

Penentuan Aktivitas Mahasiswa Menggunakan Algoritma Greedy

Ilham Prasetyo Wibowo - 13520013
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13520013@std.stei.itb.ac.id

Abstract— Algoritma greedy merupakan salah satu algoritma pemecahan persoalan. Algoritma greedy mencari solusi setiap langkah tanpa memikirkan konsekuensi di masa depan. Contoh persoalan yang bisa diselesaikan dengan algoritma greedy adalah *job-scheduling problem* dan *activity selection problem*. Mahasiswa memiliki beragam aktivitas mulai dari aktivitas yang berhubungan dengan kegiatan perkuliahan, sampai aktivitas mereka pribadi masing-masing. Aktivitas sehari-hari seperti mengerjakan tugas, belajar, bisa dimodelkan menjadi sebuah persoalan penjadwalan dengan tenggat waktu (*job-scheduling problem*) atau persoalan pemilihan aktivitas (*activity selection problem*). Makalah ini akan membahas mengenai persoalan tersebut serta implementasinya menggunakan program komputer.

Keywords—aktivitas,greedy,mahasiswa,penjadwalan

I. PENDAHULUAN

Algoritma Greedy adalah algoritma pemecahan persoalan dengan langkah per langkah. Pada setiap langkah, algoritma Greedy mengambil pilihan yang terbaik pada saat itu tanpa memperhatikan masa depan. Algoritma greedy dapat diaplikasikan dalam banyak hal. Salah satu penggunaan algoritma Greedy adalah untuk penyelesaian masalah optimisasi. Contoh permasalahan yang dapat diselesaikan dengan algoritma greedy adalah permasalahan *job-scheduling problem with deadline problem* atau penjadwalan job dengan tenggat waktu dan persoalan pemilihan aktivitas (*activity selection problem*).

Mahasiswa memiliki banyak aktivitas yang bisa dilakukan, di dalam kampus, maupun di luar kampus. Aktivitas tersebut dapat berupa mengerjakan tugas, bermain gim, berorganisasi dan lain-lain. Sehingga dengan banyaknya aktivitas tersebut mahasiswa dituntut untuk mengatur waktunya dengan baik. Waktu itu sendiri merupakan bagian penting dari kehidupan seseorang. Bahkan, bisa dikatakan kesuksesan seseorang ditentukan dari bagaimana dia menggunakan waktunya.

Pada tahun 2020, dunia termasuk Indonesia terkena pandemi Covid-19 yang menyebabkan mahasiswa untuk belajar dari rumah. Dengan belajar dari rumah, mahasiswa memiliki lebih banyak waktu dari biasanya. Manajemen waktu adalah kemampuan manusia untuk mengatur waktunya untuk berbagai kegiatan dengan efektif dan efisien. Pada makalah ini penulis akan mencoba membuat prioritas atau memilih

aktivitas sehari-hari untuk mahasiswa menggunakan pemodelan persoalan penjadwalan job dengan tenggat waktu, dan persoalan pemilihan aktivitas, yang keduanya akan diselesaikan menggunakan algoritma greedy.

II. DASAR TEORI

A. Algoritma Greedy

Algoritma greedy adalah algoritma yang memecahkan persoalan secara langkah per langkah. Pada algoritma greedy, solusi dipilih pada saat itu juga tanpa mempertimbangkan masa depan. Sehingga algoritma greedy biasanya digunakan untuk permasalahan optimisasi (maksimum atau minimum). Setelah mengambil langkah yang dirasa terbaik pada saat itu, algoritma greedy berharap agar solusi yang dipilih (solusi lokal) juga merupakan solusi global.

B. Elemen-elemen Algoritma Greedy

Algoritma greedy memiliki beberapa elemen yaitu :

1. Himpunan Kandidat

Himpunan kandidat adalah himpunan yang berisi solusi yang mungkin akan dipilih pada setiap langkah.

2. Himpunan Solusi

Himpunan solusi berisi solusi yang sudah terpilih.

