

Pemanfaatan Algoritma *Greedy* dalam Penataan Barang di Wadah Persegi Panjang Terbuka

Pemanfaatan Algoritma *Greedy by Largest Base Area, Longest Length, and Largest Length-to-Width Ratio*

Andika Naufal Hilmy - 13520098
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): andika.naufal48@gmail.com

Abstract—Penataan barang pada wadah yang memiliki kapasitas terbatas adalah masalah yang sering dijumpai ketika ingin menyimpan barang dengan jumlah sebanyak mungkin tanpa melebihi kapasitas wadah tersebut. Dalam menata barang, diperlukan sebuah strategi yang baik untuk mengefisiensi ruang penyimpanan dan memaksimalkan daya tampung wadah sehingga meminimalkan pemborosan ruang. Algoritma *greedy* merupakan salah satu algoritma yang dapat digunakan untuk menyusun barang dalam wadah dengan penekanan pada dimensi barang.

Keywords—*tata ruang, barang, wadah, greedy, daya tampung*

I. PENDAHULUAN

Penataan barang adalah sebuah permasalahan tentang bagaimana cara menata barang-barang yang akan disimpan atau ditaruh dalam suatu tempat. Penataan barang sangat menentukan bagaimana barang-barang dapat disimpan dengan baik tanpa merusak penampungnya dan membutuhkan lebih sedikit tempat dalam penyimpanannya sehingga memberi ruang yang cukup untuk barang-barang yang lain selama penampung barang masih memungkinkan untuk menyimpan barang tambahan.

Masalah penataan barang muncul ketika ada banyak barang yang harus disimpan ke dalam wadah yang memiliki luas permukaan yang terbatas. Biasanya, masalah ini muncul ketika seseorang harus memindahkan barang dengan sesedikit mungkin perpindahan yang diinginkan atau menata banyak barang dengan ruang tempat yang sangat terbatas sedemikian rupa sehingga tercipta ruang gerak yang lebih banyak. Untuk menyelesaikan masalah ini, dibutuhkan perhitungan yang tepat dalam menaruh barang-barang yang akan disimpan sehingga ruang tempat penyimpanan dapat digunakan dengan optimal.

Ada banyak cara dan metode dalam menata barang dalam ruang tempat terbatas. Khusus dalam makalah ini, pembahasan mengenai masalah penataan barang akan dibatasi oleh sejumlah asumsi, yaitu wadah merupakan tempat penampungan terbuka dengan bentuk persegi panjang dan memiliki luas alas terbatas, dimensi barang yang ditaruh tidak melebihi dimensi wadah baik dari dimensi panjang maupun

dimensi lebar, serta barang tidak boleh bertumpukan satu sama lain dalam wadah. Dalam kasus ini, barang yang disimpan diasumsikan memiliki bentuk alas persegi panjang, diputar 90 derajat, dan tidak ditata secara miring menurut wadah.



Gambar 1.1. Wadah

Sumber: https://res.cloudinary.com/ruparupa-com/image/upload/w_360,h_360,f_auto,q_auto/f_auto,q_auto:eco/v1491464881/229459_1.jpg

Dalam memilih barang yang akan disimpan dalam wadah, diperlukan sebuah strategi yang baik untuk menyiasati ruang tempat yang terbatas sehingga wadah dapat menyimpan sebanyak mungkin barang dengan sisa ruang tempat sesedikit mungkin. Khusus dalam makalah ini, strategi yang digunakan adalah strategi *greedy* yang akan digunakan untuk menyusun barang-barang pada wadah.

Algoritma *greedy* adalah salah satu jenis strategi algoritma yang sering digunakan dalam menyelesaikan persoalan yang memberikan hasil yang cukup optimal dengan waktu perumusan strategi dan keputusan yang cepat. Karena itu, algoritma *greedy* dapat diaplikasikan dalam penyelesaian masalah penyusunan barang sekaligus menjadi acuan dalam penyelesaian masalah penyusunan barang di wadah.

