

Penerapan Strategi *Greedy* Dalam Proses *Draft* Pada Olahraga Basket

Kenny Benaya Nathan - 13521023
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13521023@std.stei.itb.ac.id

Abstrak—Proses draft merupakan salah satu tahap terpenting dalam membangun tim yang kompetitif, di mana tim memilih atlet baru dengan potensi tertinggi sehingga akan menciptakan suasana liga yang lebih seimbang. Salah satu cara yang dapat digunakan untuk pemain mana yang akan dipilih adalah dengan menggunakan strategi *greedy*. Dengan menggunakan strategi ini, setiap tim dapat memanfaatkannya untuk memilih pemain terbaik yang dapat dipilih sesuai dengan apa yang dibutuhkan oleh tim, sehingga diharapkan mampu memperbaiki performa tim pada musim-musim mendatang.

Kata Kunci—*draft; value; pemain; greedy; terbaik;*

I. PENDAHULUAN

Proses *Draft* merupakan sebuah tahap di mana setiap tim dalam suatu liga memiliki kesempatan untuk memilih atlet-atlet yang baru akan bergabung dengan liga tersebut. Beberapa liga sudah mengadopsi tahapan ini dalam keberjalanan kompetisi mereka masing-masing, beberapa diantaranya adalah *National Basketball Association* (NBA), liga basket Amerika dan merupakan liga basket terpopuler serta paling bergengsi di dunia, dan juga *Indonesian Basketball League* (IBL), liga basket yang dimiliki oleh Indonesia.

Proses draft dalam olahraga basket merupakan salah satu tahap yang krusial untuk membangun tim yang kompetitif. Pada tahap ini, tim memiliki kesempatan untuk memilih atlet-atlet baru yang dianggap memiliki potensi tinggi dan dapat berkontribusi secara signifikan dalam mencapai kesuksesan tim. Namun, dengan begitu banyaknya pilihan yang tersedia, tim perlu mengembangkan strategi yang efektif untuk memilih atlet-atlet baru yang paling cocok dengan kebutuhan mereka.

Salah satu strategi yang sering digunakan dalam proses draft adalah strategi *greedy*. Strategi ini didasarkan pada prinsip memilih atlet terbaik yang tersedia pada setiap tahap pemilihan, tanpa mempertimbangkan konsekuensi jangka panjang. Dalam strategi *greedy*, tim berfokus pada keuntungan yang tersedia beserta dengan potensi individu atlet, dengan harapan bahwa kombinasi dari atlet-atlet individual tersebut akan membentuk tim yang kuat.

Dalam tulisan ini, penulis akan mencoba untuk membuat contoh penerapan strategi *greedy* sederhana dalam proses draft pada olahraga basket untuk memilih pemain terbaik yang tersedia sesuai dengan apa yang dibutuhkan oleh suatu tim basket.

II. DASAR TEORI

A. Draft

Proses *Draft* adalah sebuah acara tahunan yang biasa diadakan oleh beberapa liga di setiap awal musim baru atau di akhir musimnya. Setiap tim dalam suatu liga bisa memilih pemain baru yang ingin bergabung dengan liga tersebut dan sudah memenuhi syarat. Tujuan diadakannya *draft* adalah untuk mencapai keseimbangan dan keadilan dalam kompetisi. Persaingan yang seimbang tentunya akan menciptakan atmosfer yang lebih kompetitif dalam suatu liga. Oleh karena itulah biasanya tim-tim yang menempati peringkat bawah pada musim sebelumnya akan memiliki kemungkinan yang lebih baik untuk bisa mendapatkan kesempatan memilih yang pertama.

Performa suatu tim dalam musim sebelumnya merupakan salah satu faktor utama yang menentukan urutan pemilihan dalam *draft*. Biasanya, urutan pemilihan akan ditentukan dari posisi tim terendah hingga yang teratas. Ada juga beberapa liga yang menerapkan proses kemungkinan. Urutan pemilihan akan ditentukan dengan *lottery*, di mana seseorang akan dengan sembarang memilih sebuah tim secara sembarang dari sebuah pot yang tersedia. Perbedaannya, tim dengan posisi terendah akan mendapatkan kemungkinan terbesar di mana pilihan untuk tim tersebut akan lebih banyak tersedia dalam pot. Dengan penentuan proses pemilihan ini, tim yang kurang sukses pada musim sebelumnya akan mendapatkan kesempatan untuk mendapatkan pemain terbaik yang tersedia untuk membangun kembali tim mereka dan meningkatkan performa tim di musim selanjutnya.



