Objektif

Fungsi objektif digunakan untuk mencapai solusi maksimal dan minimal[20]. Dalam konteks penelitian ini, fungsi objektif digunakan untuk memaksimalkan jumlah tugas yang dapat diselesaikan dalam durasi waktu tertentu.

Berikutnya, dalam menyelesaikan isu masalah penentuan urutan penyelesaian tugas kuliah mahasiswa, diperlukan langkah-langkah untuk pengumpulan data yang akan digunakan dalam studi atau penelitian yang akan dijalankan.. Berikut adalah data penelitian yang dapat terlihat pada tabel 1 di bawah ini.

Tabel 1. Data Penelitian

Tugas Ke-	Mata Kuliah	Bobot Tugas (Tingkat Kesulitan Dari Skala 1-5)	Tenggat Waktu/Batas Pengerjaan (Dalam Hari)
1	Algoritma dan Pemrograman I (F1)	3	2
2	Pengantar Teknik Informatika (F2)	2	7
3	Kalkulus I (F3)	5	6
4	Bahasa Inggris (F4)	2	6
5	Sistem Digital (F5)	4	4
6	Logika Diskrit (F6)	3	3

Dapat dilihat pada tabel 1 berupa data penelitian yang berisikan tugas-tugas kuliah dengan masing-masing mata kuliah berbeda, tingkat kesulitan, dan tenggat waktu tugas. Dari data yang dapat dilihat, mahasiswa tersebut memiliki enam buah tugas yang harus diselesaikan dan memiliki waktu maksimal 7 hari dengan limitasi 6 jam per hari untuk menyelesaikan semua tugas tersebut. Dengan menggunakan algoritma greedy, fungsi seleksi yang harus diperhatikan untuk dapat menentukan urutan prioritas pengerjaan tugas kuliah tersebut sebagai berikut.

1. Tenggat waktu pengerjaan tugas. Pada algoritma greedy, efisiensi waktu adalah aspek yang lebih penting daripada hasil yang optimal. Oleh karena itu, aspek paling pertama yang harus mahasiswa tersebut perhatikan adalah tenggat waktu dari tugas, karena semakin dekat tenggat waktu dari tugas tersebut maka semakin cepat tugas itu harus dikerjakan.
2. Nilai bobot tingkat kesulitan tugas. Bobot tingkat kesulitan tugas adalah aspek kedua yang harus diperhatikan oleh mahasiswa tersebut, aspek ini hanya akan digunakan pada tugas yang memiliki tenggat waktu yang sama. Berdasarkan tabel 1, penjabaran data berupa bobot tugas sesuai dengan angkanya dituliskan pada tabel 2 di bawah ini.

Tabel 2. Penjabaran Bobot Tugas

Bobot Tugas	Estimasi Pengerjaan
1 (Sangat Mudah)	1 Jam
2 (Mudah)	2 Jam
3 (Sedang)	4 Jam
4 (Sulit)	8 Jam
5 (Sangat Sulit)	16 Jam

Tabel 2 memberikan penjelasan tentang bobot setiap tugas yang ditandai dengan angka tertentu. Angka 1 menunjukkan tugas dengan bobot sangat mudah, 2 untuk bobot mudah, 3 untuk bobot sedang, 4 untuk bobot sulit, dan 5 untuk bobot sangat sulit. Selain itu, tabel 2 juga mencakup estimasi waktu yang dibutuhkan untuk menyelesaikan setiap tugas berdasarkan bobotnya, yaitu 1 jam untuk tugas dengan bobot sangat mudah, 2 jam untuk bobot mudah, 4 jam untuk bobot sedang, 8 jam untuk bobot sulit, dan 16 jam untuk bobot sangat sulit. Informasi tentang bobot tugas dan estimasi waktu pengerjaan ini akan digunakan sebagai dasar dalam menentukan prioritas penyelesaian tugas kuliah.

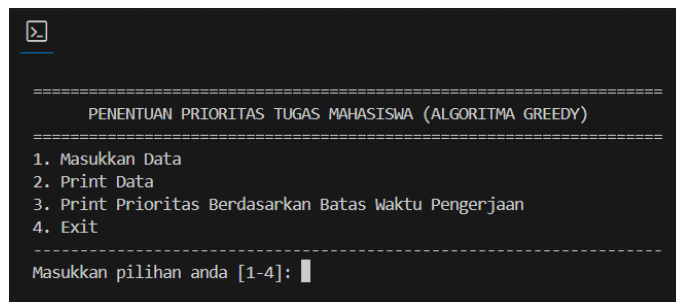
Dengan memperhatikan data penelitian, kita dapat memahami sejauh mana kompleksitas tugas yang dihadapi oleh mahasiswa. Data ini mencakup berbagai aspek, mulai dari waktu yang dibutuhkan untuk menyelesaikan tugas hingga tingkat kesulitan dari setiap tugas. Selanjutnya, estimasi pengerjaan berdasarkan bobot tugas memberikan gambaran lebih lanjut tentang bagaimana tugas-tugas tersebut harus ditangani. Dengan mempertimbangkan semua faktor ini, kita dapat menentukan prioritas pengerjaan tugas dari mahasiswa tersebut. Akhirnya, semua tugas ini dapat diselesaikan dengan

menggunakan algoritma greedy, yang dapat memastikan bahwa mahasiswa dapat menyelesaikan tugas kuliah dengan efisien dan efektif.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Implementasi Algoritma Greedy

