

# Penerapan Algoritma Greedy Untuk Memenangkan Permainan Dots and Boxes

Andreas Indra Kurniawan - 13520091

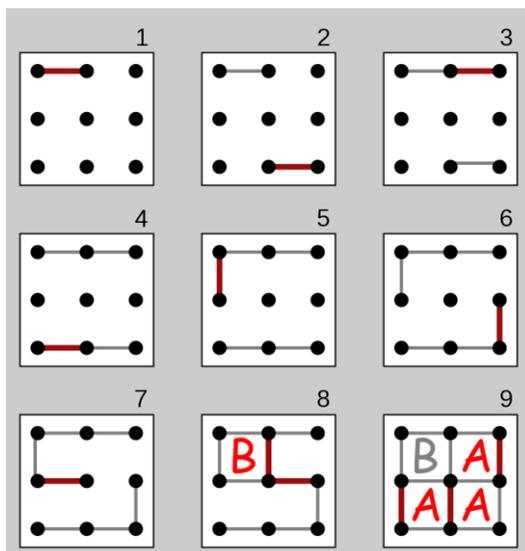
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail (gmail): 13520091@std.stei.itb.ac.id

**Abstract**—Permainan merupakan salah satu sarana hiburan bagi sebagian besar orang. Untuk memenangkan sebuah permainan biasanya diperlukan strategi yang tepat agar permainan bisa dimenangkan, contohnya *permainan Dots and Boxes* yang diangkat pada makalah ini. Dengan penggunaan strategi yang tepat maka kemenangan dalam *permainan Dots and Boxes* sudah hampir pasti bisa diprediksi. Oleh karena itu makalah ini akan memilih strategi (sebagai fungsi heuristic) yang akan digunakan untuk memenangkan *permainan Dots and Boxes* dan mengimplementasikannya menggunakan algoritma *Greedy*.

**Keywords**—*Greedy*, *Dots and Boxes*, Strategi

## I. PENDAHULUAN

Permainan merupakan sarana rekreasi yang diciptakan oleh manusia untuk menghabiskan waktu luang. Ada bermacam-macam *genre* dari permainan seperti *action*, *strategy*, *simulation*, dan *MOBA*. Masing-masing memiliki suatu strategi yang perlu digunakan agar dapat mencapai tujuan. *Dots and Boxes* merupakan salah satu permainan strategi yang dapat dimainkan secara digital maupun tradisional dengan tujuan mendapatkan kotak terbanyak.



Gambar 1 Permainan Dots and Boxes

Sumber : [https://en.wikipedia.org/wiki/Dots\\_and\\_Boxes](https://en.wikipedia.org/wiki/Dots_and_Boxes)

Permainan *Dots and Boxes* dimainkan oleh 2 orang. Papan permainan terdiri dari titik-titik yang berada pada papan

segiempat. Ukuran dari papannya sendiri beragam mulai dari 3x3 hingga sebesar mungkin. Pemain pertama akan membuat garis yang menghubungkan dua titik. Lalu pemain kedua akan melakukan hal yang sama. Sebuah kotak akan dimiliki oleh pemain jika pemain tersebut merupakan orang terakhir yang membuat garis menghubungkan 4 titik menjadi satu kotak. Pemain yang berhasil membuat satu kotak akan mendapatkan giliran kedua dan diperbolehkan untuk membuat garis selanjutnya. Permainan akan berakhir ketika seluruh titik yang ada telah dihubungkan oleh garis. Pemain dengan kotak terbanyak akan memenangkan pertandingan.

Karena sifat dari game yang *turn-based* (per giliran), maka penulis akan menggunakan pendekatan *Greedy* untuk membuat bot yang dapat memainkan permainan *Dots and Boxes* dengan cukup baik.

## II. LANDASAN TEORI

### A. Algoritma Greedy

Algoritma *Greedy* merupakan langkah-langkah pemecahan persoalan dengan mengambil pilihan terbaik (optimum lokal) pada saat itu tanpa memperhitungkan apa yang akan terjadi ke depannya. Pengambilan optimum lokal ini diharapkan akan menuntun program pada solusi optimum global.

