Pemanfaatan Algoritma Greedy dan Graf dalam Penentuan Prioritas Pengerjaan Tugas di Informatika

Kevin Andrian Liwinata 13516118
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
kevinandrianliu@gmail.com

***Abstract*—Informatika adalah ilmu yang mempelajari cara menggabungkan perangkat keras dan perangkat lunak dalam memproses suatu data menjadi informasi. Informatika merupakan ilmu yang memiliki cakupan luas sehingga sulit untuk mempelajari semua materi Informatika selama S1 sehingga paradigma yang paling memungkinkan untuk mempelajarinya adalah dengan mempelajari inti logika dari suatu materi, dan diterapkan dalam tugas-tugas yang diberikan. Ini menyebabkan banyak tugas yang harus diberikan agar mahasiswa dapat semakin paham materi di kelas, dan membuat mahasiswa kebingungan harus memilih tugas yang mana untuk diprioritaskan agar dapat memperoleh hasil yang terbaik. Makalah ini mengkaji penetapan prioritas dalam pengerjaan suatu tugas dengan metode Greedy, salah satu strategi yang digunakan agar mendapatkan hasil yang cukup optimum. Pengkajian akan dilakukan dengan mempertimbangkan berbagai kemungkinan strategi Greedy yang diterapkan**

***Keywords*—Informatika, tugas, prioritas, Greedy**

I. PENDAHULUAN

Informatika adalah ilmu yang mempelajari tentang pemrosesan suatu informasi, dimulai dari tampilan antarmukanya dan mengaplikasikan metode untuk menyimpan dan mengolah data sehingga menghasilkan suatu informasi. Menurut Dr.Parker, Informatika bukan hanya belajar untuk memprogram saja, melainkan mengutilisasi sistem kesatuan perangkat keras dan perangkat lunak sehingga dapat berfungsi lebih efektif, dan sangat terpakai dalam organisasi untuk bersaing di era IT. Informatika memiliki cakupan yang luas, seperti dasar-dasar logika dan pemanfaatannya dalam AI, penggunaan dan pengoptimalisasian basis data agar dapat sesuai dengan kebutuhan pengguna yang sedang dimodelkan, penggunaan organisasi dan arsitektur komputer dan sistem operasi untuk mengetahui keadaan komputer kita dan memungkinkan kita dapat membuat algoritma yang efisien dan tidak memberatkan komputer, sampai berbagai paradigma pemrograman agar dapat memiliki gambaran secara utuh tentang dasar-dasar perbedaan pola pemikiran dalam sekelompok bahasa pemrograman. Untuk mempelajari semua topik dalam Informatika selama 4 tahun kuliah merupakan hal yang cukup sulit sehingga pendekatan yang memungkinkan agar dapat mempelajari semua materi dengan efisien adalah belajar logikanya, bukan belajar bahasa pemrograman secara spesifik(meskipun mata kuliah yang menekankan tentang paradigma pemrograman perlu mempelajari bahasa pemrograman tertentu secara spesifik supaya lebih mudah dipraktekkan nanti). Namun, belajar Informatika tidak dapat dipisahkan dari memprogram, karena pada dasarnya inti dari semua pekerjaan di Informatika adalah membuat program yang sesuai dengan kebutuhan pengguna dalam berbagai aspek. Oleh karena itu, dibuatlah tugas-tugas yang banyak agar setiap mahasiswa/-i dapat mengeksplorasi sendiri bahasa pemrograman tertentu agar dapat membiasakan diri dengan bahasa tersebut dan semakin mengasah kemampuan logika yang sudah didapatkan ketika di kelas.

Tugas yang melimpah inilah yang menjadi pertimbangan setiap mahasiswa dalam penentuan prioritas pengerjaannya. Terkadang, banyak tugas yang harus dikerjakan secara bersamaan ketika tenggat waktu sudah di depan mata sehingga banyak mahasiswa merasa bingung untuk memprioritaskan tugas yang mana. Di sisi lain, mereka membawa tanggung jawab dan harapan orang tua mereka untuk mendapatkan nilai sebaik-baiknya sehingga mereka harus bisa mendapatkan hasil yang terbaik di setiap tugas yang mereka kerjakan. Oleh karena itu, mereka memerlukan strategi yang tepat dalam pengerjaan tugas mereka.