Gambar 1 Suasana NBA Draft 2022
(sumber: NBA.com)

Setelah urutan pemilihan ditentukan, tim-tim secara bergantian akan memilih seorang pemain dari daftar pemain yang tersedia sesuai dengan urutannya. Tentu saja pemain yang sudah dipilih tidak akan bisa dipilih kembali. Hal ini akan kembali mendukung fakta bahwa *draft* ini diadakan untuk memberikan keseimbangan kemampuan setiap tim dalam suatu liga.

Sebelum diadakan *draft*, setiap tim biasanya memiliki beberapa orang *scout* yang akan memperhatikan beberapa calon pemain yang kemungkinan akan masuk ke dalam liga tersebut. Selain itu, beberapa liga juga akan mengadakan *draft combine* untuk pemain-pemain yang sudah pasti akan mengikuti *draft* sebagai wadah resmi untuk melihat kemampuan para pemain secara lebih aktual. Hal ini dilakukan untuk melihat performa para pemain-pemain tersebut, sehingga informasi yang didapatkan bisa dijadikan acuan untuk menentukan strategi yang tepat dalam mengambil keputusan *draft* mendatang.

Untuk mencapai hasil yang terbaik dalam menentukan pemain untuk dipilih, ada beberapa faktor yang harus diperhatikan. Mulai dari potensi individu pemain, kemampuan pemain sekarang, aturan *salary cap* yang berlaku dalam liga tersebut, hingga peran pemain sehingga bisa menyesuaikan dengan kebutuhan tim tersebut. Dengan strategi yang baik, pemain yang dipilih bisa memberikan hasil yang terbaik juga tidak hanya dalam musim selanjutnya, namun juga pada musim-musim berikutnya hingga dalam jangka waktu yang panjang.

B. Algoritma Greedy

Algoritma *Greedy* adalah salah satu algoritma yang sederhana dan cukup populer untuk memecahkan permasalahan optimasi, di mana permasalahan tersebut mencari solusi yang paling optimal (*maximization* atau *minimization*). Sesuai dengan namanya (*greedy* yang dalam Bahasa Indonesia artinya rakus), algoritma *greedy* mengambil prinsip "*take what you can get now*".

Proses yang dilakukan algoritma *greedy* akan membentuk solusi langkah per langkah. Di setiap langkahnya, ada banyak pilihan yang perlu dievaluasi. Oleh karena itu, keputusan yang terbaik harus diambil pada setiap langkahnya dalam menentukan pilihan tanpa memperhatikan konsekuensi kedepannya (optimum lokal) dan tidak bisa mundur lagi ke langkah sebelumnya. Harapannya, langkah-langkah ini akan berakhir menuju solusi yang optimum global.

Beberapa elemen yang terdapat pada Algoritma Greedy antara lain:

1. Himpunan Kandidat
himpunan ini berisi kandidat elemen pembentuk solusi yang akan dipilih pada setiap langkah
2. Himpunan Solusi
himpunan ini berisi kandidat yang sudah terpilih sebagai solusi persoalan.
3. Fungsi Solusi

fungsi yang menentukan apakah kandidat yang dipilih sudah memberikan solusi. Fungsi ini akan mengembalikan nilai boolean yaitu true jika memberikan solusi lengkap dan false jika memberikan solusi belum lengkap.

4. Fungsi Seleksi
fungsi yang memilih kandidat yang paling mungkin untuk mencapai solusi berdasarkan strategi greedy tertentu yang bersifat heuristik.
5. Fungsi Kelayakan
fungsi yang memeriksa kelayakan kandidat yang terpilih apakah dapat dimasukkan ke dalam himpunan solusi atau tidak.
6. Fungsi Objektif
fungsi yang mengoptimalkan solusi (memaksimumkan atau meminimumkan).

