

PENERAPAN DECREASE AND CONQUER UNTUK MEMANAJEMEN TUGAS KULIAH INFORMATIKA

Ismail Sholeh¹

¹STMIK Muhammadiyah Paguyangan Brebes
Email: fatikhulqoli33@gmail.com.

Abstrak

Dalam pengerjaan tugas besar informatika perlu adanya manajemen yang baik dan tepat, sehingga pada saat pengerjaannya akan lebih terstruktur dan lebih efisien waktu yang digunakan. Dalam hal ini kita bisa memanfaatkan system *Decrease and Conquer*. *Decrease and Conquer* adalah sebuah metode pemecahan masalah yang mampu meminimalkan jumlah Langkah eksekusi yang di butuhkan untuk menyelesaikan masalah. Tugas besar informatika dapat di kerjakan secara efektif jika pengerjaan sesuai dengan prioritas.

Kata kunci: *Decrease and Conquer, Priority Queue, Binary Search, Tree*

1. PENDAHULUAN

Di dunia perkuliahan tidak terlepas dari yang namanya tugas, termasuk di Teknik Informatika STMIK Muhammadiyah Paguyangan Brebes. Pada perkuliahan di prodi Teknik Informatika, untuk mengasah kreativitas dan pemahaman mahasiswa dalam menerapkan materi kuliah yang diberikan, setiap mata kuliah masing-masing memberikan tugas besar.

Pada semester 4, mata kuliah yang diajarkan adalah Strategi Algoritma, Grafika Komputer, Metode numerik, Analisa dan Perancangan Sistem Informasi, Komunikasi Data dan Jaringan Studi Islam, Technopreneurship dan Sistem Operasi. Setiap mata kuliah tersebut memberikan tugas besar kepada mahasiswa dengan frekuensi masing - masing.

Setiap pada akhir semester, biasanya hampir semua tugas besar memiliki batas waktu pengumpulan yang berbeda dan biasanya saling berdekatan. Yang akhirnya hal ini membuat banyak mahasiswa harus berusaha keras menyelesaikan tugas sebelum pengumpulan. Di samping itu, sebenarnya waktu yang diberikan untuk mengerjakan tugas besar sebenarnya cukup lama.

Kebanyakan mahasiswa yang bersifat pemalas, yang selalu menganggap semua masalah adalah hal sepele dan cenderung untuk menunda-nunda bahkan lari dari masalah akademik dan kemahasiswaan tersebut. Kata lain dari gangguan psikologis ini sering disebut juga dengan procrastination. Procrastination, yaitu sebuah kebiasaan berlebih untuk menunda suatu pekerjaan hingga kesempatan untuk menyelesaikannya itu habis. Orang dengan kebiasaan berlebih seperti ini biasa disebut procrastinator atau kadang deadliner..

Kali ini akan membahas solusi terhadap kesulitan mahasiswa memberikan prioritas pengerjaan tugas kuliah dengan system decrease and conquer.

2. METODE PENELITIAN

2.1. Tugas Besar

Tugas besar (Tubes) Istilah ini merujuk pada tugas yang harus diselesaikan mahasiswa diakhir bab / semester untuk menguji pemahaman mahasiswa, sekaligus sebagai sarana pelatihan mahasiswa untuk menerapkan materi yang diajarkan dalam bentuk yang lebih nyata.

2.1.1 Atribut tugas besar

Setiap tugas besar memiliki 3 atribut yaitu:

a. Nama mata kuliah

Nama mata kuliah adalah nama mata kuliah yang memberikan tugas besar kepada mahasiswa. Nama mata kuliah yang diadakan pada semester 4 adalah Strategi Algoritma, Grafika Komputer, Komunikasi Data dan Jaringan dan Analisa dan Perancangan Sistem Informasi

b. Nama tugas besar

Nama tugas besar adalah identifikator tugas besar mata kuliah yang bersangkutan. Berikut ini nama tugas besar setiap mata kuliah yang diajarkan pada semester 4 :

- 1) Strategi Algoritma : Algoritma brute force, Algoritma Greedy, Algoritma Decrease And Conquer, Algoritma Devide and Conquer.
- 2) Grafika Komputer : Algoritma Digital Differential Analyzer (DDA), Algoritma Bresenham, Pengenalan OpenGL, System Pemrograman Grafis.
- 3) Analisa dan Perancangan System : Use Case, Kelas Diagram, Revisi, Desain Arsitektur System.
- 4) Komunikasi Data dan Jaringan : Setting mikrotik di winbox .

c. Deadline Pengumpulan

Deadline pengumpulan adalah tenggat waktu maksimal yang di berikan dari masing dosen pengempu untuk Tubes yang di berikan kepada mahasiswa.

