

Penggunaan Algoritma *Decrease & Conquer* Untuk Mencari Solusi Pada Permainan *Bridge and Torch*

Alexander / 13519090 (*Author*)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519090@std.stei.itb.ac.id

Abstract—Video Game merupakan sebuah permainan dalam bentuk elektronik dengan user interface maupun alat untuk menerima masukan user (Pengendali). Banyak video game bertemakan teka-teki dapat diselesaikan dengan strategi algoritma yang ada, salah satunya adalah permainan teka-teki ialah *Bridge and Torch*. Permainan ini dapat diselesaikan dengan strategi *Decrease and Conquer*, dimana persoalan akan terus diperkecil dan diselesaikan. Algoritma akan menggunakan 2 strategi (rekurens) untuk menentukan hasil yang optimal dalam permasalahan *Bride and Torch* dengan cara mengurangi 2 orang terlambat setiap rekursi dilakukan, sehingga setiap strategi yang dilakukan akan menghasilkan 2 orang terlambat berada di sisi *finish*, kecuali sisa pada larik start merupakan basis yaitu $n \leq 3$.

Keywords—*Bridge and Torch*; *Decrease and Conquer*; teka-teki

I. PENDAHULUAN

Pada zaman modern, *video game* merupakan salah satu perkembangan teknologi yang dapat memberikan peluang pekerjaan maupun membantu dalam meningkatkan kualitas hidup manusia. Sekarang Permainan dapat dimainkan dari banyak *platform* dan mudah untuk didapatkan, tidak memerlukan objek seperti permainan lampu (monopoli, ular tangga). Salah satunya ialah permainan teka-teki.

Permainan teka-teki merupakan permainan yang dimenangkan dengan cara mencari solusi atau menyelesaikan masalah yang diberikan oleh pembuat permainan tersebut. Penyelesaian dari permainan tersebut bisa saja banyak (solusi jamak), ataupun hanya 1 solusi yang sudah ditetapkan. Permainan ini banyak dimainkan oleh orang-orang untuk mencari *thrill* dalam memecahkan sebuah masalah, mencari cara untuk menyelesaikan masalah tersebut.

Bridge and Torch merupakan salah satu permasalahan yang sering dijadikan permainan teka-teki atau *puzzle*. Permainan ini bertujuan untuk menyebrangkan semua orang dari 1 sisi ke sisi lainnya, dengan waktu yang terbatas, dan setiap orang memiliki waktu untuk menyebrang yang berbeda beda, sementara untuk menyebrang diperlukan sebuah lampu, sehingga setelah menyebrang 1 orang harus kembali untuk mengembalikan lampu dan pemain akan memilih orang mana yang selanjutnya harus menyebrang. Permainan akan selesai jika semua orang bisa mencapai sisi sebrangnya dalam batas waktu yang ditentukan.



Gambar 1. 1 Contoh Permainan *Bridge and Torch* (*Family Bridge Crossing*).

Sumber : <https://youtube.com/InRoll>

Namun permainan *Bridge and torch* dapat diubah dan dibuat menjadi beragam, maksudnya adalah pengubahan jumlah n orang yang menyebrang, ataupun mengubah waktu dari orang untuk menyebrang. Contoh dari permainan *Bridge and Torch* dengan waktu dan jumlah orang yang berbeda.



Gambar 1. 2 Contoh lain Permainan *Bridge and Torch* (*Cross the Bridge*).

Sumber : <https://www.inwebson.com/demo/cross-the-bridge/>

Permasalahan *Bridge and Torch* ini dapat diselesaikan dengan banyak cara, salah satunya adalah *Decrease and Conquer*, dapat dibentuk 2 list, start dan finish, list tersebut berisikan orang yang ada di *start* / *finish*, yang perlu diselesaikan larik *start*, orang yang sudah di sisi sebrang (*finish*) akan diabaikan, sehingga ini memenuhi pengertian *Decrease and Conquer*.

II. TEORI DASAR

A. Puzzle

Puzzle merupakan sebuah jenis permainan yang menguji cara pikir intelektual dan memacu bagaimana seseorang dapat menyelesaikan sebuah *puzzle* atau teka-teki.



