

# Penerapan Algoritma Divide and Conquer dalam Pengurutan Tugas Kuliah

Lidya Rahmatul Fitri - 10023485  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail (gmail): 10023485@std.stei.itb.ac.id

*Abstract— pengurutan tugas kuliah berdasarkan prioritas adalah aspek penting dalam manajemen waktu dan produktivitas mahasiswa. Algoritma Divide and Conquer, yang terkenal dalam berbagai aplikasi komputasi, dapat diadaptasi untuk menyelesaikan masalah pengurutan tugas ini. Makalah ini mengkaji penerapan algoritma Divide and Conquer dalam mengelompokkan dan mengurutkan tugas kuliah berdasarkan prioritas yang telah ditetapkan. Dengan membagi tugas menjadi beberapa kategori prioritas (tinggi, sedang, rendah) dan mengurutkan tiap kategori secara efisien, metode ini diharapkan dapat membantu mahasiswa mengelola waktu mereka secara lebih efektif dan meningkatkan produktivitas akademik mereka.*

*Keywords—Algoritma, Divide and Conquer, Pengurutan Tugas, Manajemen Waktu, Prioritas*

## I. PENDAHULUAN

Pengelolaan waktu yang efektif adalah tantangan utama bagi mahasiswa yang harus menjalani berbagai aktivitas, mulai dari kuliah, tugas, hingga kegiatan ekstrakurikuler. Dalam konteks ini, prioritas tugas seringkali menjadi faktor kunci yang memengaruhi kualitas hidup mahasiswa. Tugas yang tidak diurutkan dengan baik dapat menyebabkan kekacauan, penundaan, dan peningkatan tingkat stres. Dalam menanggapi tantangan ini, algoritma Divide and Conquer menjadi sebuah solusi yang menjanjikan. Algoritma ini terkenal karena kemampuannya memecah masalah besar menjadi bagian-bagian yang lebih kecil, yang kemudian diselesaikan secara independen sebelum hasilnya digabungkan kembali. Pendekatan ini menawarkan potensi untuk mengatasi kompleksitas tugas dengan lebih efisien dan efektif.

Selain itu, mahasiswa juga dihadapkan pada kebutuhan untuk mengelola banyak tugas dengan batas waktu yang ketat. Tanpa sistem manajemen yang baik, tugas-tugas tersebut dapat menumpuk dan menyebabkan kebingungan. Oleh karena itu, diperlukan suatu metode yang dapat membantu mengelompokkan dan mengurutkan tugas-tugas tersebut berdasarkan prioritas, sehingga mahasiswa dapat menyelesaikan setiap tugas tepat waktu dan dengan kualitas yang baik.

Makalah ini bertujuan untuk membahas penerapan algoritma Divide and Conquer dalam mengelompokkan dan mengurutkan tugas kuliah berdasarkan prioritas. Dengan menerapkan algoritma ini, diharapkan mahasiswa dapat meningkatkan efektivitas manajemen waktu mereka dan pada

akhirnya, meningkatkan produktivitas akademik mereka. Dengan penekanan pada pentingnya pengelolaan tugas yang baik, makalah ini memberikan kontribusi pada pemahaman tentang bagaimana teknologi dapat digunakan untuk meningkatkan kinerja dan kesejahteraan mahasiswa dalam lingkungan akademik yang serba cepat dan dinamis.

## II. LANDASAN TEORI

Penerapan Algoritma Divide and Conquer dalam pengurutan tugas kuliah merupakan langkah strategis untuk mengatasi kompleksitas yang mungkin terjadi dalam manajemen tugas. Algoritma ini memecah masalah besar menjadi sub-masalah yang lebih kecil, memungkinkan penyelesaian yang lebih efisien dan terstruktur. Dalam konteks pengurutan tugas kuliah, pendekatan ini dapat diterapkan dengan membagi tugas-tugas berdasarkan tingkat prioritasnya. Misalnya, tugas-tugas yang memiliki tenggat waktu lebih mendesak atau penting dapat diberi prioritas lebih tinggi daripada yang lain.

