

Pencarian Jumlah Maksimum KRL yang dapat Berhenti di Stasiun Manggarai dalam Satu Hari

Ferdian Ifkarsyah/13517024

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13517024@std.stei.itb.ac.id

Abstraks—Regional Jabodetabek sebagai salah satu wilayah dengan penduduk terpadat dan teraktif di Indonesia mempunyai KRL Commuter Line sebagai salah satu transportasi umum andalannya. Saat ini, sistem KRL Commuter Line memiliki 6 jalur dan 80 stasiun dengan 6 di antaranya merupakan stasiun transit. Stasiun Manggarai adalah salah satu stasiun transit terpadat yang menghubungkan 3 jalur utama. Makalah ini akan membahas jumlah maksimum KRL berbeda yang berhenti setiap harinya di Stasiun Manggarai.

Kata Kunci—KRL, Commuter, greedy, persoalan, optimasi

I. PENDAHULUAN

KRL Commuter Line merupakan salah satu moda transportasi utama yang menghubungkan Jakarta dengan empat kota satelitnya, yaitu Kota Bogor, Kota Depok, Kota Tangerang, dan Kota Bekasi, atau yang lebih dikenal dengan singkatan Jabodetabek. KRL Commuterline beroperasi dibawah PT Kereta Commuter Indonesia(KCI) yang merupakan anak perusahaan PT Kereta Api Indonesia(KAI).

Saat ini, sistem KRL Commuter Line memiliki rangkaian rel dengan total panjang mencapai 418 km yang menjadikannya salah satu yang terpanjang di dunia. Dengan sistem seperti itu, KRL Commuter Line dapat mencapai rata-rata jumlah penumpang pada hari kerja mencapai 1 juta lebih penumpang.

Sistem rangkaian rel tersebut tersusun dari 6 jalur, yaitu Jakarta Kota--Bogor, Jatinegara--Bogor, Tanah Abang--Raangkasbitung, Jakarta Kota--Cikarang, Duri--Tangerang, dan Jakarta Kota--Tanjung Priok. Keenam jalur ini didukung oleh 80 stasiun dengan 6 di antaranya adalah stasiun transit. Stasiun transit adalah stasiun yang menjadinya tempat pertemuan lebih dari satu jalur. Keenam stasiun transit ini adalah Stasiun Duri, Stasiun Tanah Abang, Stasiun Jakarta Kota, Stasiun Kampung Bandan, Stasiun Manggarai, dan Stasiun Jatinegara.

Dari 80 stasiun tersebut, Stasiun Manggarai menjadi stasiun sekaligus stasiun transit tersibuk di Indonesia. dengan

lebih dari 100.000 penumpang naik dan turun dalam satu hari. Stasiun yang pertamakali dibuka pada 1918 itu menjadi perpotongan dari 3 jalur utama sistem KRL Commuter Line, yaitu Jakarta Kota--Bogor, Jakarta Kota--Bekasi, dan Jatinegara--Bogor. Sebagai sebuah stasiun yang cukup padat, seluruh KRL yang singgah di Stasiun Manggarai harus diatur sedemikian rupa agar tetap efektif dan efisien. Salah satu parameter keefisienan yang dapat diukur adalah jumlah KRL berbeda yang berhenti setiap harinya di Stasiun Manggarai. Dengan mengasumsikan bahwa hanya terdapat sebuah jalur untuk masing-masing jalur, kita akan mencoba untuk menggunakan algoritma *greedy* dalam mencari jumlah maksimum KRL berbeda yang berhenti setiap harinya di Stasiun Manggarai.



Gambar 1.1 Peta Rute KRL Commuter Line (Sumber: krl.co.id)

II. LANDASAN TEORI

A. Algoritma Greedy

Algoritma Greedy merupakan salah satu paradigma pemecahan masalah yang paling populer dalam persoalan optimasi. Algoritma greedy bekerja dengan menggunakan nilai optimal sementara pada setiap langkahnya untuk membangun solusi akhirnya.

Nilai optimal sementara dipilih berdasarkan apa yang terlihat paling baik pada langkah tersebut dan menghiraukan langkah lainnya. Oleh sebab itu, terdapat sebuah frasa populer yang sangat menggambarkan prinsip yang digunakan algoritma greedy, yaitu, "take what you can get now!".