Gambar 2. 1 Ilustrasi Permainan Puzzle.
Sumber : <https://freepik.com/puzzle>

Puzzle biasanya dikaitkan dengan ilustrasi diatas, namun sebenarnya *puzzle* memiliki banyak jenis, beberapanya adalah *logic puzzle*, *relational puzzle*, *number puzzle*, *crossword puzzle* dan *word-search puzzle*.

Pada zaman modern ini sangat banyak *puzzle* yang dibuat dalam bentuk elektronik, tidak memerlukan barang fisik lagi seperti dulu. Sudah banyak tersedia permainan *puzzle* pada gawai yang dapat dimainkan dengan mudah dan gratis.

Puzzle memang sebuah permainan, tapi seringkali sebuah *puzzle* diambil atau diimplementasikan dari sebuah persoalan matematika ataupun persoalan logika. Sebuah *puzzle* dapat dibuat penyelesaiannya dengan menggunakan strategi algoritma, contohnya *puzzle* geser dapat diselesaikan dengan algoritma *Branch and Bound*, dan masalah *i* yang dapat diselesaikan dengan *Decrease and Conquer*.

B. Decrease and Conquer

Decrease and Conquer merupakan salah satu metode perancangan algoritma untuk menyelesaikan masalah, dimana suatu persoalan akan dipecah menjadi dua upa-persoalan yang lebih kecil, dan yang akan diproses selanjutnya hanyalah satu sub-persoalan saja. *Decrease and Conquer* merupakan turunan dari algoritma *Divide and Conquer*, dimana pada algoritma *Divide and Conquer* kedua sub-persoalan akan di-conquer. Hal ini berbeda dengan *Decrease and Conquer* yang hanya menaklukkan salah satu

Algoritma *Decrease and Conquer* memiliki 2 tahap dalam pembentukannya, yaitu :

1. Decrease

Pada Tahap ini, sebuah masalah akan direduksi sehingga terbentuk menjadi beberapa sub-persoalan yang lebih kecil (biasanya menjadi 2 sub-persoalan) dan akan ditentukan dari 2 sub-persoalan tersebut, persoalan yang akan di conquer dan persoalan yang tidak di-conquer.

2. Conquer

Pada tahap ini sub-persoalan yang sudah dipilih dari tahap sebelumnya, akan di-conquer atau ditaklukan. Tahap ini biasanya dilakukan secara rekursif untuk menyelesaikan salah satu sub persoalan yang dipilih.

Karena hanya satu sub-persoalan yang diselesaikan maka tidak ada tahap untuk menggabungkan kedua sub-persoalan tersebut.

Decrease and Conquer dapat diimplementasikan dalam 2 bentuk, yaitu

1. Top-down approach

Pendekatan top-down berarti penggunaan decrease and conquer dalam bentuk yang rekursif, pendekatan ini sangat umum pada penggunaan strategi algoritma decrease and conquer, penulis juga mengimplementasi solusi dengan pendekatan top-down.

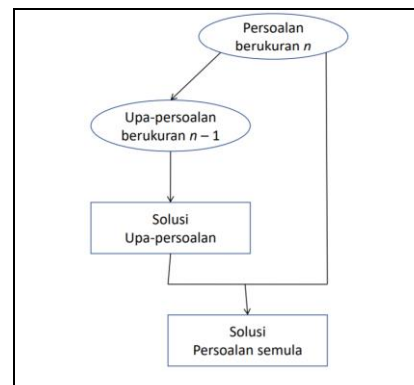
2. Bottom-up approach

Pendekatan *bottom-up*, tidak menggunakan bentuk rekursif, namun menggunakan bentuk iteratif, biasanya akan mencari solusi untuk permasalahan terkecil.

Decrease and Conquer memiliki 3 varian untuk melakukan pengurangan persoalan pada tahapan Decrease, 3 varian tersebut adalah

1. Decrease by a constant

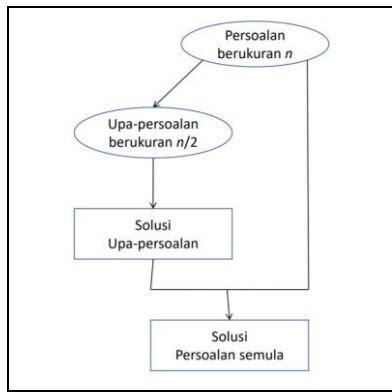
Varian ini akan mereduksi sebuah persoalan sesuai dengan konstanta, contohnya dikurangi 1 elemen setiap kali Decrease dilakukan, varian ini merupakan varian yang digunakan oleh penulis karena untuk setiap iterasi, akan dikurangi sebanyak 2 elemen.