Skema umum dari algoritma *greedy* dapat dilihat pada figure 1.

Salah satu hal yang harus diperhatikan dalam algoritma *greedy* adalah hasil solusi optimum global belum tentu merupakan solusi yang terbaik, bisa jadi merupakan solusi *sub-optimum* atau *pseudo-optimum*. Hal ini dikarenakan Algoritma greedy tidak beroperasi secara menyeluruh terhadap semua kemungkinan solusi yang ada, tidak seperti metode *exhaustive search*. Selain itu ada beberapa fungsi seleksi yang berbeda, sehingga harus memilih fungsi yang tepat jika ingin algoritma yang dihasilkan merupakan solusi yang paling optimal.

Secara umum, algoritma *greedy* adalah metode yang efisien dan mudah diimplementasikan. Namun, kekurangannya terletak pada kemungkinan menghasilkan solusi suboptimal atau tidak optimal secara global. Dalam beberapa situasi, solusi yang dihasilkan oleh algoritma *greedy* mungkin tidak mencapai solusi optimal secara keseluruhan. Karena itu, penting untuk memilih aturan atau kriteria dalam algoritma *greedy* dengan hati-hati, serta melakukan analisis yang lebih mendalam untuk memastikan bahwa solusi yang dihasilkan memenuhi kebutuhan dan persyaratan yang diberikan.

Meskipun demikian, algoritma *greedy* tetap memiliki kegunaan dalam memecahkan berbagai masalah yang melibatkan pengambilan keputusan bertingkat. Pada banyak kasus, algoritma greedy dapat memberikan solusi yang cukup memadai dengan tingkat efisiensi yang tinggi. Namun, perlu diingat bahwa untuk persoalan yang lebih kompleks dan membutuhkan optimisasi yang lebih baik, mungkin diperlukan penggunaan algoritma lain yang lebih canggih dan kompleks.

```

function greedy(C : himpunan_kandidat) → himpunan_solusi
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy }
Deklarasi
x : kandidat
S : himpunan_solusi
Algoritma:
S ← {} { inialisasi 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

```

Figur 1 Skema umum algoritma greedy dalam bentuk Pseudocode (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))

III. PENGIMPLEMENTASIAN ALGORITMA

A. Algoritma Program

Pada program kali ini, ada beberapa faktor yang dibatasi untuk diperhitungkan pada setiap pemain, karena pada sistem draft sesungguhnya, ada banyak sekali faktor yang harus diperhatikan. Faktor yang diperhitungkan pada setiap pemain adalah performa selama draft combine, kemampuan yang dilaporkan oleh scout, perhitungan seberapa cocok pemain tersebut terhadap tim, dan preferensi gaji yang dimiliki oleh setiap pemain yang harus disesuaikan dengan salary cap yang dimiliki oleh tim. Faktor performa, kemampuan, dan kecocokan pemain dihitung dengan skala 1-10 dengan angka 10 sebagai nilai yang terbaik.

Pada program kali ini juga, peneliti memiliki perhitungan sendiri sebagai penilaian keseluruhan value yang dimiliki oleh pemain. Perlu diperhatikan, bobot perhitungan ini merupakan perhitungan pribadi dan tidak mengacu sepenuhnya pada penilaian yang dimiliki oleh tim professional manapun.

Pada persoalan ini, elemen algoritma greedy terdiri dari:

1. Himpunan Kandidat
Himpunan pemain yang memenuhi syarat untuk mengikuti draft
2. Himpunan Solusi
Sebuah priority queue dari pemain yang sudah diurutkan berdasarkan perhitungan value yang dimiliki oleh pemain.
3. Fungsi Solusi
Memeriksa apakah seluruh pemain sudah dievaluasi dan sudah diurutkan berdasarkan value nya.
4. Fungsi Seleksi
Memilih pemain dengan perhitungan value yang tertinggi.
5. Fungsi Kelayakan
Memeriksa apakah preferensi gaji yang dimiliki pemain sudah mencukupi salary cap yang dimiliki oleh tim
6. Fungsi Objektif

