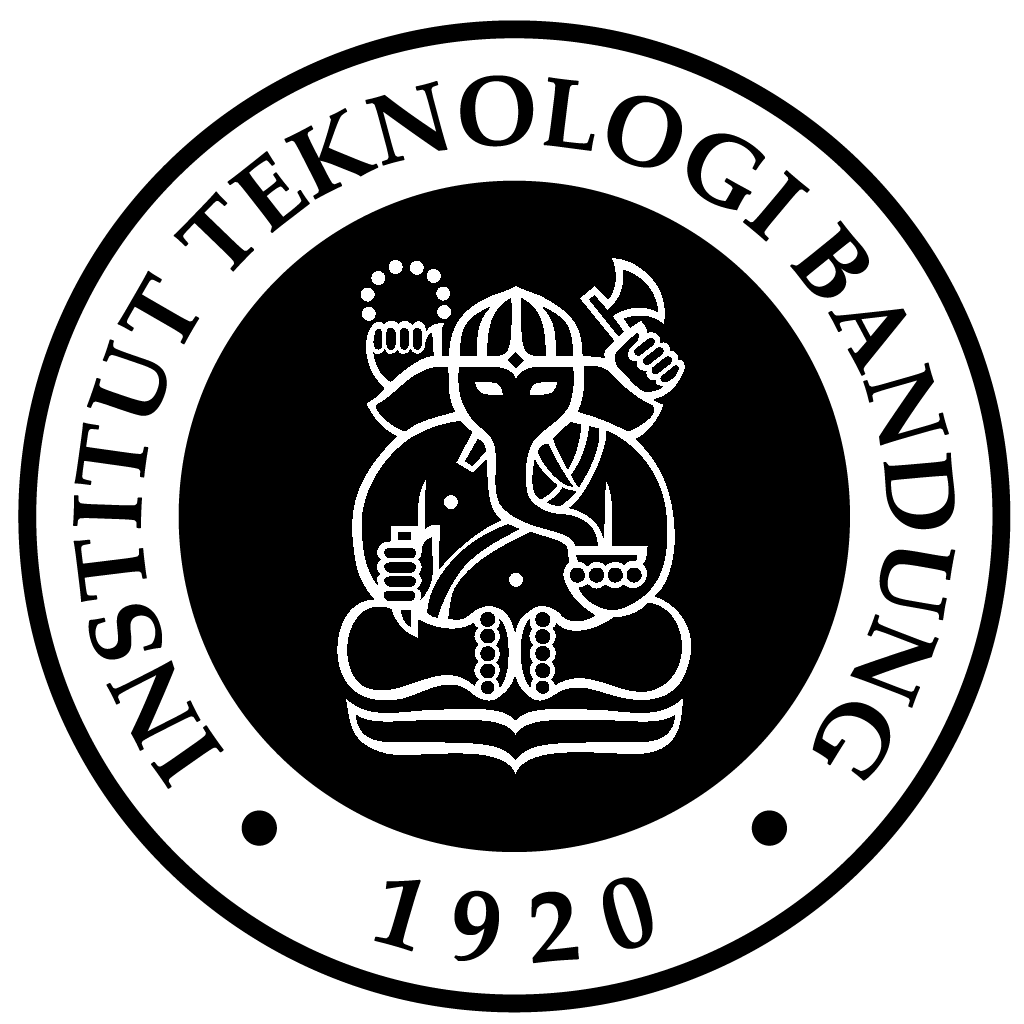
**Tugas Besar 1 IF3170 Intelegensi Buatan**

**Implementasi *Minimax Algorithm* dan *Local Search* pada Permainan Dots and Boxes**



**Disusun oleh:**

Ahmad Alfani Handoyo 13520023

Aditya Prawira Nugroho 13520049

Felicia Sutandijo 13520050

Putri Nurhaliza 13520066

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2022**

**Daftar Isi**

[A. Objective Function 1](#_Toc117015923)

[B. Minimax Alpha Beta Pruning 3](#_Toc117015924)

[C. Local Search 4](#_Toc117015925)

[D. Hasil Pertandingan 5](#_Toc117015926)

[E. Saran Perbaikan 6](#_Toc117015927)

[F. Kontribusi Anggota 6](#_Toc117015928)

# Objective Function

*Objective function* dibuat sebisa mungkin untuk merepresentasikan keadaan sebenarnya dari permainan, dengan melihat potensi menang/kalah suatu *player*. Di sini, *objective function* kami melihat seberapa banyak garis yang mengelilingi setiap kotak.

Variabel:

* L : Jumlah kotak (*box*) pada *board* permainan. Pada kasus ini L = 9.
* Bi : Kotak (*box*) 1x1 ke-i pada *board* permainan
* v(Bi) : Nilai (*value*) dari kotak 1x1 ke-i pada *board* permainan
* t : Giliran (*turn*) pada *state* tersebut. Bila giliran kita bernilai -1, bila giliran lawan bernilai 1

Nilai setiap kotak (v(Bi)):

1. Kotak dengan 0 garis : 0
2. Kotak dengan 1 garis : 5
3. Kotak dengan 2 garis : 10
4. Kotak dengan 3 garis :
5. Kotak yang berhasil dibuat oleh kita : 100
6. Kotak yang berhasil dibuat oleh musuh : -100