Gambar 2. 2 Flowchart Algoritma Decrease and Conquer Varian Decrease by a constant dengan nilai constant = 1.
Sumber : <https://informatika.stei.itb.ac.id>

2. Decrease by a constant factor

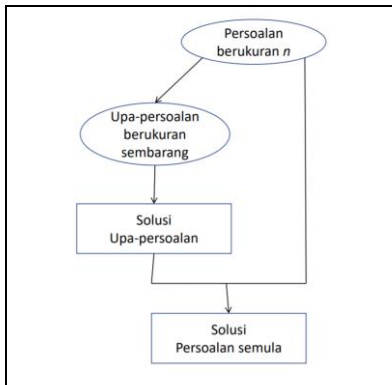
Varian ini akan mereduksi sebuah persoalan dengan suatu faktor konstanta, contohnya dibagi 2 untuk setiap pengurangan, dibagi 3 ataupun faktor lainnya.



Gambar 2. 3 Flowchart Algoritma Decrease and Conquer Variasi Decrease by a constant factor dengan nilai constant factor = 2.
Sumber : <https://informatika.stei.itb.ac.id>

3. Decrease by a variable size

Variasi ini akan mereduksi persoalan bervariasi pada setiap iterasi algoritma, sehingga jumlah reduksi yang dilakukan akan bergantung pada jumlah variabel dan iterasi yang dilakukan.



Gambar 2. 4 Flowchart Algoritma Decrease and Conquer Variasi Decrease by a variable size
Sumber : <https://informatika.stei.itb.ac.id>

C. Bridge and Torch

Bridge and Torch merupakan suatu teka-teki logika dimana ada n orang harus menyebrang ke sisi lain, dengan batas waktu tertentu dan sebuah obor yang akan digunakan untuk menyebrangi jembatan tersebut. Aturan pada permainan *Bridge and Torch* adalah :

1. Ada n orang pada sisi start. Dimana semua orang harus berpindah ke seberang atau sisi finish melalui sebuah jembatan dan alat bantu Obor.
2. Jembatan hanya dapat dilewati oleh 2 orang secara bersamaan dan harus menggunakan Obor.
3. Jumlah obor hanya 1 dan memiliki Batas waktu tertentu.
4. Setiap 2 orang yang melewati jembatan, salah satu orang tersebut harus kembali menyebrangi jembatan untuk mengembalikan obor.

5. Setiap orang memiliki waktu yang berbeda beda, ketika menyebrang maka waktu yang digunakan adalah waktu yang terlama antara 2 penyebrang tersebut.

III. PENERAPAN ALGORITMA DECREASE AND CONQUER PADA PERMASALAHAN BRIDGE AND TORCH

Untuk menyelesaikan masalah tersebut akan dituliskan beberapa spesifikasi larik untuk membantu penggunaan algoritma decrease and conquer kedepannya.

- a. Sebuah larik t yang berisikan waktu dari n orang, $n = 5$ maka $t = [t_1, t_2, t_3, t_4, t_5, t_6] = [1, 2, 4, 6, 8, 12]$ dalam satuan waktu yang sama (menit/detik). Syarat berlaku bahwa larik t sudah terurut keatas.
- b. Batas waktu (B) = 30 (menit/detik).
- c. Sebuah Himpunan solusi akhir yang berbentuk $[[t_1, t_2], t_1]$ yang menandakan bahwa pertama t_1, t_2 pergi dan t_1 kembali lagi untuk memberikan obor.
- d. Decrease akan dilakukan ketika 2 orang terlambat sudah mencapai sisi finish, sehingga akan dilakukan decrease by constant yaitu 2.

Untuk mencari penyelesaian masalah dengan algoritma *Decrease and conquer* diperlukan basis untuk masalah *Bridge and Torch*, Basis yang dibuat adalah, ketika hanya 2 orang yang ingin menyebrang sehingga hanya terdapat 2 elemen pada list t .

$$t = [t_1, t_2] \text{ atau } t = [t_1]$$

$$\text{bridgeAndTorch}(t,b) = \max(t)$$

Jika ada 2 orang atau kurang, maka secara otomatis waktu yang dibutuhkan untuk menyebrangi jembatan adalah waktu tertinggi dari orang (orang dengan waktu menyebrang paling lambat), dikarenakan setelah menyebrang tidak perlu ada yang kembali untuk mengembalikan obor.