2.1.2 *Tips Pengerjaan Tugas Besar*

Adapun tips atau cara mengerjakan tugas yang baik adalah :

- a. Niat
- b. Mengurutkan tugas dari yang lama sampe yang terbaru atau mengelist tugas
- c. Kerjakan Tubes secara urut mulai dari yang mendekati deadline
- d. Mencari referensi pengerjaan tugas
- e. Mencoret tugas yang sudah di kerjakan di list
- f. Bersemangat dalam mengerjakan tugas baru

2.2 *Decrease and Conquer*

2.2.1 *Pengertian*

Decrease and conquer adalah metode algoritma dengan mereduksi persoalan menjadi beberapa sub-persoalan yang lebih kecil, tetapi selanjutnya hanya memperoleh satu sub-persoalan saja.

2.2.2 *Langkah- Langkah*

Step by step dalam penyelesaian dengan paradigma decrease and conquer terbagi menjadi 3 langkah :

- a. Meminimalisir permasalahan ke sub permasalahan yang lebih kecil dari masalah yang sama dan memperluas solusi.
- b. Menyelesaikan sebuah permasalahan dengan memecahkan sub permasalahan yang lebih kecil.
- c. Perluas solusi pada sub permasalahan yang lebih kecil untuk mendapatkan solusi pada permasalahan asli.

Gagasan dasar teknik decrease and conquer didasarkan pada mencari hubungan antara solusi untuk permasalahan tertentu dan solusi untuk sub permasalahan yang lebih kecil. Pendekatan ini dikenal juga sebagai pendekatan inkremental atau induktif.

2.2.3 *Variasi*

Dilihat pada metode pembagian permasalahan menuju ke sub permasalahan yang lebih kecil, decrease and conquer bisa dibagi menjadi tiga variasi :

a. Decrease by Constant

Pada variasi ini, Takaran permasalahan dikurangi dengan konstanta yang sama pada setiap iterasi algoritma. Seringnya, konstanta ini sama dengan satu, walaupun pengurangan ukuran

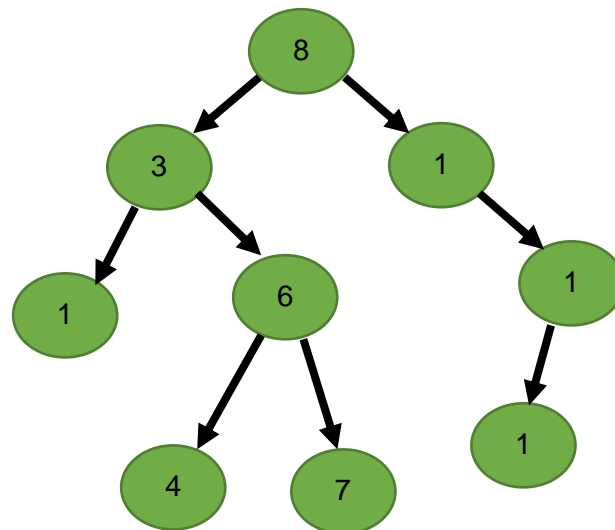
konstan lainnya dapat terjadi sesekali. Cara menyelesaikan permasalahan menggunakan decrease by constant adalah Insertion Sort, Depth First Search, Breadth First Search, dan Topological Sorting

b. Decrease by Constant Factor

Cara ini menyarankan pengurangan instance masalah dengan menggunakan faktor konstan yang serupa pada setiap iterasi algoritma. sebagian besar aplikasi, faktor konstan ini mirip dengan dua. Pengurangan dengan faktor selain dua hamper sekali jarang terjadi. Penurunan pada algoritma faktor konstan sangatlah efisien terutama ketika faktor tersebut lebih besar dari 2.

2.3 Binary Search Tree

2.3.1 Pengertian



Gambar 1. Binary Search Tree

Binary Search Tree adalah sebuah pohon biner yang setiap simpul internal menyimpan kunci (dan secara opsional, nilai yang terkait) dan masing-masing memiliki dua sub-pohon yang berbeda, dilambangkan kanan dan kiri. Pohon memenuhi properti pencarian biner, yang menyatakan bahwa kunci di setiap simpul harus lebih besar atau sama dengan kunci apa pun yang disimpan pada sub-pohon kiri, dan kurang dari atau sama dengan kunci apa pun yang disimpan pada sub-pohon kanan. Simpul akhir (daun) dari pohon tidak mengandung sub-pohon berikutnya dan tidak memiliki struktur untuk membedakannya satu dengan yang lain.

