

# Penerapan Algoritma Branch and Bound dan Statistik Expected Goal dalam Menentukan Kombinasi Passing pada Sepak Bola

Ulung Adi Putra - 13521122  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
13521122@std.stei.itb.ac.id

*Expected Goals (xG) adalah statistik yang merepresentasikan peluang suatu tendangan menjadi sebuah gol. Statistik ini bisa dikombinasikan dengan algoritma Branch and Bound untuk mendapatkan kombinasi passing saat menyerang dalam permainan sepak bola. Statistik xG digunakan dalam penentuan nilai cost tiap simpul yang dibangkitkan.*

**Keywords**—branch and bound, expected goals, xG

## I. PENDAHULUAN

Permainan sepak bola adalah salah satu olahraga yang paling terkenal dan digemari di seluruh dunia. Permainan sepak bola adalah permainan beregu yang dimainkan oleh 2 tim dengan tiap tim berisi 11 pemain. Tujuan dari permainan ini adalah menciptakan gol yaitu dengan cara memasukkan bola ke dalam gawang lawan. Tim dengan jumlah gol terbanyak akan dinyatakan sebagai pemenang dari pertandingan sepak bola.

Untuk menciptakan sebuah gol dalam permainan sepak bola, diperlukan kerja sama antar tim yang dikombinasikan dengan skill individu tiap pemain. Oleh karena itu, diperlukan suatu strategi yang tepat untuk memaksimalkan hal tersebut.

Salah teknik dasar yang ada di sepak bola adalah *passing* atau mengumpan bola ke pemain lain. *Passing* merupakan teknik yang mudah tetapi sering disepelekan oleh banyak pemain, terutama pemain amatir. Kebanyakan pemain lebih suka memamerkan skill individu mereka dengan cara menggiring bola. Walaupun sering disepelekan, *passing* merupakan landasan yang sangat penting dalam permainan sepak bola.

Permainan sepak bola hanya akan menjadi permainan individu jika *passing* tidak pernah dilakukan. Selain itu, *passing* merupakan komponen penting dalam menyusun strategi dalam permainan sepak bola. *Passing* juga bisa menjadi parameter untuk menilai apakah strategi yang digunakan sudah efektif atau tidak. Logikanya, bagaimana strategi sepak bola dinilai berhasil jika jumlah *passing* yang dilakukan cukup sedikit dan tingkat akurasi rendah.

Banyak organisasi statistik sepak bola yang melakukan analisis dalam permainan sepak bola, salah satunya adalah Opta. Salah satu statistik penting yang dirilis oleh Opta adalah *Expected Goals (xG)*. *Expected Goals* adalah nilai ekspektasi

terjadinya sebuah gol jika dilakukan tendangan ke arah gawang.

Pada zaman yang berkembang ini, semua bisa dilakukan melalui teknologi. Pada makalah ini, penulis berusaha untuk menghubungkan *Expected Goals* dengan *passing* melalui algoritma branch and bound untuk menghasilkan kombinasi yang memungkinkan untuk menghasilkan gol.

## II. LANDASAN TEORI

### A. Algoritma Branch and Bound

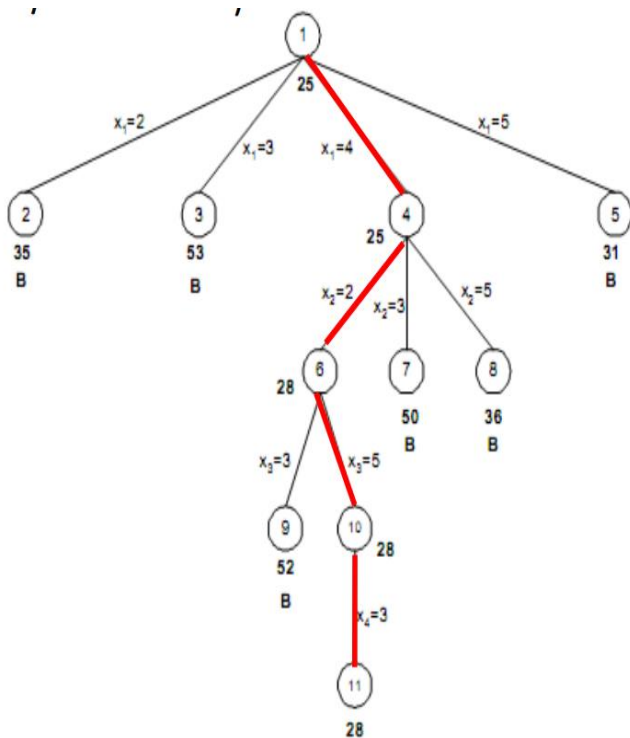
Algoritma *branch and bound* adalah algoritma yang digunakan untuk persoalan optimasi. Persoalan optimasi adalah persoalan yang solusinya harus berupa minimum atau maksimum dari fungsi objektif yang tidak melanggar batasan yang diberikan pada persoalan. Secara garis besar, Algoritma *Branch and Bound* (BnB) adalah perpaduan dari algoritma *Breath First Search* (BFS) dengan algoritma *least cost search*. Pada algoritma ini terdapat 2 prinsip dalam membangkitkan simpul simpulnya, yaitu :