Namun 1 basis tidak cukup karena bisa saja tersisa n orang dengan jumlah yang ganjil contohnya 3, maka harus ada salah satu yang kembali untuk menjemput orang pada sisi *start*.

Jika ada 3 orang yang ingin menyebrang maka, diperlukan 1 perjalanan pulang untuk mengembalikan obor agar 3 orang tersebut dapat menyebrangi jembatan tersebut. Orang yang paling cepat akan menjadi pengantar orang lain dan pengantar obor, sehingga t_1 berjalan dengan t_2 , lalu t_1 akan kembali dan menjemput t_3 untuk menyebrangi jembatan, sehingga didapat basis ke 2 yaitu

$$t = [t_1, t_2, t_3]$$

$$\text{bridgeAndTorch}(t,b) = t[0] + t[1] + t[2]$$

Lalu yang terakhir merupakan rekursi dari *Decrease and Conquer*, jika n orang dengan $n \geq 4$, akan digunakan suatu strategi untuk memindah 2 orang terlambat dari sisi *start* ke sisi *finish*, dimana orang yang sudah berada di sisi *finish* setelah algoritma ini dilakukan akan dibuang dari larik t .

Strategi yang dimaksud ialah :

1. 2 Orang tercepat akan menyebrang terlebih dahulu
2. Orang tercepat dari sisi *finish* akan kembali untuk mengembalikan obor
3. 2 Orang terlambat akan pergi menyebrang
4. Orang tercepat dari sisi *finish* akan kembali untuk mengembalikan obor
5. Lakukan Decrease and Conquer hingga basis tercapai

Hal ini dilakukan karena ketika 2 orang terlambat menyebrangi jembatan secara bersamaan maka akan dicapai efisiensi maksimal, dimana orang kedua terlambat tidak begitu tertahan oleh orang terlambat.

Dengan melakukan hal tersebut, jika diambil contoh

$$t = [t_1, t_2, t_3, t_4]$$

$$t = [1, 2, 4, 6] \quad B = 15$$

Data Waktu Penyebrangan			
No	Orang yang menyebrang	Orang yang kembali	Waktu yang Diperlukan
1	t_1, t_2	-	2
2	-	t_1	1
3	t_3, t_4	-	6
4	-	t_2	2
5	Total waktu		11
6	Lakukan Decrease and Conquer		
7	t_1, t_2	-	2
8	Total waktu		13

Tabel 3.1 Langkah penyelesaian permasalahan bridge and torch

Maka didapat solusi sebagai berikut

$$t = [[1, 2], 1, [6, 4], 2, [2, 1]]$$

$$\text{Total waktu} = 13$$

$$\text{bridgeAndTorch}(t, b) = t[1] + t[0] + t[n-1] + t[1]$$

$$= t_2 + t_1 + t_n + t_2$$

Setelah Total waktu yang pertama dihitung, dapat dilihat bahwa pada larik start berisikan $[t_1, t_2]$ dan pada larik finish berisikan $[t_3, t_4]$, sehingga larik start berhasil diperkecil, dan akan dilakukan *conquer* pada larik *finish*, dimana larik *start* memenuhi basis dengan $n = 2$, sehingga didapat waktu dari larik finish adalah 2. Maka didapat kesimpulan bahwa solusi dari permasalahan $t = [1, 2, 4, 6]$ adalah 13 satuan waktu. Solusi ini optimal dan tetap memenuhi batas waktu yang diberikan yaitu 15.

