

Perbandingan Pengaplikasian Algoritma Brute Force dengan Algoritma Greedy pada Penentuan Prioritas Pengerjaan Tugas-Tugas di Lingkungan Informatika ITB

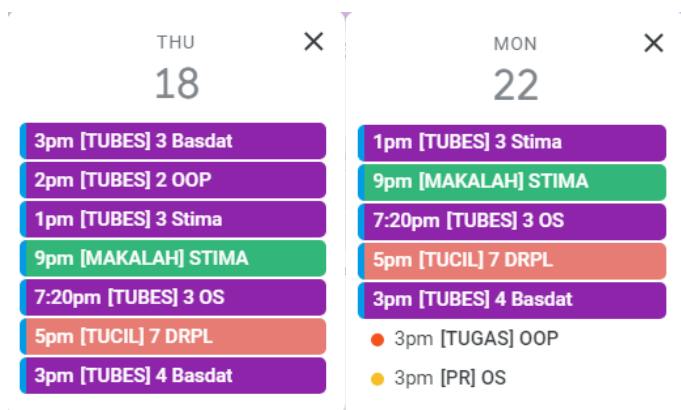
Vincent Budianto - 13517137
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517137@std.stei.itb.ac.id

Abstrak—Algoritma *brute force* dan algoritma *greedy* merupakan strategi-strategi pendekatan yang diajarkan pada mata kuliah IF2211 – Strategi Algoritma ITB. Algoritma *brute force* merupakan salah satu strategi pendekatan yang paling mudah diimplementasikan namun membutuhkan waktu yang lama dalam penyelesaiannya. Sedangkan algoritma *greedy* merupakan salah satu strategi pendekatan yang paling populer untuk memecahkan persoalan optimasi. Makalah ini akan menjelaskan sedikit perbandingan mengenai pengaplikasian algoritma *brute force* dengan algoritma *greedy* untuk menentukan prioritas pengerjaan tugas-tugas di lingkungan informatika ITB.

Kata Kunci—Algoritma *brute force*, algoritma *greedy*, lingkungan informatika ITB, penentuan prioritas, tugas.

I. PENDAHULUAN

Tidak bisa dipungkiri lagi bahwa kehidupan mahasiswa di labtek V ITB itu disertai banyak tugas-tugas. Di lingkungan informatika ITB, tugas-tugas yang diberikan dapat diklasifikasikan menjadi 3 yaitu pekerjaan rumah, tugas kecil dan tugas besar.



Picture 1 Jadwal Tugas-Tugas di Lingkungan Informatika ITB

Sumber:

<https://calendar.google.com/calendar/b/1/r/month/2019/3/1?tab=rc&pli=1&t=AKUaPmZIADw9KEVblacWCC28x7q5pZA4yf4HFuWFCQ>

[UX5asACS7_VTroTW_7GWF1Jko961vc-0aghh_t4Bs3rKOLb-dnLKKWkO%3D%3D](https://www.researchgate.net/publication/332111111) diakses pada 24 April 2019

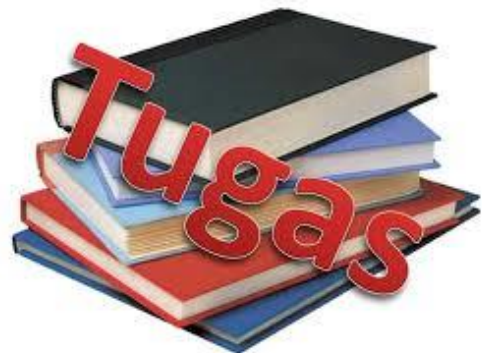
Pekerjaan rumah biasanya diberikan dalam bentuk soal-soal dan biasanya dikerjakan secara perorangan. Tugas kecil biasanya diberikan dalam bentuk suatu permasalahan yang harus dipecahkan dengan membuat suatu program dengan spesifikasi yang telah ditentukan dan biasanya dikerjakan secara perorangan atau berkelompok antara dua atau tiga orang. Tugas besar hampir mirip dengan tugas kecil namun memiliki lingkup spesifikasi yang lebih luas dan biasanya dikerjakan secara berkelompok antara tiga hingga lima orang.

II. DASAR TEORI

A. Tugas

1. Definisi Tugas

Tugas merupakan salah satu hal yang wajib dikerjakan oleh para mahasiswa. Namun terkadang banyaknya tugas membuat mahasiswa cukup kesulitan menyelesaikan semua tugas-tugas yang ada sehingga harus menggunakan banyak cara seperti menentukan prioritas untuk tugas-tugas yang ada.



