

Penerapan Algoritma Greedy dalam Membantu Penjadwalan Pengerjaan Tugas

Syamil Cholid Abdurrasyid 13519052

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail : syamilca01@gmail.com

Abstract—Di dalam kehidupan manusia, tugas atau suatu pekerjaan adalah kegiatan sehari-hari. Pertumbuhan jumlah manusia semakin hari semakin membuat persaingan dalam kehidupan semakin tinggi untuk dapat bertahan menyambung kehidupan. Terkadang manusia tanpa disadari mengambil dan atau memiliki banyak sekali tugas. Hal ini berlaku baik itu seseorang yang sudah bekerja maupun anak-anak yang masih sekolah atau kuliah. Tentu, banyaknya tugas ini membuat tekanan tersendiri bagi seseorang. Kadang kala sampai ada orang yang tidak kuat hingga akhirnya stres karena banyaknya tugas yang dia miliki. Salah satu strategi menyelesaikan tugas yang banyak adalah dengan membuat jadwal pengerjaan tugas sehingga kita dapat lebih produktif dalam kesehariannya dan juga sebagai bentuk manajemen stres diri. Dalam makalah ini akan dibahas penerapan strategi algoritma greedy sebagai salah satu alternatif untuk membantu menyusun jadwal pengerjaan tugas dengan menggunakan variabel deadline dari tugas dan estimasi waktu yang dibutuhkan untuk menyelesaikan tugas.

Keywords—tugas; stress; jadwal; greedy

I. PENDAHULUAN

Saat ini pandemi COVID-19 sudah berjalan lebih dari satu tahun. Adanya pandemi ini tidak bisa dipungkiri membawa banyak perubahan dalam basis sistem kerja menjadi berbasis daring. Pakar kesehatan dari London School of Economics, Dr. Clare Wenham, menyebutkan bahwa kehidupan normal tidak akan kembali hingga 2024. Diperlukan tiga hingga empat tahun bagi dunia untuk kembali normal. Menurutnya, hingga saat ini pandemi Corona akan terus membuat banyak negara mencatat lonjakan kasus, seiring dengan vaksinasi yang sudah dimulai. Disebutkan, kondisi bebas virus Corona terjadi ketika vaksin COVID-19 sudah diterima seluruh dunia termasuk negara berkembang [1]. Di Indonesia sendiri, tim dari Fakultas Kesehatan Masyarakat (FKM) UI menyebutkan bahwa dengan strategi vaksinasi yang cerdas dan inovatif, estimasi pandemi di Indonesia mulai terkendali adalah di September 2021. [2]

Gaya hidup daring tentu sedikit banyak mempengaruhi gaya hidup dan keseharian orang juga terutama dalam bekerja. Terkadang gaya hidup daring membuat seseorang merasa bisa mengambil suatu job atau tugas lebih banyak. Atau bisa jadi juga pada suatu waktu tugas-tugas wajib dari pekerjaan atau tugas dari sekolah memang sangatlah banyak. Sebenarnya banyaknya tugas ini bisa dianalogikan dengan PR sekolah yang mana sudah banyak risetnya mengenai dampak terlalu

banyaknya PR bagi anak-anak. Dikutip dari hellosehat.com, bahwa Penelitian yang dilakukan oleh sekelompok peneliti Australia menyelidiki hubungan antara waktu yang dihabiskan untuk pekerjaan rumah dengan prestasi akademis siswa. Hasilnya, mereka menemukan jika kebanyakan siswa yang mendapatkan terlalu banyak PR justru akan meningkatkan masalah kesehatan karena kurang tidur, stres, kurang waktu bermain, dan lain sebagainya. Terlalu banyak PR tidak membantu anak-anak mendapatkan nilai bagus di sekolah, namun pada kenyataannya hal ini justru malah membuat nilai ujian mereka anjlok. [3]

Hal tersebut berlaku juga bagi orang dewasa, banyaknya tugas atau pekerjaan akan memberikan tekanan tersendiri bagi seseorang. Masalah-masalah kesehatan bisa timbul dan sangat rentan seseorang menjadi stres apabila manajemen stres nya tidak kuat. Terlalu banyak tugas mengantri terkadang juga membuat seseorang menjadi *blank* dan mudah terdistraksi karena dia bingung harus melakukan apa terlebih dahulu. Misalnya bingung apakah ingin mengejakan yang paling mudah dulu, atau mau mencicil yang susah dulu karena takut tidak selesai, atautkah mengerjakan yang *deadline*-nya lebih dekat terlebih dahulu. Hal itu justru bisa membuat seseorang menjadi kurang produktif dan tidak segera melakukan sesuatu untuk mencicil tugas-tugas yang dia miliki. Dan layaknya *circle of despair*, tugas-tugas menjadi semakin tidak terselesaikan dan semakin memberikan tekanan yang lebih bagi orang tersebut.