II. TEORI DASAR

A. Algoritma Greedy

Algoritma *greedy* adalah salah satu algoritma populer yang digunakan untuk penyelesaian masalah optimasi. Masalah optimasi dapat dibagi menjadi dua macam, yaitu masalah maksimasi dan masalah minimasi. Algoritma *greedy* dilakukan dengan langkah per langkah sedemikian rupa sehingga pada setiap langkah, pilihan terbaik yang diperoleh saat itu akan diambil tanpa memperhatikan konsekuensi ke depan, sesuai dengan prinsip “*Take what you can get now*”. Dengan langkah seperti itu, optimum lokal yang didapat pada setiap langkah diharapkan akan berakhir dengan hasil optimum global.

Pada algoritma *greedy*, ada enam elemen berikut yang digunakan, yaitu

1. Himpunan kandidat, C
Himpunan yang berisi daftar kandidat yang akan dipilih pada setiap langkah yang diambil.
2. Himpunan solusi, S
Himpunan yang berisi daftar kandidat yang sudah dipilih.
3. Fungsi solusi
Fungsi yang akan menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
4. Fungsi seleksi
Fungsi yang akan melakukan pemilihan kandidat berdasarkan strategi *greedy* yang sudah ditentukan sebelumnya. Dengan demikian, strategi *greedy* ini bersifat heuristik.
5. Fungsi kelayakan
Fungsi yang akan memeriksa kandidat yang dipilih apakah dapat dimasukkan ke dalam suatu himpunan solusi.
6. Fungsi objektif
Fungsi yang akan memaksimalkan atau meminimumkan sesuai dengan target yang ingin dicapai pada persoalan optimasi.

Algoritma *greedy* banyak digunakan dalam menyelesaikan permasalahan optimasi. Beberapa permasalahan optimasi berikut dapat dipecahkan dengan algoritma *greedy*, antara lain

1. Persoalan penukaran uang
2. Persoalan memilih aktivitas
3. Minimasi waktu di dalam sistem

4. Persoalan *knapsack* (*knapsack problem*)
5. Penjadwalan *Job* dengan tenggat waktu
6. Pembentukan pohon merentang minimum
7. Pencarian lintasan terpendek
8. Pembentukan Kode Huffman
9. Persoalan Pecahan Mesir

```
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
```

Gambar 2.1. Skema umum algoritma *greedy*

Sumber:

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf)

B. Wadah

Secara bahasa, wadah adalah tempat untuk menyimpan atau menaruh sesuatu. Wadah dapat dikategorikan sebagai alat dasar yang meliputi ruang yang tertutup sebagian atau tertutup total untuk menyimpan dan memindahkan barang. Dalam konteks modern, wadah tidak hanya melindungi isinya dari guncangan dan kelembaban, tetapi juga mudah untuk digunakan, misalnya membuka dan menutup wadah, memasukkan dan mengeluarkan isi wadah, dan membawa wadah untuk dipindahkan.