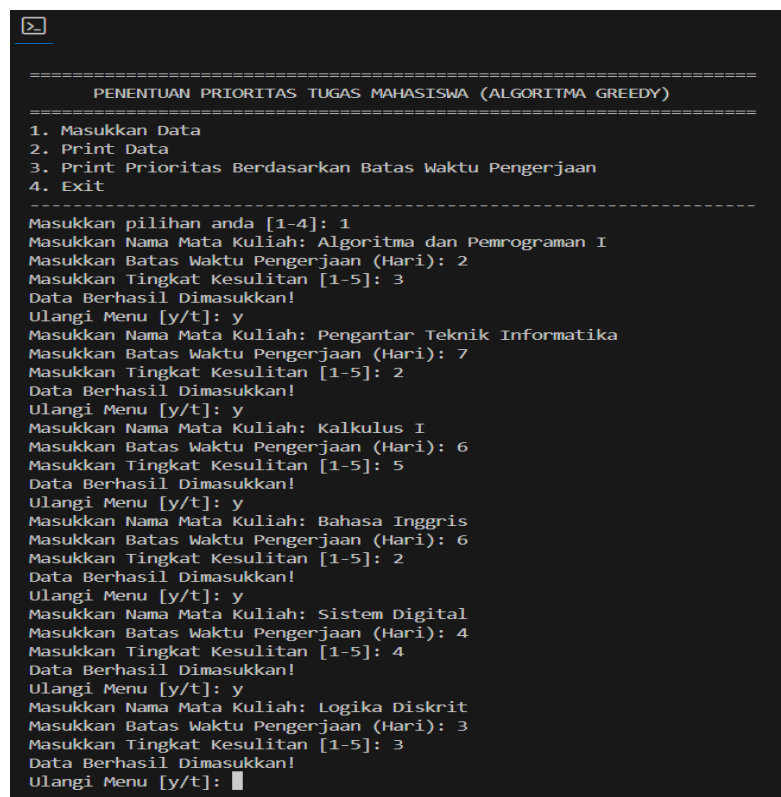
Berikut ini merupakan implementasi algoritma greedy pada sebuah program untuk memasukkan detail tugas yang diberikan berdasarkan data penelitian agar program memberikan solusi berupa prioritas yang harus dikerjakan oleh pengguna agar pengerjaannya dapat efektif dan efisien. Output program yang terdiri dari tampilan menu utama program, menu 1 untuk memasukkan data tugas, menu 2 untuk mencetak data tugas, menu 3 untuk mencetak hasil prioritas pengerjaan tugas menggunakan algoritma greedy, dan menu 4 untuk keluar program dapat dilihat pada gambar 2, gambar 3, gambar 4, dan gambar 5.



```
=====
PENENTUAN PRIORITAS TUGAS MAHASISWA (ALGORITMA GREEDY)
=====
1. Masukkan Data
2. Print Data
3. Print Prioritas Berdasarkan Batas Waktu Pengerjaan
4. Exit
=====
Masukkan pilihan anda [1-4]:
```

**Gambar 2.** Tampilan Menu Utama Program

Dapat dilihat pada gambar 2 yang merupakan tampilan jika program pertama kali dijalankan. Menu utama pada program tersebut ada 4, yaitu Masukkan Data, Print Data, Print Prioritas Berdasarkan Batas Waktu Pengerjaan, dan Exit. Pengguna diminta untuk memilih salah satu menu untuk dapat melanjutkan program.



```
=====
PENENTUAN PRIORITAS TUGAS MAHASISWA (ALGORITMA GREEDY)
=====
1. Masukkan Data
2. Print Data
3. Print Prioritas Berdasarkan Batas Waktu Pengerjaan
4. Exit
=====
Masukkan pilihan anda [1-4]: 1
Masukkan Nama Mata Kuliah: Algoritma dan Pemrograman I
Masukkan Batas Waktu Pengerjaan (Hari): 2
Masukkan Tingkat Kesulitan [1-5]: 3
Data Berhasil Dimasukkan!
Ulangi Menu [y/t]: y
Masukkan Nama Mata Kuliah: Pengantar Teknik Informatika
Masukkan Batas Waktu Pengerjaan (Hari): 7
Masukkan Tingkat Kesulitan [1-5]: 2
Data Berhasil Dimasukkan!
Ulangi Menu [y/t]: y
Masukkan Nama Mata Kuliah: Kalkulus I
Masukkan Batas Waktu Pengerjaan (Hari): 6
Masukkan Tingkat Kesulitan [1-5]: 5
Data Berhasil Dimasukkan!
Ulangi Menu [y/t]: y
Masukkan Nama Mata Kuliah: Bahasa Inggris
Masukkan Batas Waktu Pengerjaan (Hari): 6
Masukkan Tingkat Kesulitan [1-5]: 2
Data Berhasil Dimasukkan!
Ulangi Menu [y/t]: y
Masukkan Nama Mata Kuliah: Sistem Digital
Masukkan Batas Waktu Pengerjaan (Hari): 4
Masukkan Tingkat Kesulitan [1-5]: 4
Data Berhasil Dimasukkan!
Ulangi Menu [y/t]: y
Masukkan Nama Mata Kuliah: Logika Diskrit
Masukkan Batas Waktu Pengerjaan (Hari): 3
Masukkan Tingkat Kesulitan [1-5]: 3
Data Berhasil Dimasukkan!
Ulangi Menu [y/t]:
```