Salah satu cara menghadapi tugas yang sangat banyak adalah dengan membuat penjadwalan untuk mengerjakan tugas. Karena tidak semua orang bisa dengan hanya mencatat seluruh tugas kemudian secara naluri mengetahui apa yang harus dia lakukan. Bahkan justru dengan mengetahui daftar seluruh tugas yang harus diselesaikan sangat banyak, dia semakin terasa terbebani. Penjadwalan ini adalah satu langkah lebih maju lagi dari mendaftarkan seluruh tugas.

Penjadwalan untuk pengerjaan tugas bisa bermacam – macam pilihannya. Kita bisa memilih untuk mengerjakan tugas dengan pilihan-pilihan yang telah disinggung sebelumnya seperti dengan memilih tugas yang *deadline*-nya terdekat, tugas yang termudah, ataupun tugas yang paling susah terlebih dahulu. Jika subjektivitas susah atau mudah diganti dengan estimasi waktu yang diperlukan untuk menyelesaikan tugas, maka ditambah dengan waktu *deadline* kita mendapatkan variabel-variabel yang bersifat angka dan dapat dihitung secara

kuantitatif. Kondisi ini memungkinkan adanya persoalan tentang optimasi yang kemudian bisa dilakukan penyusunan menggunakan algoritma Greedy. Salah satu hasil perumusan permasalahan optimasinya adalah dengan berbagai tugas yang tersedia beserta estimasi pengerjaan dan *deadline*-nya masing-masing, bagaimana cara menyusun jadwal sedemikian rupa sehingga didapatkan tugas yang dapat diselesaikan adalah paling banyak. Sebenarnya masih ada perumusan masalah lain dan penggunaan algoritma Greedy dalam penjadwalan ini yang nanti akan disinggung lebih lanjut.

II. LANDASAN TEORI

A. Tugas dan Stres Kerja / Burnout Syndrome

Tugas tentu telah menjadi keseharian kita. Mulai dari sekolah, kuliah, kerja, bahkan hingga dewasa nanti, tugas adalah hal yang tidak terpisahkan dari kehidupan. Tugas sendiri sebenarnya memiliki peran penting bagi manusia, yaitu sebagai *productivity driver*, selain sebagai memenuhi tujuan dari manusia itu sendiri. Jika manusia tidak memiliki tugas sama sekali, tentu tidak bisa kita bayangkan akan seperti apa kehidupan yang dilakukan manusia pada sehari-harinya.

Namun, di lain sisi memang seringkali tugas sendiri adalah suatu beban dan kewajiban bagi manusia yang harus dia selesaikan. Apapun itu, jika seseorang mendapatkan atau harus melakukan sesuatu yang bukan atas keinginannya sendiri, hal tersebut memiliki potensi memberikan suatu beban pada orang tersebut. Dan memang tidak bisa dipungkiri juga, meskipun tugas itu berdampak positif, mayoritas manusia tidak ingin diberikan tugas terutama yang bukan keinginannya. Apalagi ketika kondisi suatu tugas yang dimiliki seseorang telah menumpuk begitu banyak. Tentu hal tersebut menjadi tekanan tersendiri baginya dan bahkan dapat berdampak buruk. Hal – hal yang sering terjadi akibat terlalu berlebihannya pekerjaan antara lain adalah stres dan gangguan psikis lain dan menurunnya kondisi kesehatan tubuh yang biasanya diakibatkan oleh kurangnya istirahat.