*Greedy* sering digunakan untuk persoalan optimasi seperti masalah penukaran uang, masalah pemilihan aktivitas, dan minimisasi waktu di dalam sistem. Hal yang perlu diperhatikan dalam algoritma *Greedy* adalah algoritma ini tidak menjamin akan memberikan hasil yang paling optimal pada seluruh permasalahan karena yang diambil hanyalah optimum lokal dari setiap langkah. Hal ini diakibatkan sifat dari algoritma *Greedy* yang tidak bisa kembali lagi setelah mengambil keputusan. Namun kelebihan dari algoritma ini adalah kecepatannya dalam mengambil keputusan meskipun hasilnya tidak seoptimal algoritma lain seperti *backtracking* atau *branch and bound* atau algoritma lainnya yang melakukan pengecekan ke banyak node. Oleh karena itu *greedy* cocok digunakan untuk permasalahan yang tidak membutuhkan hasil paling optimum dan dibutuhkan waktu proses yang relative cepat seperti contohnya pada pembuatan bot permainan *Dots and Boxes*.

Algoritma ini dibangun oleh 6 elemen yaitu:

- Himpunan kandidat

Himpunan berisi kandidat yang akan dipilih untuk setiap permasalahan.

b. Himpunan solusi

Himpunan berisi kandidat yang terpilih setelah disaring oleh fungsi.

c. Fungsi solusi

Fungsi yang digunakan untuk mengecek jika himpunan kandidat merupakan solusi yang dicari.

d. Fungsi seleksi

Fungsi yang digunakan untuk memilih kandidat menggunakan strategi *Greedy*. Strategi *Greedy* yang digunakan memiliki sifat heuristic.

e. Fungsi kelayakan

Fungsi yang digunakan untuk mengecek jika himpunan kandidat yang dipilih dapat dimasukkan ke himpunan solusi.

f. Fungsi obyektif

Fungsi yang digunakan untuk meminimumkan atau memaksimumkan hasil dari permasalahan yang diinginkan.