2.3.2 Operasi

Binary Search Tree mempunyai 3 operasi yaitu :

a. Searching

Mencari nilai simpul Binary Search Tree untuk nilai tertentu dan dapat diprogram secara berulang atau rekursif. Diawali dengan memeriksa simpul akar. Jika pohon adalah nol, kunci yang dicari tidak terdapat di pohon. Jika tidak, ketika kunci sama dengan akar, maka pencarian berhasil dan dikembalikan nilai simpul. Jika kunci lebih kecil dari akar, maka dicari subtree kiri. Demikian juga, jika kunci lebih besar dari pada akar, maka dicari subtree kanan.

Proses ini diulang-ulang hingga kunci didapatkan atau subtree yang tersisa adalah nol. Ketika kunci yang dicari tidak di dapatkan setelah subtree nol tercapai, maka kunci tersebut tidak terdapat pada pohon. Ini dapat dinyatakan sebagai algoritma rekursif .

b. Insertion

Penyisipan diawali dari simpul akar. ketika kuncinya tidak sama dengan akar, maka dicari pada sub pohon kiri atau kanan seperti sebelumnya. Sehingga, akan mencapai simpul eksternal dan menambahkan pasangan nilai kunci baru sebagai anak kiri atau kanannya, tergantung pada kunci simpul tersebut. Dengan kata lain, diperiksa diawali dari simpul akar dan secara rekursif menyisipkan simpul baru ke subtree kiri ketika kuncinya kurang dari akar, atau subtree kanan ketika kuncinya lebih besar dari atau sama dengan akar.

c. Deletion

Proses deletion dimulai dari simpul akar. Proses ini dilakukan secara rekursif dengan mengunjungi simpul anak dari simpul akar. Ketika simpul kanan lebih besar dari simpul akar operasi dilanjutkan ke arah subtree kanan dari akar, sebaliknya operasi dilanjutkan ke arah simpul kiri dari akar. ketika nilai sama dengan kunci yang akan dihapus, maka simpul akan dihapus dari Binary Search Tree.

2.3.3 Kelebihan

Binary Search Tree sering digunakan dalam penerapan suatu struktur data pada pemrograman. keuntungan memiliki struktur data Binary Search Tree yaitu :

- a. BST menampilkan data secara struktur
- b. BST memberikan proses Insertion, Deletion, serta Searching secara optimal
- c. Pohon yang di bentuk bersifat fleksibel

2.3.4 Kompleksitas

Kompleksitas yang paling tepat dari pencarian binary search tree adalah $O(1)$. Syarat untuk mendapatkan Kompleksitas ini saat akar di binary search tree merupakan elemen yang dicari. Kompleksitas terburuk pada pencarian binary search tree adalah jika seluruh simpul berada di kanan atau kiri. Kompleksitas pada kasus kali ini yaitu proses pencarian diharuskan menelusuri semua simpul sehingga di dapati kompleksitas berjumlah $O(n)$. Kompleksitas rata – rata memiliki arti jika simpul di sisi kanan dan kiri seimbang. Kompleksitas rata – rata mempunyai waktu sebesar $O(\log n)$.

2.4 Priority Queue

2.4.1 Pengertian

Di dalam ilmu komputer, priority queue adalah tipe data abstrak seperti struktur data queue, akan tetapi setiap elemen mempunyai "prioritas" yang masing-masing telah ditetapkan . Pada priority queue, elemen dengan prioritas lebih tinggi disajikan sebelum elemen dengan prioritas lebih rendah. Pada beberapa implementasi, ketika dua elemen memiliki prioritas yang sama, maka mereka dilayani sesuai dengan urutan di mana mereka masuk, akan tetapi dalam implementasi lainnya, penempatan elemen dengan prioritas yang sama tidak di izinkan.

2.4.2 Operasi

Pada Priority Queue mempunyai tiga operasi yaitu top, pop, dan push. Operasi top adalah untuk mengetahui nilai yang mempunyai prioritas paling tinggi pada priority queue. Operasi pop adalah mengeluarkan nilai yang mempunyai prioritas tertinggi pada priority queue. Operasi push adalah menginputkan elemen baru pada priority queue.

2.4.3 Implementasi

Priority Queue dapat diimplementasikan melalui:

a. Implementasi Naif

Ini merupakan implementasi dengan cara yang paling simpel. Implementasi naif memiliki kompleksitas yang tidak efisien dalam melakukan operasi pop elemen. Setiap elemen yang baru ditambahkan disimpan dalam array yang tidak terurut. Setiap perintah pop dipanggil, proses yang dilakukan adalah mencari nilai terbesar dalam array tersebut lalu mengembalikan nilai tersebut.