1. Setiap simpul akan diberi nilai cost yang merupakan taksiran nilai perkiraan dari simpul saat ini sampai ke simpul tujuan.
2. Simpul yang akan diekspan tidak berdasarkan urutan pembangkitan simpul (FIFO). Akan tetapi, simpul yang diekspan diambil berdasarkan nilai cost yang paling kecil (untuk persoalan minimasi) atau nilai cost paling besar (untuk persoalan maksimasi).

Algoritma *Branch and Bound* juga menerapkan “pemangkasan” simpul pada simpul yang dianggap tidak mengarah ke solusi seperti pada algoritma *backtracking*. Secara umum, pemangkasan simpul dapat dilakukan dengan berbagai cara, seperti :

1. Nilai simpul tidak lebih baik dari nilai terbaik saat ini (tidak lebih kecil pada persoalan minimasi atau tidak lebih besar pada persoalan optimasi)
2. Simpul tidak menunjukkan solusi yang *feasible* karena melanggar batasan yang ada pada persoalan.

3. Solusi simpul yang akan dipangkas hanya terdiri dari satu titik saja, dengan kata lain hanya terdapat satu pilihan saja.



**Gambar 2.1** Penyelesaian Persoalan TSP dengan BnB

(sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian2.pdf> diakses pada 21 Mei 2023 pukul 20.21)

Algoritma *branch and bound* mengkalkulasi biaya dari setiap simpul yang dibangkitkan. Biaya yang dibangkitkan adalah taksiran nilai yang diperlukan untuk mencapai simpul tujuan (goal). Biaya dari tiap simpul dilambangkan dengan notasi  $\hat{c}(i)$  dengan  $i$  adalah simpul status ke  $i$ . Kemudian simpul akan diekspan lagi berdasarkan simpul dengan nilai biaya terbaik. Pada contoh gambar 2.1, simpul yang akan diekspan adalah simpul dengan cost terendah.

### B. *Expected Goals* ( $xG$ )

Dalam dunia sepak bola, terdapat banyak sekali cara untuk menganalisis suatu pertandingan. Analisis permainan sepak bola bisa dilakukan dengan melihat jalannya pertandingan, analisis strategi, sampai menganalisis data statistik permainan sepak bola. Pada umumnya, data statistik yang populer untuk dibahas adalah statistik penguasaan bola, gol, assist, jumlah umpan, akurasi umpan, dan lain lain. Akan tetapi, terdapat satu data statistik yang mungkin asing di telinga orang awam. Statistik tersebut adalah *Expected Goals*.

*Expected Goals* atau yang biasanya disingkat  $xG$  adalah suatu statistik yang merepresentasikan peluang suatu tendangan berbuah menjadi gol. Statistik ini bisa untuk mendeskripsikan kualitas dari peluang suatu serangan sepak bola. Nilai *expected*

*goals* dari suatu tembakan berada dalam kisaran 0 sampai 1. Nilai 0 merepresentasikan bahwa tendangan tersebut tidak mungkin untuk membuahkan gol, sedangkan nilai 1 merepresentasikan bahwa suatu tendangan pasti membuahkan gol. Akan tetapi, tidak ada hal yang pasti dalam dunia sepak bola, sehingga nilai *expected goals* tidak mungkin berada pada nilai 0 ataupun 1.

