Analisis Strategi Optimal pada Permainan NIM Menggunakan Aljabar Boolean dan Operasi XOR

An-Dafa Anza Avansyah - 13524038
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: andafaanzaa@gmail.com , 13524038@std.stei.itb.ac.id

***Abstrak*—Permainan NIM merupakan salah satu permainan kombinatorial klasik yang memiliki struktur logika deterministik dan dapat dianalisis secara matematis. Dalam makalah ini, dibahas strategi optimal dalam permainan NIM menggunakan konsep aljabar Boolean, khususnya operasi logika XOR (exclusive OR). Operasi ini digunakan untuk menghitung NIM-sum, yang menjadi dasar penentuan posisi menang atau kalah dalam setiap giliran permainan. Selain pemaparan teori, makalah ini juga menyajikan implementasi algoritma dalam bahasa pemrograman Python sebagai simulasi untuk memverifikasi strategi yang dibangun. Hasil simulasi menunjukkan bahwa pendekatan berbasis XOR mampu memandu pemain dalam mengambil keputusan optimal secara sistematis. Penelitian ini menegaskan pentingnya peran matematika diskrit dalam penyelesaian masalah strategi permainan dan membuka peluang penerapan lebih luas dalam teori permainan dan pengambilan keputusan berbasis logika.**

***Kata Kunci*—Permainan NIM; Matematika Diskrit; XOR; Aljabar Boolean; NIM-sum; Strategi Optimal; Simulasi Python.**

I. PENDAHULUAN

Permainan kombinatorial telah lama menarik perhatian para matematikawan karena sifatnya yang sederhana namun menyimpan berbagai analisis yang kompleks dan mendalam. Salah satu permainan kombinatorial yang beken adalah permainan NIM, sebuah permainan sederhana yang dimainkan oleh dua pemain secara bergantian mengambil objek dari beberapa tumpukan. Aturan dasar permainan ini memungkinkan setiap pemain mengambil sejumlah objek dari satu tumpukan dalam satu giliran, dengan tujuan utama pemain untuk mengambil objek terakhir agar memenangkan permainan.

Di balik kesederhanaan aturan tersebut, permainan NIM menyimpan struktur matematis yang menarik, terutama dalam konteks aljabar Boolean dengan penggunaan operasi XOR (exclusive OR). Operasi XOR memiliki peran penting dalam menentukan posisi menang dan kalah dalam permainan ini. Secara matematis, suatu posisi permainan dianggap sebagai posisi menang jika hasil operasi XOR dari jumlah objek pada setiap tumpukan tidak bernilai nol, dan sebaliknya, posisi permainan dianggap sebagai posisi kalah jika hasil XOR tersebut bernilai nol.

Makalah ini bertujuan untuk menyajikan analisis terhadap strategi optimal dalam permainan NIM melalui pendekatan aljabar Boolean dan operasi XOR. Selain analisis teoritis, makalah ini juga akan dilengkapi dengan implementasi simulasi menggunakan bahasa pemrograman untuk menguji kevalidan dan efektivitas strategi yang diperoleh dari analisis tersebut.

Ruang lingkup makalah ini akan dibatasi pada versi standar permainan NIM dengan beberapa tumpukan objek, dengan jumlah objek pada masing-masing tumpukan dalam kisaran yang mudah disimulasikan. Metodologi yang digunakan terdiri dari analisis teoritis aljabar Boolean dan implementasi simulasi komputer untuk memvalidasi strategi optimal secara praktis.

II. LANDASAN TEORI

A. Permainan NIM

Permainan NIM sudah dikenal sejak zaman dahulu dan dipercaya berasal dari permainan tradisional Tiongkok kuno yang bernama *Tsyanshidzi* (jiǎn-shízǐ (捡石子) yang artinya "mengambil batu". Permainan serupa Nim telah dimainkan di berbagai belahan dunia selama berabad-abad; catatan tertulis pertama di Eropa tercatat muncul sekitar awal abad ke-16. Nama "Nim" sendiri mulai digunakan secara resmi pada tahun 1901 oleh Charles L. Bouton, seorang profesor matematika di Universitas Harvard, yang kala itu mengembangkan teori lengkap untuk strategi permainan ini. Istilah "Nim" kemungkinan diambil dari kata perintah *"nimm"* dalam bahasa Jerman yang berarti "ambil". Hal ini sesuai dengan karakter permainan NIM di mana pemain secara bergantian mengambil objek-objek hingga habis[3].