Kandidat yang dipilih diurutkan berdasarkan nilai value yang tertinggi

Secara garis besar, algoritma pada program ini adalah sebagai berikut:

1. Menghitung value yang dimiliki oleh setiap pemain. Rumus dari perhitungan value tersebut adalah

$$v = 35\% * p + 35\% * k + 30\% * c$$

di mana

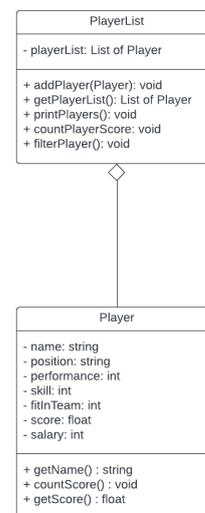
- v = value pemain
- p = performa draft combine
- k = kemampuan pemain
- c = kecocokan dalam tim

2. Mengurutkan pemain berdasarkan value yang dimiliki mulai dari yang terbesar hingga yang terkecil
3. Memeriksa apakah salary cap mencukupi preferensi gaji yang dimiliki oleh setiap pemain. Apabila tidak mencukupi, maka pemain tersebut akan dihapus dari priority queue

Hasil dari program ini adalah sebuah priority queue. Jadi apabila pemain yang ada sudah dipilih oleh tim lain, maka pemain tersebut akan dihapus. Jika sudah ada di giliran tim tersebut, maka pemain yang dipilih adalah pemain pada index ke-0, di mana pemain yang memiliki value tertinggi bagi tim tersebut.

B. Lingkungan Implementasi

Pengimplementasian dilakukan pada lingkungan Bahasa pemrograman Python versi 3.11.1. Pengimplementasian juga menggunakan paradigma pemrograman objek. Kelas yang dibuat terdiri dari 2 kelas, yaitu kelas Player dan PlayerList.



Gambar 2 Diagram Kelas Program draft decider (sumber: pribadi)

C. Hasil Implementasi

Berikut merupakan hasil pengimplementasian dari perancangan *draft decider*.

```
class Player:
    def __init__(self, name, performance, skill, position,
fitInTeam, salary):
        self.name = name
        self.position = position
        self.performance = performance
        self.skill = skill
        self.fitInTeam = fitInTeam
        self.score = 0
        self.salary = salary

    def __str__(self):
        return self.name + " pos: " + self.position + " perf: " +
str(self.performance) + " skill: " + str(self.skill) + " fit: " +
str(self.fitInTeam) + " salary: " + str(self.salary)

    def getName(self):
        return self.name

    def countScore(self):
        self.score = (35 * self.performance) + (35 / 100 *
self.skill) + (30 / 100 * self.fitInTeam)

    def getScore(self):
        return self.score

class PlayerList:
    def __init__(self, players):
        self.playerList = players

    def addPlayer(self, player):
        self.playerList.append(player)

    def getPlayerList(self):
        return self.playerList

    def printPlayers(self):
        for player in self.playerList:
            print(player)
            print("Score: " + str(player.getScore()) + "\n")

    def countPlayerScore(self):
        for player in self.playerList:
            player.countScore()

    def filterPlayer(self):
        self.playerList.sort(key=lambda x: x.getScore(),
reverse=True)
        for player in self.playerList:
```

```
if player.salary > salaryCap:
    self.playerList.remove(player)
```

Figur 2 Perancangan Kelas pada program *draft decider* (sumber: pribadi)

D. Pengujian

Untuk mengujikan algoritma ini, maka dibuatlah sebuah fungsi *main* dari program tersebut yakni sebagai berikut

```
if __name__ == "__main__":
    salaryCap = 100

    # Create Eligible Players
    Player1 = Player("Player1", 8, 9, "PG", 7, 90)
    Player2 = Player("Player2", 7, 8, "SG", 8, 75)
    Player3 = Player("Player3", 9, 7, "SF", 9, 120)
    Player4 = Player("Player4", 6, 9, "PF", 7, 50)
    Player5 = Player("Player5", 8, 7, "C", 8, 105)

    players = [Player1, Player2, Player3, Player4, Player5]
    playerList = PlayerList(players)

    # print the drafted players
    playerList.countPlayerScore()
    print("ELIGIBLE PLAYERS TO DRAFT\n")
    playerList.printPlayers()

    print("=====")

    # Sort the players based on their score and filter the
players
    playerList.filterPlayer()

    # print the Priority Players Queue to draft
    print("PRIORITY PLAYERS TO DRAFT\n")
    playerList.printPlayers()
```