Statistik ini sepertinya cukup mengubah cara pandang *pundit* sepak bola ataupun penggemar dalam menilai pertandingan. Sebelum statistik ini diperkenalkan oleh Opta (perusahaan analisis olahraga), orang-orang selalu menilai bahwa suatu tim bermain lebih baik dari tim lainnya jika tim tersebut melakukan tembakan lebih banyak daripada lawannya. Sebagai contoh, orang-orang menilai bahwa Tim A yang melakukan 5 tembakan lebih baik daripada Tim B yang hanya melakukan 3 tembakan. Mereka berasumsi bahwa semakin banyak tembakan yang dilakukan, maka peluang yang diciptakan oleh suatu tim akan semakin banyak juga. Hal ini dapat merujuk kepada efektivitas strategi dalam menghasilkan peluang. Penilaian tersebut tidak sepenuhnya salah, tetapi tidak spesifik dalam memberi penilaian. Bisa saja Tim A menembakan 5 tendangan yang berasal dari jarak jauh semua, sedangkan Tim B menembakan 3 tendangan yang semuanya dari dalam kotak penalti. Hal ini menunjukkan bahwa Tim B membuat lebih banyak peluang daripada Tim A. Oleh karena itu, statistik *expected goals* diperkenalkan untuk meluruskan persoalan ini.

Statistik *expected goals* didapat dari hasil perhitungan dan analisis pertandingan-pertandingan yang telah dienggarakan. Angka  $xG$  (*expected Goals*) didapatkan melalui analisis yang dilakukan pada database Opta. Opta mengklaim bahwa mereka telah menganalisis lebih dari 300 ribu tendangan yang tersimpan pada database mereka. Setiap data tembakan memiliki atribut dan karakteristik masing-masing, seperti jarak dan sudut tendangan terhadap gawang, sundulan atau tendangan, tipe umpan, posisi pemain lawan, kaki untuk menendang, dan lain-lain. Atribut dan karakteristik tersebut kemudian menjadi faktor yang menentukan nilai  $xG$ .

Pada penentuan nilainya, Opta mengklasifikasikan tendangan dengan atribut dan karakteristik yang serupa. Kemudian, persentase dari gol yang dihasilkan akan dijadikan nilai dari  $xG$  untuk tendangan dengan tipe tersebut. Sebagai contoh, terdapat 10 ribu tembakan dengan jarak, sudut, kaki, tipe umpan tertentu. Diantara 10 ribu tendangan tersebut, ternyata tercatat 1000 diantaranya membuahkan gol. Maka dapat diambil kesimpulan bahwa nilai  $xG$  untuk tipe tembakan tersebut bernilai sekitar 0.1. Secara garis besar, perhitungan nilai  $xG$  didapat dengan cara tersebut. Akan tetapi, terdapat beberapa analisis dan pengolahan data tambahan untuk menghasilkan data yang lebih akurat.

#	Team	MP	xG	xGA	xGD	GF	GA	xG vs Actual
1	AS Monaco FC	2	2.74	1.17	+1.57	1.50	2.00	-1.24
2	SSC Napoli	10	2.33	1.29	+1.04	2.60	0.80	+0.27
3	Manchester City FC	12	2.31	1.02	+1.29	2.58	0.42	+0.27
4	Club Atlético de Madrid	6	2.30	1.51	+0.79	0.83	1.50	-1.47
5	FC Bayern München	10	2.27	1.37	+0.90	2.20	0.60	-0.07
6	Liverpool FC	8	2.24	1.26	+0.98	2.38	1.50	+0.14
7	FC Barcelona	6	2.05	1.51	+0.54	2.00	2.00	-0.05
8	Real Madrid CF	12	1.98	1.48	+0.50	2.17	1.08	+0.19
9	Maccabi Haifa FC	12	1.96	1.58	+0.38	1.75	2.33	-0.21
10	Chelsea FC	10	1.93	1.28	+0.65	1.20	0.90	-0.73