3. Fungsi Seleksi

Fungsi untuk memilih kandidat berdasarkan strategi greedy tertentu.

4. Fungsi Solusi

Fungsi untuk menentukan apakah kandidat yang dipilih sudah memberikan solusi.

5. Fungsi Kelayakan

Fungsi yang digunakan untuk memeriksa apakah kandidata yang dipilih dapat dimasukkan ke dalam himpunan solusi.

6. Fungsi Obyektif.

Fungsi obyektif merupakan fungsi untuk memaksimalkan atau meminimumkan (optimasi).

Berikut adalah skema umum algoritma greedy.

```
function greedy(C : himpunan_kandidat) → himpunan_solusi
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy }
Deklarasi
x : kandidat
S : himpunan_solusi

Algoritma:
S ← {} { inisialisasi S dengan kosong }
while (not SOLUSI(S) and (C ≠ {})) do
  x ← SELEKSI(C) { pilih sebuah kandidat dari C }
  C ← C - {x} { buang x dari C karena sudah dipilih }
  if LAYAK(S ∪ {x}) then { x memenuhi kelayakan untuk dimasukkan ke dalam himpunan solusi }
    S ← S ∪ {x} { masukkan x ke dalam himpunan solusi }
  endif
endwhile
{ SOLUSI(S) or C = {} }

if SOLUSI(S) then { solusi sudah lengkap }
  return S
else
  write('tidak ada solusi')
endif
```

Sumber: [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)

Algoritma greedy belum tentu menghasilkan solusi yang optimal. Hal ini karena algoritma greedy tidak beroperasi secara menyeluruh terhadap semua kemungkinan. Fungsi seleksi pada algoritma greedy juga berbeda dapat menghasilkan solusi yang berbeda. Namun, algoritma greedy jauh lebih cepat daripada algoritma pencarian exhaustive search.

C. Penggunaan Algoritma Greedy

Algoritma greedy dapat menyelesaikan beberapa persoalan sebagai berikut.

1. Persoalan penukaran uang
2. Persoalan memilih aktivitas
3. Minimisasi waktu di dalam sistem
4. Perosalan knapsack
5. Penjadwalan job dengan tenggat waktu
6. Pohon merentang minimum
7. Lintasan terpendek
8. Kode Huffman
9. Pecahan Mesir

Salah satu algoritma yang sangat terkenal yang menggunakan strategi greedy adalah algoritma dijkstra. Algoritma dijkstra merupakan algoritma yang dibuat oleh Edsger W. Dijkstra. Algoritma ini merupakan algoritma optimal untuk mencari lintasan terpendek (*shortest path*). Lintasan terpendek dibangun langkah per langkah. Pada dasarnya, strategi greedy pada algoritma ini adalah pada setiap langkah, pilih lintasan berbobot minimum yang menghubungkan simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Lintasan dari simpul asal ke simpul baru haruslah merupakan lintasan yang terpendek diantara semua lintasannya ke simpul-simpul yang belum terpilih.

D. Penjadwalan Job dengan Tenggat Waktu

Penjadwalan job dengan tenggat waktu (*job scheduling with deadline*) adalah sebuah persoalan untuk menentukan job yang dapat dikerjakan dalam satu waktu dengan memaksimalkan profit. Persoalan biasanya diberi sebanyak n buah job, kemudian job hanya bisa diproses satu waktu.

Kemudian kita harus menentukan urutan job yang bisa dikerjakan agar memberikan profit yang maksimal.

Persoalan ini dapat diselesaikan menggunakan algoritma greedy, dengan fungsi obyektif adalah maksimisasi profit.

$$F = \Sigma \text{profit}$$

Solusi layaknya adalah sebuah himpunan yang berisi urutan job sedemikian sehingga setiap job di dalam himpunan selesai dikerjakan sebelum tenggat waktunya.