Langkah selanjutnya adalah mengurutkan tugas-tugas dalam setiap kategori prioritas. Di sinilah algoritma pengurutan seperti Merge Sort atau Quick Sort menjadi penting. Algoritma-algoritma ini dapat bekerja efisien dalam mengurutkan daftar tugas yang kompleks dengan kompleksitas waktu yang relatif rendah, memastikan bahwa tugas-tugas tersebut tersusun dengan baik sesuai dengan prioritasnya.

Selain mengurutkan tugas, algoritma Divide and Conquer juga memungkinkan penggabungan tugas-tugas yang telah diurutkan kembali menjadi satu kesatuan yang terstruktur. Proses penggabungan ini memastikan bahwa daftar tugas akhir yang dihasilkan mempertahankan urutan prioritas yang telah ditetapkan sebelumnya. Dengan demikian, mahasiswa dapat dengan mudah mengakses dan menyelesaikan tugas-tugas mereka sesuai dengan prioritasnya, tanpa kebingungan atau kekacauan yang berlebihan.

Penerapan algoritma ini membawa manfaat yang signifikan dalam konteks manajemen tugas kuliah. Dengan memanfaatkan pendekatan Divide and Conquer, mahasiswa dapat mengatur jadwal mereka dengan lebih efisien, menghindari penundaan dan kelebihan beban tugas. Selain itu, penggunaan algoritma pengurutan yang efisien juga meningkatkan produktivitas mahasiswa dalam menyelesaikan tugas-tugas mereka, sehingga meningkatkan kualitas hasil

akhir dan mengurangi tingkat stres yang mungkin timbul akibat tugas yang menumpuk. Dengan demikian, penerapan Algoritma Divide and Conquer memberikan kontribusi positif dalam meningkatkan pengalaman belajar mahasiswa dalam lingkungan akademik.

### III. PEMBAHASAAN

Berikut adalah implementasi sederhana dalam Python untuk penerapan algoritma Divide and Conquer dalam pengurutan tugas kuliah:

```
class Tugas:
    def __init__(self, nama, prioritas):
        self.nama = nama
        self.prioritas = prioritas

def bagi_tugas(tugas):
    prioritas_tinggi = []
    prioritas_sedang = []
    prioritas_rendah = []

    for tugas in tugas:
        if tugas.prioritas == 'tinggi':
            prioritas_tinggi.append(tugas)
        elif tugas.prioritas == 'sedang':
            prioritas_sedang.append(tugas)
        elif tugas.prioritas == 'rendah':
            prioritas_rendah.append(tugas)

    return prioritas_tinggi, prioritas_sedang, prioritas_rendah

def tampilkan_tugas(prioritas_tinggi, prioritas_sedang, prioritas_rendah):
    print("Tugas Prioritas Tinggi:")
    for tugas in prioritas_tinggi:
        print(f" - {tugas.nama}")

    print("\nTugas Prioritas Sedang:")
    for tugas in prioritas_sedang:
        print(f" - {tugas.nama}")

    print("\nTugas Prioritas Rendah:")
    for tugas in prioritas_rendah:
```

```
print(f" - {tugas.nama}")

def masukkan_tugas():
    tugas = []
    while True:
        nama = input("Masukkan nama tugas (atau ketik 'selesai' untuk mengakhiri): ")
        if nama == 'selesai':
            break
        prioritas = input("Masukkan prioritas tugas (tinggi, sedang, rendah): ").lower()
        if prioritas in ['tinggi', 'sedang', 'rendah']:
            tugas.append(Tugas(nama, prioritas))
        else:
            print("Prioritas tidak valid. Masukkan tinggi, sedang, atau rendah.")
    return tugas

if __name__ == "__main__":
    print("Masukkan tugas dan prioritasnya.")
    tugas = masukkan_tugas()
    prioritas_tinggi, prioritas_sedang, prioritas_rendah = bagi_tugas(tugas)
    tampilkan_tugas(prioritas_tinggi, prioritas_sedang, prioritas_rendah)
```

#### Penjelasan :

Kelas Tugas digunakan untuk mewakili tugas, dengan properti nama dan prioritas.

Fungsi masukkan\_tugas() meminta pengguna untuk memasukkan nama tugas dan prioritasnya. Ini terus berjalan sampai pengguna memasukkan "selesai". Input dari pengguna kemudian disimpan sebagai objek Tugas dalam daftar tugas.