**Gambar 2.2** Statistik xG Pada Liga Champions

(sumber : [UEFA Champions League xG \(Expected Goals\) | FootyStats](#) diakses pada 22 Mei 2023 pukul 14.27)



**Gambar 2.3** Pemain dengan total xG tertinggi pada Piala Dunia 2022

(sumber : Sofascore Instagram, diakses pada 22 Mei 2023 pukul 14.43)

Dalam penerapannya, nilai xG sangat berguna untuk melakukan analisis pertandingan. Nilai xG dapat berguna untuk menganalisa performa tim maupun individu pemain. Dengan ini, tim dengan nilai xG yang lebih tinggi menunjukkan bahwa tim tersebut bermain lebih baik daripada lawannya. Hal ini dikarenakan Tim tersebut menghasilkan lebih banyak peluang sehingga nilai gol yang diekspetasikan menjadi lebih tinggi. Selain itu, pemain dengan nilai xG tertinggi menunjukkan bahwa pemain tersebut banyak mendapat atau menciptakan peluang serangan.



**Gambar 2.4** Analisis xG Pada Gol Lionel Messi Pada Piala Dunia 2022

(sumber : Sofascore Instagram, diakses pada 22 Mei 2023 pukul 14.43)

Perlu diingat, nilai xG yang lebih besar atau banyaknya peluang yang diciptakan tidak dapat menjamin kemenangan suatu tim. Sebagai contoh, pada gambar 2.4 ditunjukkan statistik xG dari gol Lionel Messi ke gawang Mexico pada Piala Dunia 2022. Dari analisis yang dilakukan oleh Sofascore, nilai xG yang diperoleh Messi pada tembakan tersebut hanya sebesar 0.02. Nilai peluang yang didapatkan oleh Messi tergolong salah satu yang paling kecil selama kompetisi berlangsung. Namun, Messi dapat mengkonversikan peluang yang hanya sebesar 0.02 menjadi sebuah gol. Hal ini memperkuat bahwa nilai xG tidak menjamin kemenangan karena terdapat perbedaan kualitas pemain antar tim. Namun perlu diingat kembali, xG digunakan untuk menilai kualitas permainan.

### III. PEMBAHASAN

Dalam persoalan ini, nilai xG dapat dimodelkan sebagai nilai cost dari tiap simpul. Nilai xG dapat dijadikan nilai cost dari simpul yang dibangkitkan karena, nilai xG merepresentasikan peluang untuk terjadinya gol ketika pemain menembakan bola dari posisinya. Hal ini merupakan nilai taksiran untuk mencapai suatu gol (tujuan). Semakin tinggi nilai xG yang dimiliki pemain, maka semakin dekat menjadi sebuah gol. Maka urutan pembangkitan simpul berdasarkan nilai xG terbesar.

Pada penyelesaian persoalan ini, bola mula mula dipegang oleh penjaga gawang (simpul 0). Kemudian akan dimodelkan persoalan ini sebagai kombinasi umpan antar pemain yang dapat menghasilkan gol. Bola akan diumpan secara terus menerus ke pemain lain hingga terdapat pemain menerima bola dengan nilai xG lebih dari 0.9 jika melakukan tembakan. Hal tersebut meruapak *goal state* yang ingin dicapai dari persoalan ini.

### A. Pemain

```
class Player:
    def __init__(self, name, role, position, xG, skill):
        self.name = name
        self.xG = xG
        self.skill = skill
        if(role == "GK"):
            self.role = 0
        elif(role == "Back"):
            self.role = 1
        elif(role == "Midfielder"):
            self.role = 2
        elif(role == "Foward"):
            self.role = 3

        if(position == "Left"):
            self.position = 1
        elif(position == "Right"):
            self.position = 3
        elif(position == "Center"):
            self.position = 2
```

Gambar 3.1 Pendefinisian Pemain pada Class Player

Pada awal program akan didefinisikan terlebih dahulu 11 pemain dengan *constructor* seperti pada gambar 3.1. nilai *xG* yang dimasukkan adalah nilai *xG* suatu pemain jika melakukan tembakan pada posisi awal tim akan melakukan serangan. Skill adalah nilai kemampuan dari suatu pemain. Nilai dari skill ini berkisar antara 1 samapai 3. Nilai ini nantinya akan digunakan untuk menentukan perubahan nilai *xG* tiap pemain. Selain itu, nilai ini juga akan digunakan untuk menentukan simpul mana saja yang bisa dibangkitkan dalam program (dalam konteks nyata, mengindikasikan pemain yang dapat diberi umpan).