Untuk perbandingan, persoalan juga dapat diselesaikan menggunakan exhaustive search, dengan mencari semua kemungkinan urutan job dan mencari profit maksimum. Misalkan terdapat empat job dengan profit (p1,p2,p3,p4) = (50,10,15,30) dan deadline (d1,d2,d3,d4) = (2,1,2,1). Pencarian solusi menggunakan algoritma exhaustive search akan menghasilkan barisan job {1,4} dengan total keuntungan 80.

Strategi greedy untuk menyelesaikan permasalahan ini adalah memilih job dengan keuntungan terbesar pada setiap langkah, untuk menaikkan fungsi obyektif. Sehingga solusi akan berbentuk sebagai berikut.

Langkah	J	Keuntungan	Fungsi Kelayakan
0	{}	0	-
1	{1}	50	Layak
2	{4,1}	80	Layak
3	{4,1,3}	-	Tidak layak
4	{4,1,2}	-	Tidak layak

Pada kasus ini, solusi yang ditemukan sama dengan pencarian menggunakan exhaustive search, yaitu {4,1}. Sehingga solusi merupakan solusi lokal sekaligus solusi global. Pseudocode algoritma penjadwalan ini adalah sebagai berikut.

```
function JobSchedulling(d[1..n] : array of integer, p[1..n] : array of integer) → himpunan_job
{ Menghasilkan barisan job yang akan diproses oleh mesin }
```

```
Deklarasi
i, k : integer
J : himpunan_job { solusi }
```

```
Algoritma
J ← {}
for i ← 1 to n do
  k ← job yang mempunyai profit terbesar
  if (semua job di dalam J ∪ {k} layak) then
    J ← J ∪ {k}
  endif
endfor
return J
```

Sumber: [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)

E. Persoalan Pemilihan Aktivitas

Permasalahan pemilihan aktivitas merupakan persoalan mengenai pemilihan aktivitas untuk menggunakan *resource* yang hanya bisa digunakan satu aktivitas tiap waktu. Persoalan pemilihan aktivitas biasanya diberikan n aktivitas dengan waktu mulai (si) dan waktu selesai(fi). Solusi dari persoalan tersebut adalah mencari himpunan aktivitas terbanyak yang bisa dilakukan.

Berikut adalah contoh dari persoalan pemilihan aktivitas.

No	Start	Finish
1	1	4
2	3	5
3	4	6
4	5	7
5	3	8
6	7	9
7	10	11
8	8	12
9	8	13
10	2	14
11	13	15

Penyelesaian menggunakan exhaustive search dapat dilakukan dengan menentukan semua himpunan bagian (*subset*) dengan n aktivitas. Kemudian evaluasi himpunan bagian aktivitas yang kompatibel dan pilih himpunan yang memiliki jumlah aktivitas terbanyak.

Skema umum penyelesaian persoalan aktivitas menggunakan algoritma greedy adalah sebagai berikut.

```
function Greedy-Activity-Selector( $s_1, s_2, \dots, s_n$ : integer,  $f_1, f_2, \dots, f_n$ : integer) → set of integer
{ Asumsi: aktivitas sudah diurut terlebih dahulu berdasarkan waktu selesai:  $f_1 \leq f_2 \leq \dots \leq f_n$  }
Deklarasi
   $i, j, n$ : integer
  A: set of integer
Algoritma:
   $n \leftarrow \text{length}(s)$ 
  A ← {1} { aktivitas nomor 1 selalu terpilih }
  j ← 1
  for i ← 2 to n do
    if  $s_i \geq f_j$  then
      A ← A ∪ {i}
      j ← i
    endif
  endif
```