Kompleksitas dalam melakukan pemasukan elemen adalah $O(1)$ namun kompleksitas mengembalikan nilai pop adalah $O(n)$ karena harus dilakukan iterasi terhadap keseluruhan elemen.

b. Implementasi Binary Search Tree

Implementasi yang lebih baik dilakukan dengan menggunakan binary search tree. Implementasi yang dilakukan adalah :

- 1) Push: Elemen yang akan disisipkan akan ditaruh pada simpul yang tepat menggunakan penyisipan pada Binary Search Tree
- 2) Pop: Pengambilan elemen dilakukan dengan cara mengambil elemen paling kanan menggunakan modifikasi fungsi pencarian pada Binary Search Tree. Kompleksitas dalam implementasi Binary Search Tree yaitu mirip dengan teori Binary Search Tree yaitu $O(\log n)$.

3. HASIL DAN PEMBAHASAN

3.1 Ide Penyelsaian

Permasalahan Tugas Besar dapat diselesaikan dengan priority queue dengan menetapkan sistem prioritas sebagai tugas dengan deadline terdekat.

3.2 Operasi

Terdapat 3 operasi yang dapat dikerjakan dalam penyelesaian tugas besar, yaitu :

a. Menambah Tubes baru

Operasi tambah tubes baru dapat diselesaikan dengan cara melakukan perintah push pada priority queue pada daftar tugas besar. Operasi ini dapat diimplementasikan dalam bentuk binary search tree. Tugas baru yang akan ditambahkan dibandingkan dengan tugas pada simpul akar. Jika tugas baru memiliki deadline pengumpulan lebih dekat dari pada simpul akar, maka secara rekursif akan menelusuri bagian subtree kiri dari simpul akar, sebaliknya jika tugas baru memiliki .deadline pengumpulan lebih jauh dari simpul akar, maka secara rekursif akan menelusuri subtree kanan dari simpul akar. Penelusuran dilakukan secara rekursif hingga simpul mencapai simpul daun dari binary search tree. tubes baru akan ditempatkan sesuai dengan posisi yang tepat sebagai simpul anak dari simpul daun terakhir.

b. Melihat Tubes yang harus di selsaikan

Karena tubes yang harus diselesaikan adalah tubes yang memiliki deadline terdekat, maka ketika operasi ini dijalankan akan mengeluarkan tubes yang memiliki deadline terdekat. Operasi ini dapat diimplementasikan menggunakan operasi searching pada Binary Search Tree. Pimulai dari simpul akar, maka proses pencarian berjalan. Dikarenakan, operasi ini diharuskan mencari tubes dengan deadline tanggal terdekat, pencarian akan dilakukan mengunjungi subtree dari simpul akar hingga mencapai simpul daun. Jika sudah mencapai simpul daun,

kembalikan tubes yang berada di simpul daun dan inialisasikan sebagai tubes yang harus diselesaikan.

c. Menyelesaikan Tubes

Operasi menyelesaikan tubes akan menghapus tubes yang memiliki deadline terdekat dalam priority queue tugas besar. Operasi ini dapat diimplementasikan menggunakan operasi delete pada Binary Search Tree. Proses pencarian dimulai dari simpul akar. Dikarenakan, operasi ini haruslah mencari tubes dengan deadline tanggal terdekat, pencarian akan dilakukan mengunjungi subtree dari simpul akar hingga mencapai simpul daun. Jika sudah mencapai simpul daun, kembalikan tubes yang berada di simpul daun dan delete tubes tersebut.

3.3 Kompleksitas

Ide penyelesaian dari permasalahan tugas besar adalah menggunakan priority queue yang diterapkan dalam bentuk binary search tree .

Setiap operasi pada penyelesaian tugas besar sama dengan operasi pada Binary Search Tree, yaitu membagi permasalahan menjadi 2 sub permasalahan yang lebih kecil dengan ukuran pembagian selalu sama dengan konstan factor sama dengan 2. Kompleksitas Decrease and Conquer dengan konstan faktor sama dengan 2 adalah $O(\log n)$

3.4 Data Tugas Besar Teknik Informatika Semester 4

Dalam penyusunan makalah ini, kita mengambil data tugas besar teknik informatika semester 4 tahun ajaran 2020/2021. Tugas besar yang dijadikan sampel data adalah tugas besar yang memiliki tenggat waktu deadline pengumpulan pada bulan Juli dan Agustus 2021.