**Gambar 3.** Tampilan Menu Nomor 1

Dapat dilihat pada gambar 3 yang merupakan tampilan jika pengguna memilih menu nomor 1 yang digunakan untuk memasukkan data. Pengguna dapat memasukkan mata kuliah, batas waktu pengerjaan dalam format hari, dan tingkat kesulitan tugas dengan detail pada program. Jika data berhasil dimasukkan, pengguna akan diberikan pertanyaan apakah ingin mengulang menu atau tidak. Pada gambar 3, data yang dimasukkan pengguna disesuaikan dengan data penelitian yang telah dideklarasikan sebelumnya.

```
=====
PENENTUAN PRIORITAS TUGAS MAHASISWA (ALGORITMA GREEDY)
=====
1. Masukkan Data
2. Print Data
3. Print Prioritas Berdasarkan Batas Waktu Pengerjaan
4. Exit
-----
Masukkan pilihan anda [1-4]: 2
-----
1. Algoritma dan Pemrograman I (Batas Waktu: 2 Hari, Difficulty: 3 -> Durasi Pengerjaan: 4 Jam)
2. Pengantar Teknik Informatika (Batas Waktu: 7 Hari, Difficulty: 2 -> Durasi Pengerjaan: 2 Jam)
3. Kalkulus I (Batas Waktu: 6 Hari, Difficulty: 5 -> Durasi Pengerjaan: 16 Jam)
4. Bahasa Inggris (Batas Waktu: 6 Hari, Difficulty: 2 -> Durasi Pengerjaan: 2 Jam)
5. Sistem Digital (Batas Waktu: 4 Hari, Difficulty: 4 -> Durasi Pengerjaan: 8 Jam)
6. Logika Diskrit (Batas Waktu: 3 Hari, Difficulty: 3 -> Durasi Pengerjaan: 4 Jam)
-----
Tekan Enter Untuk Kembali Ke Menu Utama...|
```

**Gambar 4.** Tampilan Menu Nomor 2

Seperti yang terlihat pada gambar 4, ini adalah tampilan yang muncul ketika pengguna memilih menu 2 setelah sebelumnya memilih menu 1. Di menu ini, program akan menampilkan data yang telah diinput oleh pengguna sebelumnya dan menampilkannya di layar. Data yang ditampilkan mencakup nama mata kuliah, tenggat waktu, dan tingkat kesulitan tugas. Berdasarkan gambar 4, terdapat 6 data yang telah diinput, yaitu Algoritma dan Pemrograman I dengan tenggat waktu 2 hari dan tingkat kesulitan 3 (sedang) dengan durasi pengerjaan selama 4 jam, Pengantar Teknik Informatika dengan tenggat waktu 7 hari dan tingkat kesulitan 2 (mudah) dengan durasi pengerjaan selama 2 jam, Kalkulus I dengan tenggat waktu 6 hari dan tingkat kesulitan 5 (sangat sulit) dengan durasi pengerjaan selama 16 jam, Bahasa Inggris dengan tenggat waktu 6 hari dan tingkat kesulitan 2 (mudah) dengan durasi pengerjaan selama 2 jam, Sistem Digital dengan tenggat waktu 4 hari dan tingkat kesulitan 4 (sulit) dengan durasi pengerjaan selama 8 jam, dan terakhir Logika Diskrit dengan tenggat waktu 3 hari dan tingkat kesulitan 3 (sedang) dengan durasi pengerjaan selama 4 jam.

```
=====
PENENTUAN PRIORITAS TUGAS MAHASISWA (ALGORITMA GREEDY)
=====
1. Masukkan Data
2. Print Data
3. Print Prioritas Berdasarkan Batas Waktu Pengerjaan
4. Exit
-----
Masukkan pilihan anda [1-4]: 3
Masukkan Jangka Waktu Penyelesaian (Hari): 7
PRIORITAS DEADLINE:
-----
Urutan prioritas tugas yang dapat diselesaikan dalam 7 hari (1 hari atau 0 jam tersisa):
Algoritma dan Pemrograman I
Logika Diskrit
Sistem Digital
Bahasa Inggris
Kalkulus I
Pengantar Teknik Informatika
-----
Tekan Enter Untuk Kembali Ke Menu Utama|
```