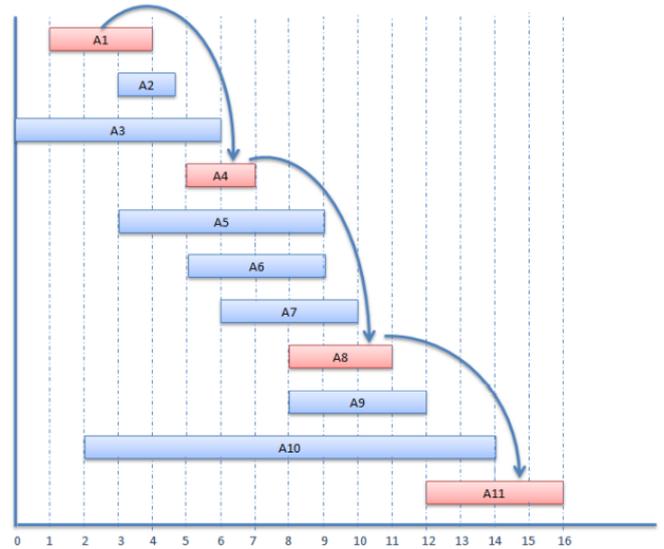
Sumber:

[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)

Penyelesaian menggunakan algoritma greedy dengan mengurutkan terlebih dahulu semua aktivitas berdasarkan waktu selesainya. Kemudian pada setiap langkah, pilih aktivitas yang waktu mulainya lebih besar atau sama dengan waktu selesai aktivitas yang sudah dipilih sebelumnya. Dari persoalan sebelumnya, solusi yang diperoleh adalah 1, 3, 6, 7, dan 11 (5 aktivitas).

No	Start	Finish
1	1	4
2	3	5
3	4	6
4	5	7
5	3	8
6	7	9
7	10	11
8	8	12
9	8	13
10	2	14
11	13	15

Berikut adalah ilustrasi dari penyelesaian persoalan menggunakan algoritma greedy.



Sumber

[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)

F. Aktivitas Mahasiswa

Mahasiswa adalah seseorang yang sedang dalam proses menimba ilmu ataupun belajar dan terdaftar sedang menjalani pendidikan pada salah satu bentuk perguruan tinggi yang terdiri dari akademik, politeknik, sekolah tinggi, institut dan universitas (Hartaji, 2012: 5).

Aktivitas yang dilakukan oleh mahasiswa dapat berupa aktivitas yang berhubungan dengan perkuliahan, maupun tidak. Aktivitas berhubungan dengan perkuliahan dapat meliputi mengerjakan tugas, kerja kelompok, belajar ujian, dan lain sebagainya. Sedangkan aktivitas lainnya seperti bermain, membaca buku, dan lain sebagainya.

III. PENENTUAN AKTIVITAS MAHASISWA MENGGUNAKAN ALGORITMA GREEDY

Penentuan aktivitas mahasiswa akan menggunakan algoritma greedy. Selain itu, akan digunakan dua persoalan untuk merepresentasikan aktivitas mahasiswa. Persoalan pertama adalah dengan *job-scheduling problem*, sedangkan persoalan kedua adalah *activity selection problem*. Kedua persoalan akan dibuat program untuk memecahkannya. Program akan ditulis menggunakan bahasa Python, mengikuti *pseudocode* yang sudah tertera pada dasar teori dengan modifikasi untuk masukan dan luaran.

A. Menggunakan Pemodelan Job-scheduling Problem

Job-scheduling problem memiliki tenggat waktu dan keuntungan. Pada masalah ini, tenggat waktu akan tetap sebagai tenggat waktu, sedangkan keuntungan akan diubah sebagai poin. Poin digunakan untuk menguku prioritas sebuah aktivitas, misal aktivitas dengan poin 100, akan didahulukan daripada aktivitas dengan poin 10.

Dalam penyelesaian masalah ini, akan digunakan pemisalan kumpulan aktivitas sebagai berikut.