1) Elemen dalam Algoritma Greedy

Dalam proses penyelesaian permasalahannya, algoritma greedy dibantu oleh lima buah elemen, yaitu:

- Himpunan kandidat, C , yang berisi elemen-elemen calon pembentuk solusi.
- Himpunan solusi, S , yang merupakan elemen-elemen terpilih dari himpunan kandidat, C .
- Fungsi seleksi, yang digunakan dalam setiap langkahnya untuk menentukan solusi optimal lokal.
- Fungsi kelayakan, yang memeriksa suatu kandidat layak atau tidak.
- Fungsi obyektif, yang merupakan tujuan akhir yang berupa tujuan optimasi program.

2) Skema Umum Algoritma Greedy

Dengan kelima elemen tersebut, kita dapat mengatakan bahwa:

"Algoritma greedy menyelesaikan permasalahan pencarian sebuah himpunan solusi, S , yang merupakan himpunan bagian dari himpunan kandidat, C , dengan syarat bahwa S harus memenuhi fungsi seleksi dan fungsi kelayakan dan S dioptimasi melalui fungsi obyektif."

```
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 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
```

Gambar 2.1 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))

B. Activity Selection Problem

Salah satu permasalahan klasik yang dapat diselesaikan dengan algoritma greedy adalah persoalan pemilihan aktivitas atau *Activity Selection Problem*. Persoalan ini memiliki deskripsi sebagai berikut:

"Diberikan n buah aktivitas, $S = \{a_1, a_2, \dots, a_n\}$. Setiap aktivitas memiliki waktu mulai dan waktu berakhir, sebut s_i dan f_i , dimana $s_i \leq f_i$. Tentukan banyak aktivitas maksimum yang dapat dilakukan oleh seseorang dengan asumsi dia hanya dapat melaksanakan satu aktivitas setiap waktunya."

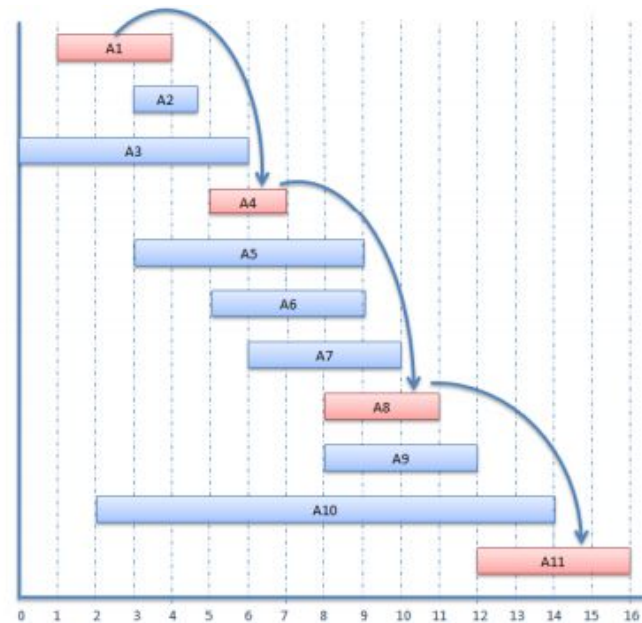
Contoh, misalkan terdapat 11 aktivitas beserta waktu mulai dan akhirnya, yaitu:

$activity = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$

$start = [1, 3, 0, 5, 3, 5, 6, 8, 8, 2, 12]$

$finish = [4, 5, 6, 7, 9, 9, 10, 11, 12, 13, 16]$

Solusi untuk instan persoalan di atas adalah memilih 4 aktivitas, yaitu $activity[1]$, $activity[4]$, $activity[8]$, dan $activity[11]$. Solusi ini diilustrasikan sebagai berikut:



Gambar 2.2 Solusi contoh persoalan activity selection problem

(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))

Gambar 3.1 Sebagai jadwal KRL yang singgah di Stasiun Manggarai.
(Sumber: <http://www.krl.co.id/>)