**Gambar 5.** Tampilan Menu Nomor 3

Dapat dilihat pada gambar 5 yang merupakan tampilan jika pengguna memilih menu 3 yang digunakan untuk menampilkan prioritas pengerjaan tugas berdasarkan batas waktu pengerjaan yang telah dimasukkan pengguna sebelumnya. Sebelum menerima hasil prioritas pengerjaan tugas, pengguna harus memasukkan jangka waktu penyelesaian tugas dengan format hari ke dalam program. Pada penelitian ini, jangka waktu penyelesaian yang digunakan adalah 7 hari. Karena limitasi pengerjaan setiap hari adalah 6 jam, maka dalam 7 hari mahasiswa hanya memiliki waktu penyelesaian tugas selama 42 jam.

Berdasarkan output pada gambar 5, berikut detail tahapan pengerjaan tugas yang dilakukan mahasiswa berdasarkan prioritas batas waktu pengerjaan.

1. Pada hari pertama, mahasiswa terlebih dahulu mengerjakan tugas Algoritma dan Pemrograman I selama 4 jam, karena tingkat kesulitannya 3. Setelah itu, mahasiswa tersebut dapat mengerjakan tugas Logika Diskrit selama 2 jam pada hari pertama, yang berarti tugas ini memiliki sisa waktu 2 jam yang harus diselesaikan pada hari berikutnya
2. Pada hari kedua, mahasiswa terlebih dahulu melanjutkan pengerjaan tugas Logika Diskrit selama 2 jam. Setelah itu, mahasiswa tersebut dapat mengerjakan tugas Sistem Digital selama 4 jam pada hari kedua. Namun, tugas Sistem Digital harus dilanjutkan 4 jam pada hari ketiga

3. Pada hari ketiga, mahasiswa melanjutkan pengerjaan tugas Sistem Digital selama 4 jam. Setelah itu, Ia dapat mengerjakan tugas Bahasa Inggris selama 2 jam.
4. Pada hari keempat, mahasiswa terlebih dahulu mengerjakan tugas kalkulus I selama 6 jam, yang berarti pada hari kelima mahasiswa ini harus melanjutkan tugas ini selama 6 jam lagi.
5. Pada hari kelima, mahasiswa melanjutkan pengerjaan tugas Kalkulus I selama 6 jam, yang berarti pada hari keenam mahasiswa tersebut harus melanjutkan tugas ini selama 4 jam sebelum mengerjakan tugas selanjutnya.
6. Pada hari keenam, mahasiswa melanjutkan pengerjaan tugas Kalkulus I selama 4 jam. Setelah itu, mahasiswa ini dapat melanjutkan untuk mengerjakan tugas Pengantar Teknik Informatika selama 2 jam. Dengan demikian, seluruh tugasnya kuliah dari mahasiswa tersebut dapat diselesaikan semuanya dalam waktu 6 hari dan menyisakan waktu 1 hari atau 0 jam untuk hari ke-6.

Pada hasil prioritas pengerjaan tugas berdasarkan batas waktu seperti yang terlihat pada gambar 5 maupun uraian yang terdapat di atas, dapat terlihat bahwa tugas Bahasa Inggris dan Kalkulus I memiliki batas waktu pengerjaan yang sama, yaitu 6 hari. Namun, tugas Bahasa Inggris harus dikerjakan terlebih dahulu karena tugas ini memiliki tingkat kesulitan tugas yang lebih rendah daripada tugas Kalkulus I, yang berarti durasi waktu yang digunakan untuk mengerjakan tugas Bahasa Inggris lebih singkat dibandingkan dengan tugas Kalkulus I.

### 3.2 Pembahasan

Dalam dunia akademik, beragam jenis tugas diberikan kepada mahasiswa dengan batas waktu yang berbeda. Ada tugas yang harus diselesaikan dalam waktu singkat, sementara ada juga yang memiliki batas waktu yang lebih panjang. Banyak mahasiswa mengalami tantangan dalam mengatur waktu mereka secara efektif untuk dapat menuntaskan semua tugas mereka dengan baik dan tepat waktu. Sebuah cara untuk menangani masalah ini adalah dengan menggunakan algoritma greedy. Algoritma greedy merupakan teknik dalam desain algoritma yang digunakan untuk menyelesaikan masalah optimasi. Metode ini berkerja dengan memilih solusi terbaik pada setiap tahapannya tanpa mempertimbangkan keseluruhan solusi secara detail terlebih dahulu. Walaupun metode ini tidak selalu memberikan hasil terbaik, namun sering kali dapat memberikan solusi yang memadai dengan kecepatan dan efisiensi yang tinggi. Oleh karena itu, algoritma greedy dapat digunakan untuk mencari solusi yang efektif bagi mahasiswa dalam mengatur waktu mereka dalam menyelesaikan tugas kuliah. Dengan menerapkan algoritma greedy, mahasiswa dapat menyelesaikan tugas-tugas mereka dengan lebih efisien dan tepat waktu.