II. LANDASAN TEORI

Strategi Greedy adalah strategi optimalisasi yang menggunakan strategi secara langsung sesuai dengan sifat intiuitif manusia yang ingin mengambil pilihan yang terbaik dari sekumpulan plihan yang ada. Pada dasarnya, strategi Greedy membagi masalah menjadi sekumpulan pilihan, yang pada setiap tahapannya akan dipilih pilihan yang dapat memberikan hasil paling optimal di antara pilihan yang lain.[1] Pengambilan pilihan dilakukan yang terbaik pada saat itu, sehingga strategi Greedy tidak akan mencari semua kemungkinan solusi dalam penyelesaian suatu masalah. Harapannya adalah ketika didapatkan nilai terbaik(atau disebut optimum lokal) pada setiap tahapannya, akan menghasilkan optimum global. Pada strategi Greedy, didefinisikan elemen-elemen sebagai berikut:

1. Himpunan kandidat
Himpunan ini adalah semesta yang dibicarakan, artinya semua

pilihan yang mungkin. Misalkan dalam TSP, himpunan kandidatnya adalah semua simpul yang harus dilalui.

2. Himpunan solusi
Bagian dari himpunan kandidat yang mengarah ke solusi.

3. Fungsi Seleksi
Fungsi yang menentukan pilihan terbaik yang harus diambil. Fungsi ini merupakan hal paling utama dalam strategi Greedy, dan perlu mempertimbangkan beberapa kemungkinan fungsi seleksi yang mungkin agar menadapatkan hasil paling optimum.

4. Fungsi Kelayakan
Fungsi yang menentukan apakah suatu solusi dapat diterima atau tidak sesuai dengan batasan kondisi yang ditetapkan dalam suatu permasalahan

5. Fungsi Objektif
Fungsi tujuan yang harus dicapai dalam suatu permasalahan.

Misalkan kita mengambil contoh kasus TSP:
1. Himpunan Kandidat
Semua simpul dalam graf

2. Himpunan Solusi
$S = \{s_1,s_2,s_3,\ldots,s_n,s_1\}$ dengan s menyatakan simpul yang harus dipilih pada urutan pertama dan seterusnya.

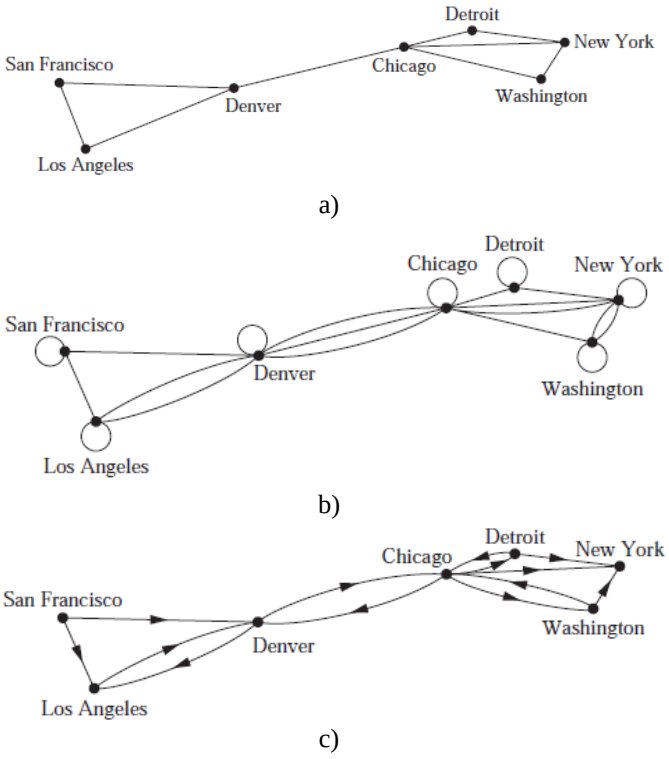
3. Fungsi Seleksi
Ada berbagai macam kemungkinan, seperti fungsi yang menentukan jalan dengan ongkos paling minimal, atau fungsi yang menentukan jalan dengan jumlah jalan tersedikit, baru bobot paling minimum. Misalkan dari simpul ada 2 pilihan, simpul B dan simpul C. Ada 2 jalan ke B dengan bobot 10 dan 15, sedangkan hanya ada 1 jalan ke C dengan bobot 20. Hasil dari fungsi seleksi adalah jalan ke C.