Berikut disertakan skema umum dari 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
  
```

Gambar 2 Pseudocode Skema Umum 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)

B. Heuristik

Heuristik diambil dan diturunkan dari Bahasa Yunani yaitu “eureka” yang memiliki arti “menemukan”. Dengan arti tersebut Heuristik bisa diartikan sebagai seni dan ilmu menemukan sesuatu.

Teknik yang digunakan pada heuristik biasanya didapatkan dari pengalaman seseorang dalam mengerjakan sesuatu sehingga teknik ini tidak menjamin akan memberikan solusi yang paling optimal. Heuristik ini sendiri tidak bisa disamakan dengan algoritma karena heuristik merupakan suatu panduan pengerjaan yang bisa diambil agar suatu algoritma dapat berjalan dengan sedikit lebih baik sedangkan algoritma merupakan langkah-langkah untuk menyelesaikan persoalan.

Sebuah heuristik yang baik dapat mempersingkat waktu pengerjaan persoalan secara drastis dengan cara mengeliminasi kombinasi solusi yang sebenarnya tidak perlu diambil. Meskipun heuristik tidak dapat menjamin hasil paling optimal, teknik ini cukup baik dalam menyelesaikan persoalan khususnya ketika mencari solusi menggunakan exhaustive search. Heuristik merupakan teknik yang sering digunakan pada bidang intelegensia buatan contohnya pada algoritma A\*.

C. State dari Kotak

Kondisi dari kotak dapat dibagi menjadi 2 yaitu kotak yang aman diisi serta kotak yang tidak aman untuk diisi. Kotak yang aman diisi berarti kotak tersebut memiliki kurang dari 2 sisi yang tergambar. Kotak disebut tidak aman untuk Digambar ketika terdapat 2 atau lebih sisi yang telah digambar karena ketika kita menggambar di kotak ini lawan akan mendapat kesempatan untuk mengambil kotak tersebut.

Kotak ini sendiri bisa saja terhubung dengan kotak lainnya sehingga membentuk rantai kotak. Syarat terjadinya rantai kotak ini adalah kotak tersebut berada pada kondisi tidak aman diisi yang menyambung dengan kotak lainnya seperti pada Gambar 3 pada bab III.

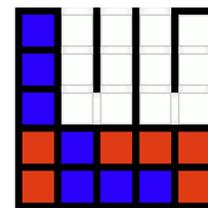
Untuk penghitungan pada bab III maka satu rantai kotak akan dihitung sebagai satu kesatuan sebagai kotak yang tidak aman untuk ditulis.

III. ANALISIS DAN PEMBAHASAN

Dots and Boxes memiliki tujuan menghubungkan setiap titik dengan garis sehingga pemain memiliki jumlah kotak terbanyak ketika seluruh titik telah dihubungkan. Oleh karena itu pemain perlu berhati-hati agar tidak membuat garis yang membentuk kotak yang hampir jadi.

Strategi yang umum dilakukan pada permainan ini adalah mencoba agar dapat melakukan gerakan menutup kotak di akhir karena semakin terakhir melakukan gerakan menutup kotak kemungkinan akan ada rantai garis yang berisi banyak kotak yang bisa ditutup sekaligus dalam satu giliran.

Strategi lain yang dilakukan adalah membuat rantai kotak yang belum bisa ditutup lalu memaksa pemain lain untuk mengambil langkah lebih dulu sehingga kita bisa mengambil kotak di dalam rantai kotak tersebut.



Gambar 3 Contoh Rantai Kotak(bagian yang belum diwarnai)

Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

Pada makalah ini baris akan merepresentasikan sisi pada papan. Sebagai contoh baris 1 merupakan sisi yang

menghubungkan titik teratas dan satu titik di bawahnya. Sedangkan kolom 1 merupakan sisi yang menghubungkan titik ter kiri dan satu titik di kanannya.

A. Pemetaan Elemen Greedy

Untuk membangun algoritma Greedy yang baik langkah pertama yang perlu kita ambil adalah memetakan seluruh elemen greedy yang akan digunakan dalam menyeleksi solusi dari permainan.

a. Himpunan kandidat