III. PEMBAHASAN PENATAAN BARANG DENGAN ALGORITMA GREEDY

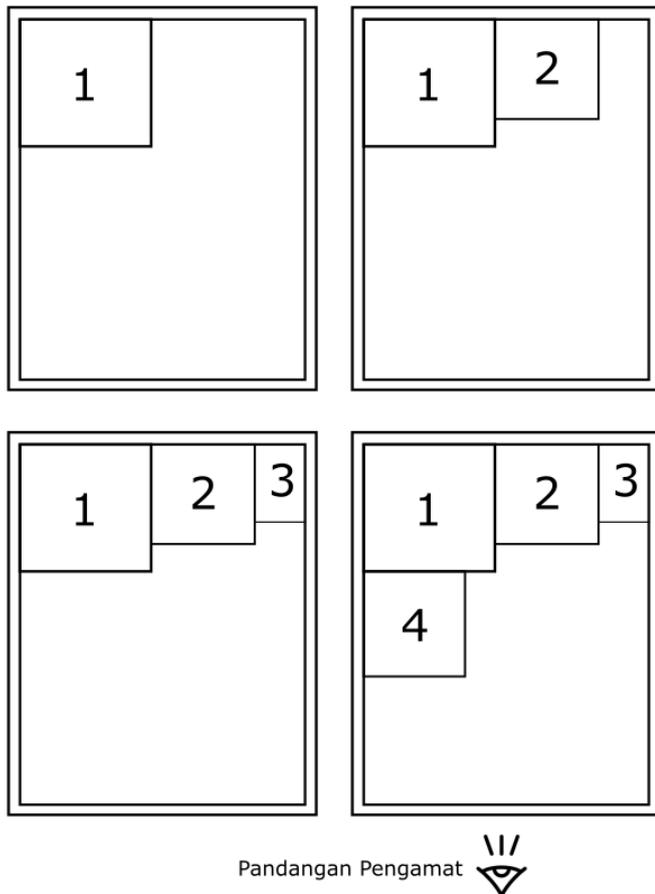
Konfigurasi penataan barang akan berpengaruh pada ruang yang dibutuhkan oleh barang-barang untuk disimpan di sebuah wadah. Dengan penataan yang baik, wadah dapat menampung lebih banyak barang dengan sisa ruang sekecil mungkin dan membuat penyimpanan barang lebih rapi. Dalam pembahasan ini, strategi penyusunan barang akan disusun dengan algoritma *greedy*.

A. Batasan Masalah

Dalam persoalan penataan barang di wadah, akan diterapkan berbagai aturan penataan barang, yaitu

- Wadah yang akan digunakan berbentuk persegi atau persegi panjang
- Panjang dan lebar dari semua barang yang akan ditaruh di wadah tidak boleh melebihi panjang atau lebar dari wadah yang akan ditempati

- Barang tidak dapat disusun bertumpukan dengan barang lainnya
- Barang tidak dapat diletakkan keluar dari wadah
- Barang tidak dapat disusun miring menurut wadah
- Barang disusun secara memanjang ke samping atau ke belakang. Selama sisa ruang di atas masih mencukupi, maka diusahakan barang disusun memanjang ke samping.
- Peletakan barang akan dilakukan dengan sisi wadah yang lebih pendek menghadap ke pengamat dan dimulai dari pojok kiri sisi belakang menurut pengamat
- Peletakan barang dilakukan dari pojok kiri sampai ke kanan selama “baris” belum penuh. Jika “baris” sudah penuh, maka pengisian barang dimulai lagi dari pojok kiri dengan posisi menempel barang pertama pada “baris” sebelumnya.
- Posisi barang yang baru diusahakan diletakkan berimpitan dengan objek sebelumnya dengan orientasi kiri atas. Jika tidak memungkinkan, barang yang baru dapat disusun berimpitan dengan ujung terdepan pada “baris” sebelumnya.



Gambar 3.1. Skema pengisian barang di wadah persegi panjang
 Sumber: Dokumen pribadi

B. Tinjauan Algoritma Greedy

Dalam pembahasan ini, strategi *greedy* yang akan akan berfokus pada dimensi barang yang akan dimasukkan. Mengingat dimensi barang yang akan dimasukkan diperhatikan dari panjang dan lebar barang, maka ada sejumlah parameter heuristik yang akan diperhatikan dalam menerapkan algoritma ini, yaitu

1. *Greedy by largest base area*, yaitu strategi *greedy* yang akan mengutamakan barang dengan luas alas terbesar terlebih dahulu memasuki wadah.
2. *Greedy by longest length*, yaitu strategi *greedy* yang akan mengutamakan barang dengan sisi yang lebih panjang yang paling panjang. Jika barang memiliki panjang yang sama, maka benda yang lebih lebar akan masuk terlebih dahulu
3. *Greedy by largest length-to-width ratio*, yaitu strategi *greedy* yang akan mengutamakan benda yang memiliki rasio panjang banding lebar yang paling besar. Jika rasio panjang banding lebar dua benda adalah sama, maka benda yang akan dimasukkan terlebih dahulu adalah benda yang memiliki luas alas terbesar.