Dalam kerangka kerja algoritma greedy, ada beberapa elemen penting yang mempengaruhi urutan penyelesaian tugas. Pertama, kita memiliki himpunan kandidat, yang mana di dalam penelitian ini berisi daftar tugas kuliah yang harus dikerjakan oleh mahasiswa, seperti Tugas Algoritma dan Pemrograman I (F1), Tugas Pengantar Teknik Informatika (F2), Tugas Kalkulus I (F3), Tugas Bahasa Inggris (F4), Tugas Sistem Digital (F5), dan Tugas Logika Diskrit (F6). Selanjutnya, kita memiliki himpunan solusi, yang awalnya kosong dan akan diisi oleh algoritma dengan urutan tugas yang dipilih untuk dikerjakan. Fungsi kelayakan mengevaluasi apakah suatu tugas layak untuk dipilih atau tidak, yang dalam kasus ini bisa berupa pengecekan apakah tenggat waktu tugas masih memungkinkan tugas tersebut untuk diselesaikan dalam waktu 7 hari dan dalam sehari mahasiswa hanya memiliki waktu 6 jam untuk mengerjakan tugas. Kemudian, fungsi seleksi menentukan tugas mana yang harus dipilih selanjutnya. Dalam konteks ini, fungsi seleksi bisa berupa pemilihan tugas dengan tenggat waktu terdekat atau bobot tugas tertinggi. Terakhir, fungsi objektif adalah tujuan yang ingin dicapai oleh algoritma, yang bisa berupa penyelesaian semua tugas dalam waktu yang paling efisien atau mencapai bobot total tugas yang paling tinggi. Semua elemen-elemen tersebut akan bekerja bersama dalam algoritma greedy untuk membantu mahasiswa menentukan urutan prioritas tugas kuliah mereka dengan tepat dan efektif. Dengan pendekatan ini, mahasiswa dapat merasakan manfaat langsung dari penelitian ini dalam kehidupan akademik mereka, mengurangi stres dan meningkatkan produktivitas mereka.

Penelitian ini akan mengulas lebih lanjut tentang bagaimana pengimplementasian dalam program Python yang dirancang untuk membantu mahasiswa dalam mengatur prioritas pengerjaan tugas kuliah mereka. Program ini terdiri dari beberapa fungsi utama. Pertama, fungsi `__init__`, yang bertugas untuk menginisialisasi objek `CourseAssignment` dengan atribut-atribut yang relevan seperti nama tugas, tenggat waktu, dan tingkat kesulitan. Inisialisasi ini sangat penting karena memungkinkan program untuk memanipulasi dan membandingkan tugas-tugas yang ada. Kedua, fungsi `get_duration`, yang berfungsi untuk menghitung estimasi waktu yang diperlukan untuk menyelesaikan suatu tugas berdasarkan tingkat kesulitan yang diberikan. Estimasi waktu ini menjadi kunci dalam menentukan prioritas pengerjaan tugas-tugas dengan tingkat kesulitan yang berbeda. Ketiga, fungsi `__lt__`, yang memungkinkan perbandingan antara dua objek `CourseAssignment` berdasarkan deadline dan tingkat kesulitan. Hal ini penting dalam proses pengurutan tugas-tugas berdasarkan kriteria tertentu seperti deadline atau tingkat kesulitan. Keempat, fungsi `add_assignments`, yang memiliki tugas untuk meminta input dari pengguna untuk menambahkan tugas baru ke dalam daftar tugas. Setelah tugas ditambahkan, program akan menghitung estimasi waktu pengerjaan dan menyimpan tugas tersebut dalam daftar. Kelima, fungsi `print_assignments`, yang bertanggung jawab untuk mencetak daftar tugas beserta detailnya seperti nama, tenggat waktu, dan tingkat kesulitan. Keenam, fungsi `print_prioritized_assignments`, yang memiliki tugas untuk mencetak daftar tugas yang diprioritaskan berdasarkan kriteria tertentu seperti deadline atau tingkat kesulitan, dengan mempertimbangkan waktu yang tersedia. Sebelum mencetak, daftar tugas akan diurutkan terlebih dahulu sesuai dengan kriteria yang dipilih atau disediakan. Terakhir, fungsi utama dari program ini adalah fungsi `main`, yang memiliki tugas untuk menjalankan logika utama yang meliputi interaksi dengan pengguna, pemrosesan input, dan pemanggilan fungsi-fungsi lainnya sesuai dengan pilihan yang dibuat pengguna.