Berikut daftar tugas besar teknik informatika semester 4 tahun ajaran 2020/2021 :

Mata kuliah	Nama Tubes	Deadline
Strategi Algoritma	Tugas 1	Rabu, 28 Juli 2021
Grafika komputer	Tugas 1	Selasa, 27 Juli 2021
Apsi	Tugas akhir	Selasa, 2 Agustus 2021
Komunikasi Data & Jaringan	Tugas 1	Sabtu, 31 Juli 2021

Tabel 1. Daftar Tubes Informatika Semester 4

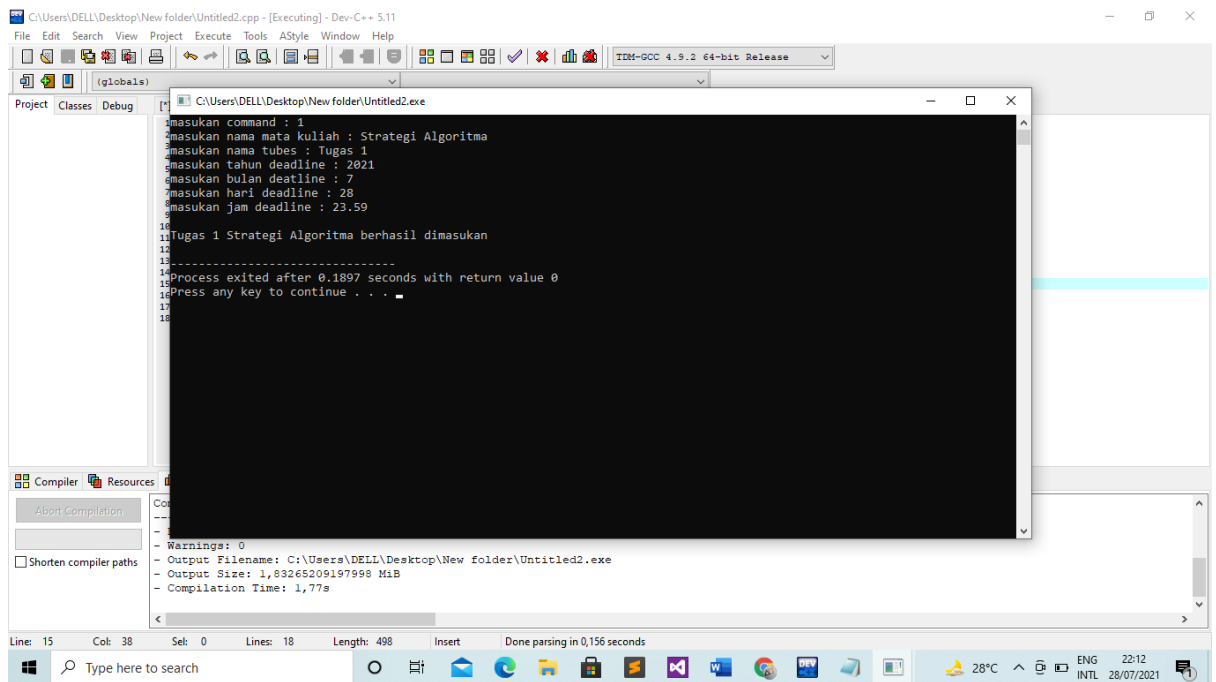
3.5 Simulasi Pengerjaan Tugas Besar

Penulis melakukan simulasi pengerjaan tugas besar melalui program yang telah dibuat. Daftarperintah yang dapat dilakukan adalah:

- Tambah tubes baru ke dalam daftar
- Lihat Tubes yang harus diselsaikan sekarang
- Konfirmasi tubes sekarang selesai