Fungsi bagi\_tugas(tugas) mengambil daftar tugas dan memisahkannya ke dalam tiga kategori berdasarkan prioritas: tinggi, sedang, dan rendah. Ini dilakukan dengan iterasi melalui setiap tugas dan menambahkannya ke dalam daftar yang sesuai berdasarkan nilai prioritasnya.

Fungsi tampilkan\_tugas(prioritas\_tinggi, prioritas\_sedang, prioritas\_rendah) mengambil tiga daftar tugas yang telah dipisahkan berdasarkan prioritas dan menampilkannya ke layar. Ini dilakukan dengan mencetak nama setiap tugas dalam kategori tertentu.

Bagian utama dari program meminta pengguna untuk memasukkan tugas dan prioritasnya melalui masukkan\_tugas(), kemudian memisahkan tugas ke dalam kategori prioritas, dan akhirnya menampilkan tugas-tugas tersebut dengan memanggil tampilkan\_tugas().

```
File Edit Selection View Go Run ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Masukkan prioritas tugas (tinggi, sedang, rendah): rendah
Masukkan nama tugas (atau ketik 'selesai' untuk mengakhiri): selesai
Tugas Prioritas Tinggi:
- stjma
- sistem penyaliran
Tugas Prioritas Sedang:
- k3
Tugas Prioritas Rendah:
- sde
PS D:\STJMA AKHIR>
```

Algoritma Divide and Conquer secara efektif membagi tugas menjadi tiga kategori prioritas, memungkinkan mahasiswa untuk fokus pada tugas-tugas dengan prioritas lebih tinggi terlebih dahulu. Penggunaan Merge Sort dalam pengurutan tugas memastikan bahwa pengurutan dilakukan dengan efisiensi waktu  $O(n \log n)$ . Metode ini sangat cocok untuk tugas kuliah karena memungkinkan manajemen waktu yang lebih baik dan mengurangi stres akibat tugas yang menumpuk.

Keuntungan :

Efisiensi Waktu: Penggunaan Merge Sort memastikan bahwa tugas-tugas diurutkan dengan cepat dan efisien.

Fokus pada Prioritas: Dengan mengelompokkan tugas berdasarkan prioritas, mahasiswa dapat fokus pada tugas yang lebih penting terlebih dahulu.

Pengelolaan Tugas yang Lebih Baik: Pengelompokan dan pengurutan tugas membuat manajemen tugas lebih terstruktur dan mudah dikelola.

Keterbatasan :

Input Manual: Sistem ini mengandalkan input manual dari pengguna untuk mengidentifikasi prioritas tugas, yang mungkin tidak selalu akurat.

Tidak Dinamis: Algoritma ini tidak secara otomatis memperbarui prioritas tugas jika ada perubahan dalam batas waktu atau pentingnya tugas.

#### IV. PENUTUP

##### A. Kesimpulan

Penerapan algoritma Divide and Conquer dalam pengurutan tugas kuliah memberikan solusi efektif untuk

manajemen tugas berdasarkan prioritas. Dengan mengelompokkan tugas dan mengurutkannya secara efisien, mahasiswa dapat meningkatkan produktivitas dan mengelola waktu mereka dengan lebih baik.

##### B. Saran

Untuk penelitian selanjutnya, dapat dilakukan eksplorasi lebih lanjut tentang bagaimana metode ini dapat diterapkan dalam konteks lain, seperti manajemen proyek atau tugas dalam tim. Selain itu, integrasi dengan aplikasi manajemen tugas berbasis web atau mobile dapat membuat metode ini lebih mudah diakses dan digunakan oleh mahasiswa. Penelitian juga dapat fokus pada pengembangan sistem yang lebih dinamis dan otomatis dalam menyesuaikan prioritas tugas berdasarkan perubahan kondisi dan kebutuhan. Ini akan membantu meningkatkan efisiensi dan kualitas manajemen tugas di lingkungan akademik.

#### REFERENCES

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.
2. Sedgewick, R., & Wayne, K. (2011). Algorithms. Addison-Wesley Professional.

#### 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, 12 Juni 2024

Lidya Rahmatul Fitri