

Penerapan Greedy dalam Pengerjaan Tugas Besar pada jurusan Informatika Institut Teknologi Bandung

Angger Ilham Amanullah -13521001
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13521001@itb.ac.id

Astract—Tugas Besar (Tubes) pada jurusan Informatika (IF) Institut Teknologi Bandung (ITB) adalah suatu tugas proyek dimana mengharuskan mahasiswanya untuk mengerjakan suatu tugas proyek bersama dengan mahasiswa lainnya sebagai bahan penilaian terhadap mahasiswanya. Mahasiswa seringkali dihadapkan permasalahan yaitu pada tubes dari beberapa mata kuliah diberikan dalam satu waktu dan dengan *deadline* yang terbatas sehingga membuat mahasiswa harus mengalokasikan waktunya dengan baik agar menyelesaikan semua tugasnya. Selain itu juga dengan banyaknya spesifikasi wajib yang diberikan pada tubes maka diperlukannya strategi untuk mengalokasikan pembagian tugas agar tubes bisa dikerjakan semaksimal mungkin. Makalah ini bertujuan untuk melakukan penerapan algoritma *greedy* dalam pengerjaan tubes pada jurusan IF di ITB.

Keywords—Tugas Besar, Proyek, Pemrograman, Informatika, ITB, Greedy, Algoritma Greedy.

I. PENDAHULUAN

Tugas besar (Tubes) merupakan salah satu tugas pada jurusan Informatika Institut Teknologi Bandung yang sebagian besar tugasnya adalah mengerjakan suatu proyek membuat program berkelompok dengan kelompok lain. Tubes ini merupakan salah satu komponen penting dalam penilaian kemampuan mahasiswa dan bertujuan untuk mengembangkan kemampuan mahasiswa dalam menerapkan konsep materi yang telah dipelajari dalam perkuliahan lalu disajikan dalam bentuk perangkat lunak.

Dalam lingkungan akademik yang padat dimana sumber daya manusia maupun waktu yang sangat terbatas, penting bagi mahasiswa untuk mengatur pengerjaan tugas besar secara efektif sehingga dapat menyelesaikan semua tugas besar semaksimal mungkin. Terkadang juga terdapat beberapa mata kuliah yang memberikan tugas besar kepada mahasiswanya dalam satu waktu sehingga diperlukannya pengelolaan pengerjaan tugas besar.

Dalam makalah kali ini, penulis mencoba menggunakan penerapan algoritma *greedy* dalam penentuan prioritas tugas besar yang akan dikerjakan dan pembagian kerja dalam tim sehingga dapat mendapat hasil yang optimal.

Algoritma *greedy* merupakan salah satu algoritma yang digunakan dalam pengambilan keputusan untuk memaksimalkan hasil dengan cara mempertimbangkan kriteria Makalah IF2211 Strategi Algoritma – Sem. II Tahun 2022/2023

yang didapat. Jadi pada makalah ini akan dibahas bagaimana mahasiswa dapat menerapkan algoritma *greedy* dalam memprioritaskan tugas besar mana dulu yang akan dikerjakan dan juga bagaimana pembagian pekerjaan dapat dilakukan secara efisien dalam suatu tugas besar.

Diharapkan hasil dari penelitian dalam makalah ini dapat memberikan rekomendasi praktis kepada mahasiswa dalam menghadapi tugas besar berupa suatu proyek dan pengerjaannya berkelompok sehingga melibatkan mahasiswa lain. Selain itu, diharapkan makalah ini bisa menjadi dasar dalam pengembangan strategi yang efektif dalam mengoptimalkan penentuan prioritas pengerjaan tugas dan juga pembagian tugas di dalam tugas besar.

II. DASAR TEORI

A. Tugas Besar

Tugas Besar 2 IF2210 Pemrograman Berorientasi Objek sem. 2 2022/2023
Spesifikasi Tugas Besar 2: Manajemen Usaha BNMO
IF2210 Pemrograman Berorientasi Objek
Deadline: 04Mer202302:00 WIB PMMer202302:00 WIB 03 Mei 10:10 WIB
Latar Belakang Persoalan



"Toko sebelum direnovasi"
Setelah gagal pada bisnis mereka sebelumnya, Indra dan Dori ingin membuka bisnis baru yang mencakup banyak produk seperti restoran makanan, kafe, toko DIY, apotik dan sebagainya. Untuk menunjang bisnis mereka ini, mereka ingin menggunakan BNMO sebagai sistem POS (Point of Sale). Akan tetapi, BNMO belum memiliki fitur POS yang mereka inginkan. Oleh karena itu, Indra dan Dori ingin meminta kalian, mahasiswa IF '21, untuk menambahkan fitur baru pada BNMO agar mereka dapat menjalankan bisnis baru mereka dengan lancar.