4. Fungsi Kelayakan
Fungsi yang menentukan himpunan solusinya memiliki jumlah simpul yang sama lebih dari satu kali atau tidak.

5. Fungsi Objektif
Terbentuk himpunan solusi dengan ongkos paling minimal.

Strategi Greedy tidak selalu memberikan solusi terbaik karena tidak mendaftarkan dan menyeleksi semua kemungkinan solusi, tetapi dijadikan sebagai hampiran dari solusi.[1] Strategi Greedy diterima oleh para ilmuwan dan masyarakat saat ini yang membutuhkan waktu yang singkat dalam penentuan suatu keputusan dengan kesalahan yang tidak terlalu besar.

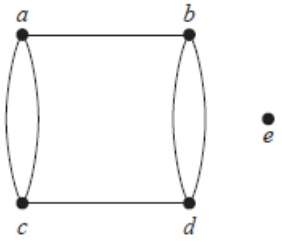
Graf adalah suatu struktur data yang terdiri dari simpul dan sisi yang menghubungkan antara dua simpul. Berdasarkan ada atau tidaknya gelang atau kalang dan sisi ganda, graf dibagi menjadi graf sederhana (graf yang tidak memiliki kalang dan sisi ganda), dan graf tak sederhana(graf yang memiliki kalang dan/atau sisi ganda). Sisi ganda adalah sisi yang jumlahnya lebih dari satu tetapi menghubungkan sepasang simpul yang sama. Kalang tau gelang adalah sisi yang menghubungkan suatu sisi dengan dirinya sendiri. Berdasarkan tanda panah, graf dibagi menjadi graf berarah(memiliki tanda panah, yang menunjukkan arah yang dapat dilewati pada suatu sisi, yaitu simpul memiliki ekor panah menuju simpul yang memiliki kepala panah) dan graf tak-berarah(tidak memiliki tanda panah sehingga sisi yang menghubungkan kedua simpul dapat dilewati dari simpul yang berhubungan dengan sisi tersebut menuju simpul lainnya yang berhubungan dengan sisi tersebut).



Gambar 1 Contoh graf, terdiri dari a) Graf sederhana, b) Graf tak-sederhana yang memiliki kalang pada setiap simpulnya dan sisi ganda seperti pada Los Angeles-Denver, Denver-Chicago, Chicago-New York, dan New York-Washington, dan c) Graf Berarah (sumber: Rosen, *Discrete Mathematics and Its Applications*, 7th Ed.)

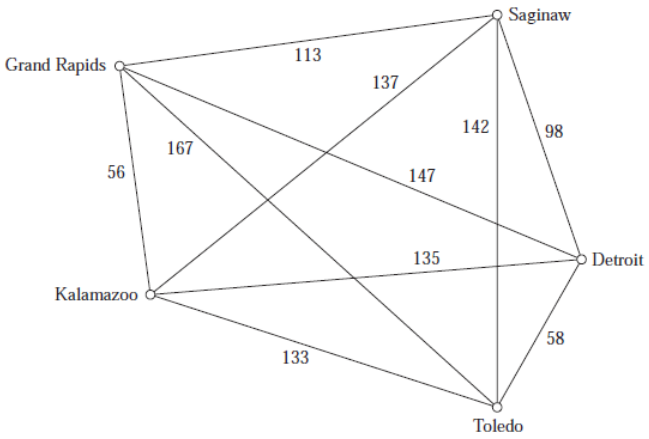
Beberapa istilah dalam graf yang digunakan dalam makalah ini adalah sebagai berikut:
1. Ketetanggaan, yaitu simpul-simpul yang dihubungkan oleh sisi tertentu
2. Bersisian, yaitu sisi-sisi yang melewati simpul-simpul tertentu
3. Simpul terpencil, yaitu simpul yang tidak terhubung oleh sisi manapun.