Solusi di atas didapatkan dari algoritma berikut:

```
def max_activity_selected(start, finish):
    count_activity = len(start)
    i = 0
    count_selected = 1
    for j in range(count_activity):
        if start[j] >= finish[i]:
            count_selected += 1
            i = j
    return count_selected

# activity has been sorted first by finish time
print(max_activity_selected(start=[1, 3, 0, 5, 3, 5, 6, 7, 7, 2, 12],
                             finish=[4, 5, 6, 7, 9, 9, 10, 11, 12, 13, 16]
                             ))
```

Gambar 2.3 Penyelesaian *activity selection problem*
(Sumber: pribadi)

Skema umum penyelesaian persoalan *Activity Selection Problem* adalah sebagai berikut:

- Urutkan aktivitas berdasarkan waktu berakhirnya.
- Pilih aktivitas pertama dan tambahkan *counter* aktivitas dengan 1.
- Jika waktu mulai aktivitas sekarang lebih lama atau sama dengan waktu berakhir aktivitas yang telah dipilih sebelumnya, pilih aktivitas yang sekarang.

III. IMPLEMENTASI PENCARIAN JUMLAH MAKSIMUM KRL YANG DAPAT BERHENTI DI STASIUN MANGGARAI

A. Penggambaran dan Perumusan Masalah

Setiap stasiun dapat kita ketahui KRL apa saja yang singgah dan jam berapa di stasiun tersebut dari website <http://www.krl.co.id/jadwal.html>. Gambar di bawah adalah sebagian jadwal KRL yang singgah di Stasiun Manggarai.

Relasi

Stasiun Keberangkatan

Stasiun Singgah

Range Waktu Hingga Jam

No KA	Relasi	Stasiun Keberangkatan	Stasiun Persinggahan	Waktu Datang	Waktu Berangkat	LS	Status
1053	BOGOR-JAKARTAKOTA	BOGOR	MANGGARAI	05:43:00	05:45:00	N	CENTRAL LINE
1054	JAKARTAKOTA-BOGOR	JAKARTAKOTA	MANGGARAI	06:43:00	06:45:00	N	CENTRAL LINE
1055	BOGOR-JAKARTAKOTA	BOGOR	MANGGARAI	06:00:00	06:02:00	N	CENTRAL LINE
1056	JAKARTAKOTA-BOGOR	JAKARTAKOTA	MANGGARAI	06:58:00	07:00:00	N	CENTRAL LINE
1057	DEPOK-JAKARTAKOTA	DEPOK	MANGGARAI	06:08:00	06:10:00	N	CENTRAL LINE
1058	JAKARTAKOTA-BOGOR	JAKARTAKOTA	MANGGARAI	07:13:00	07:15:00	N	CENTRAL LINE

Dari gambar di atas, dapat kita lihat bahwa terdapat kolom “No KA”, “Relasi”, “Stasiun Keberangkatan”, “Stasiun Persinggahan”, “Waktu Datang”, “Waktu Berangkat”, “LS”, dan “Status”. Kolom yang kita perlukan dalam persoalan pencarian jumlah maksimum KRL yang boleh berhenti setiap saat di Stasiun Manggarai hanya lah kolom “Waktu Datang”, “Waktu Berangkat”, dan “Status”, dengan masing-masing penjelasannya adalah:

- Waktu Datang : Waktu KRL Commuter Line tertentu tiba di Stasiun Manggarai
- Waktu Berangkat : Waktu KRL Commuter Line tertentu berangkat dari Stasiun Manggarai
- Status : Lajur rel yang digunakan oleh KRL Commuter Line saat berhenti di Stasiun Manggarai. Terdapat 7 line atau lajur yang tersedia, yaitu BEKASI LINE, LOOP LINE, CENTRAL LINE, CIKARANG LINE, FEEDER, SERPONG LINE, TANGERANG LINE. Kita dapat menamai kembali kolom ini dengan “Lajur”.

Setelah mendapatkan semua contoh data yang dibutuhkan, kita dapat menceritakan persoalan secara lebih deskriptif, yaitu sebagai berikut:

“Stasiun Manggarai akan dilewati oleh m buah KRL dengan waktu datang dan waktu keberangkatannya masing-masing. Stasiun Manggarai terdiri atas n buah lajur.

Setiap lajur hanya dapat ditempati oleh 1 KRL setiap saatnya. Setiap KRL memiliki lajur khusus tempatnya berhenti di stasiun tersebut (misalnya: CENTRAL LINE, LOOP LINE, dll).