Picture 2 Ilustrasi Tugas

Sumber: <http://4.bp.blogspot.com/>

2. Parameter Tugas

Untuk menentukan prioritas tugas apa yang harus diprioritaskan tentu dibutuhkan parameter-parameter untuk membandingkannya. Berikut parameter-parameter perbandingan untuk menentukan prioritas suatu tugas:

- Waktu pengerjaan
Waktu tersisa hingga batas pengumpulan tugas.
- Satuan kredit semester (SKS) mata kuliah tugas yang bersangkutan
Beban yang harus dilaksanakan oleh mahasiswa.
- Jenis tugas
Jenis tugas yang diberikan dapat berupa pekerjaan rumah (PR), tugas kecil (Tucil), dan tugas besar (Tubes).
- Tingkat kesulitan
Derajat kesulitan suatu tugas. dapat diklasifikasikan menjadi mudah, cukup mudah, biasa, sulit, dan sangat sulit.

B. Algoritma Brute Force

1. Definisi Algoritma Brute Force

Algoritma *brute force* merupakan strategi pendekatan yang *straightforward* untuk memecahkan suatu permasalahan. Algoritma ini biasanya didasarkan pada pernyataan yang ada di persoalan dan definisi konsep yang digunakan persoalan. Algoritma *brute force* biasa disebut juga sebagai algoritma naif.

Algoritma *brute force* memecahkan permasalahan dengan sangat sederhana, langsung dan jelas. Oleh karena itu algoritma *brute force* memiliki prinsip “*Just do it!*” atau “*Just solve it!*”.

- Algoritma *brute force* biasanya tidak “cerdas” dan efisien karena membutuhkan algoritma ini membutuhkan jumlah komputasi yang banyak dan waktu *runtime* yang lama untuk menyelesaikan suatu persoalan.

2. Kelebihan Algoritma Brute Force

- Dapat digunakan untuk memecahkan hampir semua masalah.
- Sederhana dan mudah untuk dipahami.
- Menghasilkan algoritma yang layak untuk beberapa masalah.
- Menghasilkan algoritma yang baku untuk tugas-tugas komputasi.

3. Kekurangan Algoritma Brute Force

- Jarang menghasilkan algoritma yang efektif.

- Beberapa algoritma memakan waktu yang sangat lama sehingga tidak dapat diaplikasikan.
- Kurang konstruktif jika dibandingkan dengan strategi pendekatan yang lain.

C. Algoritma Greedy

1. Definisi Algoritma Greedy

Algoritma *greedy* merupakan strategi pendekatan yang paling populer untuk memecahkan suatu persoalan optimasi (persoalan mencari solusi optimum). Persoalan optimasi sendiri dipecah menjadi dua yaitu persoalan optimasi yang mencari solusi maksimum, maksimasi, dan persoalan optimasi yang mencari solusi minimum, minimasi.

Algoritma *greedy* memiliki prinsip “*Take what you can get now!*”. Untuk menyelesaikan suatu persoalan, algoritma *greedy* membentuk solusi langkah per langkah (step by step). Oleh karena itu, pada setiap langkah harus diambil keputusan terbaik dalam menentukan pilihan melalui cara membuat pilihan optimum lokal dengan harapan pada langkah-langkah berikutnya dapat mencapai solusi optimum global.

Pada setiap langkah, algoritma *greedy* mengambil pilihan terbaik yang tersedia pada saat itu tanpa memperhatikan konsekuensi ke depannya. Oleh sebab itu, solusi akhir yang dihasilkan oleh algoritma *greedy* tidak selalu berupa solusi optimum global namun masih mendekati solusi optimum global.