Maka dari basis dan data diatas dapat dibentuk Algoritma *Decrease and Conquer*, dalam bahasa *python* sebagai berikut

```
def totalTime(t,n):
# Asumsi bahwa larik t sudah terurut menaik
if (n < 3): # Elemen dibawah 3
if (n == 2):
result.append([t[n-1],t[n-2]])
else :
result.append(t[n-1])
return t[n - 1]
elif (n == 3):
# Jika larik hanya berisikan 3 elemen
result.append([t[0],t[1]])
result.append(t[0])
result.append([t[0],t[2]])
return t[0] + t[1] + t[2]
# larik berisikan lebih atau sama dengan 4 elemen
else :
time = t[1] + t[0] + t[n-1] + t[1]
result.append([t[0],t[1]])
result.append(t[0])
result.append([t[n-1],t[n-2]])
result.append(t[1])

return time + totalTime(t[:-2], len(t[:-2]))
```

Algoritma diatas akan memberikan jumlah waktu yang diperlukan untuk permasalahan *Bridge and Torch*, Algoritma tersebut tidak hanya menghasilkan jumlah waktu, namun juga memberikan Langkah solusi yang diperlukan untuk mencapai jumlah waktu tersebut.

Dapat dilihat pada rekursi di bagian else terakhir, bahwa dilakukan *decrease* terhadap 2 elemen terakhir, karena 2 elemen terakhir sudah tidak akan diperhitungkan pada rekursi selanjutnya, dikarenakan 2 elemen terakhir sudah berada pada larik *finish*.

Namun bila diteliti lebih lanjut maka akan ditemukan bahwa ada kasus ketika orang tercepat pertama dan kedua memiliki selisih waktu yang cukup besar maka algoritma tersebut tidak menghasilkan nilai yang optimal, maka akan dilakukan optimasi untuk membuat semua permasalahan yang akan dihitung, akan menghasilkan solusi yang optimal.

Contoh kasusnya adalah sebagai berikut :

$$t = [t_1, t_2, t_3, t_4]$$

$$t = [1, 22, 24, 26] \quad B = 75$$

Data Waktu Penyebrangan			
No	Orang yang menyebrang	Orang yang kembali	Waktu yang Diperlukan
1	t_1, t_2	-	22

2	-	t ₁	1
3	t ₃ , t ₄	-	26
4	-	t ₂	22
5	Total waktu		71
6	Lakukan <i>Decrease and Conquer</i>		
7	t ₁ , t ₂	-	22
8	Total waktu		93

Tabel 3.2 Langkah penyelesaian permasalahan bridge and torch kasus tidak optimal

t = [[1, 22], 1, [26, 24], 22, [22, 1]]
Total waktu = 93

Didapat Hasil dari algoritma tersebut ialah 93, namun selain tidak memenuhi batas B, hasil yang didapat juga tidak optimal. Solusi yang Optimal adalah sebagai berikut

No	Data Waktu Penyebrangan		
	Orang yang menyebrang	Orang yang kembali	Waktu yang Diperlukan
1	t ₁ , t ₄	-	26
2	-	t ₁	1
3	t ₁ , t ₃	-	24
4	-	t ₁	1
5	Total waktu		52
6	Lakukan <i>Decrease and Conquer</i>		
7	t ₁ , t ₂	-	22
8	Total waktu		74

Tabel 3.3 Langkah penyelesaian permasalahan bridge and torch kasus optimal

Maka akan dibentuk strategi kedua, yang akan dibandingkan pada Strategi pertama, strategi kedua merupakan :

1. Orang tercepat akan mengantar Orang terlambat
2. Orang tercepat kembali
3. Ulangi Hingga 2 orang terlambat sudah di sisi finish, dan orang tercepat sudah di sisi start
4. Lakukan Decrease and Conquer hingga basis tercapai

Dengan adanya tambahan strategi ini, maka semua hasil dari algoritma ini akan menghasilkan hasil yang optimal, tapi belum tentu berhasil jika B yang diberikan memang tidak mencukupi (batas B terlalu kecil yang tidak memungkinkan solusi yang mungkin).

Maka dari itu kode yang sudah ada sebelumnya akan dilakukan optimasi dengan strategi kedua dan diberikan nama time2, maka setelah dioptimasi (Kode yang ditambahkan adalah kode dengan highlight berwarna kuning), Penambahan strategi 2