Kondisi masalah tubuh yang diakibatkan oleh pekerjaan biasanya disebut stres kerja atau yang biasa juga disebut *burnout* atau *burnout syndrome*. *Burnout syndrome* adalah salah satu kondisi stres yang berhubungan dengan pekerjaan. Itu sebabnya, kondisi kesehatan yang satu ini juga dikenal sebagai *occupational burnout* atau *job burnout*. Menurut WHO sendiri *burnout* termasuk dalam *11th Revision of the International Classification of Diseases (ICD-11)* sebagai *occupational phenomenon* fenomena pekerjaan. *Burnout* tidak diklasifikasikan sebagai kondisi medis. Hal ini dijelaskan dalam bab 'Faktor yang mempengaruhi status kesehatan atau kontak dengan layanan kesehatan' - yang mencakup alasan orang menghubungi layanan kesehatan tetapi tidak digolongkan sebagai penyakit atau kondisi kesehatan. [4]

Dalam ICD-11, *burnout* dijelaskan sebagai sindrom yang dikonseptualisasikan sebagai akibat dari stres kronis di tempat kerja yang belum berhasil dikelola, yang dicirikan oleh tiga hal yaitu perasaan kehabisan energi atau kelelahan, peningkatan *mental distance* dari pekerjaan seseorang, atau perasaan negativisme atau sinisme terkait dengan pekerjaan seseorang, dan pengurangannya efikasi profesional. Burn-out mengacu

secara khusus pada fenomena dalam konteks pekerjaan dan tidak bisa diterapkan untuk menggambarkan pengalaman di bidang kehidupan lain. [4]

B. Algoritma Greedy

Dalam kehidupan sehari-hari, banyak terdapat persoalan yang menuntut pencarian solusi optimum. Persoalan tersebut dinamakan persoalan optimasi (*optimization problems*). Pada persoalan optimasi, kita diberikan sejumlah kendala (*constraint*) dan fungsi optimasi. Solusi yang memenuhi semua kendala adalah solusi layak (*feasible solution*). Solusi layak yang mengoptimalkan fungsi optimasi disebut solusi optimum.

Algoritma greedy memiliki prinsip "*take what you can get now!*". Algoritma greedy merupakan algoritma yang memecahkan masalah secara langkah per langkah. Algoritma greedy ini akan mengambil suatu keputusan yang terbaik dalam setiap langkah pemecahan masalah. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya.

Pada setiap langkah di algoritma greedy dilakukan :

- Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip "*take what you can get now!*")
- Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global. [5]

Algoritma greedy melibatkan pencarian sebuah himpunan bagian, S , dari himpunan kandidat, C ; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimisasi oleh fungsi obyektif.[1]

1) Himpunan Kandidat (C)

Berisi elemen-elemen pembentuk solusi. Contohnya adalah himpunan koin, himpunan job yang akan dikerjakan, himpunan simpul di dalam graf, dan lainlain. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.

2) Himpunan Solusi (S)

Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi adalah himpunan bagian dari himpunan kandidat.

3) Fungsi Solusi

Menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi

4) Fungsi Seleksi (SELEKSI)

Fungsi yang memilih kandidat yang paling memungkinkan emncapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah berikutnya

5) Fungsi Kelayakan (LAYAK)

Fungsi memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

6) Fungsi Objektif

Fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain). [6]

Dengan kata lain, persoalan optimasi yang diselesaikan dengan algoritma Greedy melibatkan pencarian sebuah himpunan bagian S , dari himpunan kandidat C ; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimisasi oleh fungsi objektif.

Skema umum yang digunakan pada kebanyakan algoritma greedy dapat dirumuskan sebagai berikut :

- 1) Inisialisasi S dengan kosong.
- 2) Pilih sebuah kandidat (dengan fungsi seleksi) dari C .
- 3) Kurangi C dengan kandidat yang sudah terpilih dari langkah di atas.
- 4) Periksa apakah kandidat yang dipilih tersebut bersamasama dengan himpunan solusi membentuk solusi yang layak atau feasible (dilakukan oleh fungsi kalayak). Jika ya, masukkan kandidat tersebut ke dalam himpunan solusi; jika tidak, buang kandidat tersebut dan tidak perlu dipertimbangkan lagi.
- 5) Periksa apakah himpunan solusi sudah memberikan solusi yang lengkap (dengan menggunakan fungsi solusi). Jika ya, berhenti (selesai); jika tidak, ulangi lagi dari langkah poin 2. [6]