2. Elemen-Elemen Algoritma Greedy

- Himpunan kandidat
Kumpulan kandidat solusi.
- Himpunan solusi
Kumpulan solusi.
- Fungsi seleksi (*Selection function*)
Digunakan untuk menentukan kandidat terbaik untuk dimasukkan ke himpunan solusi.
- Fungsi kelayakan (*feasible*)
Digunakan untuk menentukan apakah suatu kandidat dapat digunakan berkontribusi pada solusi.
- Fungsi obyektif
Memberikan nilai pada solusi atau solusi parsial.

```

function greedy(input C: himpunan_kandidat) → himpunan_kandidat
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy
Masukan: himpunan kandidat C
Keluaran: himpunan solusi yang bertipe himpunan_kandidat
}
Deklarasi
x : kandidat
S : himpunan_kandidat
Algoritma:
S ← {} (inisialisasi S dengan kosong)
while (not SOLUSI(S) and (C ≠ {})) do
x ← SELEKSI(C) (pilih sebuah kandidat dari C)
C ← C - {x} (elemen himpunan kandidat berkurang satu)
if LAYAR(S ∪ {x}) then
S ← S ∪ {x}
endif
endwhile
(SOLUSI(S) or C = {})
if SOLUSI(S) then
return S
else
write('tidak ada solusi')
endif

```

Picture 3 Skema Umum Algoritma Greedy

Sumber:

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Algoritma-Greedy-\(2019\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Algoritma-Greedy-(2019).pdf) diakses pada 25 April 2019

3. Kelebihan Algoritma Greedy

- Fast response.
- Tidak mempertimbangkan konsekuensi dari apa yang diputuskan.

4. Kekurangan Algoritma Greedy

- Hasil akhir yang tidak selalu optimum.
- Konsekuensi pemilihan pada suatu langkah tidak dapat dipertanggung jawabkan secara penuh.

III. PENGAPLIKASIAN ALGORITMA BRUTE FORCE DAN ALGORITMA GREEDY

A. Tipe Data yang Dibutuhkan

Untuk pengaplikasian algoritma brute force dan algoritma greedy, program membutuhkan beberapa tipe data sebagai data untuk diproses. Tipe-tipe data yang dibutuhkan akan dijelaskan di bawah ini:

- List yang digunakan sebagai data awal
 - {'Name': 'Nama Tugas', 'Waktu': Jumlah Hari Menuju Deadline, 'SKS': Jumlah Beban SKS Tugas Bersangkutan, 'Jenis': Jenis Tugas, 'Kesulitan': Tingkat Kesulitan}, ...]
 - ❖ Jika 'Jenis' bernilai 1 maka tugas yang bersangkutan adalah pekerjaan rumah (PR).
 - ❖ Jika 'Jenis' bernilai 2 maka tugas yang bersangkutan adalah tugas kecil (Tucil).
 - ❖ Jika 'Jenis' bernilai 3 maka tugas yang bersangkutan adalah tugas besar (Tubes).
- List yang akan digunakan oleh program
 - ['Jumlah Hari Menuju Deadline', 'Waktu yang Dibutuhkan untuk Pengerjaan', 'Nama Tugas', 'Poin']
 - ❖ Rumus yang digunakan untuk menghitung 'Waktu yang Dibutuhkan untuk Pengerjaan' adalah $(C * 'Jenis' * 'Kesulitan')$ dengan nilai C bergantung pada jenis tugas yang bersangkutan (0.7 untuk pekerjaan rumah (PR), 0.85 untuk tugas kecil

(Tucil) dan 1 untuk tugas besar (Tubes)). Namun jika hasil perhitungan melebihi jumlah hari menuju deadline maka nilai 'Waktu yang Dibutuhkan untuk Pengerjaan' diubah menjadi sama dengan jumlah hari menuju deadline.

❖ Rumus yang digunakan untuk menghitung nilai 'Poin' adalah $(0.3 * 'SKS' + 0.3 * 'Jenis' + 0.4 * 'Kesulitan')$.

- Variabel waktu
Variabel yang dibutuhkan sebagai penanda waktu yang dimiliki untuk mengerjakan tugas.

B. Asumsi yang Digunakan

Program penentu prioritas tugas ini menggunakan asumsi-asumsi berikut:

1. Pengerjaan setiap tugas membutuhkan waktu sebanyak 1 hari terlepas dari seberapa sulit tugas tersebut.
2. Urutan pemberian tugas sudah terurut dari awal.

C. Algoritma Brute Force

Untuk persoalan *scheduling* kali ini, strategi pendekatan *brute force* yang dipakai adalah mengerjakan tugas yang telah diberikan terlebih dahulu sebelum melanjutkan ke tugas berikutnya.