C. Penerapan Algoritma Greedy

Berdasarkan batasan masalah dan tinjauan algoritma yang telah dipaparkan, algoritma *greedy* dapat diterapkan dengan langkah-langkah berikut:

1. Sortir barang sesuai prioritas barang yang akan dimasukkan ke dalam wadah menurut heuristik algoritma *greedy* yang telah ditentukan.
2. Tinjau titik terbelakang dari pandangan pengamat. Jika ada, ambil titik terkiri sebagai patokan untuk memasukkan barang.
3. Dari titik tersebut, ambil titik kiri belakang dari pandangan pengamat pada barang sebagai patokan dalam memasukkan barang. Hadapkan sisi barang yang lebih panjang ke mata pengamat. Jika barang tersebut diletakkan pada wadah dalam posisi tersebut tidak membuat barang keluar dari wadah atau menumpuk barang lainnya, maka taruh barang tersebut.
4. Jika tidak memungkinkan, maka coba putar barang tersebut sejauh 90 derajat. Jika barang tersebut diletakkan pada wadah dalam posisi yang sudah diputar tidak membuat barang keluar dari wadah atau menumpuk barang lainnya, maka taruh barang tersebut.
5. Jika masih tidak memungkinkan, ambil posisi terdepan dari susunan barang yang telah masuk ke wadah. Taruh barang ke wadah dengan sisi belakang menempel dengan barang yang sudah dimasukkan di belakangnya dan impitkan barang ke sisi kiri wadah jika barang tersebut ditaruh tidak membuat barang tersebut keluar wadah atau menumpuk benda lain.

6. Sebaliknya, jika barang sudah tidak memungkinkan untuk ditaruh dalam susunan apapun, maka tolak barang tersebut.
7. Lakukan langkah 2 sampai langkah 6 pada barang lainnya sampai semua barang sudah dicoba.

D. Analisis Elemen Algoritma Greedy

Berdasarkan algoritma *greedy* yang telah dipaparkan sebelumnya, terdapat elemen-elemen algoritma *greedy* dengan detail seperti berikut.

1. Himpunan Kandidat

Himpunan kandidat pada persoalan ini adalah himpunan berisi semua barang yang akan dimasukkan ke wadah.

2. Himpunan Solusi

Himpunan solusi pada persoalan ini adalah sebuah himpunan berisi daftar barang apa saja yang dimasukkan ke dalam wadah beserta orientasi penyusunan barang (potret atau lansekap) dan posisi ujung kiri atas sebagai patokan posisi barang. Sebagai tambahan, himpunan solusi ini dapat digambarkan sebagai tata letak barang dalam wadah dalam bentuk visual.

3. Fungsi Solusi

Fungsi solusi pada persoalan ini adalah mengecek apakah masih ada barang yang bisa dimasukkan ke dalam wadah menurut fungsi seleksi.

4. Fungsi Seleksi

Fungsi seleksi pada persoalan ini adalah untuk memutuskan apakah barang layak dimasukkan ke dalam wadah dan di posisi mana barang tersebut diletakkan menurut ujung kiri belakang barang serta orientasi penyusunan barang apakah dalam bentuk potret atau lansekap.

5. Fungsi Kelayakan

Fungsi kelayakan yang digunakan pada persoalan ini adalah untuk mengecek apakah barang yang akan diletakkan menurut fungsi seleksi tidak menyebabkan penumpukan barang satu dengan barang lainnya atau membuat barang yang diletakkan keluar wadah.

6. Fungsi Objektif

Fungsi objektif yang digunakan pada persoalan ini adalah untuk mengecek manakah solusi yang menghasilkan sisa ruang kosong pada wadah yang sudah ditempati sekecil mungkin.