No	Deskripsi Aktivitas	Tenggat Waktu	Poin
1	Mengerjakan Tugas Besar 1	3	100
2	Menonton film Avengers : Endgame	2	20
3	Bermain Genshin Impact	3	15
4	Mengerjakan Tugas Kecil 3	1	85
5	Nongkrong	5	40
6	Belajar UAS	5	60
7	Belajar Kuis	6	80

Penyelesaian menggunakan algoritma greedy akan mencari aktivitas dengan poin terbesar terlebih dahulu. Sehingga kegiatan “Mengerjakan Tugas Besar 1” akan dimasukkan dan layak (tidak ada tenggat waktu yang bertabrakan). Setelah itu, akan diambil kegiatan “Mengerjakan Tugas Kecil 1”. Kemudian akan diambil kegiatan kedua yaitu “Belajar Kuis”, dan dimasukkan kedalam hasil. Selanjutnya “Belajar UAS”, setelah itu akan diambil “Nongkrong”, tetapi kegiatan tersebut bertabrakan deadlinenya dengan “Belajar UAS” sehingga “Nongkrong” tidak dimasukkan ke dalam himpunan solusi.

Tabel penyelesaian dari permasalahan tersebut adalah sebagai berikut.

Langkah	J	Keuntungan	Fungsi Kelayakan
0	{}	0	-
1	{1}	100	Layak
2	{1,4}	185	Layak
3	{1,4,7}	265	Layak
4	{1,4,7,6}	325	Layak
5	{1,4,7,6,5}	375	Tidak layak
6	{1,4,7,6,2}	345	Layak
7	{1,4,7,6,2,3}	360	Tidak layak

Dapat disimpulkan bahwa solusinya adalah kegiatan nomor 1,4,7,6,2,3. Yaitu mengerjakan tugas besar 1, mengerjakan tugas kecil 3, belajar kuis, belajar UAS, dan menonton film Avengers:Endgame.

Untuk memastikan algoritma, akan dibuat sebuah program dengan bahasa python. Program akan memproses list tenggat waktu dan keuntungan. Kemudian menampilkan aktivitas yang seharusnya dilakukan oleh mahasiswa. Source code dari program adalah sebagai berikut.

```
def layak(solution, deadline) :
    if (deadline in solution):
        return False
    else :
        return True
```

```
#asumsi profits sudah terurut dari besar ke kecil
```

```
def jobshced(deadlines, profits):
    j = []
    jd = []
    indices = []
    for i in range(len(deadlines)):
        if (layak(jd, deadlines[i])) :
            j.append(profits[i])
            jd.append(deadlines[i])
            indices.append(i)

    return j, jd, indices
```

```
aktivitas = ["Mengerjakan Tugas Besar 1", "Mengerjakan Tugas Kecil 3", "Belajar Kuis", "Belajar UAS", "Nongkrong", "Menonton film Avengers : Endgame", "Bermain Genshin Impact", ]
```

```
dl = [3,1,6,5,5,2,3]
prft = [100,85,80,60,40,20,15]
```

```
print("Aktivitas yang mungkin dilakukan adalah : ")
deadline,profit,index = jobshced(dl,prft)
for i in index:
    print(aktivitas[i])
```

```
Luaran program :
Aktivitas yang mungkin dilakukan adalah :
Mengerjakan Tugas Besar 1
Mengerjakan Tugas Kecil 3
Belajar Kuis
Belajar UAS
Menonton film Avengers : Endgame
```

Sumber : dokumentasi pribadi

Dapat disimpulkan bahwa aktivitas yang dipilih dengan program yang ditulis dengan aktivitas dari pencarian solusi secara manual adalah sama.

B. Menggunakan Activity Selection Problem

Menggunakan activity selection problem akan membuat mahasiswa mengerjakan aktivitas sebanyak mungkin tanpa melihat prioritas dari aktivitas tersebut. Misalkan kegiatan pemodelan kegiatan mahasiswa adalah sebagai berikut.

No	Aktivitas	Mulai	Selesai
----	-----------	-------	---------

1	Bermain Genshin Impact	3	4
2	Berolahraga	5	6
3	Tidur	6	7
4	Mencicil Tugas Kuliah	5	8
5	Makan	4	9
6	Membaca buku	7	10
7	Berorganisasi	8	14
8	Berdemo	11	16
9	Bermain Instagram	12	17
10	Belajar Web Development	19	22
11	Belajar Artificial Intelligence	20	24

```
indices = activityselector(starts,ends)
print("Aktivitas : ")
for i in indices:
    print(lines[i])
```

Program menghasilkan luaran :

```
Aktivitas :
Berolahraga
Tidur
Membaca buku
Berdemo
Belajar Web Development
```

Sumber : dokumentasi pribadi

Dapat disimpulkan bahwa aktivitas yang dipilih dengan program yang ditulis dengan aktivitas dari pencarian solusi secara manual adalah sama.