Fungsi dibawah bertujuan untuk menghasilkan tugas-tugas yang dapat dikerjakan selama waktu masih mencukupi dimulai dari tugas yang diberikan pertama kali. Jika suatu tugas membutuhkan waktu pengerjaan yang lebih besar dari waktu tersisa yang dimiliki maka tugas tersebut dilompati dan program akan mengecek tugas berikutnya.

```

def bruteforce(tugas, waktu):
    global totalBF

    count = 1
    totalBF = 0

    for i in range(len(tugas)):
        waktu -= tugas[i][2]

        if (waktu >= 0):
            print(count, end = ': ')
            print(tugas[i][3])
            totalBF += (tugas[i][0])
            count += 1
        else:
            waktu += tugas[i][2]

    print()
    print('Poin total:',
          round(totalBF, 2))

```

Table 1 Fungsi Brute Force

D. Algoritma Greedy

Untuk persoalan *scheduling* kali ini, strategi pendekatan *greedy* yang dipakai adalah mengurutkan secara mengecil tugas-tugas yang ada berdasarkan poin yang dimiliki oleh masing-masing tugas.

Fungsi dibawah bertujuan untuk menghasilkan tugas-tugas yang dapat dikerjakan selama waktu masih mencukupi dimulai dari tugas yang diberikan pertama kali. Jika suatu tugas membutuhkan waktu pengerjaan yang lebih besar dari waktu tersisa yang dimiliki maka tugas tersebut dilompati dan program akan mengecek tugas berikutnya.

```
def greedy(tugas, waktu):
    global totalG

    count = 1
    totalG = 0

    skor = []

    for i in range(len(tugas)):
        skor.append(tugas[i][0])

    x = sorted(range(len(tugas)),
               key=lambda k:skor[k],
               reverse = True)

    for j in range(len(skor)):
        waktu -= tugas[x[j]][2]

        if (waktu >= 0):
            print(count, end = ': ')
            print(tugas[x[j]][3])
            totalG += (tugas[x[j]][0])
            count += 1
        else:
            waktu += tugas[x[j]][2]

    print()
    print('Poin total:',
          round(totalG, 2))
```

Table 2 Fungsi Greedy

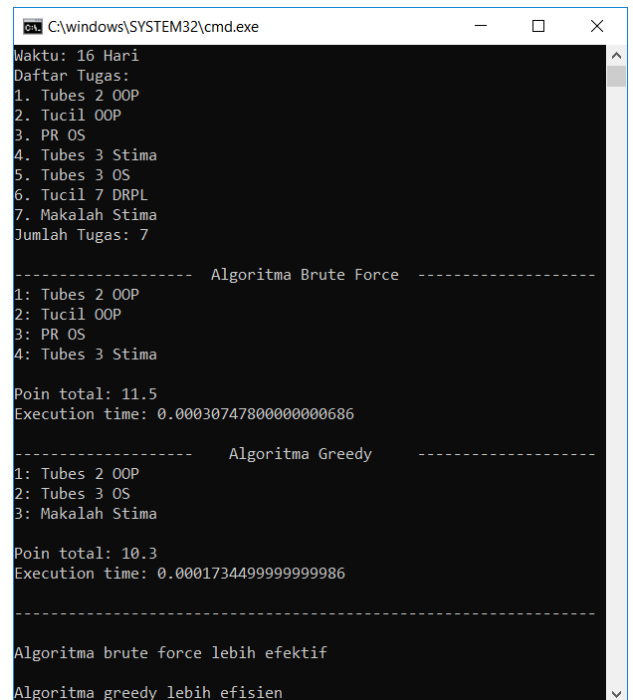
E. Perbandingan Algoritma Brute Force Dengan Algoritma Greedy

Perbedaan algoritma *brute force* dengan algoritma *greedy* dapat dilihat dengan jelas pada kedua fungsi di atas. Pada fungsi yang menggunakan strategi pendekatan algoritma *brute force*, program langsung mengiterasi satu persatu *list* tugas. Sedangkan pada fungsi yang menggunakan strategi pendekatan algoritma *greedy*, program mengurutkan tugas-tugas berdasarkan poinnya dahulu baru diiterasi satu persatu sesuai urutan hasil pengurutan poin.