Figur 3 Fungsi *main* program *draft decider* (sumber: pribadi)

Hasil dari fungsi *main* ini dapat dilihat pada gambar di bawah ini.

```
ELIGIBLE PLAYERS TO DRAFT
Player1 pos: PG perf: 8 skill: 9 fit: 7 salary: 90
Score: 285.25
Player2 pos: SG perf: 7 skill: 8 fit: 8 salary: 75
Score: 250.20000000000002
Player3 pos: SF perf: 9 skill: 7 fit: 9 salary: 120
Score: 320.15
Player4 pos: PF perf: 6 skill: 9 fit: 7 salary: 50
Score: 215.25
Player5 pos: C perf: 8 skill: 7 fit: 8 salary: 105
Score: 284.84999999999997
=====
PRIORITY PLAYERS TO DRAFT
Player1 pos: PG perf: 8 skill: 9 fit: 7 salary: 90
Score: 285.25
Player2 pos: SG perf: 7 skill: 8 fit: 8 salary: 75
Score: 250.20000000000002
Player4 pos: PF perf: 6 skill: 9 fit: 7 salary: 50
Score: 215.25
```

Gambar 1 Hasil pengujian Program *draft decider* (sumber: pribadi)

Pada fungsi *main* sesuai dengan figur 3, diberikan *salary cap* dari sebuah tim yaitu 100. Kemudian, dibuatlah lima orang pemain yang memenuhi syarat untuk mengikuti *draft*. Setelah itu, dibuatlah sebuah *Instance PlayerList* yang terdiri dari kelima pemain tersebut.

Sesuai dengan alur algoritma yang sudah dijelaskan sebelumnya, program akan menghitung terlebih dahulu nilai yang dimiliki oleh pemain. Kemudian, program akan melakukan sorting pada *PlayerList*. Setelah itu, program akan menguji fungsi kelayakannya, yaitu apakah *salary cap* mencukupi preferensi gaji yang dimiliki oleh pemain.

Terlihat hasil pengujiannya pada gambar, ada 3 orang yang bisa dipilih oleh tim tersebut. Diurutkan berdasarkan nilai *value*-nya, maka Player1, Player2, dan Player4 lah kandidat pemain yang akan dipilih. Preferensi gaji yang juga dimiliki oleh masing-masing pemain sudah mencukupi *salary cap* yang dimiliki oleh tim. Jika Player1 nantinya sudah dipilih oleh tim lain, maka tim akan memerhatikan Player2. Apabila sudah mendapatkan giliran, maka tim baru akan memilih Player2.

IV. SIMPULAN DAN SARAN

Dari hasil pengimplementasian program *draft decider*, dapat ditarik kesimpulan sebagai berikut,

- Dalam proses draft pada olahraga basket, penerapan strategi greedy dapat menjadi pendekatan yang efisien untuk memilih atlet baru. Strategi ini berfokus pada memilih atlet terbaik yang tersedia pada setiap tahap pemilihan, tanpa mempertimbangkan konsekuensi jangka panjang atau memperhatikan kebutuhan tim secara keseluruhan. Meskipun strategi greedy

memiliki keuntungan dalam memperoleh atlet berkualitas tinggi secara cepat, namun terdapat kelemahan potensial yaitu dapat menghasilkan solusi yang suboptimal secara global.

- Dalam beberapa kasus, solusi yang dihasilkan oleh strategi greedy mungkin tidak mencapai solusi yang optimal secara keseluruhan. Oleh karena itu, pemilihan aturan atau kriteria dalam strategi greedy harus dilakukan dengan hati-hati, dengan mempertimbangkan kebutuhan dan persyaratan tim secara keseluruhan. Analisis yang lebih mendalam diperlukan untuk memastikan bahwa solusi yang dihasilkan sesuai dengan tujuan jangka panjang tim.