Nim adalah permainan strategi matematika yang dimainkan oleh dua orang pemain secara bergiliran. Di awal permainan, disiapkan beberapa tumpukan objek (misalnya batu kerikil, korek api, atau benda sejenis) dengan jumlah objek pada tiap tumpukan boleh berbeda. Kedua pemain kemudian bergantian melakukan langkah; pada setiap giliran, seorang pemain harus mengambil sekurang-kurangnya satu objek dari satu tumpukan yang dipilihnya. Pemain hanya boleh mengambil dari satu tumpukan per giliran, tetapi boleh mengambil lebih dari satu objek asalkan berasal dari tumpukan yang sama. Setelah objek diambil, objek tersebut dikeluarkan dari tumpukan (sehingga tumpukan berkurang). Permainan berlanjut hingga tidak ada lagi objek yang tersisa di semua tumpukan.

Terdapat dua variasi kondisi kemenangan dalam permainan NIM, yaitu “misère” dan normal. Pada aturan misère yang tradisional, pemain yang mengambil objek terakhir justru kalah, sehingga sebaiknya menghindari mengambil objek terakhir. Sebaliknya, dalam versi normal (biasa), pemain yang berhasil mengambil objek terakhir itulah yang menang. Versi normal inilah yang mengikuti aturan permainan pada umumnya, sehingga disebut *normal play*. Dalam makalah ini, fokus pembahasan adalah Nim versi normal, di mana pemain terakhir yang mengambil objek memenangkan permainan. Aturan Nim versi normal ini sejalan dengan kondisi kemenangan pada permainan strategi sederhana lain, sehingga lebih mudah dipahami oleh pemula.

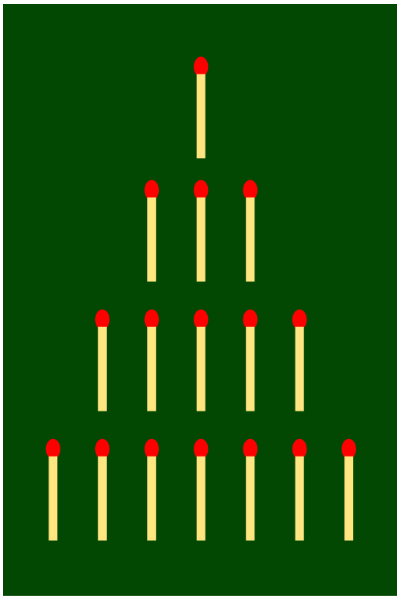


Fig. 1. Permainan NIM, diambil dari [4]

Sebagai ilustrasi, berikut adalah contoh permainan NIM versi normal dengan tiga tumpukan awal berisi 2, 3, dan 4 objek. Misalkan dua pemain, sebut saja Pemain A dan Pemain B, akan bermain dengan aturan Nim seperti dijelaskan di atas. Pemain A mengambil giliran pertama secara acak. Urutan langkah permainan dapat berlangsung sebagai berikut:

TABLE I. CONTOH PERMAINAN NIM

Tumpukan A	**Tumpukan B**	**Tumpukan C**	**Langkah**
2	3	4	Permainan dimulai
2	3	2	Pemain A mengambil 2 objek dari Tumpukan C
2	1	2	Pemain B mengambil 2 objek dari Tumpukan B
1	1	2	Pemain A mengambil 1 objek dari Tumpukan A
1	1	1	Pemain B mengambil 1 objek dari Tumpukan C
1	0	1	Pemain A mengambil 1 objek dari Tumpukan B
0	0	1	Pemain B mengambil 1 objek dari Tumpukan A
0	0	0	Pemain A mengambil 1 objek terakhir dari Tumpukan C. Pemain A menang.