Program ini menawarkan empat fitur utama yang dirancang untuk membantu pengguna dalam mengelola tugas-tugas mereka dengan efisien. Pertama, fitur "Masukkan Data" memungkinkan pengguna untuk dengan mudah memasukkan tugas baru ke dalam sistem. Pengguna diminta untuk memasukkan informasi penting tentang tugas tersebut, termasuk nama tugas, deadline, dan tingkat kesulitan. Berdasarkan tingkat kesulitan yang dipilih oleh pengguna, program secara otomatis menentukan durasi yang diperlukan untuk menyelesaikan tugas tersebut. Dengan adanya fitur ini, pengguna dapat dengan cepat dan sistematis mencatat setiap tugas yang perlu mereka selesaikan. Kedua, fitur "Print Data", fitur ini memberikan pengguna kemampuan untuk melihat semua tugas yang telah mereka masukkan ke dalam sistem. Informasi detail tentang setiap tugas, seperti nama, deadline, dan tingkat kesulitan, ditampilkan sehingga pengguna dapat dengan cepat melihat dan memperbarui daftar tugas mereka. Fitur ini membantu pengguna untuk tetap terorganisir dan memiliki pemahaman yang jelas tentang tugas-tugas yang harus mereka selesaikan. Selanjutnya, fitur "Print Prioritas Berdasarkan Waktu Pengerjaan", fitur ini berfungsi untuk mencetak urutan prioritas pengerjaan tugas yang sudah diurutkan menggunakan algoritma greedy, di dalam fitur ini pengguna akan diminta memasukkan jangka waktu pengerjaan tugas terlebih dahulu. Barulah urutan pengerjaan tugas mahasiswa dicetak. Dengan fitur ini, pengguna dapat menentukan strategi yang lebih efisien tentang bagaimana mereka harus memanfaatkan waktu mereka untuk menyelesaikan tugas kuliah mereka dengan urutan yang telah ditetapkan. Terakhir, fitur "Exit", fitur ini memungkinkan pengguna untuk mengakhiri sesi penggunaan program dengan aman. Jika pengguna telah selesai menggunakan fitur-fitur yang tersedia atau ingin keluar dari program, mereka dapat memilih opsi ini untuk menutup program dengan baik. Hal ini memberikan pengguna kemudahan dalam mengelola waktu mereka saat menggunakan program, karena mereka dapat menyelesaikan tugas-tugas mereka dan keluar dari program tanpa masalah. Alur kerja program dimulai dengan menjalankan fungsi main, yang merupakan titik awal dari interaksi antara pengguna dan program. Pengguna kemudian diberikan pilihan untuk menggunakan salah satu fitur yang disediakan oleh program, seperti menambahkan tugas baru, melihat daftar tugas, memprioritaskan tugas, atau keluar dari program. Program akan merespons sesuai dengan pilihan yang dibuat oleh pengguna.

Dalam kode program yang telah dibuat ini, algoritma greedy diterapkan dalam fungsi `print_prioritized_assignments` (`assignments`, `sort_by`, `completion_time`). Fungsi ini menerima daftar tugas (`assignments`), kriteria pengurutan (`sort_by`), dan waktu penyelesaian yang tersedia (`completion_time`). Pertama-tama, fungsi ini menyortir daftar tugas berdasarkan kriteria yang diberikan, entah itu berdasarkan deadline atau tingkat kesulitan. Setelah itu, fungsi ini menghitung waktu yang tersisa untuk menyelesaikan tugas-tugas dengan memperhitungkan waktu yang diperlukan untuk setiap tugas (`completion_time * 6`). Kemudian, fungsi ini memilih tugas-tugas yang dapat diselesaikan dalam waktu yang tersedia berdasarkan deadline. Hal ini dilakukan dengan iterasi melalui daftar tugas yang sudah diurutkan, dan memilih tugas-tugas yang deadline-nya masih berada dalam batas waktu yang tersisa. Tugas-tugas yang dipilih kemudian dimasukkan ke dalam daftar `selected_assignments`, dan waktu yang tersisa dikurangi dengan durasi setiap tugas yang dipilih. Setelah itu, fungsi ini mencetak daftar tugas-tugas yang telah dipilih, beserta estimasi waktu yang tersisa. Jika tidak ada tugas yang dapat diselesaikan dalam batas waktu yang diberikan, program akan mencetak pesan yang menyatakan bahwa tidak ada tugas yang dapat diselesaikan dalam waktu yang tersedia.