4. Derajat, yaitu banyaknya sisi yang terhubung dengan simpul tersebut. Pada graf berarah, didefinisikan dua jenis derajat, yaitu derajat masuk atau banyaknya sisi yang kepala panahnya menuju ke simpul tersebut, dan derajat keluar, yaitu banyaknya sisi yang ekor panahnya menuju ke simpul tersebut.
5. Lintasan adalah kumpulan simpul-simpul yang setiap pasang simpul yang bersebelahan merupakan tetangga.
6. Siklus adalah lintasan yang memiliki simpul awal dan simpul akhir yang sama.
7. Terhubung, yaitu suatu keadaan pada dua simpul jika terdapat lintasan yang melalui kedua simpul tersebut.
8. Graf berbobot adalah graf yang sisinya mempunyai nilai bobot tertentu.
9. Graf lengkap adalah graf yang semua simpulnya terhubung dengan simpul lainnya(K_n).
10. Lintasan Hamilton, yaitu lintasan yang menghubungkan setiap simpul dalam graf tepat sekali. Syarat cukup adanya lintasan Hamilton adalah setiap pasang simpul u dan v yang tidak bertetangga dalam suatu graf, jumlah kedua derjatatnya ditambah derajat minimum dari kedua simpul tersebut paling sedikit n+1 menurut syarat Rahman-Kaykobad[2].



Gambar 2. Contoh Graf (sumber: Rosen, *Discrete Mathematics and Its Applications*, 7th Ed.)

Misalkan pada contoh graf pada gambar 2, kita mendapatkan infromasi sebagai berikut:
1. a bertetangga dengan b dan c, b bertetangga dengan a dan d, c bertetangga dengan a dan d, d bertetangga dengan b dan c, dan e tidak bertetangga dengan simpul manapun.
2. Sisi (a,b) bersisian dengan simpul a dan b, sisi (a,c) bersisian dengan simpul a dan c, sisi (b,c) berisisian dengan simpul b dan c, sisi (b,d) bersisian dengan simpul b dan d, dan sisi (c,d) bersisian dengan simpul c dan d.
3. e merupakan simpul terpencil
4. Simpul a,b,c, dan d memiliki derajat 3, sedangkan simpul e memiliki derajat 0.
5. a-b-d-c adalah suatu lintasan, sedangkan a-b-e-d-c bukan suatu lintasan.
6. a-b-d-c-a adalah suatu sirkuit, sedangkan b-d-c bukan sebuah sirkuit meskipun adalah sebuah lintasan.
7. Simpul a,b,c, dan d terhubung dengan semua simpul yang ada dalam graf ini kecuali simpul e, sedangkan simpul e tidak terhubung ke simpul manapun selain simpulnya sendiri.



Gambar 3. Contoh Graf (sumber: Rosen, *Discrete Mathematics and Its Applications*, 7th Ed.)

Contoh dari lintasan Hamilton adalah graf pada gambar 3. Graf ini adalah graf berbobot, serta memiliki sirkuit Hamilton dengan bukti bahwa graf tersebut adalah graf teratur(K_5) karena semua simpulnya berhubungan dengan setiap simpul lainnya pada graf tersebut sehingga derajatnya adalah 4. Menurut syarat Rahman-Kaykobad, karena setiap simpulnya memiliki derajat 4+4 >= 4+1, graf tersebut pasti memiliki lintasan Hamilton.