05/04/2023

pg. 1 of 24

Halaman utama Spesifikasi Tugas Besar
Sumber : Tugas Besar PBO IF ITB 2022/2023

Pada jurusan Informatika ITB, terdapat tugas besar yang merupakan tugas untuk menguji pemahaman mahasiswa terhadap materi yang dipelajari dalam bentuk pengembangan suatu perangkat lunak. Tugas besar ini biasanya dikerjakan secara berkelompok dengan mahasiswa lainnya dalam sebuah

tim. Hal ini bertujuan untuk melatih kemampuan mahasiswa baik dalam hal komunikasi maupun berkolaborasi dalam tim untuk mencapai tujuan bersama.

Selain itu, dalam tugas besar juga biasanya akan diberikan batasan waktu untuk menyelesaikan proyek tersebut. Jika mengumpulkan melebihi dari waktu batasan yang telah ditentukan hal ini akan memengaruhi nilai akhir dari tugas besar tersebut hingga parahnya dapat membuat nilai dari tugas besar menjadi 0.

Melalui tugas besar ini, mahasiswa diharapkan dapat mengembangkan kemampuan dalam pemecahan masalah, komunikasi, manajemen waktu, serta kemampuan untuk bekerja dalam tim. Selain itu, tugas besar juga dapat membantu

mahasiswa mengasah kreativitas dan juga bisa berinovasi dalam pengembangan perangkat lunak. Tugas besar ini juga bisa digunakan untuk mempercantik portofolio mahasiswa untuk pekerjaannya kelak.

Dengan demikian, tugas besar dalam jurusan Informatika memiliki peran penting membekali mahasiswa dengan segudang keterampilan dan pengetahuan yang diperlukan untuk menjadi seorang *programmer* yang berkompeten di bidang teknologi informasi dan komunikasi.

B. Algoritma Greedy

Algoritma greedy merupakan salah satu pendekatan heuristic yang biasanya digunakan untuk memecahkan masalah optimisasi. Pendekatan ini berdasarkan pada prinsip memilih pilihan terbaik pada setiap langkahnya dengan harapan dapat menghasilkan solusi akhir yang paling optimal. Pengaplikasian algoritma *greedy* sering terdapat pada situasi dimana dibutuhkan pengambilan keputusan pada setiap langkahnya tanpa mempertimbangkan kedepannya.

Sesuai namanya, algoritma greedy memiliki ciri khas dimana ia akan mengambil keputusan secara serakah disetiap langkahnya. Pada setiap langkah/iterasinya, algoritma greedy akan mengambil keputusan terbaik berdasarkan kriteria yang telah ditentukan tanpa memperhitungkan konsekuensi di masa depan. Keputusan yang diambil adalah yang terbaik pada saat itu, dengan informasi yang tersedia pada saat itu, tanpa memperhatikan keadaan keseluruhan.

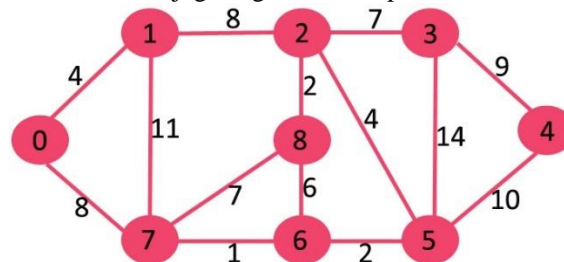
Selain itu, dikarenakan keputusan yang telah diambil disetiap langkah akan dilanjutkan dan tidak akan berubah sampai tercapainya hasil akhir, algoritma greedy memiliki kompleksitas waktu yang relative rendah karena tidak memerlukan pengulangan kembali langkah – langkah sebelumnya.