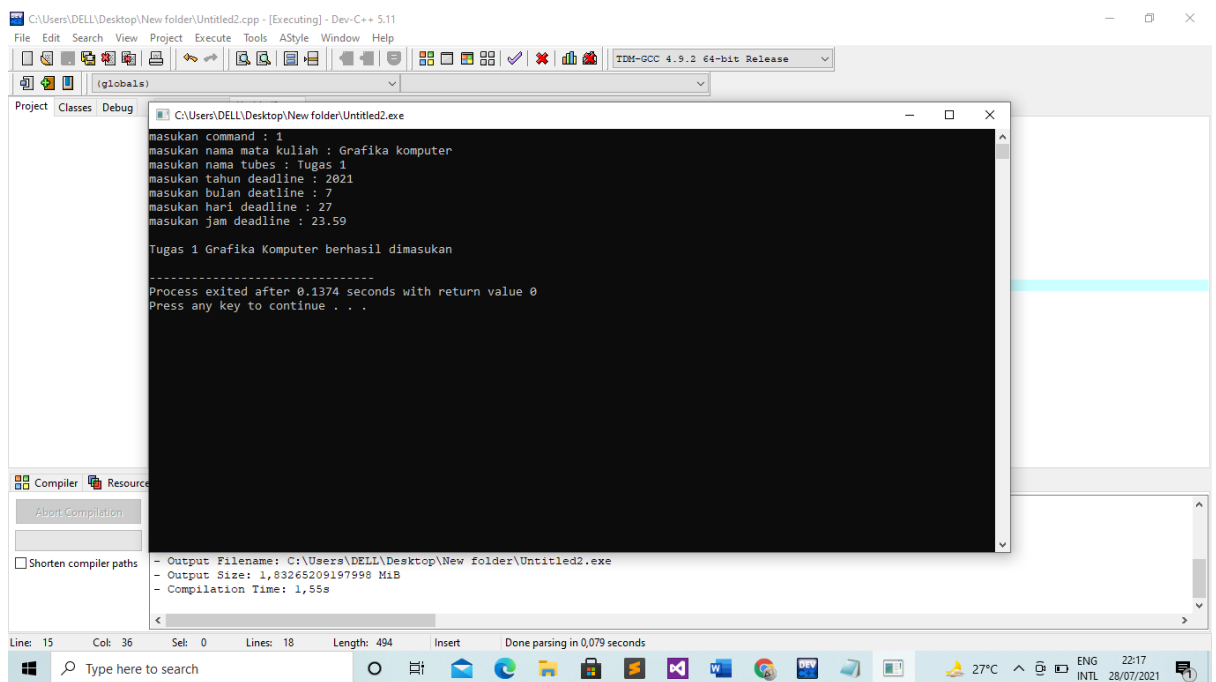
Skenario yang dibuat adalah:

- Masukan Tugas baru berupa Tugas 1 Strategi Algoritma dengan deadline pengumpulan 28 Juli 2021 jam 12.00



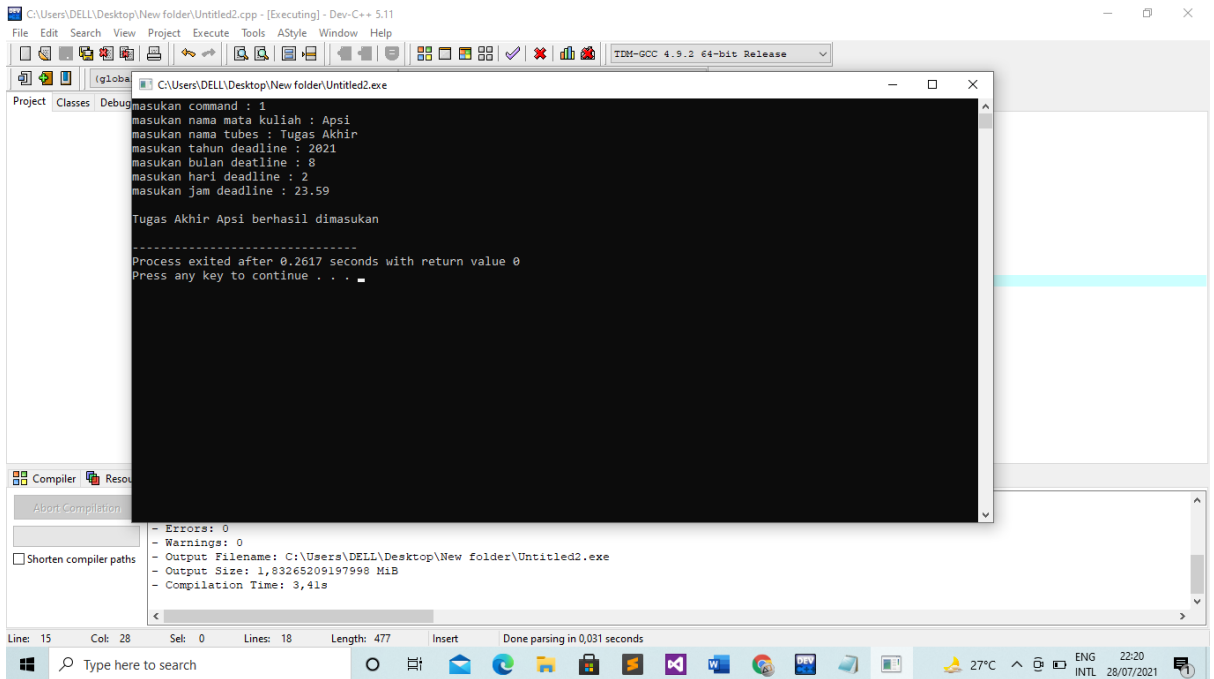
Gambar 2. Simulasi 1

- b. Masukan tubes baru berupa Tugas 1 Grafika komputer dengan Deadline Selasa, 27 Juli 2021 jam 23.59