F. Hasil Uji Satu

```
data = [{ 'Name': 'Tubes 2 OOP',
          'Waktu': 3, 'SKS': 3,
          'Jenis': 3, 'Kesulitan': 5},
        { 'Name': 'Tucil OOP',
          'Waktu': 4, 'SKS': 3,
          'Jenis': 2, 'Kesulitan': 3},
        { 'Name': 'PR OS',
          'Waktu': 4, 'SKS': 3,
          'Jenis': 1, 'Kesulitan': 2},
        { 'Name': 'Tubes 3 Stima',
          'Waktu': 4, 'SKS': 3,
          'Jenis': 3, 'Kesulitan': 3},
        { 'Name': 'Tubes 3 OS',
          'Waktu': 6, 'SKS': 3,
          'Jenis': 3, 'Kesulitan': 4},
        { 'Name': 'Tucil 7 DRPL',
          'Waktu': 6, 'SKS': 2,
          'Jenis': 2, 'Kesulitan': 4},
        { 'Name': 'Makalah Stima',
          'Waktu': 8, 'SKS': 3,
          'Jenis': 2, 'Kesulitan': 4}]
```

Table 3 Data Uji 1



```
C:\windows\SYSTEM32\cmd.exe
Waktu: 16 Hari
Daftar Tugas:
1. Tubes 2 OOP
2. Tucil OOP
3. PR OS
4. Tubes 3 Stima
5. Tubes 3 OS
6. Tucil 7 DRPL
7. Makalah Stima
Jumlah Tugas: 7

----- Algoritma Brute Force -----
1: Tubes 2 OOP
2: Tucil OOP
3: PR OS
4: Tubes 3 Stima

Poin total: 11.5
Execution time: 0.0003074780000000686

----- Algoritma Greedy -----
1: Tubes 2 OOP
2: Tubes 3 OS
3: Makalah Stima

Poin total: 10.3
Execution time: 0.0001734499999999986

-----
Algoritma brute force lebih efektif
Algoritma greedy lebih efisien
```

Picture 4 Hasil Uji Satu

G. Hasil Uji Dua

```
data = [{ 'Name': 'Tucil OOP',
          'Waktu': 4, 'SKS': 3,
          'Jenis': 2, 'Kesulitan': 3},
        { 'Name': 'PR OS',
          'Waktu': 4, 'SKS': 3,
          'Jenis': 1, 'Kesulitan': 2},
        { 'Name': 'Tubes 3 Stima',
          'Waktu': 4, 'SKS': 3,
          'Jenis': 3, 'Kesulitan': 3},
```

```
{'Name': 'Tubes 3 OS',
  'Waktu': 6, 'SKS': 3,
  'Jenis': 3, 'Kesulitan': 4},
{'Name': 'Tucil 7 DRPL',
  'Waktu': 6, 'SKS': 2,
  'Jenis': 2, 'Kesulitan': 4},
{'Name': 'Makalah Stima',
  'Waktu': 8, 'SKS': 3,
  'Jenis': 2, 'Kesulitan': 4},
{'Name': 'Tubes 4 Basdat',
  'Waktu': 8, 'SKS': 3,
  'Jenis': 3, 'Kesulitan': 5}]
```

Table 4 Data Uji 2

```
C:\windows\SYSTEM32\cmd.exe
Waktu: 16 Hari
Daftar Tugas:
1. Tucil OOP
2. PR OS
3. Tubes 3 Stima
4. Tubes 3 OS
5. Tucil 7 DRPL
6. Makalah Stima
7. Tubes 4 Basdat
Jumlah Tugas: 7

----- Algoritma Brute Force -----
1: Tucil OOP
2: PR OS
3: Tubes 3 Stima
4: Tubes 3 OS

Poin total: 11.1
Execution time: 0.0004563480000000225

----- Algoritma Greedy -----
1: Tubes 4 Basdat
2: Tubes 3 OS
3: PR OS

Poin total: 9.2
Execution time: 0.0001924640000000033

-----
Algoritma brute force lebih efektif
Algoritma greedy lebih efisien
```

Picture 5 Hasil Uji Dua

H. Hasil Uji Tiga

```
data = [{'Name': 'PR Probstat',
  'Waktu': 7, 'SKS': 3,
  'Jenis': 1, 'Kesulitan': 2},
{'Name': 'Tubes 2 OS',
  'Waktu': 7, 'SKS': 3,
  'Jenis': 3, 'Kesulitan': 5},
{'Name': 'Tucil 3 Stima',
  'Waktu': 7, 'SKS': 3,
  'Jenis': 2, 'Kesulitan': 2},
{'Name': 'Tubes 3 Basdat',
  'Waktu': 8, 'SKS': 3,
  'Jenis': 3, 'Kesulitan': 3},
{'Name': 'Tucil 6 DRPL',
  'Waktu': 9, 'SKS': 2,
  'Jenis': 2, 'Kesulitan': 3},
{'Name': 'Tubes 1 OOP',
  'Waktu': 11, 'SKS': 3,
  'Jenis': 3, 'Kesulitan': 3},
```

```
{'Name': 'Tubes Probstat',
  'Waktu': 15, 'SKS': 3,
  'Jenis': 3, 'Kesulitan': 3}]
```