Tentukan, berapa jumlah maksimum KRL berbeda yang dapat berhenti dalam satu hari di Stasiun Manggarai?”

Kita dapat mengilustrasikan solusi persoalan ini dengan kasus yang lebih kecil. Misalnya, kita instantiasi persoalan dengan tabel berikut:

No KA	Datang	Berangkat	Lajur
1	10:00	10:30	1
2	10:10	10:30	1
3	10:10	10:20	2
4	10:30	12:30	2
5	12:00	12:30	3
6	09:00	10:05	1

Solusi dari instan persoalan di atas adalah maksimum terdapat 5 KRL yang dapat berhenti secara bersamaan di Stasiun Manggarai.

Penjelasannya adalah tidak mungkin keenam-enamnya ada di stasiun pada saat yang bersamaan. Namun, jika KA1 telah berangkat, lajur 1 yang ditinggalkan KA1 dapat ditempati oleh KA2 dan KA6. Kemudian, KA3 dan KA4 dapat dimasukkan ke lajur 2. Terakhir, KA5 tetap di lajur 3.

B. Perancangan Struktur Data

KRL direpresentasikan sebagai sebuah kelas KRL. Dengan waktu tiba dan berangkat memiliki tipe data *datetime* dan lajur memiliki tipe data *int*.

```
class KRL:
    def __init__(self, tiba, berangkat, lajur):
        self.tiba = tiba
        self.berangkat = berangkat
        self.lajur = lajur
```

Gambar 3.2 Kelas yang merepresentasikan KRL (Sumber: pribadi)

Lajur dalam kelas direpresentasikan sebagai sebuah angka untuk alasan kemudahan dengan daftar pemetaannya sebagai berikut:

- Lajur 0 = BEKASI LINE
- Lajur 1 = LOOP LINE
- Lajur 2 = CENTRAL LINE
- Lajur 3 = CIKARANG LINE
- Lajur 4 = FEEDER
- Lajur 5 = SERPONG LINE
- Lajur 7 = TANGERANG LINE

```
KRLs = []
KRLs.append(KRL(s2tm('10:10'), s2tm('10:30'), 1))
KRLs.append(KRL(s2tm('10:10'), s2tm('10:30'), 1))
KRLs.append(KRL(s2tm('10:00'), s2tm('10:20'), 2))
KRLs.append(KRL(s2tm('10:30'), s2tm('12:30'), 2))
KRLs.append(KRL(s2tm('12:10'), s2tm('12:30'), 3))
KRLs.append(KRL(s2tm('09:00'), s2tm('10:05'), 1))
```

Gambar 3.3 Contoh pembuatan objek dari kelas KRL (Sumber: pribadi)

Untuk memastikan bahwa kita sedang berada dalam pemikiran yang sama, pastikan Anda mengerti bahwa:

- KRLs[3] akan tiba pukul 10:30 di lajur LOOP LINE(1), dan berangkat pukul 12:30.
- KRLs[5] akan tiba pukul 09:00 di lajur BEKASI LINE(0), dan berangkat pukul 10:05.

C. Perancangan dan Pengimplementasian Algoritma

Setelah melihat persoalan secara lebih jelas, kita dapat menentukan bahwa persoalan pencarian jumlah maksimum KRL di stasiun ini adalah varian dari *Activity Selection Problem*. Artinya, kita dapat menggunakan pendekatan *greedy* sejenis yang digunakan dalam penyelesaian *Activity Selection Problem* untuk menyelesaikan persoalan ini.

Algoritma yang dipakai adalah sebagai berikut:

- Kelompokkan KRL berdasarkan lajur tempat dia akan berhenti.
- Inisialisasi *count_max* dengan 0. '*count_max*' adalah yang menghitung berapa banyak KRL maksimum yang bisa sampai saat ini.
- Untuk tiap lajur, lakukan hal berikut:
 - Urutkan KRL berdasarkan waktu berangkatnya.
 - Pilih KRL pertama dan tambahkan nilai '*count_max*' dengan 1.
 - Jika waktu tiba KRL sekarang lebih besar atau sama dengan waktu berangkat KRL yang telah dipilih sebelumnya, pilih KRL yang sekarang.

```
def max_KRL_in_station(KRLs, n):
    lajurs = {}
    for krl in KRLs: # group KRLs by its platform
        if krl.lajur in lajurs:
            lajurs[krl.lajur].append(krl)
        else:
            lajurs[krl.lajur] = [krl]
    count_max = 0
    for lajur, krl in lajurs.items():
        krl_sorted = sorted(krl, key=attrgetter('berangkat'))
        count_max += 1 # select first KRL in each platform
        x = 0 # current KRL examined
        for j in range(1, len(krl_sorted)):
            if krl_sorted[j].tiba >= krl_sorted[x].berangkat:
                x = j
                count_max += 1
    return count_max
```

Gambar 3.3 Algoritma penyelesaian (Sumber: pribadi)

```
KRLs = []
KRLs.append(KRL(s2tm('10:10'), s2tm('10:30'), 1))
KRLs.append(KRL(s2tm('10:10'), s2tm('10:30'), 1))
KRLs.append(KRL(s2tm('10:00'), s2tm('10:20'), 2))
KRLs.append(KRL(s2tm('10:30'), s2tm('12:30'), 2))
KRLs.append(KRL(s2tm('12:10'), s2tm('12:30'), 3))
KRLs.append(KRL(s2tm('09:00'), s2tm('10:05'), 1))

print('Maksimum berhenti: ' + str(max_KRL_in_station(KRLs, 6)))
```

Gambar 3.4 Pengujian dengan kasus uji yang kecil. (Sumber: pribadi)

```

ifkarsyah@Acer475G:~/Desktop/KRL-statio
n-optimization$ python3 "/home/ifkarsya
h/Desktop/KRL-station-optimization/gree
dy.py"
Maksimum berhenti: 5

```

Gambar 3.5 Hasil pengujian.
(Sumber: pribadi)

IV. HASIL DAN ANALISIS

A. Hasil

Setelah mendapatkan hasil yang kita harapkan pada sampel kasus uji, kita siap untuk menguji algoritma ini dengan data lengkapnya yang didapat dari <http://www.krl.co.id/jadwal.html>. Sebagian datanya terlihat seperti gambar dibawah:

1	22:20,22:21,6
2	23:00,23:00,6
3	11:28,11:30,5
4	15:58,16:00,5
5	05:28,05:30,1
6	06:28,06:30,1
7	05:53,05:55,1
8	07:03,07:05,1
9	06:23,06:25,1
10	07:23,07:25,1

Gambar 4.1 Sebagian data dengan kolom yang akan digunakan sudah disaring.
(Sumber: pribadi)

Kita dapat mem-parsing data tersebut dengan bantuan kelas *built-in csv* Python. Kode untuk mengkonversi 'data.csv' ke list *KRLs* dapat dilihat seperti gambar di bawah:

```

KRLs = []
with open('src/data.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        KRLs.append(KRL(s2tm(row[0]),
                        s2tm(row[1]),
                        int(row[2])))

```

Gambar 4.2 Pengonversian data dari file 'data.csv' ke list 'KRLs'
(Sumber: pribadi)

Dengan menjalankan kembali algoritma *greedy* untuk list *KRLs* yang baru, kita mendapatkan fakta bahwa dari 475 perjalanan KRL yang melewati Stasiun Manggarai, terdapat 382 KRL Commuter Line berbeda yang dapat berhenti di Stasiun Manggarai dalam satu hari.

```

45 print('Maksimum berhenti: ' + str(max_KRL_in_station(KRLs, 7)))
46
ifkarsyah@Acer475G:~/Desktop/KRL-station-optimization$ python3 "/home/ifkarsyah/
optimization/src/greedy.py"
Maksimum berhenti: 382
ifkarsyah@Acer475G:~/Desktop/KRL-station-optimization$

```

Gambar 4.3 Hasil komputasi algoritma untuk file 'data.csv'
(Sumber: pribadi)

Perlu diingat bahwa hasil tersebut berlaku jika setiap jalur hanya memiliki sebuah lajur. Namun, dalam kenyataannya, PT KCI menyediakan lebih dari sebuah lajur untuk jalur yang tergolong padat. Misalnya, untuk LOOP LINE JakartaKota--Bogor memiliki 2 buah jalur pada jam pulang kerja, yaitu jalur 6 dan jalur 8.