Secara keseluruhan, program ini merupakan contoh nyata penerapan algoritma greedy dalam mengoptimalkan prioritas tugas mahasiswa. Dengan menyediakan fitur-fitur yang bermanfaat dan menggunakan algoritma yang cerdas, program ini dapat memberikan dukungan yang berharga bagi mahasiswa dalam mengelola tugas-tugas mereka dengan lebih efisien dan efektif. Namun, penting untuk diingat bahwa keberhasilan program ini tergantung pada bagaimana mahasiswa menggunakannya. Oleh karena itu, program ini sebaiknya digunakan sebagai alat bantu yang mendukung upaya manajemen tugas, sekaligus tetap mengembangkan keterampilan manajemen waktu yang diperlukan dalam lingkungan akademis dan profesional. Dengan memanfaatkan sumber daya dan alat yang tersedia dengan bijaksana, mahasiswa dapat meningkatkan produktivitas mereka dan meraih kesuksesan dalam pendidikan dan karir mereka.

#### 4. KESIMPULAN

Berdasarkan dari temuan yang didapatkan dari penelitian mengenai urutan prioritas pengerjaan tugas kuliah mahasiswa menggunakan algoritma greedy. Dengan upaya agar dapat membantu mahasiswa untuk menentukan prioritas dari pengerjaan tugas-tugas kuliah. Bisa terlihat dari data pemisalan yang ada menunjukkan tugas-tugas yang diberikan kepada mahasiswa ini bersama dengan tenggat waktu pengumpulan dan juga nilai bobot kesulitan dari tugas-tugas tersebut. Dengan menggunakan algoritma greedy, aspek paling pertama yang harus mahasiswa tersebut perhatikan adalah deadline atau tenggat waktu pengumpulan dari tugas, karena algoritma greedy akan lebih mengutamakan hal yang lebih menguntungkan dalam kondisi saat ini, tentu saja semakain dekat deadline atau tenggat waktu pengerjaan tugas maka semakin cepat mahasiswa tersebut mengerjakan tugas yang deadline-nya singkat, maka semakin menguntungkan bagi mahasiswa tersebut, dari aspek pertama itu tugas yang harus dikerjakan paling pertama adalah tugas Algoritma dan Pemrograman I dan yang paling terakhir adalah tugas Pengantar Teknik Informatika. Lalu, aspek paling terakhir yang harus diperhatikan adalah nilai bobot kesulitan tugas. Aspek ini hanya akan digunakan apabila ada tugas-tugas yang memiliki deadline atau tenggat waktu pengerjaan tugas yang sama, seperti contohnya tugas Bahasa Inggris lebih diprioritaskan lebih dulu daripada tugas Kalkulus I karena tugas Bahasa Inggris ini memiliki nilai bobot kesulitan yang lebih rendah dari tugas Kalkulus I. Dari penelitian yang telah dilakukan, algoritma greedy ini cukup efisien jika ingin

digunakan dalam waktu yang singkat, akan tetapi hasil akhir atau solusi yang di hasilkan masih kurang optimal karena hanya memperhatikan hal yang paling menguntungkan dalam kondisi saat ini.