Seluruh sisi yang dapat digunakan untuk menghubungkan node(titik) yang ada. Jika sebuah papan memiliki m baris dan n kolom, total gerakan yang dapat dibuat dalam permainan berjumlah  $m + 2mn + n$  gerakan. Angka tersebut didapat dari  $(m+1)n$  gerakan yang mungkin dilakukan oleh sisi horizontal dan  $m(n+1)$  gerakan yang mungkin dilakukan oleh sisi vertikal.

b. Himpunan solusi

Himpunan solusi pada permainan Dots and Boxes adalah gerakan-gerakan yang dipilih pada setiap giliran pemain.

c. Fungsi solusi

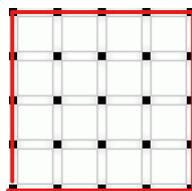
Fungsi menghitung jika jumlah kotak yang dimiliki oleh pemain lebih besar dari lawan.

d. Fungsi seleksi

Fungsi mencari jika setiap kotak yang belum jadi dapat diselesaikan dengan menambah satu sisi baru atau jika gerakan tersebut tidak akan memberikan kesempatan bagi lawan.

e. Fungsi kelayakan

Fungsi digunakan untuk mengecek jika pengguna melakukan gerakan yang legal atau tidak. Gerakan yang legal adalah pembuatan garis di bagian dalam titik-titik terluar.



Gambar 4 Range Gerakan Legal

Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

Selain itu garis yang sudah digambar tidak boleh ditimpa oleh pemain.

f. Fungsi obyektif

Fungsi digunakan untuk memaksimalkan jumlah kotak yang didapat dengan pemilihan sisi yang benar.

B. Ekplorasi Fungsi Solusi

Fungsi solusi ini digunakan untuk menentukan jika pemain masih bisa menang dari lawan dengan menghitung jumlah kotak yang dimiliki lawan serta jumlah kotak yang masih tersedia untuk diambil di papan.

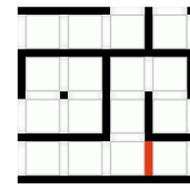
C. Eksplorasi Fungsi Seleksi

Fungsi seleksi memiliki peran yang penting dalam memenangkan permainan Dots and Boxes. Sehingga perlu menggunakan heuristik yang tepat agar dapat memberikan solusi yang optimum. Berikut strategi greedy yang akan digunakan.

a. Greedy by free box

Strategi ini pertama akan mencari sisi yang belum terisi namun tidak akan memberikan poin pada lawan. Strategi ini digunakan dengan mengecek tiap kotak apakah terdapat kurang dari sama dengan 2 sisi yang telah digambar. Jika terdapat sisi dengan kondisi di atas, maka akan dimasukkan sebagai kandidat.

Terdapat kasus khusus pada strategi ini yaitu ketika menutup kotak yang aman untuk digambar seperti pada gambar 5 akan memunculkan kotak yang tidak aman pada kotak lain. Oleh karena itu perlu dicari penggambaran yang akan membuat kotak tidak aman sekecil mungkin. Karena kotak yang masih aman tidak akan dihitung sebagai rantai kotak maka akan dicek satu persatu mulai dari kiri kanan dan atas yang mana memiliki rantai terpendek.



Gambar 5 Kasus Khusus Greedy by Free Box

Sumber :

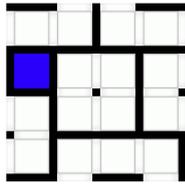
<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

b. Greedy by smallest chain box

Pada strategi ini akan dicari rantai kotak dengan jumlah kotak paling sedikit untuk ditutup. Hal ini dilakukan untuk memberikan lawan rantai kotak dengan jumlah kotak yang lebih sedikit.

Strategi ini dapat dibagi lagi menjadi 4 kasus, yaitu ketika 2 rantai terbawah jumlahnya sama, ketika 2 rantai terbawah jumlahnya berbeda, ketika hanya tersisa 2 rantai dan kasus ketika 2 rantai terbawah terdapat bentuk segiempat.

a. Kasus 2 rantai terbawah jumlahnya sama



Gambar 6 Contoh Kasus Rantai Terbawah 1  
Sumber :

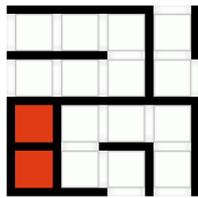
<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

Ketika terjadi hal di atas kita akan mengutamakan mengisi rantai dengan sisi yang berada di dalam dan bukan di luar hingga 2 rantai terbawah tidak lagi sama jumlahnya.

b. Kasus 2 rantai terbawah jumlahnya berbeda

Pada kasus ini jika jumlah kotak dalam rantai sama dengan 2 maka kita bisa mendapatkan rantai yang lebih banyak, namun jika jumlah kotak tidak sama dengan 2 lebih baik kita merelakan rantai dengan kotak lebih banyak agar bisa mendapat rantai yang lebih kecil 100%.