Langkah-langkah algoritma greedy didasarkan pada prinsip bahwa keputusan terbaik yang diambil pada setiap langkah akan menuntun kita ke solusi terbaik secara keseluruhan. Namun, perlu diingat bahwa algoritma greedy tidak selalu menghasilkan solusi

optimal untuk setiap masalah. Terdapat situasi dimana pendekatan ini menghasilkan solusi yang kurang optimal atau bahkan bisa jauh dari hasil yang diinginkan.

Pemilihan tindakan terbaik pada algoritma greedy ini dilakukan berdasarkan kriteria yang telah ditentukan sebelumnya. Kriteria ini dapat berupa biaya, waktu, keuntungan, atau parameter lain yang relevan dengan permasalahan yang sedang dihadapi. Tujuan utama dari algoritma greedy adalah memaksimalkan hasil dengan cara memaksimalkan atau meminimalkan kriteria yang telah ditentukan.

Ada beberapa contoh penerapan algoritma greedy yang terkenal. Contohnya yaitu Dijkstra Algorithm, Algoritma Prim, Algoritma Kruskal, dan juga Algoritma Knapsack.



Graf

Sumber :

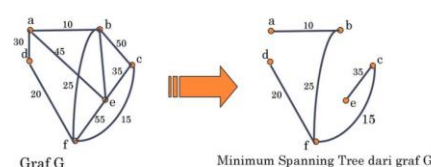
<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>

Dijkstra Algorithm merupakan algoritma yang digunakan untuk mencari jalur terpendek antara dua simpul dalam suatu graf berbobot. Algoritma ini ditemukan oleh Edsger W. Dijkstra yang sangat berguna dalam pemecahan masalah rute terpendek antara 2 simpul.

Prinsip dasar dari algoritma Dijkstra adalah membangun jalur terpendek secara bertahap, dimulai dari simpul awal menuju simpul – simpul lain hingga mencapai simpul akhir dalam graf. Algoritma ini menggunakan algoritma greedy dengan cara memilih simpul dengan jarak terkecil pada setiap langkahnya.

Algoritma Prim merupakan algoritma yang digunakan dalam mencari minimum spanning tree, yaitu mencari subgraph yang terhubung dengan bobot total terkecil yang menghubungkan semua simpul dalam graf berbobot yang tak terarah. Algoritma ini menggunakan pendekatan greedy dengan cara memilih sisi dengan bobot terkecil pada setiap langkahnya.

CONTOH



Bobot minimum spanning tree yang diperoleh dengan menggunakan algoritma Prim :

$$10 + 25 + 15 + 20 + 35 = 105$$

Algoritma Prim

Sumber :
<https://slideplayer.info/slide/12944399/>

Algoritma Kruskal juga digunakan untuk mencari minimum spanning tree seperti Prim. Algoritma ini berbeda dengan algoritma prim karena perbedaan kegunaan algoritma prim yang digunakan untuk membangun pohon penutup minimum.

CONTOH
 Selesaikan dengan menggunakan algoritma Kruskal

Graf G Minimum Spanning Tree dari graf G

Edge graf diurut menaik berdasarkan bobotnya :

Sisi	(a,b)	(c,f)	(d,f)	(b,f)	(a,d)	(c,e)	(b,e)	(a,e)	(b,c)	(e,f)
Bobot	10	15	20	25	30	35	40	45	50	55

Algoritma Kruskal

Sumber :
<https://slideplayer.info/slide/12944399/>

Algoritma Knapsack Greedy diharuskan untuk memecahkan masalah yang berkaitan dengan knapsack, dimana setiap benda memiliki bobot dan value tertentu harus dipilih untuk dimasukkan kedalam knapsack dengan kapasitas tertentu untuk mendapatkan value semaksimal mungkin.



Knapsack Problem

Sumber :
https://www.researchgate.net/figure/Gambar-1-Illustrasi-Knapsack-Knapsack-problem-secara-secara-matematis-dapat-ditulis_fig1_324078835

Meskipun algoritma greedy seringkali efisien dalam penyelesaian permasalahan yang berkaitan dengan optimisasi, terkadang dapat beberapa masalah yang perlu diperhatikan.