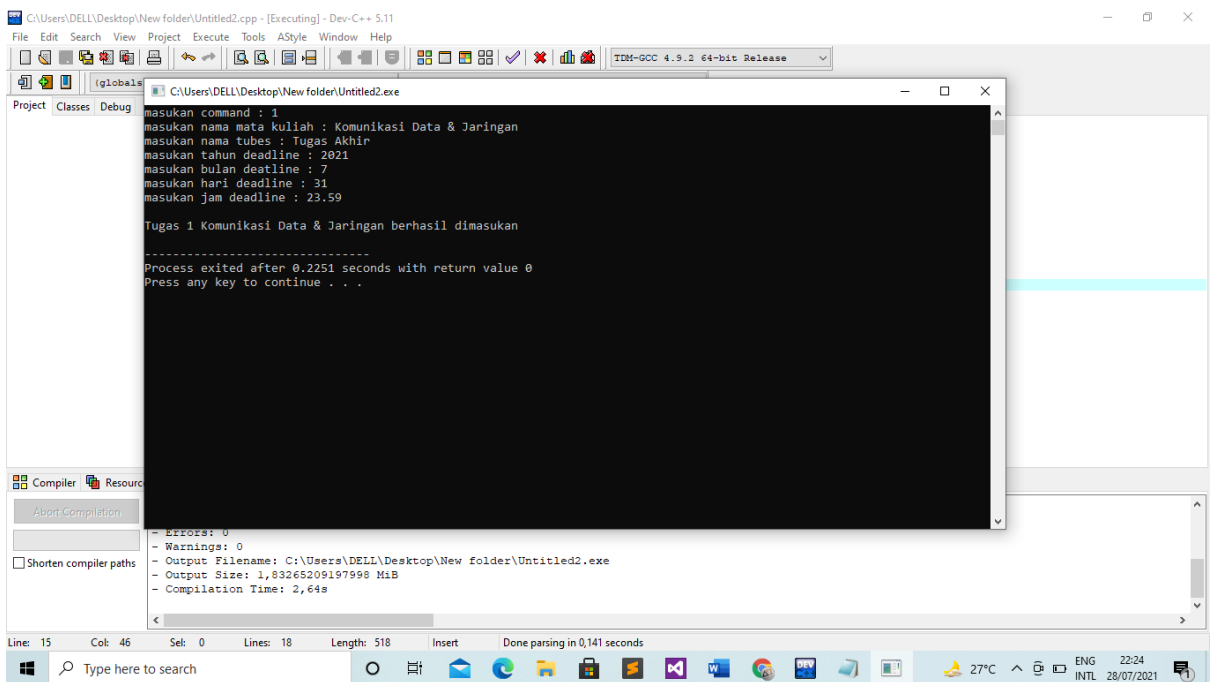
Gambar 3. Simulasi 2

- c. Masukan tubes baru berupa Tugas Akhir Apsi dengan Deadline Selasa, 2 Agustus 2021