III. PENGKAJIAN BEBERAPA ALTERNATIF FUNGSI SELEKSI DALAM STRATEGI GREEDY DENGAN MEMANFAATKAN GRAF DALAM PENENTUAN PRIORITAS PENGERJAAN TUGAS BESAR DI INFORMATIKA

Dalam penentuan prioritas pengerjaan tugas besar di informatika ini, terdapat data yang diambil pada hari Kamis, 19 April 2018:

No	Nama	SKS	Bobot tugas (dalam %)	Tenggat Waktu (dalam hari)	Estimasi Waktu Pengerjaan (dalam jam)
1	Strategi Algoritma	3	10	4	6
2	Pemrograman Berorientasi Objek	3	10	6	12
3	Pemrograman Berorientasi Objek	3	5	8	4
4	Sistem Operasi	3	10	6	8
5	Basis Data	3	5	5	4
6	Dasar Rekayasa Perangkat Lunak	2	5	8	12

Tabel 1. Contoh tabel yang berisi informasi mengenai tugas-tugas yang memiliki tenggat waktu sesudah 19 April 2018

Informasi-informasi tersebut dijadikan pertimbangan untuk fungsi seleksi nantinya. Seorang mahasiswa dengan keadaan tugas seperti itu, hanya memiliki 30 jam untuk mengerjakan semua tugas yang diberikan. Ia berharap agar ia mampu mengerjakan sebanyak-banyaknya tugas dalam waktu luangnya tersebut.
Pertama, didefinisikan terlebih dahulu elemen-elemen dari strategi Greedy:
1. Himpunan Kandidat, yaitu semua tugas yang diberikan
2. Himpunan Solusi, yaitu urutan {T1, T2, …, Tn} dengan Ti menunjukkan tugas ke-i
3. Fungsi Seleksi, akan dijelaskan kemudian
4. Fungsi Kelayakan, yaitu apakah total waktu untuk pengerjaan tugas melebihi total waktu yang ada.
5. Fungsi Objektif memaksimumkan jumlah tugas yang dapat dikerjakan dalam waktu tertentu

Ini dimasukkan ke dalam suatu *list* yang berisi *list* yang terdiri dari nama tugas, SKS, bobot tugas, dan tenggat waktu. Setiap list akan dipilih berdasarkan urutan secara berturut-turut sebagai berikut:
1. Berdasarkan SKS terbesar terbesar, bobot terbesar terbesar, dan tenggat waktu terkecil
2. Berdasarkan SKS terbesar terbesar, tenggat waktu terkecil, dan bobot terbesar terbesar
3. Berdasarkan bobot terbesar terbesar, SKS terbesar terbesar, dan tenggat waktu terkecil
4. Berdasarkan bobot terbesar, tenggat waktu terkecil, dan SKS terbesar
5. Berdasarkan tenggat waktu terkecil, SKS terbesar, dan bobot terbesar
6. Berdasarkan tenggat waktu terkecil, bobot terbesar, dan SKS terbesar
7. Berdasarkan bobot*SKS terbesar
8. Berdasarkan deadline/bobot terkecil
9. Berdasarkan deadline/SKS terkecil

Hubungan dengan graf didefinisikan sebagai berikut:
Persoalan ini dapat didefinisikan sebagai pembuatan graf yang dibuat dalam bentuk *list of list* , dengan setiap simpul berisi elemen dengan nilai yang berbeda. Lalu, grafnya memiliki nilai ke simpul lain sebesar nilai yang akan diseleksi dalam fungsi seleksi(SKS, tenggat waktu, dan bobot). Selanjutnya, dari simpul awal, akan dipilih simpul pertama yang memiliki nilai paling optimum sesuai dengan fungsi seleksi, lalu akan membuat lintasan Hamilton (bukan sirkuit Hamilton karena tidak ada pengerjaan tugas yang sama sebanyak dua kali) dengan total ongkos(dalam hal ini waktu total) tidak melebihi 30.