Adakalanya optimum global belum tentu merupakan solusi optimum (terbaik), tetapi dapat merupakan solusi suboptimum atau pseudo-optimum. Hal ini bisa dijelaskan dari dua faktor berikut :

- Algoritma greedy tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada sebagaimana pada metode *exhaustive search*.
- Pemilihan fungsi seleksi biasanya didasarkan pada fungsi objektif. Terdapat beberapa jenis fungsi objektif yang berbeda-beda oleh karena itu kita harus memilih fungsi yang tepat jika kita ingin algoritma bekerja dengan benar dan menghasilkan solusi yang benar-benar optimum. [5]

Pseudo-code algoritma greedy secara umum :

1) Fungsi predikat SELEKSI

```
function SELEKSI (C : himpunan_kandidat) →
kandidat
{ me-return sebuah kandidat yang dipilih dari
C berdasarkan kriteria yang di pilih }
```

2) Fungsi predikat SOLUSI

```
function SOLUSI (S : himpunan_solusi) →
boolean
{ true jika S adalah Solusi dari persoalan;
sebaliknya false jika S belum menjadi solusi }
```

3) Fungsi predikat LAYAK

```
function LAYAK (S : himpunan_kandidat) →
boolean
{ bernilai true jika S merupakan solusi yang
tidak melanggar kendala; sebaliknya bernilai
false jika S melanggar kendala }
```

4) Algoritma Greedy

```
function greedy (C : himpunan_kandidat) →
himpunan_solusi
{ Mengembalikan solusi dari persoalan
optimasi dengan algoritma greedy
Masukan : himpunan kandidat C
Keluaran : himpunan solusi yang bertipe
himpunan_kandidat }
```

Deklarasi :

x : kandidat
 S : himpunan_solusi

Algoritma :

```
S ← {} {inisialisasi S dengan kosong }
while ( not SOLUSI(S)) and (C ≠ {} ) do
  x ← SELEKSI(C)
  C ← C - {x}
  if LAYAK ( S ∪ {x} ) then
    S ← S ∪ {x}
  endif
endwhile
{ SOLUSI(S) or C = {} }
```

```
if SOLUSI(S) then
  return S
else
  write('tidak ada solusi')
endif
```

III. HASIL DAN PEMBAHASAN

Permasalahan penjadwalan ini sebenarnya adalah bentuk lain atau salah satu bentuk dari permasalahan *Job Scheduling with Deadlines*. Seperti yang sempat disinggung di bagian pendahuluan, bentuk dasar dari perumusan masalah adalah dengan membuat estimasi pengerjaan dari setiap tugas yang

tersedia. Dengan disertai informasi *deadline* dari setiap tugas, maka kita dapat menjadikan masalah penjadwalan ini menjadi masalah yang bisa dikerjakan dengan algoritma greedy. Optimasi yang dilakukan adalah memaksimalkan jumlah tugas yang dapat diselesaikan.

Penggunaan algoritma greedy dalam permasalahan yang dibahas ini memiliki 2 tujuan dengan kondisi yang berbeda :

- 1) Kondisi pertama yaitu ketika daftar tugas adalah suatu tugas pilihan yang dapat kita ambil sehingga optimasinya adalah seperti biasa yaitu bagaimana kita memaksimalkan pengambilan tugas yang mana tentu semua bisa selesai sebelum *deadline*.
- 2) Kondisi yang kedua adalah ketika daftar tugas yang tersedia merupakan tugas telah diterima yang berarti semua tugas haruslah bisa diselesaikan. Dengan begitu, himpunan solusi yang dihasilkan akan memiliki penilaian berbeda yaitu apabila himpunan solusi tidak mencakup semua tugas yang ada, berarti kita akan mengetahui di tugas mana yang diperkirakan tidak akan selesai memenuhi *deadline* sehingga yang dapat disimpulkan adalah pengalokasian waktu estimasi pengerjaan masih belum tepat dan perlu adanya perubahan dengan pertimbangan tertentu.

Data yang akan digunakan dalam makalah ini adalah sebagai berikut :

Tugas	Estimasi Pengerjaan (jam)	Deadline (jam)
1	10	20
2	64	168
3	24	72
4	7	25
5	14	36
6	36	104
7	6	12
8	72	124

Deadline didapatkan dengan cara mengkonversikan waktu *deadline* dalam bentuk *timestamp* menjadi bentuk jam yang merupakan jarak *deadline* terhadap waktu pembuatan jadwal dalam jam. Satuan sebenarnya bisa bebas menggunakan satuan apa tetapi pada kali ini digunakan satuan jam.