Pemain juga akan diinisialisasi nilai dari posisi dan perannya di dalam tim. Nilai *role* 0 akan diberikan jika suatu pemain berperan sebagai penjaga gawang, 1 untuk pemain yang berperan sebagai pemain bertahan, 2 untuk pemain yang berperan sebagai pemain tengah, dan 3 untuk pemain yang berperan sebagai posisi depan. Nilai *position* merepresentasikan letak terhadap perannya. Nilai *position* 1 diberikan kepada pemain yang berada di kiri, 2 untuk pemain yang berada di tengah, dan 3 untuk pemain yang berada di kanan. Sebagai contoh, pemain *left back* akan mendapat nilai *role* 1 dan *position* 1.

```
Buffon = Player("Buffon", "GK", "Center", 0.1, 3)
Alves = Player("Alves", "Back", "Right", 0.31, 3)
Maldini = Player("Maldini", "Back", "Center", 0.24, 2)
Ramos = Player("Ramos", "Back", "Center", 0.25, 2)
Carlos = Player("Carlos", "Back", "Left", 0.23, 2)
Pirlo = Player("Pirlo", "Midfielder", "Center", 0.35, 3)
Xavi = Player("Xavi", "Midfielder", "Center", 0.25, 3)
Iniesta = Player("Iniesta", "Midfielder", "Center", 0.36, 3)
Messi = Player("Messi", "Foward", "Right", 0.71, 3)
Cristiano = Player("Cristiano", "Foward", "Left", 0.69, 3)
Ronaldo = Player("Ronaldo", "Foward", "Center", 0.67, 2)

Team = [Buffon, Alves, Maldini, Ramos, Carlos, Pirlo, Xavi, Iniesta, Messi, Cristiano, Ronaldo]
```

Gambar 3.2 Contoh deklarasi pemain dan tim

### B. Pembangkitan Simpul

Pada pemodelan ini tiap simpul menyatakan pemain yang sedang memegang bola. Pada kasus nyata, simpul dibangkitkan dengan pemain mengoper bola kepada rekan satu timnya. Simpul yang dapat dibangkitkan dari simpul ekspan (pemain yang memegang bola) adalah pemain lain yang berada dalam jangkauan umpan dari pemain saat ini. Berikut adalah kode yang digunakan untuk menentukan apakah pemain lain berada di dalam jangkauan umpan pemain saat ini (simpul ekspan)

```
def isPassAble(self, player):
    return (abs(player.role-self.role) + abs(player.position-self.position)) <= self.skill
```

Gambar 3.3 Potongan kode untuk menentukan suatu pemain berada dalam jangkauan umpan pemain lain

Penentuan jangkauan umpan pemain dilakukan menggunakan perhitungan *Manhattan distance*. *Manhattan distance* adalah suatu teknik untuk mengukur jarak yang dirumuskan :

$$d = |x_1 - x_2| + |y_1 - y_2|$$

Pada kasus ini nilai *role* pada player berperan sebagai *x* dan nilai *position* pada player berperan sebagai *y*.

Semakin jauh *Manhattan distance* dari dua pemain, maka semakin sulit juga untuk pemain mengumpan ke pemain lainnya. Untuk mengumpan dengan jarak yang jauh, dibutuhkan akurasi umpan yang bagus dari pemain agar umpan tersebut sampai dengan tepat ke target. Oleh karena itu, jangkauan umpan dibatasi oleh skill yang dimiliki oleh pemain. Semakin tinggi skill pemain, maka jangkauan umpan akan semakin jauh juga. Hal ini mengindikasikan pemain dengan skill yang tinggi dapat membangkitkan simpul lebih banyak.