B. Analisis

Kita dapat melihat bahwa pengelompokan KRL berdasarkan jalurnya memiliki kompleksitas $O(m)$, dengan m adalah jumlah KRL yang lewat. Kemudian, pengurutan KRL berdasarkan waktu berangkatnya memiliki kompleksitas $O(m \log m)$ dengan fungsi *sorted()* bawaan Python.

Pemrosesan pemilihan KRL di setiap lajurnya memiliki kompleksitas $O(m)$. Kita dapat melihat bahwa banyaknya lajur atau n tidak berpengaruh karena n hanya dibagi menjadi kelompok-kelompok m dan bukan digandakan.

Dari pembahasan di atas, kita dapat menyimpulkan bahwa algoritma *greedy* ini berjalan dengan total kompleksitas $O(m \log m)$.

Kompleksitas tersebut sama dengan kompleksitas algoritma *greedy* klasik untuk penyelesaian persoalan *Activity Selection Problem*. Hal ini dapat mengindikasikan bahwa memang benar persoalan ini adalah varian dari persoalan klasik tersebut. Dalam persoalan ini, kita dapat mengandaikan bahwa sebuah KRL Commuter Line yang akan datang ke Stasiun Manggarai adalah sebuah *activity* yang akan datang.

V. KESIMPULAN

Algoritma *greedy* yang digunakan dalam persoalan pencarian jumlah maksimum KRL yang dapat berhenti di Stasiun Manggarai memiliki kompleksitas $O(m \log m)$.

Kompleksitas tersebut sama dengan kompleksitas algoritma *greedy* klasik untuk penyelesaian persoalan *Activity Selection Problem* yang menjadi salah satu indikator bahwa memang algoritma dan persoalan ini adalah varian dari persoalan klasik tersebut.

Dari 475 perjalanan KRL yang melewati Stasiun Manggarai, terdapat 382 KRL Commuter Line berbeda yang dapat berhenti di Stasiun Manggarai dalam satu hari. Jumlah ini dapat berbeda karena kita mengasumsikan hanya terdapat sebuah lajur untuk satu jalur. Namun, di Stasiun Manggarai sendiri terdapat sebuah jalur yang terdiri atas lebih dari satu lajur. Misalnya, lajur 6 dan lajur 8 yang digunakan untuk jalur JakartaKota--Bogor.

UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terimakasih kepada Bapak Rinaldi Munir selaku Dosen Strategi Algoritma ITB K-3 atas ilmu dan pengetahuan dari mata kuliah ini. Kemudian, penulis juga ingin berterimakasih kepada para penulis yang artikel, tulisan atau bukunya penulis jadikan referensi dalam pengerjaan makalah ini. Terakhir, penulis berterimakasih kepada teman-teman mahasiswa yang menjadi tempat bertukar pikiran dan memberikan sudut pandang unik yang sangat membantu dalam mewujudkan makalah ini.

LAMPIRAN

Semua *source code* yang berupa *screenshot* dalam makalah ini dapat diakses melalui halaman <https://github.com/Ifkarsyah/KRL-station-optimization>

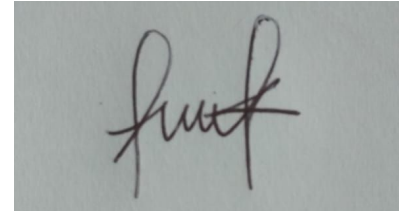
REFERENSI

- [1] Munir, Rinaldi. Matematika Diskrit. Bandung:ITB,2006.
- [2] Rosen, Kenneth H. 2006. Discrete Mathematics and Its Applications, 6th Edition. Singapore: McGraw-Hill.
- [3] [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)
Tanggal akses: 25 April 2019, pukul 20:30 WIB
- [4] <https://www.geeksforgeeks.org/activity-selection-problem-greedy-algo-1>
Tanggal akses: 25 April 2019, pukul 21:30 WIB
- [5] <http://www.krl.co.id/>
Tanggal akses: 25 April 2019, pukul 22:00 WIB
- [6] <http://www.krl.co.id/jadwal.html>
Tanggal akses: 25 April 2019, pukul 22:30 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



Ferdian Ifkarsyah
13517024