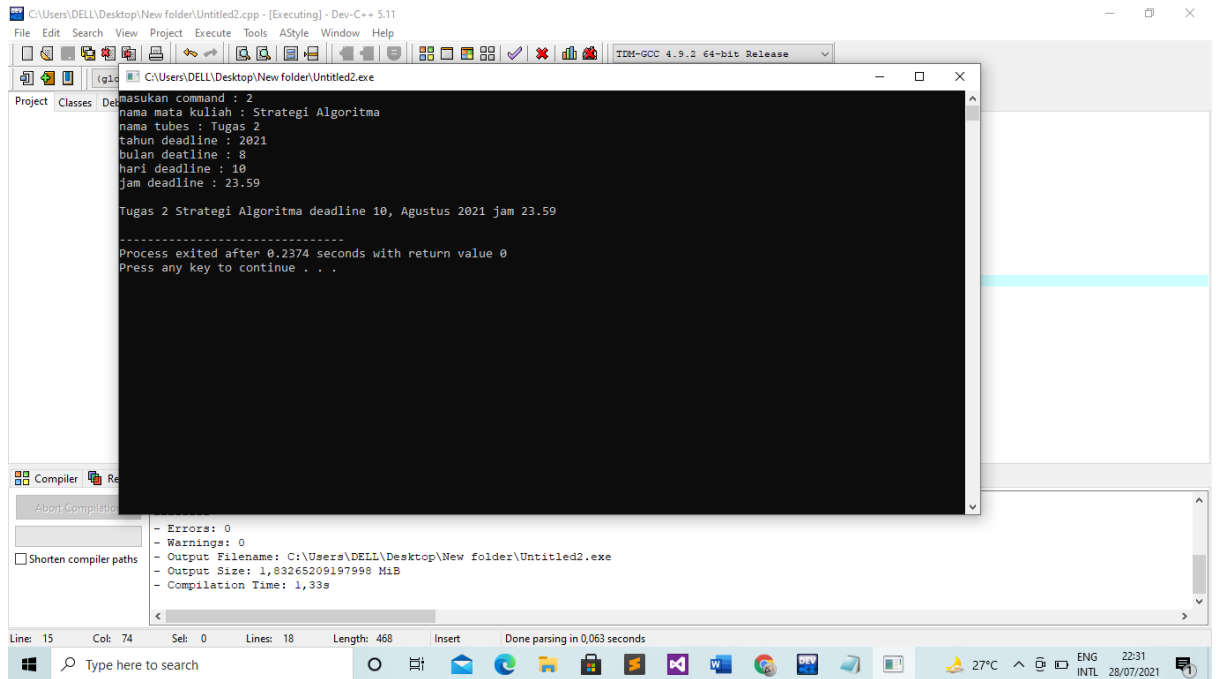
Gambar 4. Simulasi 3

- d. Masukan tubes baru berupa Tugas 1 Komunikasi Data & Jaringan dengan Deadline Sabtu, 31 Juli 2021



Gambar 5. Simulasi 4

- e. Lihat Tubes yang harus di selsaikan



Gambar 6. Simulasi 5

4. KESIMPULAN

Dalam Permasalahan tugas besar mahasiswa Teknik Informatika 4 dapat diatasi dengan menyusun skala prioritas pengerjaan. Salah satu parameter yang dapat dipakai adalah prioritas berdasarkan deadline pengumpulan yang terdekat.

Dalam ilmu komputer, hal ini dapat diterapkan dalam bentuk struktur data priority queue. Untuk memaksimalkan pemrosesan query, dapat diterapkan dalam bentuk binary search tree yang merupakan salah satu teknik penyelesaian masalah dengan paradigma decrease and conquer.

DAFTAR PUSTAKA

- [1] S. Halim and F. Halim, "Competitive Programming, 3rd Edition" Singapore, April 2013.
- [2] M. Rinaldi, Algoritma dan Pemrograman Ed. 6, Penerbit Informatika, Februari 2016
- [3] S. Robert and W. Kevin, "Algorithms (4th Edition)", Addison-Wesley Professional, Maret 2011
- [4] Esther Irawati S. , Gunawan, Indra Maryati, Joan Santoso, Rossy P.C.2012.
- [5] Rasool, A., Tiwari, A., Khare, G.S.N. 2012. String Matching Methodologies: A Comparative Analysis. International Journal of Computer Science and Information Technologies ((IJCSIT), 3(2), 3394 - 3397
- [6] Pressman, R.S. 2011. Software Engineering a Practitioner's Approach. 5th Edition. Mc Graw Hill. New York – USA
- [7] Didik Dwi Prasetyo,2005, "Mengolah Database dengan Visual Basic.NET dan MySQL Server", PT. Elex Media Komputindo, Jakarta
- [8] Fatmi, 2011, "Katalog Perpustakaan", Andi Yogyakarta.
- [9] Nugroho, Bunafit, 2005, "Database Relasional Dengan MySQL", Andi, Jakarta
- [10] Sarno Riyanto, dkk, 2012, "Semantic Search Pencarian Berdasarkan Konten", Andi Yogyakarta.
- [11] Victor Siahaan, Faran Faryadi & Yudi Fitranto Tamtoko(2018), Analisa kbutuhan sistem informasi penjualan online pada toko distri Jaulah Pontianak. *Seminar Nasional Sistem Informasi dan Teknologi Informasi*. STMIK Pontianak. SENSITEK 2018
- [12] Debbie Defrina&Dewi Putrie Lestari(2017), Aplikasi pemesanan makanan dan minuman online berbasis mobile browser pada restoran tiga saudara, Vol.22 No.3, 158-170
- [13] Bayu Aditya, Alex Wijaya & Qoriani Widayati, Rancang bangun data penjualan dan pembelian barang menggunakan konsep SOM
- [14] Miwan Kurniawan Hidayat & Rtno Catur Pangestu Ningrum(2017), Sistem informasi penjualan

- online pada toko yusuf Bekasi, Vol.2 No.2, 24-30
- [15] Heny Pratiwi, Ita Arfyanti& Dendy Kurniawan(2016), Implementasi algoritma brute force dalam aplikasi kamus istilah kesehatan, Vol.2 No.2, 119-125