### C. Bounding Function

*Cost* atau biaya suatu simpul pada algoritma Branch and Bound adalah nilai taksiran untuk suatu simpul mencapai simpul target. Dalam persoalan ini target yang dimaksud adalah simpul yang dapat memberikan peluang cukup tinggi untuk menghasilkan gol. Dengan kata lain, simpul target adalah pemain dengan nilai *expected goals* tertinggi jika melakukan tembakan. *Cost* dari suatu simpul dari suatu simpul dirumuskan dengan :

$$c^{\wedge}(S) = f(S)$$

dengan *f(S)* adalah nilai *expected goals* pemain (simpul) yang sudah diupdate.

Pengupdatetan data *xG* tiap pemain seharusnya dilakukan dengan analisis *real time* secara terus menerus. Untuk setiap perubahan kondisi di lapangan, seperti perpindahan lokasi pemain,antisipasi musuh, dan lain lain, akan dilakukan



perhitungan nilai xG. Sebagai contoh, misal terdapat kasus suatu pemain memiliki nilai xG mula mula 0.5. Namun karena pemain tersebut berlari menuju gawang, maka nilai xG pemain tersebut akan diupdate meningkat karena lebih dekat ke gawang. Jika sebaliknya pemain tersebut berlari menjauhi gawang untuk membuka ruang, maka nilai xG pemain tersebut akan diupdate menurun dari nilai xG sebelumnya. Terdapat banyak sekali kasus yang dapat mengubah nilai xG dari suatu pemain. Sehingga pengupdatean nilai xG harus diambil dari kalkulasi semua data dengan kondisi serupa pada database statistik sepak bola seperti Opta. Namun, dikarenakan *training data* untuk kalkulasi xG dan cara perhitungannya secara aktual tidak dipublikasikan oleh badan statistik sepak bola, maka penulis tidak bisa memodelkan penentuan xG secara tepat. Selain itu, karena batasan pada mata kuliah strategi algoritma, penentuan nilai xG secara aktual berdasar data pemain di lapangan tidak bisa dilakukan. Hal ini dikarenakan, metode pengambilan data secara aktual mungkin akan masuk ke lingkup pembelajaran *Artificial Intelligence* atau *Machine Learning*.

```
def updateExpectedGoal(self):
    randomBool = random.randint(0,100)
    if(randomBool < 75):
        randomInt = random.randint(0, self.role*self.skill)
        temp = self.xG + randomInt*0.01
        if(isValidXG(temp)):
            self.xG = temp
    else:
        randomInt = random.randint(0, 10)
        temp = self.xG - randomInt*0.01
        if(isValidXG(temp)):
            self.xG = temp
```

**Gambar 3.4** Kode yang digunakan untuk mengupdate nilai expected goals

Penentuan nilai cost atau xG tiap simpul dibangkitkan dengan fungsi random yang bergantung pada posisi dan skill pemain. Rentang penambahan nilai xG dipengaruhi oleh skill dan peran pemain di dalam lapangan. Pemain dengan posisi yang semakin depan akan cenderung untuk mendapat peluang mencetak gol lebih tinggi, sehingga semakin depan peran pemain maka semakin besar juga rentang penambahan nilai xG. Penentuan rentang penamabahan nilai xG juga dipengaruhi oleh kemampuan dari pemain. Pemain dengan kemampuan tinggi berpotensi lebih tinggi untuk menciptakan kesempatan mencetak gol. Sehingga semakin tinggi skill, maka akan semakin tinggi rentang penambahan xG pemain. Selain itu, terdapat kondisi yang menyebabkan nilai xG berkurang. Pada kasus nyata di lapangan, hal ini disebabkan oleh beberapa faktor seperti tekanan dari pemain lawan, dan sebagainya. Oleh karena itu, penulis memodelkan peluang untuk nilai xG berkurang adalah sebesar 25%.