Pseudocode dari algoritma ini adalah sebagai berikut:

```

type Barang: <panjang: int, lebar: int>
type Wadah: <panjang: int, lebar: int>
type Posisi: <x: int, y: int>
type tupleBarang: <barang: Barang, posisi: Posisi, orientasi: String>

{ Prosedur isi_ruang dan fungsi cek_ruang dianggap sudah terdefinisi sebelumnya. }
{ Prosedur isi_ruang mengisi ruang pada wadah dengan barang yang dimasukkan pada patokan posisi tertentu, dalam hal ini adalah posisi pojok kiri belakang barang serta orientasi penyusunan apakah berbentuk potret atau lansekap }
{ Fungsi cek_ruang mengecek apakah ada ruang kosong pada wadah dengan barang yang akan dimasukkan dengan patokan titik mulai pada posisi tertentu, dalam hal ini adalah calon posisi pojok kiri belakang barang serta orientasi penyusunan apakah berbentuk potret atau lansekap }

function taruhBarang(wadah: Wadah, himpunan_kandidat: list of Barang) -> list of tupleBarang
{ Fungsi ini menerima masukan berupa himpunan kandidat yakni semua barang yang tersedia dan ukuran wadah yang akan ditempati. Fungsi ini mengembalikan daftar tupleBarang berupa barang yang akan menempati, posisi patokan, dan orientasi peletakan barang }

Deklarasi
solusi: list of tupleBarang
sorted_barang: list of Barang
tupleBarang: tupleBarang
petak: list of list of char
x, y: integer

Algoritma
solusi <- []
for y from 0 to wadah.lebar - 1
  for x from 0 to wadah.panjang - 1
    petak[y][x] <- ' '
  end for
end for
sorted_barang <- sortir(himpunan_kandidat)
x <- 0
y <- 0
{ Fungsi sortir dilakukan berdasarkan heuristik }
for each (barang in sorted_barang)
  { cek apakah barang bisa diletakkan secara landscape }
  if (barang.panjang + x <= wadah.panjang and barang.lebar + y <= wadah.lebar and cek_ruang(barang, x, y, 'landscape')) then
    tupleBarang <- (barang, Posisi(x, y), 'landscape')
    isi_ruang(petak, Posisi(x, y), barang, 'landscape')
    solusi <- solusi + [tupleBarang]
    x <- x + barang.panjang
  end if
  { cek apakah barang bisa diletakkan secara portrait }
  if (barang.lebar + x <= wadah.panjang and barang.panjang + y <= wadah.lebar and cek_ruang(barang, x, y, 'portrait')) then
    tupleBarang <- (barang, Posisi(x, y), 'portrait')
    isi_ruang(petak, Posisi(x, y), barang, 'portrait')
    solusi <- solusi + [tupleBarang]
    x <- x + barang.lebar
  end if
  { cek dimensi barang berikutnya jika tidak bisa diletakkan }
  { geser y sampai didapat ruang yang cukup untuk panjang barang berikutnya }
end for

-> solusi

```

Gambar 3.1. Algoritma utama program Greedy

E. Percobaan Penerapan Algoritma Greedy

Dalam menjalankan algoritma *greedy*, akan digunakan dua kasus tes, yaitu wadah ukuran 20 x 15 dengan sepuluh benda acak dan wadah ukuran 15 x 20 dengan sepuluh benda acak.

yang paling sedikit dengan hanya menyisakan 111 petak kosong pada wadah berukuran 20 x 15 petak.

Sementara untuk kasus tes kedua, digunakan tiga algoritma *greedy* yang sama untuk mengisi barang-barang ke dalam wadah berukuran 15 x 20 petak.

paling sedikit dengan hanya menyisakan 49 petak pada wadah berukuran 15 x 20 petak.

Dari kedua kasus tes yang diujikan, algoritma *greedy by longest length* adalah algoritma yang paling efisien dalam memanfaatkan ruang pada wadah untuk menyimpan barang.