Algoritma greedy tidak selalu menghasilkan hasil optimal. Terkadang hasil yang didapat merupakan hasil yang tidak optimal sehingga diperlukan analisis lebih
 Makalah IF2211 Strategi Algoritma – Sem. II Tahun 2022/2023

terhadap kriteria yang dibentuk agar solusi yang didapat merupakan solusi paling optimal.

III. APLIKASI

A. Pengaplikasian algoritma greedy dalam penentuan prioritas tugas besar

Dalam penanganan tugas besar yang beragam dalam satu waktu pada jurusan informatika di Institut Teknologi Bandung, kita dapat mengaplikasikan algoritma greedy dapat membantu dalam mengatur urutan tugas besar yang akan dikerjakan. Kriteria – kriteria yang akan digunakan pada algoritma greedy kali ini adalah batas waktu penyelesaian, tingkat kesulitan, dan bobot nilai.

Dalam proses penentuan prioritas, kita harus mengidentifikasi dan memahami spesifikasi dari setiap tugas besar yang diberikan. Setiap tugas besar tentunya memiliki tingkat kesulitan tersendiri, batas waktu pengerjaan, bobot nilai tugas besar itu pada nilai akhir mata kuliah, dan sebagainya. Dalam hal ini, kita bisa memanfaatkan algoritma greedy untuk membantu kita dalam memilih tugas besar mana yang memiliki prioritas tertinggi untuk dikerjakan pada saat itu berdasarkan kriteria yang relevan.

Setiap tugas besar memiliki *deadline* atau dengan kata lain memiliki batas waktu penyelesaian tugas yang telah ditentukan. Untuk dapat mengaplikasikan algoritma greedy kedalamnya, langkah pertama yang akan kita ambil adalah mengidentifikasi *deadline* dari penyelesaian untuk setiap tugas besar yang ada pada saat itu. Setelah itu, tugas besar dengan *deadline* yang paling dekat dapat kita prioritaskan terlebih dahulu. Hal ini untuk memastikan bahwa tugas – tugas besar yang mendekati batas waktu penyelesaian dapat dikerjakan terlebih dahulu agar dapat menghindari keterlambatan saat mengumpulkan pekerjaan.

Tentunya pada tiap matkul memiliki perbedaan tingkat kesulitan dalam tugas besarnya. Dalam algoritma greedy, kita akan membedakan setiap tugas besar berdasarkan tingkat kesulitannya. Perbedaan dari tingkat kesulitan ini berdasarkan beberapa kriteria yaitu seperti kompleksitas tugas yang diberikan, jumlah fitur atau spesifikasi wajib yang perlu dibuat, maupun pemahaman anggota kelompok mengenai spesifikasi – spesifikasi yang diminta. Tugas besar dengan tingkat kesulitan yang lebih tinggi akan diberikan prioritas yang lebih tinggi dalam pemilihan pengerjaannya.

Selain itu, setiap tugas besar akan memiliki bobot nilai terhadap nilai akhir mata kuliah terkait. Bobot ini dapat menggambarkan seberapa pentingnya tugas besar tersebut dalam penilaian terhadap mahasiswanya. Algoritma greedy dapat menggunakan bobot nilai ini untuk memberikan prioritas lebih tinggi pada tugas besar yang memiliki kontribusi yang lebih besar terhadap nilai akhir mata kuliah.

IF 2210 Pemrograman Berorientasi Objek

Deadline: 3 Mei 2023 22:40 WIB 7 Mei 2023 22:40 WIB 8 Mei 10:10 WIB

menghubungi asisten. Asistensi paling lambat dilakukan Minggu, 30 April 2023

- c. Deadline pengumpulan tugas ini adalah Jumat, 5 Mei 2023, pukul 22.40 WIB

Deadline Tugas Besar yang berdekatan

Sumber : Tugas Besar PBO dan BasDat IF ITB 2022/2023

Dalam pengaplikasiannya, algoritma greedy dapat digunakan untuk mempertimbangkan ketiga kriteria ini. Pertama, kita dapat memilih tugas besar yang mendekati batas waktu penyelesaian terlebih dahulu. Selanjutnya, jika tugas – tugas besar ini memiliki batas waktu yang dekat (hanya selisih 1 – 2 hari), maka kita dapat menentukan tugas mana dulu yang

memiliki tingkat kesulitan yang lebih tinggi. Dan yang terakhir, jika terdapat tugas besar dengan tingkat kesulitan yang sama, maka kita dapat memilih tugas besar yang memiliki bobot nilai yang lebih tinggi pada nilai akhir mata kuliah terkait.