#### D. Implementasi Algoritma BnB

```
def BnB(team):
    expandNode = team[0]
    goal = False
    passList = []
    passList.append(expandNode.name)
    while(not(goal)):
        liveNode = []
        passList.append(expandNode.name)
        if(expandNode.xG > 0.9):
            goal = True

    for player in team:
        if(expandNode.name != player.name and expandNode.isPassAble(player)):
            player.updateExpectedGoal()
            liveNode.append(player)

    liveNode.sort(key=lambda x: x.xG)
    liveNode[len(liveNode)-1].print()
    expandNode = liveNode[len(liveNode)-1]
    return passList

passList = BnB(team)
print(passList)
```

**Gambar 3.5** Implementasi algoritma BnB

Pada awalnya, simpul yang diekspan pertama adalah penjaga gawang (asumsi serangan dimulai dari penjaga gawang). Kemudian dibuat array passList yang digunakan untuk menampung perpindahan bola antar pemain. Dilakukan iterasi hingga nilai goal menjadi true, nilai goal mengindikasikan simpul yang sedang diekspan memiliki nilai cost yang cukup. Simpul dengan nilai cost yang cukup merepresentasikan pemain yang sedang memegang bola memiliki xG yang cukup besar (dalam hal ini diambil nilai 0.9) jika melakukan tembakan.

Pada tiap iterasi akan dibuat array baru bernama liveNode yang akan berisi simpul simpul yang dapat diekspan dari ekspanNode. Kemudian ekspanNode sekarang dimasukkan kedalam array passList. Kemudian dilakukan pengecekan apakah simpul yang dievaluasi sekarang memiliki nilai xG lebih besar dari 0.9, jika iya maka nilai goal akan diset menjadi true. Namun jika nilai xG tidak lebih dari 0.9, maka akan dilakukan iterasi pada semua pemain pada tim. Jika Pemain yang ada di tim berada di jangkauan umpan dari pemain yang sedang memegang bola, maka pemain tersebut masuk kedalam simpul hidup. Nilai xG dari pemain tersebut akan diupdate terlebih dahulu untuk mengindikasikan terdapat perubahan kondisi dilapangan yang mengubah nilai xG. Simpul simpul yang berhasil dibangkitkan kemudian diurutkan berdasarkan nilai xGnya. Kemudian simpul atau pemain yang memiliki nilai xG terbesar akan dijadikan expandNode.

Berikut adalah contoh hasil komputasi menggunakan deklarasi pemain pada gambar 3.2.

```
Buffon 0.1
Ronaldo 0.61000000000000001
Cristiano 0.76
Messi 0.74
Cristiano 0.8
Messi 0.73
Cristiano 0.82000000000000001
Messi 0.8
Cristiano 0.82000000000000001
Messi 0.87000000000000001
Cristiano 0.91
```

**Gambar 3.6** Hasil komputasi Algoritma

Pada gambar 3.6, terlihat urutan umpan bola yang dilakukan oleh pemain. Bola pertama dipegang oleh buffon sebagai penjaga gawang dan akhirnya bola berada pada Cristiano dengan nilai xG 0.91.

#### IV. PENUTUP

##### A. Kesimpulan

Strategi algoritma Branch and Bound yang dipadukan dengan expected goals sebagai biaya tiap simpul dapat diterapkan dalam menentukan kombinasi umpan terbaik dalam bermain sepak bola.

##### B. Saran

Untuk kedepannya makalah ini bisa dikembangkan lagi dengan meningkatkan program yang dibuat. Program bisa dikembangkan dengan menggunakan machine learning atau *artificial intelligence*.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah menguatkan penulis sehingga penulis bisa menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada seluruh dosen mata kuliah Strategi Algoritma 2022/2023 karena telah memberikan dan memaparkan ilmu sehingga penulis dapat menulis makalah ini. Tidak lupa penulis juga berterima kasih pada diri sendiri

untuk tidak pernah menyerah, selalu berjuang, dan tetap menyelesaikan penulisan makalah ini walaupun kata hati ingin berlibur.

#### REFERENCES

- [1] Munir, Rinaldi. 2021. Algoritma branch and bound (Bagian 1) <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian1.pdf>. Diakses pada 21 Mei 2023.
- [2] Munir, Rinaldi. 2021. Algoritma branch and bound (Bagian 2) <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian2.pdf>. Diakses pada 21 Mei 2023.
- [3] PanditFootball. Mengupas Statistik Expected Goals (xG) <https://www.panditfootball.com/sains-bola/213838/PND/220813/mengupas-statistik-expected-goals-xg> Diakses pada 22 Mei 2023
- [4] <https://footystats.org/europe/uefa-champions-league/xg> diakses pada 22 Mei 2023

#### 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



Ulung Adi Putra 13521122