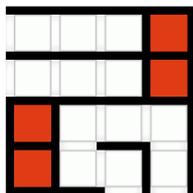
Berikut hal yang perlu dilakukan ketika terdapat rantai berjumlah 2.



Gambar 7 Contoh Kasus Rantai Terbawah 2  
Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

Kita harus memaksa lawan untuk menutup rantai yang isi kotaknya lebih banyak. Oleh karena itu kita akan memilih rantai terkecil yang ada dengan menggambar garis di dalam.

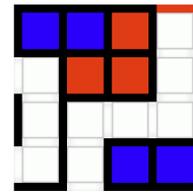


Gambar 8 Contoh Kasus Rantai Terbawah 3  
Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

Pada gambar 8 lawan terpaksa menutup sehingga kita mendapatkan rantai dengan jumlah kotak yang lebih banyak.

Untuk kasus di mana kotak rantai terkecil tidak sama dengan 2 seperti pada gambar 9 maka kita perlu merelakan rantai terbesar dan menutupnya



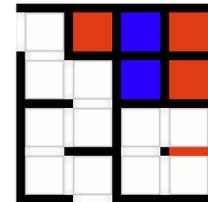
Gambar 9 Contoh Kasus Rantai Terbawah 4  
Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

Dengan begitu ada kemungkinan lawan tidak mengambil satupun rantai dengan isi 3 kotak.

c. Kasus 2 rantai terbawah terdapat bentuk segiempat

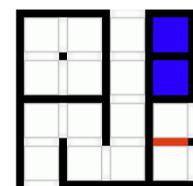
Ketika bentuk segiempat merupakan rantai terpendek maka akan menggambar sisi secara acak pada segiempat.



Gambar 10 Contoh Kasus Terdapat Bentuk Segiempat 1  
Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

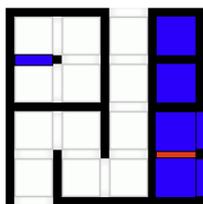
Namun ketika segiempat merupakan rantai terpanjang dari 2 rantai terpendek yang ada maka bentuk segiempat ini perlu dihindari agar bukan kita yang melakukan gerakan pertama seperti pada gambar 11.



Gambar 11 Contoh Kasus Terdapat Bentuk Segiempat 2  
Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

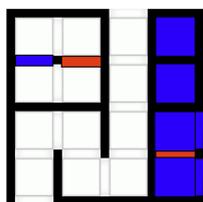
Jika setelah gerakan tersebut lawan menggambar sisi seperti pada gambar 12. Maka akan digambar sisi pada rantai segiempat sehingga segiempat terbagi 2 seperti pada gambar 13.



Gambar 12 Contoh Kasus Terdapat Bentuk Segiempat 3

Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.es.html>



Gambar 13 Contoh Kasus Terdapat Bentuk Segiempat 4

Sumber :

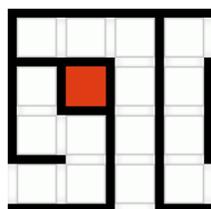
<https://www.math.ucla.edu/~tom/Games/dots&boxes.es.html>

c. *Greedy by to-be-claimed box*

Strategi ini memiliki 2 fase yaitu fase selagi masih ada free space yang bisa diisi dan fase ketika seluruh kotak sudah memiliki 2 sisi. Hal ini perlu dibedakan karena ketika seluruh kotak sudah memiliki 2 sisi, kita dapat memberikan lawan rantai kotak yang lebih besar jika tidak berhati-hati seperti pada gambar di atas.

Secara sederhana strategi ini akan mengambil kotak sebanyak mungkin tanpa memberikan lawan kesempatan untuk mengambil kotak yang lebih banyak.

Pada fase pertama ketika ada kotak yang bisa *diclaim* tidak akan dilakukan pengecekan dan program akan langsung mengambil kotak tersebut. Namun fase kedua perlu dilakukan pengecekan karena setelah mengambil box tersebut kita perlu memberikan box/rantai box kepada lawan seperti contoh pada gambar 14.



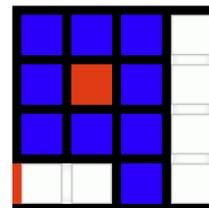
Gambar 14 Contoh Kasus *to-be-claimed-box 1*

Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

Ketika terjadi hal seperti di atas kita tentu saja menginginkan kotak sebanyak-banyaknya. Oleh karena

itu akan dicek ada berapa rantai di dalam papan. Jika ada lebih dari satu rantai dan rantai terkecil di papan jumlahnya lebih dari 2 maka kita akan melakukan manipulasi gerakan dengan cara memaksa lawan mengambil kotak yang lebih kecil seperti pada gambar 15.



Gambar 15 Contoh Kasus *to-be-claimed-box 2*

Sumber :

<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

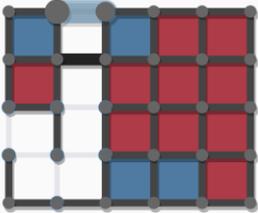