Dalam makalah ini, digunakan kondisi permasalahan yang pertama supaya lebih jelas. Fungsi SOLUSI yang digunakan adalah semua tugas yang tersedia telah dicek (himpunan kandidat sudah kosong). Fungsi SELEKSI yang digunakan adalah sesuai strategi greedy nya. Fungsi LAYAK atau kendala yang digunakan adalah menggunakan informasi *deadline* yang dimiliki setiap tugas sebagai berikut (d_i adalah *deadline* ke- i) :

$$\sum_{i=1}^n w_i x_i \leq d_i$$

Selain itu, pada kondisi sebenarnya diperlukan waktu rata-rata istirahat dalam sehari y sehingga setiap akumulasi waktu ($w_i x_i$) melewati kelipatan $24 - y$, akan langsung ditambahkan sebanyak y karena pada kenyataannya kita tidak mungkin bekerja 24 non-stop. Akan tetapi, hal tersebut diabaikan terlebih dahulu agar lebih mudah dalam membahas konsepnya.

Strategi greedy yang digunakan dalam konteks ini adalah tiga macam yaitu *greedy by nearest deadline*, *greedy by least estimated time*, dan *greedy by most estimated time*.

A. Greedy by Nearest Deadline

1) Iterasi ke-1

Akumulasi waktu sebelumnya = 0

Fungsi seleksi mengambil tugas 7.

Karena $0 + 6 < 12$, maka tugas 7 diambil.

Himpunan solusi $S : \{7\}$

2) Iterasi ke-2

Akumulasi waktu sebelumnya = 6

Fungsi seleksi mengambil tugas 1.

Karena $6 + 10 < 20$, maka tugas 1 diambil.

Himpunan solusi $S : \{7, 1\}$

3) Iterasi ke-3

Akumulasi waktu sebelumnya = 16

Fungsi seleksi mengambil tugas 4.

Karena $16 + 7 < 25$, maka tugas 4 diambil.

Himpunan solusi $S : \{7, 1, 4\}$

4) Iterasi ke-4

Akumulasi waktu sebelumnya = 23

Fungsi seleksi mengambil tugas 5.

Karena $23 + 14 > 36$, maka tugas 5 tidak diambil.

Himpunan solusi $S : \{7, 1, 4\}$

5) Iterasi ke-5

Akumulasi waktu sebelumnya = 23

Fungsi seleksi mengambil tugas 3.

Karena $23 + 24 < 72$, maka tugas 3 diambil.

Himpunan solusi $S : \{7, 1, 4, 3\}$

6) Iterasi ke-6

Akumulasi waktu sebelumnya = 47

Fungsi seleksi mengambil tugas 6.

Karena $47 + 36 < 104$, maka tugas 6 diambil.

Himpunan solusi $S : \{7, 1, 4, 3, 6\}$

7) Iterasi ke-7

Akumulasi waktu sebelumnya = 81

Fungsi seleksi mengambil tugas 8.

Karena $81 + 72 > 124$, maka tugas 8 tidak diambil.

Himpunan solusi S : {7, 1, 4, 3, 6}

8) *Iterasi ke-8*

Akumulasi waktu sebelumnya = 81

Fungsi seleksi mengambil tugas 2.

Karena $81 + 64 < 168$, maka tugas 2 diambil.

Himpunan solusi S : {7, 1, 4, 3, 6, 2}

Karena himpunan kandidat telah kosong, iterasi dihentikan dan didapatkan himpunan solusi akhir {7, 1, 4, 3, 6, 2}

Hasil tersebut menunjukkan bahwa kita harus mengambil tugas seperti pada himpunan solusi dan mengerjakan sesuai dengan urutan seperti pada himpunan juga

B. *Greedy by Least Estimated Time*

1) *Iterasi ke-1*

Akumulasi waktu sebelumnya = 0

Fungsi seleksi mengambil tugas 7.

Karena $0 + 6 < 12$, maka tugas 7 diambil.