Dari hasil pengimplementasian ini pula ada beberapa saran yang mampu diberikan oleh peneliti baik kepada penelitian selanjutnya maupun kepada tim-tim yang akan melakukan proses *draft*. Saran-saran yang dapat diberikan sebagai berikut,

- Pertimbangkan Kebutuhan Tim secara Keseluruhan: Saat menerapkan strategi greedy dalam proses draft, penting untuk tidak hanya mempertimbangkan kemampuan individu atlet, tetapi juga memperhatikan kebutuhan dan keseimbangan tim secara keseluruhan. Pemilihan atlet harus dilakukan dengan mempertimbangkan peran dan posisi yang diperlukan untuk memperkuat tim secara keseluruhan.
- Evaluasi Potensi Jangka Panjang: Meskipun strategi greedy cenderung fokus pada keuntungan instan, penting untuk mempertimbangkan potensi jangka panjang atlet yang dipilih. Pertimbangkan kemampuan mereka untuk berkembang dan beradaptasi dengan permainan serta kontribusi jangka panjang mereka dalam mencapai kesuksesan tim.
- Gabungkan dengan Pendekatan Lain: Strategi greedy dapat digabungkan dengan pendekatan lain, seperti strategi berbasis kebutuhan tim atau strategi yang mempertimbangkan aspek-aspek lain seperti keseimbangan tim dan kemampuan kolaboratif. Menggabungkan pendekatan ini dapat membantu mencapai hasil yang lebih optimal dalam proses draft.
- Analisis Lebih Lanjut dan Penelitian: Untuk memastikan keberhasilan strategi greedy dalam proses draft, diperlukan analisis yang lebih mendalam dan penelitian terkait potensi atlet, kebutuhan tim, dan evaluasi kinerja jangka panjang. Dengan pemahaman yang lebih baik tentang faktor-faktor ini, tim dapat membuat keputusan yang lebih cerdas dan mengoptimalkan hasil draft mereka.

UCAPAN TERIMA KASIH

Puji dan syukur diucapkan kepada Tuhan yang Maha Esa sehingga makalah ini dapat diselesaikan. Tidak lupa kepada Orang tua penulis yang telah mendukung proses Pendidikan penulis sehingga makalah ini dapat disusun, kepada Ir. Rila Mandala, M. Eng., Ph.D. sebagai dosen pengampu IF2211 yang telah menurunkan ilmunya tentang Matematika Diskrit dan kepada seluruh teman-teman yang telah menyemangati proses pembuatan makalah ini.

REFERENSI

- [1] Munir, R. (2021) Bahan Kuliah IF2211 strategi algoritma algoritma greedy bagian (1). Available at: [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) (Accessed: 21 May 2023).
- [2] Munir, R. (2021) Bahan Kuliah IF2211 strategi algoritma algoritma greedy bagian (2). Available at: [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). (Accessed: 21 May 2023).
- [3] PLAYER ELIGIBILITY AND NBA DRAFT. Available at: <https://web.archive.org/web/20130907202714/http://nbpa.com/sites/nbpa.org/files/ARTICLE%20X.pdf>. (Accessed: 21 May 2023).
- [4] Berman, J. (2020). NBA Draft: The Hidden Cost of International Prospects. Available at: <https://www.babcockhoops.com/post/nba-draft-the-hidden-cost-of-international-prospects>. (Accessed: 21 May 2023).
- [5] NBA.com staff (2023). NBA Draft Lottery: Odds, history and how it works. Available at: <https://www.nba.com/news/nba-draft-lottery-explainer>. (Accessed: 21 May 2023).
- [6] Staw, B. M., & Hoang, H. (1995). Sunk Costs in the NBA: Why Draft Order Affects Playing Time and Survival in Professional Basketball. *Administrative Science Quarterly*, 40(3), 474–494. <https://doi.org/10.2307/2393794>

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 2023



Kenny Benaya Nathan 13521023