Table 5 Data Uji 3

```
C:\windows\SYSTEM32\cmd.exe
Waktu: 16 Hari
Daftar Tugas:
1. PR Probstat
2. Tubes 2 OS
3. Tucil 3 Stima
4. Tubes 3 Basdat
5. Tucil 6 DRPL
6. Tubes 1 OOP
7. Tubes Probstat
Jumlah Tugas: 7

----- Algoritma Brute Force -----
1: PR Probstat
2: Tubes 2 OS
3: Tucil 3 Stima
4: Tucil 6 DRPL

Poin total: 10.5
Execution time: 0.00027594200000000124

----- Algoritma Greedy -----
1: Tubes 2 OS
2: Tubes 3 Basdat
3: PR Probstat

Poin total: 8.8
Execution time: 0.0001734499999999986

-----
Algoritma brute force lebih efektif
Algoritma greedy lebih efisien
```

Picture 6 Hasil Uji Tiga

IV. KESIMPULAN

Dari tiga hasil uji yang dilakukan pada bab sebelumnya dapat disimpulkan bahwa untuk batas waktu dan jumlah tugas yang sama namun berbeda bobotnya dihasilkan algoritma *brute force*, yang menggunakan cara mengerjakan tugas yang diberikan terlebih dahulu, lebih efektif dalam aspek poin yang didapatkan. Namun dengan menggunakan algoritma *greedy* kita dapat menentukan prioritas tugas-tugas lebih efisien walaupun tidak selalu mendapat poin tertinggi.

V. UCAPAN TERIMA KASIH

Saya ingin berterima kasih kepada Tuhan Yang Maha Esa, karena rahmat-Nya makalah ini dapat diselesaikan tanpa kesulitan yang berarti. Saya juga ingin mengucapkan terima kasih kepada kedua orang tua saya karena mendukung saya dan Dr. Masayu Leylia Khodra., ST., MT. sebagai dosen mata pelajaran Strategi Algoritma, yang mengajar saya di kelas. Akhirnya, saya ingin mengucapkan terima kasih kepada Dr. Ir. Rinaldi Munir, MT. dan para pengembang dan teman-teman saya yang telah memberikan banyak informasi berguna untuk makalah ini.

REFERENSI

[1] <https://brilliant.org/wiki/greedy-algorithm/> diakses pada 25 April 2019 pukul 19.24 WIB.

- [2] <https://dosenit.com/ilmu-komputer/pengertian-algoritma-brute-force-dan-greedy> diakses pada 25 April 2019 pukul 19.24 WIB.
- [3] <https://www.geeksforgeeks.org/greedy-algorithms/> diakses pada 25 April 2019 pukul 19.24 WIB.
- [4] <https://guide.freecodecamp.org/algorithms/brute-force-algorithms/> diakses pada 25 April 2019 pukul 19.24 WIB.
- [5] <https://guide.freecodecamp.org/algorithms/greedy-algorithms/> diakses pada 25 April 2019 pukul 19.24 WIB.
- [6] <https://www.hackerearth.com/problem/algorithm/dummy-26/> diakses pada 25 April 2019 pukul 19.24 WIB.
- [7] <https://www.hackerearth.com/practice/algorithms/greedy/basics-of-greedy-algorithms/tutorial/> diakses pada 25 April 2019 pukul 19.24 WIB.
- [8] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-\(2016\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-(2016).pdf) diakses pada 24 April 06 2019 pukul 19.04 WIB.
- [9] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Algoritma-Greedy-\(2019\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Algoritma-Greedy-(2019).pdf) diakses pada 24 April 06 2019 pukul 19.04 WIB.
- [10] <https://ltpb.itb.ac.id/akademik/satuan-kredit-semester-sks/> diakses pada 25 April 2019 pukul 19.24 WIB.
- [11] <https://stackoverflow.com/questions/31695202/the-greedy-algorithm-and-implementation> diakses pada 25 April 2019 pukul 19.24 WIB.
- [12] <https://stackoverflow.com/questions/8103050/what-exactly-is-the-brute-force-algorithm> diakses pada 25 April 2019 pukul 19.24 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, 26 April 2019



Vincent Budianto - 13517137