Himpunan solusi S : {7}

2) *Iterasi ke-2*

Akumulasi waktu sebelumnya = 6

Fungsi seleksi mengambil tugas 4.

Karena $6 + 7 < 25$, maka tugas 4 diambil.

Himpunan solusi S : {7, 4}

3) *Iterasi ke-3*

Akumulasi waktu sebelumnya = 13

Fungsi seleksi mengambil tugas 1.

Karena $13 + 10 > 20$, maka tugas 1 tidak diambil.

Himpunan solusi S : {7, 4}

4) *Iterasi ke-4*

Akumulasi waktu sebelumnya = 13

Fungsi seleksi mengambil tugas 5.

Karena $13 + 14 < 36$, maka tugas 5 diambil.

Himpunan solusi S : {7, 4, 5}

5) *Iterasi ke-5*

Akumulasi waktu sebelumnya = 27

Fungsi seleksi mengambil tugas 3.

Karena $27 + 24 < 72$, maka tugas 3 diambil.

Himpunan solusi S : {7, 4, 5, 3}

6) *Iterasi ke-6*

Akumulasi waktu sebelumnya = 51

Fungsi seleksi mengambil tugas 6.

Karena $51 + 36 < 104$, maka tugas 6 diambil.

Himpunan solusi S : {7, 4, 5, 3, 6}

7) *Iterasi ke-7*

Akumulasi waktu sebelumnya = 87

Fungsi seleksi mengambil tugas 2.

Karena $87 + 64 < 168$, maka tugas 2 diambil.

Himpunan solusi S : {7, 4, 5, 3, 6, 2}

8) *Iterasi ke-8*

Akumulasi waktu sebelumnya = 81

Fungsi seleksi mengambil tugas 8.

Karena $151 + 72 > 124$, maka tugas 8 tidak diambil.

Himpunan solusi S : {7, 4, 5, 3, 6, 2}

Karena himpunan kandidat telah kosong, iterasi dihentikan dan didapatkan himpunan solusi akhir {7, 4, 5, 3, 6, 2}

C. *Greedy by Most Estimated Time*

1) *Iterasi ke-1*

Akumulasi waktu sebelumnya = 0

Fungsi seleksi mengambil tugas 8.

Karena $0 + 72 < 124$, maka tugas 8 diambil.

Himpunan solusi S : {8}

2) *Iterasi ke-2*

Akumulasi waktu sebelumnya = 72

Fungsi seleksi mengambil tugas 2.

Karena $72 + 64 < 168$, maka tugas 2 diambil.

Himpunan solusi S : {8, 2}

3) *Iterasi ke-3*

Akumulasi waktu sebelumnya = 138

Fungsi seleksi mengambil tugas 6.

Karena $138 + 36 > 104$, maka tugas 6 tidak diambil.

Himpunan solusi S : {8, 2}

4) *Iterasi ke-4*

Akumulasi waktu sebelumnya = 138

Fungsi seleksi mengambil tugas 3.

Karena $138 + 24 > 72$, maka tugas 3 tidak diambil.

Himpunan solusi S : {8, 2}

5) Iterasi ke-5

Akumulasi waktu sebelumnya = 138

Fungsi seleksi mengambil tugas 5.

Karena $138 + 14 > 36$, maka tugas 5 tidak diambil.

Himpunan solusi S : {8, 2}

6) Iterasi ke-6

Akumulasi waktu sebelumnya = 138

Fungsi seleksi mengambil tugas 1.

Karena $138 + 10 > 20$, maka tugas 1 tidak diambil.

Himpunan solusi S : {8, 2}

7) Iterasi ke-7

Akumulasi waktu sebelumnya = 138

Fungsi seleksi mengambil tugas 4.

Karena $138 + 7 > 25$, maka tugas 4 tidak diambil.

Himpunan solusi S : {8, 2}

8) Iterasi ke-8

Akumulasi waktu sebelumnya = 138

Fungsi seleksi mengambil tugas 7.

Karena $138 + 6 > 12$, maka tugas 7 tidak diambil.

Himpunan solusi S : {8, 2}

Karena himpunan kandidat telah kosong, iterasi dihentikan dan didapatkan himpunan solusi akhir {8, 2}. Strategi ini mungkin terlihat tidak efektif. Akan tetapi, apabila diasumsikan tugas dengan estimasi waktu terlama memiliki poin / reward paling besar, maka strategi ini bisa untuk diperhitungkan.