Dengan menerapkan algoritma greedy dengan tiga kriteria ini, mahasiswa dapat memastikan bahwa tugas besar yang mendesak, sulit, dan memiliki kontribusi besar terhadap nilai akhir suatu mata kuliah dapat diprioritaskan. Hal ini membuat mahasiswa dapat mengelola waktu lebih baik dan juga mendapatkan hasil dan penyelesaian spesifikasi tugas besar secara maksimal.

B. Pengaplikasian algoritma greedy dalam pengalokasian pembagian tugas pada tugas besar

Mekanisme & Fitur
Program yang Anda buat harus memenuhi beberapa mekanisme. Berikut merupakan mekanisme yang harus Anda implementasikan di program Anda.

- **Halaman Utama**
Fitur halaman utama, terdapat sebuah logo aplikasi kalian, NIM, Nama dari setiap anggota kelompok kalian dan Jam Digital.
Jam digital memiliki format sebagai berikut:
14:26
Kerangka jam digital yang digunakan merupakan pemrosesan waktu secara real-time, maka jam digital tersebut harus diimplementasikan menggunakan threading supaya bisa membaca menu utama dan menampilkan secara real-time.
Jika kalian ingin menampilkan kapasitas di halaman ini, maka diperbolehkan dengan syarat tidak ada kebaruan dari apa yang diminta. Dengan kata lain kalian tidak menambah fitur baru untuk mengupdate. Contoh kapasitas logo aplikasi tidak akan, seperti logo barisan atau barisan dengan threading.
- **Menu dan Tab**
Untuk setiap halaman yang tersedia, ditulis dalam sebuah menu sebagai berikut. Susunan halaman yang tersedia dengan control seperti berikut.

Spesifikasi Tugas Besar yang Sangat Banyak
Sumber : Tugas Besar PBO IF ITB 2022/2023

- **Customer, Member dan VIP**
Customer adalah pelanggan yang tidak memiliki keanggotaan. Pelanggan ini hanya memiliki ID yang digunakan untuk pencarian yang dibuat. Tidak ada nama karena customer ini tidak terdaftar.
Sedangkan Member adalah pelanggan yang memiliki ID, nama, dan nomor telepon yang sudah terdaftar di dalam sistem. Keanggotaan dari Member adalah setiap pasien yang dibuat dapat menggunakan poin. Poin didapatkan 1% dari harga sewa yang dibayarkan, maksimal 5000 poin per pasien. Poin dapat digunakan untuk diskon dengan aturan 1 poin = 1 rupiah diskon dan hanya bisa dipakai pada transaksi berikutnya. Customer hanya dapat menjadi Member dengan mendaftar dengan kartu sudah pernah transaksi.
Terdapat status dan jenis VIP. VIP adalah pelanggan yang ingin dengan Member. Setiap keanggotaannya lebih baik lagi karena mendapatkan fitur yang dibuat khusus. VIP dan memiliki semua poin yang diberikan serta dengan Member. Customer juga bisa menjadi VIP dengan bayar yang sama dengan Member.
- 1. Halaman pendaftaran Member
Terdapat menu agar BRMAD dapat menambahkan Member. Transaksi tersebut yang dibuat oleh customer dengan informasi akan dipaparkan kepada transaksi pertama di Member. Informasi pendaftaran Member dan VIP harus memiliki informasi hal, seperti nomor dan nomor telepon untuk dibayarkan ke dalam sistem.
- 2. Halaman update Member/VIP
Terdapat juga halaman update data Member/VIP. BRMAD dapat melakukan update data member atau member yang ada atau dapat mengupdate status member ke VIP dan sebaliknya.
- 3. Dashboard Admin Managemen

Spesifikasi Tugas Besar yang Sangat Banyak

Sumber : Tugas Besar PBO IF ITB 2022/2023

Dalam penanganan tugas besar yang beragam dalam satu waktu pada jurusan informatika, terkadang pada satu tugas besar kita mengalami kewalahan akibat banyak spesifikasi yang diberikan.