TABEL III. HASIL PENGUJIAN KASUS TES KEDUA

Nomor Subtes	Parameter Tes		
	Heuristik yang Digunakan	Hasil Visual	Dimensi dan Posisi Barang
2A	<i>Greedy by largest base area</i>	<pre> ##### AAAAAAAAAAAA IIIIIIII ##### </pre>	12 x 10 pada (0,0), landscape
			11 x 9 pada (0,10), landscape
			9 x 1 pada (0,19), landscape
			72 petak kosong
2B	<i>Greedy by largest length-to-width ratio</i>	<pre> ##### AAAAAAAAABBCC EEEEEE BBBCC EEEEEE BBBCC EEEEEE BBBCC GGG BBBCC GGG BBB BBB BBB BBB BBB BBB ##### </pre>	9 x 1 pada (0,0), landscape
			12 x 3 pada (9,0), portrait
			5 x 2 pada (12,0), portrait
			6 x 3 pada (0,1), landscape
			3 x 2 pada (0,4), landscape
			221 petak kosong
2C	<i>Greedy by longest length</i>	<pre> ##### AAAAAAAAAAAA GGGGGGGGH##### JJJ H##### JJJ H##### ##### </pre>	14 x 7 pada (0,0), landscape
			12 x 10 pada (0,7), landscape
			9 x 1 pada (0,17), landscape
			6 x 4 pada (9,17), landscape
			3 x 2 pada (0,18), landscape
			49 petak kosong

Berdasarkan tiga algoritma *greedy* yang digunakan, algoritma *greedy by longest length* menghasilkan sisa ruang

IV. KESIMPULAN

Algoritma *greedy* adalah sebuah algoritma untuk memecahkan masalah optimasi dengan memilih pilihan terbaik di tiap langkahnya dengan harapan tiap langkah yang diambil menyusun solusi optimal secara global. Masalah penataan barang di wadah persegi panjang terbuka merupakan salah satu contoh penerapan algoritma *greedy* karena masalah ini membutuhkan cara untuk mengoptimalkan ruang yang tersedia dalam wadah dengan sebisa mungkin menyisakan ruang kosong yang paling kecil. Algoritma *greedy* menjadi salah satu strategi yang tepat dalam menyusun barang di dalam wadah karena algoritma ini akan memilih barang mana yang akan disimpan dan bagaimana cara peletakan barang tersebut dengan melihat ukuran dari barang yang akan disimpan sehingga penyusunan barang berlangsung lebih efisien dan tidak boros tempat.

Walaupun penggunaan algoritma *greedy* sudah cukup menghasilkan solusi yang baik, tetapi solusi yang diberikan masih kurang baik karena pada kasus tertentu, ruang kosong yang masih tersisa masih sangat mungkin untuk ditempati barang baru dengan menata ulang sejumlah barang yang telah ditaruh di dalam wadah.

Harapannya, program ini bisa ditingkatkan dengan metode penataan barang yang lebih baik dan peninjauan ruang kosong dapat dimaksimalkan untuk memberi ruang yang cukup pada barang baru yang akan dimasukkan ke dalam wadah.

UCAPAN TERIMA KASIH

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas rahmat dan petunjuk-Nya yang membuat penulis bisa menyelesaikan penulisan makalah ini. Penulis berterima kasih kepada semua pihak yang telah membantu penulis, baik secara langsung maupun secara tidak langsung dalam penulisan makalah ini. Penulis juga mengucapkan terima kasih sebesar-besarnya atas bimbingan dari para dosen pengampu Mata Kuliah Strategi Algoritma: Bu Ulfa, Bu Masayu, dan Pak Rinaldi selama satu semester ini yang sangat membantu penulis dalam menyelesaikan makalah ini.

REFERENSI

- [1] [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)
- [2] "Wadah," *Hasil Pencarian - KBBI Daring*. [Online]. Available: <https://kbbi.kemdikbud.go.id/entri/wadah>. [Accessed: 22-May-2022].
- [3] J. A. Moore and A. S. Keene, *Archaeological Hammers and Theories*. New York: Academic Press, 1983.
- [4] C. Gamble, *Origins and revolutions: Human identity in earliest prehistory*. Cambridge: Cambridge University Press, 2009.

- [5] A. F. Buono and H. Savall, Socio-economic intervention in organizations: The intervener-researcher and the seam approach to organizaional analysis. Charlotte, NC: IAP, 2007.
- [6] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
- [7] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].

Bandung, 23 Mei 2022



Andika Naufal Hilmy 13520098

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.