## REFERENCES

- [1] A. R. Riswaya, J. Sudrajat, and D. A. Budiman, "Perancangan Penjadwalan Pinjaman Fixed Asset Desa dengan Penerapan Algoritma Greedy," *INTERNAL (Information System Journal)*, vol. 6, no. 1, pp. 79-87, Jun. 2023, doi: 10.32627.
- [2] Y. C. Oktaviani and Y. F. Riti, "Perbandingan Algoritma Welch Powell dan Algoritma Greedy dalam Optimasi Penjadwalan Ruang Kuliah Semester Genap Fakultas Teknik," *JIMP: Jurnal Informatika Merdeka Pasuruan*, vol. 7, no. 3, pp. 87-95, Dec. 2022.
- [3] H. Sunandar and Pristiwanto, "Optimalisasi Implementasi Algoritma Greedy dalam Fungsi Penukaran Mata Uang Rupiah," *Jurnal Teknik Informatika Unika St. Thomas (JTIUST)*, vol. 4, no. 2, pp. 193-201, Dec. 2019.
- [4] Y. Darnita and R. Toyib, "Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Instansi-Instansi Penting Di Kota Argamakmur Kabupaten Bengkulu Utara," *Jurnal Media Infotama*, vol. 15, no. 2, pp. 57-64, Sep. 2019.
- [5] W. Satria, M. Sofitra, and N. H. Djanggu, "Solusi Heuristik untuk Permasalahan Rute Inventori Berkala pada Produk Segar," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 8, no. 1, pp. 111-120, Apr. 2022.
- [6] U. Maftukhah, S. Amiroch, and M. S. Pradana, "Implementasi Algoritma Greedy Pada Pewarnaan Wilayah Kecamatan Sukodadi Lamongan," *UJMC*, vol. 6, no. 2, pp. 29-38, 2020.
- [7] E. Darnila, M. Ula, and C. D. A. Soraya, "Optimasi Kelayakan Kondisi Pembangunan Jalan di Kota Lhokseumawe Menggunakan Algoritma Greedy," *Jurnal Komputer dan Informatika (JUKI)*, vol. 1, no. 1, pp. 9-14, May. 2019.
- [8] E. S. Laela, W. Gata, and J. J. Purnama, "OPTIMALISASI ALGORITMA GREEDY DALAM PENYUSUNAN JADWAL PELAJARAN PADA SMK NURUL ISLAM CIANJUR," *Syntax Literate: Jurnal Ilmiah Indonesia*, vol. 7, no. 12, pp. 18451-18460, Dec. 2022.
- [9] I. Purnamasari and O. Komarudin, "OPTIMALISASI MANAJEMEN PENERIMAAN & PENYIMPANAN DOKUMEN FAX MENGGUNAKAN ALGORITMA GREEDY," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 6, no. 6, pp. 625-636, Dec. 2019, doi: 10.25126/jtiik.20196976.
- [10] R. F. Ginting, A. Situmorang, and S. Manurung, "Sistem Informasi Gografis Pencarian Lokasi Sekolah Terdekat Pada Kota Medan Dengan Algoritma Greedy Berbasis Web," *Methodika : Jurnal Ilmiah Teknik Informatika*, vol. 2, no. 2, pp. 85-91, Oct. 2022.
- [11] S. Rianti, H. Lubis, and R. Pahlevi, "SISTEM PENUNJANG KEPUTUSAN OPTIMALISASI BARANG DENGAN ALGORITMA GREEDY PADA PT SENTRALINDO TEGUH GEMILANG," *Jurnal Sistem Informasi (JSI)*, vol. 7, no. 2, pp. 123-132, 2020.
- [12] N. N. Sania and I. Sari, "IMPLEMENTASI RENCANA PERJALANAN WISATA DI KOTA BOGOR MENGGUNAKAN ALGORITMA GREEDY BERBASIS WEBSITE," *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 24, no. 2, pp. 114-130, Aug. 2019, doi: <http://dx.doi.org/10.35760/tr.2019.v24i2.2390>.
- [13] A. C. Wibowo and A. D. Wowor, "Perancangan Distribusi Hasil Produk Textil dengan Rute Terdekat dengan Algoritma Greedy," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 7, no. 1, pp. 287-298, Mar. 2023.
- [14] F. Ikhwan and H. Kurniawan, "Pemanfaatan Algoritma Greedy Pada Aplikasi Pemesanan Jasa Grooming Kucing Berbasis Android," *Jurnal ENTER*, vol. 2, pp. 103-113, Aug. 2019.
- [15] M. Irfansyahfalalah and R. Djutalov, "Sistem Penentuan Jarak Terdekat Menuju Fasilitas Kesehatan Dengan Metode Algoritma Greedy," *Biner : Jurnal Ilmu Komputer, Teknik dan Multimedia*, vol. 1, no. 3, pp. 692-701, Aug. 2023.
- [16] L. V. Hignasari, "KOMPARASI ALGORITMA CHEAPEST INSERTION HEURISTIC (CIH) DAN GREEDY DALAM OPTIMASI RUTE PENDISTRIBUSIAN BARANG," *VASTUWIDYA*, vol. 2, no. 2, pp. 31-39, Aug. 2019-Jan. 2020.
- [17] H. P. Kekal, W. Gata, S. Nurdian, A. J. S. Rini, and D. S. Wita, "Analisa Pencarian Rute Tercepat Menuju Tempat Wisata Pulau Kumala Kota Tenggara Menggunakan Algoritma Greedy," *Jurnal Ilmiah Ilmu Komputer*, vol. 7, no. 1, pp. 9-15, Apr. 2021.
- [18] L. M. Pramesti, Prajoko, and Asriyanik, "Penerapan Metode Algoritma Greedy Untuk Menentukan Rute Terdekat Pada Objek Wisata Palabuhanratu," *Jurnal Teknologi Informatika dan Komputer MH. Thamrin*, vol. 7, no. 2, pp. 70-81, Sep. 2021.
- [19] A. P. Harianja, "PERANCANGAN PERANGKAT LUNAK VISUALISASI ALGORITMA GREEDY," *Jurnal Sistem Informasi Kaputama (JSIK)*, vol. 3, no. 1, pp. 27-39, Jan. 2019.
- [20] S. F. Saputra, W. Gata, D. N. Sulistyowati, F. Rozi, and A. R. Kadafi, "IMPLEMENTASI PENGIRIMAN BARANG DENGAN ALGORITMA GREEDY PT. ASTRA INTERNATIONAL, HONDA KALIMANTAN TIMUR," *Jurnal Informatika Sains dan Teknologi (INSTEK)*, vol. 6, no. 2, pp. 160-169, Oct. 2021.