Algoritma greedy dapat digunakan dalam proses pengalokasian pembagian tugas antara anggota tim. Pengaplikasian algoritma greedy dapat membantu menentukan urutan dan juga pembagian tugas besar yang harus dikerjakan terlebih dahulu oleh anggota tim. Terdapat beberapa langkah yang dapat diambil dengan menggunakan penerapan algoritma greedy yaitu sebagai berikut.

Langkah pertama yaitu membaca spesifikasi dari tugas besar. Setiap tugas besar memiliki spesifikasi yang harus terpenuhi dalam pengerjaannya. Dengan pemahaman tentang spesifikasi dari tugas besar, kita dapat lanjut ke langkah selanjutnya dalam pembagian tugas menggunakan algoritma greedy untuk mengoptimalkan hasil yang diperoleh.

Lalu, kita harus mengevaluasi anggota tim kita. Setiap anggota tim tentunya memiliki kemampuan dan juga kapasitas yang berbeda – beda. Hal ini bisa dilakukan dengan cara melihat dari hasil – hasil tubes sebelumnya. Dengan demikian, kita bisa menggunakan algoritma greedy dalam mempertimbangkan kemampuan setiap anggota tim dalam membagi tugas besar.

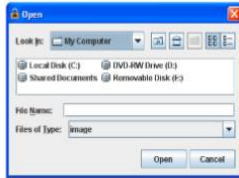
Kita juga perlu membagi spesifikasi tugas besar tadi menjadi ke beberapa sub-tugas. Lalu dari beberapa sub-tugas tersebut kita harus memprioritaskannya, dari bagian yang paling mudah hingga yang paling susah. Selain itu juga, terdapat bagian yang mungkin akan diperlukan untuk dikerjakan lebih dahulu agar bagian lain dapat diperiksa kebenarannya itu akan lebih diprioritaskan dibandingkan bagian yang lainnya. Bahkan tak jarang ada bagian yang harus dikerjakan bersama sama sehingga dapat dilakukan *testing* untuk bagian lainnya.

Setelah kita membagi bagian bagian tugas tadi menjadi lebih kecil, kita dapat mengurutkannya berdasarkan prioritasnya. Bagian tugas dengan tingkat kesulitan yang lebih rendah, dan juga ketergantungan bagian tugas lainnya kepada bagian itu akan mendapatkan prioritas lebih tinggi. Setelahnya, kita dapat menyesuaikan sub-tugas tadi berdasarkan prioritasnya kepada anggota tim kita berdasarkan hasil evaluasi dari anggota tim.

- Plugin

Plugin adalah perangkat lunak yang menambahkan fitur tertentu pada suatu program tanpa harus mengubah program itu sendiri. Biasanya, plugin digunakan agar pengguna dapat memilih komposisi fitur pada program yang tersedia. Sebagai contoh, *wordpress* adalah salah satu perangkat lunak berbasis website yang menggunakan sistem plugin, sehingga pengguna dapat memilih plugin yang dibutuhkan dan memungkinkan *developer* untuk membuat plugin baru tanpa mengubah *wordpress* itu sendiri.

Tugas anda adalah membuat sistem plugin pada tugas besar ini. Secara umum, terdapat sebuah menu plugin yang dapat membuka sebuah panel baru untuk membaca file. Tampilan memilih file adalah sebagai berikut.



Sumber Gambar: [Wuu.MIT](#)

File yang bisa dibaca hanyalah *file jar* yang spesifikasinya dijelaskan pada bagian berikutnya. Untuk menjalankan JAR yang dibaca secara dinamis, dapat memanfaatkan *Java Reflection* dan *Java Class Loader*. Berikut adalah bahan bacaan yang mungkin dapat membantu anda untuk mengerjakan bagian ini:

1. [Java Reflection: Dynamic Class Loading](#)
2. [Load JAR File Dynamically](#)

Spesifikasi dengan tingkat kesulitan paling tinggi

Sumber : *Tugas Besar PBO IF ITB 2022/2023*

Saat ditengah keberjalanan pengerjaan tugas, terkadang terdapat hal-hal yang tidak diduga. Contohnya seperti ada bagian yang ternyata lebih susah daripada yang dikira. Jika hal tersebut terjadi, maka upaya yang kita lakukan adalah mengulang lagi langkah – langkah evaluasi dengan menggunakan algoritma greedy sebelumnya. Hal ini dilakukan dengan cara mengevaluasi lagi akan tetapi hanya pada bagian yang belum selesai dikerjakan. Kita dapat melakukan pengamatan dari keberjalanan dan perkembangan proyek tugas besar kita. Dengan demikian kita dapat memastikan bahwa pembagian tugas tetap optimal sepanjang proyek berlangsung.