Algoritma secara umum yang digunakan adalah *list* akan memproses sesuai fungsi seleksi, memilih salah satu elemen yang paling sesuai dengan fungsi seleksi, lalu mengelurakannya dari *list* tersebut sampai *list* awal kosong. Ini menghasilkan suatu *list* yang terurut berdasarkan fungsi seleksi yang ada. Selanjutnya, setiap elemen *list* ini akan diproses berdasarkan waktunya. Urutan elemen yang diambil adalah elemen pertama dan seterusnya sampai elemen terakhir dan menunjukkan bahwa elemen dengan prioritas yang lebih tinggi yang akan diambil. Jika ada suatu elemen yang tidak memenuhi syarat, elemen tersebut akan dilewati dan dilanjutkan pencarian sampai tidak ada satupun tugas yang dapat dikerjakan dalam waktu yan tersisa. Dalam notasi algoritma, algoritmanya adalah sebagai berikut:

```
temp = {}
while C ≠ {}
  x <- fungsi_seleksi(C)
  temp <- temp ∪ {x}
  C <- C - {x}
endwhile
continue <- true
i = 0
while  i < n(S) – 1 and continue do
  continue <- false
  if amount ≥ time (x) then
    S <- S ∪ {x}
    temp <- temp - {x}
    continue <- true
  endif
endwhile
```

Gambar 4. Notasi Algoritmik untuk Program Scheduler

continue merupakan variable bertipe Boolean yang berfungsi untuk menentukan apakah masih ada elemen yang memiliki waktu pengerjaan lebih kecil atau sama dengan waktu tersisa. amount ≥ time (x) merupakan fungsi kelayakan karena akan membatasi himpunan solusi yang ada. Aloritma ini menyeleksi tugas-tugas yang dapat dikerjakan dalam waktu yang diberikan sehingga terdapat kemungkinan kasus-kasus tertentu saat ada tugas yang tidak dapat dikerjakan karena waktu yang kurang. Ini dapat dilihat dari S yang merupakan himpunan semua tugas yang dikerjakan dan temp yang merupakan himpunan semua tugas yang tidak dapat dikerjakan karena kurangnya waktu.

Selain itu, untuk membuktikan kecepatan algoritma Greedy yang dipakai, dibuatlah waktu dari awal pembentukan himpunan temp sampai mendapatkan himpunan solusi dan himpunan tugas yang tidak dapat dikerjakan. Hasilnya adalah sebagai berikut:

```
Tentukan pilihan prioritas:
1. Berdasarkan SKS,bobot,deadline
2. Berdasarkan SKS,deadline,bobot
3. Berdasarkan bobot,SKS,deadline
4. Berdasarkan bobot,deadline,SKS
5. Berdasarkan deadline,SKS,bobot
6. Berdasarkan deadline,bobot,SKS
7. Berdasarkan bobot*SKS
8. Berdasarkan deadline/bobot
9. Berdasarkan deadline/SKS
-9999. Sudah tidak memilih lagi.
Pilih pilihan: 1
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
3. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming
4. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Eksplorasi Bahasa Pemrograman Berorientasi Objek
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.008033 detik
Pilih pilihan kembali: 2
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks
3. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
4. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Eksplorasi Bahasa Pemrograman Berorientasi Objek
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.00303 detik
Pilih pilihan kembali: 3
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
3. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming
4. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Eksplorasi Bahasa Pemrograman Berorientasi Objek
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.001002 detik
```

a)

```
Pilih pilihan kembali: 4
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
3. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming
4. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Eksplorasi Bahasa Pemrograman Berorientasi Objek
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.016961 detik
Pilih pilihan kembali: 5
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks
3. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
4. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Eksplorasi Bahasa Pemrograman Berorientasi Objek
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.015149 detik
Pilih pilihan kembali: 6
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks
3. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
4. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Eksplorasi Bahasa Pemrograman Berorientasi Objek
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.029082 detik
Pilih pilihan kembali: 7
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
3. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming
4. Eksplorasi Bahasa Pemrograman Berorientasi Objek

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.047468 detik
```