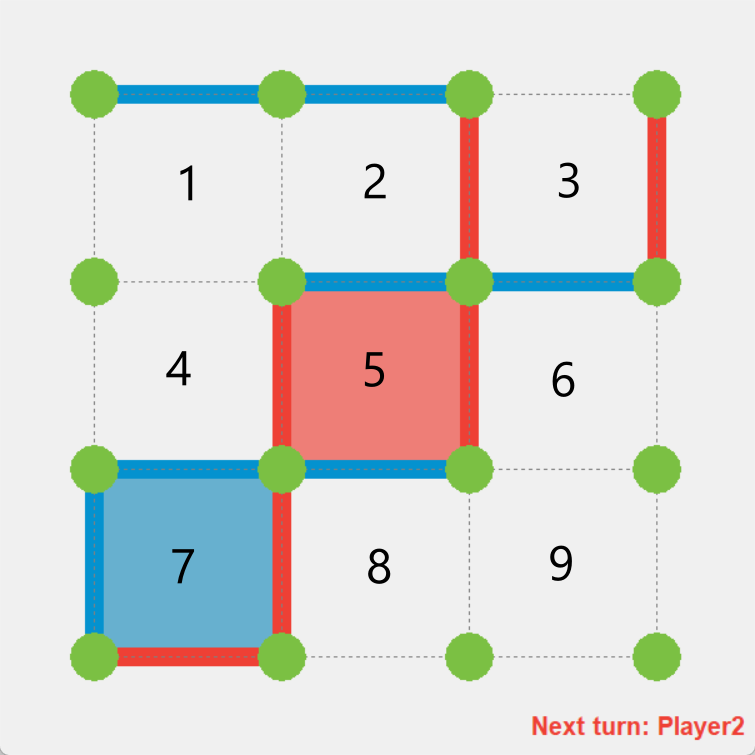
*Objective function*:

*Objective function* dihitung ketika membangkitkan suatu *state* setelah salah satu pemain meletakkan satu garis. Digunakan heuristik untuk menghitung *objective function* dengan melihat keuntungan berdasarkan garis pada setiap kotak. Pada kasus kotak dengan 3 garis, diberikan parameter tambahan yaitu gilian saat ini. Jika giliran saat ini adalah giliran kita, maka kotak dengan 3 garis akan memberikan keuntungan pada lawan di giliran setelahnya. Sebaliknya, jika saat ini adalah giliran lawan, maka kita memiliki kesempatan besar untuk memenuhi kotak pada giliran setelahnya.

Nilai untuk kotak dengan jumlah garis yang lain ditentukan menimbang bahwa ketika suatu kotak berisi 0 garis, masih tidak memberi keuntungan bagi kedua pemain. Ketika kotak berisi 1 garis tentunya lebih menguntungkan dibanding 0 garis karena artinya hanya perlu 3 garis lagi untuk memenuhi kotak, dan selanjutnya juga untuk kotak dengan 2 garis.

Penalti atau pengaruh terbesar secara logis harusnya ketika kotak berhasil dibuat oleh kita atau musuh. Tentunya saat kotak dibuat oleh kita berdampak positif karena membawa kita lebih dekat ke kemenangan sehingga direpresentasikan dengan nilai positif yang besar. Namun, bila kotak dibuat oleh musuh tentu berdampak negatif karena membawa kita semakin dekat ke kekalahan dan direpresentasikan dengan nilai-nilai negatif yang besar.

Contoh perhitungan *objective function* terdapat pada konfigurasi di bawah ini.



Misal baru saja adalah giliran kita dan kita meletakkan satu garis sehingga *t* bernilai -1.

* Kotak 1 bernilai 5 karena diisi oleh 1 garis. (v(B1) = 5)
* Kotak 2 bernilai -20 karena diisi oleh 3 garis. (v(B2) = = = -25)
* Kotak 3 bernilai -20 karena diisi oleh 3 garis. (v(B3) = 25 \* t = = -25)
* Kotak 4 bernilai 10 karena diisi oleh 2 garis. (v(B4) = 10)
* Kotak 5 bernilai -100 karena berhasil dibuat oleh musuh. (v(B5) = -100)
* Kotak 6 bernilai 10 karena diisi oleh 2 garis. (v(B6) = 10)
* Kotak 7 bernilai 100 karena berhasil dibuat oleh kita. (v(B7) = 100)
* Kotak 8 bernilai 10 karena diisi oleh 2 garis. (v(B8) = 10)
* Kotak 9 bernilai 0 karena belum diisi oleh garis. (v(B9) = 0)

Sehingga total skor untuk *objective function*:

# Minimax dengan Alpha-Beta Pruning

# Local Search

*Local search* untuk penyelesaian permainan *Dots and Boxes* dapat dilakukan dengan metode *hill-climbing search* dengan variasi *sideways move* yang memilih gerakan untuk menghasilkan *state* baru berdasarkan *objective function*-nya.

# Hasil Pertandingan

# Saran Perbaikan

# Kontribusi Anggota

|  |  |
| --- | --- |
| **Nama - NIM** | **Kontribusi** |
| Ahmad Alfani Handoyo – 13520023 |  |
| Aditya Prawira Nugroho – 13520049 |  |
| Felicia Sutandijo – 13520050 |  |
| Putri Nurhaliza – 13520066 |  |

**REFERENSI**

GeeksforGeeks. 2021. *Minimax Algorithm in Game Theory | Set 4 (Alpha-Beta Pruning)*. Diakses pada 12 September 2022, dari https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/

Khodra, Masayu Leylia. 2021. “Beyond Classical Search: Classical Search vs Local Search”. Bandung: Institut Teknologi Bandung.

Russell, Stuart. Peter Norvig. 2010. *Artificial Intelligence a Modern Approach Third Edition*. New Jersey: Pearson Education, Inc.