Prioritas *greedy* di atas perlu diurutkan agar mendapatkan hasil yang diinginkan. Prioritas pertama adalah *greedy by to-be-claimed box* disusul dengan *greedy by free box* dan terakhir *greedy by smallest chain box*.

D. Simulasi Algoritma Greedy

Penulis melakukan simulasi pada website <https://gametable.org/games/dots-and-boxes/> secara manual dengan mengikuti ketiga strategi *greedy* yang telah dijelaskan di atas melawan *bot* dengan kesulitan *medium*. Garis berwarna biru merupakan langkah yang diambil oleh penulis, langkah dari *bot website* akan langsung ditampilkan di langkah selanjutnya. Kotak yang diambil oleh lawan berwarna merah dan penulis berwarna biru.

Langkah	Gambar	Strategi
1		<i>Greedy by free box</i>
2		<i>Greedy by free box</i>
3		<i>Greedy by free box</i>

4		<i>Greedy by free box</i>	12		<i>Greedy by free box</i>
5		<i>Greedy by free box</i>	13		<i>Greedy by to-be-claimed box</i>
6		<i>Greedy by free box</i>	14		<i>Greedy by free box</i> (kasus special)
7		<i>Greedy by free box</i>	15		<i>Greedy by to-be-claimed box</i> (sudah tidak terdapat kotak yang aman)
8		<i>Greedy by free box</i>	16		<i>Greedy by smallest chain box</i> (Ketika bertemu keadaan di atas algoritma akan memilih memberikan chain yang lebih besar agar mendapat control pada rantai yang lebih kecil sehingga bisa mendapat rantai terpanjang satunya)
9		<i>Greedy by free box</i>	17		<i>Greedy by to-be-claimed box</i>
10		<i>Greedy by free box</i>	20		<i>Greedy by to-be-claimed box</i> (kasus khusus ketika terdapat rantai yang lebih besar)
11		<i>Greedy by free box</i>			

21		<p><i>Greedy by to-be-claimed box</i> (Seluruh rantai akan diambil karena pada kasus ini rantai hanya bersisa satu)</p>
end		<p>Game berakhir seri</p>

#### IV. KESIMPULAN DAN SARAN

Berdasarkan hasil simulasi di atas, dapat disimpulkan bahwa algoritma greedy yang digunakan sudah mencapai tingkat menengah karena berhasil seri melawan *bot medium* dari *website gametable*. Oleh karena itu algoritma *greedy* yang digunakan dinilai cukup layak sebagai lawan dalam permainan Dots and Boxes.

Agar algoritma dapat menjadi lebih bagus lagi dapat dicari kasus-kasus yang lebih spesifik sehingga dapat menangkis gerakan lawan dengan baik. Selain itu bisa juga digunakan perhitungan matematika untuk mencari tahu suatu rantai akan diambil oleh pemain mana hingga bisa dilakukan tindakan pencegahan terlebih dahulu.

#### VIDEO LINK AT YOUTUBE

[https://www.youtube.com/watch?v=9oqARx-a-pU&ab\\_channel=Andre](https://www.youtube.com/watch?v=9oqARx-a-pU&ab_channel=Andre)

#### UCAPAN TERIMAKASIH

Penulis memanjatkan puji syukur kepada Tuhan Yang Maha Esa karena berkat kehendak-Nya penulis bisa menyelesaikan makalah ini dengan baik dan tepat waktu. Penulis juga mengucapkan terima kasih kepada dosen pengampu mata kuliah Strategi Algoritma K-1 Dr. Masayu Leylia Khodra, S.T., M.T. karena berkat bimbingan beliau selama satu semester ini penulis bisa mengerti bahan *greedy* yang dibahas pada makalah ini.

Selain itu penulis ingin mengucapkan terimakasih kepada pihak-pihak yang turut membantu dalam membuat makalah ini khususnya para pemilik referensi yang diambil oleh penulis.

#### REFERENCES

- [1] [https://en.wikipedia.org/wiki/Dots\\_and\\_Boxes](https://en.wikipedia.org/wiki/Dots_and_Boxes). Diakses pada tanggal 20 Mei 2022.
- [2] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag2.pdf). Diakses pada tanggal 21 Mei 2022.
- [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 tanggal 21 Mei 2022.
- [4] <https://www.math.ucla.edu/~tom/Games/dots&boxes.html>. Diakses pada tanggal 21 Mei 2022.

#### 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, 22 Mei 2022



Andreas Indra Kurniawan 13520091