KESIMPULAN

Pemilihan aktivitas mahasiswa sehari-hari dapat menggunakan algoritma greedy. Pemodelan permasalahan dapat menggunakan *job-scheduling problem*, ataupun *activity selection problem*. Namun, pada activity selection problem tidak bisa memasukkan prioritas dari kegiatan yang ingin dilakukan. Sehingga algoritma akan mencari aktivitas sebanyak mungkin yang bisa dilakukan dan tidak berbenturan dengan aktivitas lainnya. Sedangkan menggunakan *job-scheduling algorithm*, dapat dituliskan prioritas dari aktivitas yang ingin dilakukan, sehingga algoritma akan mencari aktivitas dengan prioritas tertinggi terlebih dahulu. Kemudian akan mencari aktivitas dengan prioritas dibawahnya kecuali tenggat waktunya bertabrakan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih dan puji syukur terhadap Tuhan Yang Maha Esa atas rahmatnya, penulis bisa menyelesaikan makalah ini dengan tepat waktu. Penulis juga mengucapkan terima kasih kepada kedua orang tua yang sudah memberikan dukungan. Penulis juga mengucapkan terima kasih kepada Ibu Masayu Leylia Khodra selaku dosen pengajar mata kuliah Strategi Algoritma semester 2 tahun 2021/2022 .

REFERENCES

- [1] Dyla Fajhriani. N, "Manajemen Waktu Belajar Di Perguruan Tinggi Pada Masa Pandemi Covid-19". Journal of Islamic Educational Management, 2020.
- [2] Hartaji, Damar Adi. "Motivasi Berprestasi Pada Mahasiswa Yang Berkuliah Dengan Jurusan Pilihan Orang Tua". Fakultas Psikologi Universitas Gunadarma.
- [3] [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 21 Mei 2022
- [4] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag2.pdf) . Diakses pada 21 Mei 2022
- [5] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-\(2022\)-Bag3.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-(2022)-Bag3.pdf) . Diakses pada 21 Mei 2022

Diakses pada 21 Mei 2022

Aktivitas yang masuk dalam solusi adalah aktivitas nomor 1. Kemudian akan dicari aktivitas selanjutnya yang memiliki waktu mulai lebih dari atau sama dengan 4, ditemukan aktivitas nomor 2. Proses dilanjutkan sampai diperoleh hasil himpunan aktivitas {1,2,5,7,9}. Dari hasil tersebut diperoleh aktivitas yang dapat dilakukan adalah berolahraga, tidur, membaca buku, berdemo, dan belajar web development.

Langkah	J
0	{}
1	{1}
2	{1,2}
3	{1,2,5}
4	{1,2,5,7}
5	{1,2,5,7,9}

Sama halnya seperti pada job-scheduling problem, akan dibuat program untuk mencari aktivitas menggunakan bahasa python. Program yang dibuat adalah sebagai berikut.

```
#aktivitas sudah terurut berdasarkan waktu selesai
def activityselector (starts, ends):
    n = len(starts)
    a = [1]
    j = 1
    i = 1
    for i in range(n):
        if (starts[i] >= ends[j]):
            a.append(i)
            j = i
    return a
starts = [3,5,6,5,4,7,8,11,12,19,20]
ends = [4,6,7,8,9,10,14,16,17,22,24]
with open("aktivitas.txt") as file:
    lines = file.readlines()
    lines = [line.rstrip() for line in lines]
```

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, 21 Mei 2022

A handwritten signature in black ink, consisting of a large, stylized 'I' followed by 'P' and 'W' with a horizontal line extending to the left.

Ilham Prastetyo Wibowo 13520013