menyebabkan adanya 2 strategi, sehingga dari kedua strategi tersebut dipilih strategi dengan *cost* terendah, sehingga kode akan berubah menjadi :

```
def totalTime(t,n):
# Asumsi bahwa larik t sudah terurut menaik
if (n < 3): # Elemen dibawah 3
    if (n == 2):
        result.append([t[n-1],t[n-2]])
    else :
        result.append(t[n-1])
    return t[n - 1]
elif (n == 3):
# Jika larik hanya berisikan 3 elemen
    result.append([t[0],t[1]])
    result.append(t[0])
    result.append([t[0],t[2]])
    return t[0] + t[1] + t[2]
# larik berisikan lebih atau sama dengan 4 elemen
else :
    time = t[1] + t[0] + t[n-1] + t[1]
    time2 = t[n-1] + t[0] + t[n-2] + t[0]

    if time2 < time :
        result.append([t[0],t[n-1]])
        result.append(t[0])
        result.append([t[0],t[n-2]])
        result.append(t[0])
        time = time2
    else :
        result.append([t[0],t[1]])
        result.append(t[0])
        result.append([t[n-1],t[n-2]])
        result.append(t[1])

return time + totalTime(t[:-2], len(t[:-2]))
```

IV. KESIMPULAN

Algoritma *Decrease and Conquer* dapat digunakan untuk menyelesaikan banyak hal, salah satunya adalah permasalahan *Bridge and Torch*, permasalahan tersebut dapat dicari solusi dengan menggunakan algoritma *Decrease and Conquer*. Dengan solusi yang optimal, sesuai yang diinginkan. Algoritma ini berjalan dengan menggunakan 2 basis (2 dan 3 elemen) dan 2 strategi untuk jumlah orang ≥ 4 , Strategi 2 orang terlambat menyebrang bersamaan, dan strategi orang tercepat mengantar dan menjemput semuanya.

VIDEO LINK AT YOUTUBE

Video penjelasan yang diunggah oleh penulis dengan judul “Penerapan Algoritma Decrease and Conquer pada permasalahan Bridge and Torch” dapat diakses pada pranala https://youtu.be/kW1_jatBoRk

Untuk *source code* dapat dilihat di pranala

<https://github.com/Zenovore/Bridge-Torch-Solver>

V. UCAPAN TERIMA KASIH

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa, atas berkat dan rahmatnya sehingga penulis dapat menyelesaikan makalah ini dengan sebaik mungkin. Terima kasih juga untuk Dr. Nur Ulfa Maulidevi, S.T., M.Sc., selaku pengajar mata kuliah Strategi Algoritma dan pembimbing dalam pembuatan makalah ini, Beserta para dosen pengajar IF2211 Strategi Algoritma lainnya, yaitu Prof.Ir. Dwi Hendratmo Widyantoro, M.Sc.,Ph.D., Dr. Ir. Rinaldi, M.T., dan Ir. Rila Mandala, M.Eng.,Ph.D. Saya ingin mengucapkan terimakasih pada para dosen IF2211 Semester II 2020/2021 yang telah mengajarkan saya mengenai strategi algoritma, dan menugaskan makalah ini untuk memperdalam pemahaman mengenai aplikasi strategi algoritma pada kehidupan sehari-hari. Saya harap ilmu yang dipelajari akan saya gunakan di masa mendatang.

Penulis juga ingin mengucapkan terimakasih pada keluarga dan teman yang mendukung dalam proses pembuatan makalah ini. Memberikan pendapat dan kritik pada makalah ini, agar makalah ini menjadi lebih baik. Saya ucapkan terimakasih pada permasalahan *Bridge and Torch* yang sudah menjadi pokok bahasan pada makalah ini, beserta *driver gojek* yang sudah mengantarkan makanan untuk membantu proses berpikir disaat pengerjaan makalah ini.

REFERENSI

- [1] Munir, Rinaldi. 2009. Diktat Kuliah IF2211 Strategi Algoritma. Bandung : Teknik Informatika Institut Teknologi Bandung
- [2] Levitin ,Anany. 2003. Introduction to the Design & Analysis of Algorithms, Addison-Wesley,.
- [3] Munir, Rinaldi. Algoritma Decrease and Conquer. Diakses pada tanggal 10 mei 2021
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian1.pdf>
dan
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>
- [4] Rote G (2002) Crossing the bridge at night. BEATCS 78:241–246
- [5] Saloni1297, Decrease and Conquer. Diakses pada tanggal 10 mei 2021
<https://www.geeksforgeeks.org/decrease-and-conquer/>
- [6] Gleb Gribakin, Some simple and not so simple maths problems. Diakses pada tanggal 10 mei 2021
<http://web.am.qub.ac.uk/users/g.gribakin/problems.html>

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, 10 Mei 2021



Alexander 13519090