b)

```
Pilih pilihan kembali: 8
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
3. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming
4. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Eksplorasi Bahasa Pemrograman Berorientasi Objek
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.016392 detik
Pilih pilihan kembali: 9
Urutan tugas:
1. Tugas Besar 3 Strategi Algoritma - Spam dengan String Matching
2. Tugas Besar Milestone 4 Basis Data - Normalisasi dan Indeks
3. Tugas Besar 2 Pemrograman Berorientasi Objek- Translasi Java
4. Tugas Besar Milestone 3 Sistem Operasi - Konkurensi dan Multiprogramming

Urutan tugas yang tidak dapat dikerjakan karena kurangnya waktu:
1. Eksplorasi Bahasa Pemrograman Berorientasi Objek
2. Tugas 8 Dasar Rekayasa Perangkat Lunak - Tugas Final
Hasil ini dalam 0.015683 detik
```

c)

Gambar 4. Hasil keluaran program, a) untuk fungsi seleksi 1 sampai 3, b) untuk fungsi seleksi 4 sampai 7, dan c) untuk fungsi seleksi 8 dan 9

Keluaran program memberikan beberapa kesimpulan sebagai berikut:

1. Secara hasil, jumlah tugas yang dapat dikerjakan berdasarkan setiap fungsi seleksi yang diberikan adalah 4 dengan tugas pertama yang dikerjakan adalah tugas dari Strategi Algoritma, dan ada 2 tugas yang tidak dapat dikerjakan, dengan salah satunya selalu tugas dari Dasar Rekayasa Perangkat Lunak
2. Waktu keluaran yang berbeda-beda untuk setiap program, diduga karena arsitektur komputer yang digunakan sehingga berpengaruh pada performansi dari CPU ketika memproses algoritma yang diberikan.
3. Belum adanya fungsi seleksi yang paling baik, dikarenakan preferensi dari setiap mahasiswa itu sendiri. Mahasiswa mungkin dapat mempertimbangkan beberapa hal di luar kemampuan program, seperti preferensi pemilihan satu tugas dari salah satu mata kuliah atau waktu estimasi yang berbeda dengan waktu pengerjaan.

IV. SARAN

Saran untuk pengembangan makalah ini adalah sebagai berikut:

1. Penggunaan algoritma yang lebih efisien

2. Penggunaan arsitektur komputer yang lebih baik, dan membandingkannya dengan beberapa arsitektur komputer lainnya

3. Penggunaan fungsi seleksi yang beragam dan lebih kompleks, terutama fungsi seleksi yang dapat mempertimbangkan adanya preferensi dari pengguna

4. Penggunaan fungsi seleksi yang menggunakan tanggal, dan alokasi waktu setiap tanggalnya agar menghasilkan keadaan yang semakin mendekati dunia nyata.

5. Penggunaan *library* dengan gambar agar dapat digambarkan lebih baik seleksi yang dihasilkan kepada pengguna

REFERENSI

[1] Munir, Rinaldi. Strategi Algoritma. edisi 2018. Bandung:Penerbit ITB

[2] M. S. Rahman and M. Kaykobad, "On Hamiltonian cycles and Hamiltonian paths," Information Processing Letters, vol. 94, no. 1, pp. 37–41, 2005.

[3] Rosen, Kenneth H.*Discrete Mathematics and Its Applications*, seventh Edition.2012.New York:McGraw-Hill.

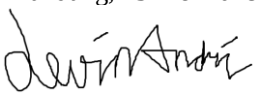
UCAPAN TERIMA KASIH

Terima kasih kepada Sdr. Jason Wiguna yang telah memberi beberapa saran pengembangan ide yang diusulkan oleh penulis.

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, 13 Mei 2018



Kevin Andrian Liwinata