IV. KESIMPULAN

Secara umum, strategi memang memiliki kemungkinan yang cukup besar untuk menghasilkan solusi yang bukan solusi optimal. Dalam makalah ini, strategi yang paling baik adalah *greedy by nearest deadline* dan *greedy by least estimated time*. Namun, secara logika penggunaan strategi *greedy by nearest deadline* adalah yang paling aman karena dalam hal ini estimasi pengerjaan tidak mungkin lebih dari *deadline*-nya sehingga algoritma ini juga akan memberikan kemungkinan yang sangat besar untuk menghasilkan solusi yang optimal untuk permasalahan ini.

Penggunaan strategi *greedy* dalam masalah penyusunan jadwal ini perlu pengembangan lebih lanjut untuk membuat produk jadi yang berupa jadwal keseharian. Pertama, seperti yang telah disebutkan tadi, manusia tidak mungkin bekerja 24 jam non-stop sehingga jam istirahat perlu diperhitungkan. Kedua, diperlukan algoritma teknis untuk pengalokasian jadwal untuk tiap periode 24 jamnya. Selain itu, variabel masih bisa ditambahkan sebagai contoh yaitu ditambahkan variabel *reward* atau poin dari menyelesaikan tugas tersebut. Dengan begitu, strategi *greedy* nya dapat lebih bervariasi lagi.

LINK VIDEO DI YOUTUBE

https://www.youtube.com/watch?v=oYRu7u_ylhM&t=3s

UCAPAN TERIMA KASIH

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas terselesaikannya makalah ini karena dengan karunia-Nya lah penulis dapat menyelesaikan makalah yang berjudul "Penerapan Algoritma Greedy dalam Membantu Penjadwalan Pengerjaan Tugas" ini. Makalah ini menjadi salah satu tugas mata kuliah Strategi Algoritma IF2211 pada Semester Genap Tahun Akademik 2020/2021 di Institut Teknologi Bandung.

Terima kasih penulis ucapkan sebanyak-banyaknya kepada:

1. Orang tua yang telah memberikan fasilitas, doa, serta dukungan kepada penulis dalam menyelesaikan makalah ini.
2. Bapak Rila Mandala selaku dosen mata kuliah Strategi Algoritma IF2211 yang telah membimbing dan mengajari penulis selama satu semester ini sehingga penulis mendapatkan ilmu yang sangat bermanfaat.
3. Bapak Dr. Ir. Rinaldi Munir, M.T. selaku salah satu dosen Strategi Algoritma yang mengurus website slide mata kuliah yang telah menjadi acuan belajar utama bagi penulis selama satu semester untuk mata kuliah ini.
4. Teman-teman saya yang telah memberikan dukungan dan semangat serta mengingatkan penulis ketika ada tugas atau ada kuis terkhusus tugas makalah ini.

REFERENSI

- [1] <https://health.detik.com/berita-detikhealth/d-5356455/prediksi-baru-para-pakar-soal-kapan-corona-berakhir-dan-bisa-hidup-true-normal>. Diakses pada tanggal 11 Mei 2021.
- [2] <https://health.detik.com/berita-detikhealth/d-5367826/bukan-10-tahun-tim-fkm-ui-prediksi-pandemi-corona-ri-selesai-september-2021>. Diakses pada tanggal 11 Mei 2021.
- [3] <https://hellosehat.com/parenting/anak-6-sampai-9-tahun/perkembangan-anak/dampak-banyak-pr-untuk-anak/>. Diakses pada tanggal 11 Mei 2021.
- [4] <https://www.who.int/news/item/28-05-2019-burn-out-an-occupational-phenomenon-international-classification-of-diseases#:~:text=%E2%80%9CBurn%2Dout%20is%20a%20syndrome,related%20to%20one's%20job%3B%20and>. Diakses pada tanggal 11 Mei 2021.
- [5] Munir, Rinaldi. Diktat Kuliah IF2211 Strategi Algoritma. Program Studi Teknik Informatika ITB. 2018
- [6] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf). Diakses pada tanggal 11 Mei 2021

PERYATAAN

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.

Yogyakarta, 11 Mei 2021



Syamil Cholid Abdurrasyid 13519052