Dengan memanfaatkan Algoritma Greedy dalam pengerjaan Tugas besar, mahasiswa dapat memperoleh banyak kemudahan seperti :

1. Algoritma greedy dapat membantu memaksimalkan penggunaan waktu dan sumber daya manusia dengan menempatkan anggota tim pada tugas besar yang sesuai dengan kemampuan dan kapasitas mereka sehingga dapat menghindari waktu yang terbuang sia sia.
2. Setelah menerapkan cara ini dalam pengerjaan tugas maka kita akan mengurangi risiko keterlambatan pengumpulan tugas.
3. Algoritma greedy ini akan memperhitungkan kemampuan dari setiap anggota tim dalam pembagian tugas sehingga setiap anggota mendapatkan distribusi beban kerja yang seimbang dan proporsional.

IV. KESIMPULAN

Pada makalah ini, penulis telah menjelaskan penerapan algoritma greedy dapat digunakan untuk membantu dalam pengerjaan tugas besar di jurusan Informatika Institut Teknologi Bandung. Penulis menunjukkan bahwa dengan menggunakan teknik ini, mahasiswa dapat menentukan prioritas dari tugas besar mana yang dikerjakan terlebih dahulu dan juga dapat menentukan pembagian tugas dalam suatu tugas besar. Penulis juga menunjukkan bagaimana teknik ini dapat digunakan untuk menganalisis kriteria pembobotan apa saja yang diperlukan seperti batas waktu penyelesaian, tingkat kesulitan tugas, dan bobot nilai dari tugas besar. Dengan memperhatikan itu, algoritma greedy dapat mengurutkan tugas besar mana yang paling mendesak. Selain itu, algoritma greedy juga dapat diterapkan dalam pengalokasian pembagian tugas dimana algoritma greedy akan mempertimbangkan keahlian tiap anggota kelompok dan juga bagaimana spesifikasi tugas akan dibagi. Dengan demikian, penulis menyimpulkan bahwa algoritma greedy tersebut efektif dalam mengoptimalkan proses pengambilan keputusan dalam prioritas tugas besar dan pembagian tugas didalamnya sehingga diharapkan setiap mahasiswa khususnya mahasiswa informatika Institut Teknologi Bandung terbantu dalam mengerjakan proyek tugas besar.

V. PENUTUP

Pada bagian penutup ini, penulis ingin menekankan bahwa algoritma greedy dapat menjadi alat yang bermanfaat bagi mahasiswa dalam keberjalanan kuliah khususnya dalam pengerjaan tugas besar. Dengan menggunakan teknik ini, mahasiswa dapat membuat keputusan yang lebih tepat dalam pengerjaan tugas besar. Penulis juga ingin menyampaikan bahwa teknik ini masih dapat dikembangkan lebih lanjut untuk meningkatkan keefektifan dalam mengerjakan tugas besar.

Namun, penulis percaya bahwa dengan penggunaan yang tepat, algoritma greedy dapat berguna bagi mahasiswa dalam meningkatkan penyelesaian tugas besar mereka.

REFERENCES

- [1] Munir, Rinaldi, Slide Perkuliahan IF 2211 Strategi Algoritma Greedy I, diakses pada 21 Mei 2023, 20.00 WIB.
- [2] Munir, Rinaldi, Slide Perkuliahan IF 2211 Strategi Algoritma Greedy II, diakses pada 21 Mei 2023, 21.00 WIB.
- [3] Munir, Rinaldi, Slide Perkuliahan IF 2211 Strategi Algoritma Greedy III, diakses pada 21 Mei 2023, 21.00 WIB.

PERNYATAAN

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

Bandung, 22 Mei 2023



Angger Ilham Amanullah 13521001