B. *Aljabar Boolean*

Aljabar Boolean adalah salah satu cabang dari matematika diskrit yang berfokus pada operasi logika dengan dua nilai kebenaran, yaitu 0 (false) dan 1 (true). Sistem ini pertama kali diperkenalkan oleh matematikawan Inggris George Boole pada pertengahan abad ke-19 melalui karyanya *The Mathematical Analysis of Logic* (1847) dan kemudian dikembangkan lebih lanjut dalam *An Investigation of the Laws of Thought* (1854)[5].

Tidak seperti aljabar biasa yang menangani angka dan operasi seperti penjumlahan atau perkalian, aljabar Boolean bekerja dengan pernyataan logika dan operasi logika, yang menjadi fondasi bagi perancangan rangkaian digital, pemrograman, basis data, serta sistem kecerdasan buatan.

Terdapat tiga operasi dasar dalam aljabar Boolean, yaitu AND ($\wedge$), OR ($\vee$), dan NOT ($\neg$). Operasi AND akan menghasilkan nilai 1 hanya jika kedua operand bernilai 1. OR akan bernilai 1 jika setidaknya salah satu operand bernilai 1. Sedangkan operasi NOT membalik nilai operand: jika operand bernilai 1 maka hasilnya 0, dan sebaliknya. Selain tiga operasi dasar ini, terdapat operasi tambahan yang sangat penting dalam konteks pemrosesan biner dan permainan strategi seperti NIM, yaitu operasi XOR (exclusive OR).

Operasi XOR adalah operasi logika yang menghasilkan nilai 1 jika dua operand memiliki nilai yang berbeda. Jika kedua operand sama, maka hasilnya adalah 0. Dalam bentuk tabel kebenaran, operasi XOR dijelaskan sebagai berikut:

TABLE II. OPERASI XOR

A	**B**	**A $\oplus$ B**
0	0	0
0	1	1
1	0	1
1	1	0

Sifat-sifat penting dari operasi XOR antara lain:

$$A \oplus 0 = A \text{ (Identitas)}$$

$$A \oplus A = 0 \text{ (Inversi diri)}$$

$$A \oplus B = B \oplus A \text{ (Komutatif)}$$

$$(A \oplus B) \oplus X = A \oplus (B \oplus X) \text{ (Asosiatif)}$$

Sifat-sifat inilah yang membuat XOR sangat berguna dalam berbagai aplikasi seperti kriptografi, pengkodean data, dan strategi permainan kombinatorial.

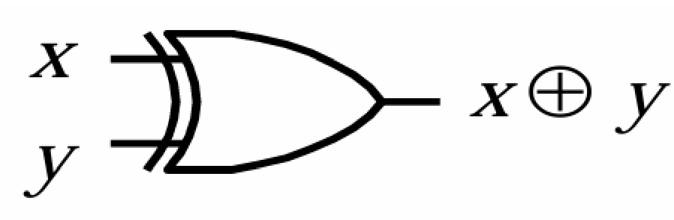


Fig. 2. Gerbang Logika Operasi XOR, diambil dari [5]

Dalam permainan NIM, operasi XOR menjadi kunci utama dalam menentukan strategi optimal. Permainan NIM terdiri dari beberapa tumpukan objek, dan pemain secara bergiliran mengambil sejumlah objek dari satu tumpukan pada setiap giliran. Pemenang adalah pemain yang berhasil mengambil objek terakhir. Untuk menganalisis posisi permainan, digunakan konsep yang disebut sebagai NIM-sum, yaitu hasil operasi XOR dari semua jumlah objek di setiap tumpukan.

III. Analisis Matematis Permainan NIM

A. *Konfigurasi Permainan sebagai Bilangan Biner*

Permainan NIM memiliki keunikan karena dapat dianalisis secara matematis dengan menggunakan representasi bilangan biner dan operasi logika. Pendekatan ini memungkinkan kita untuk menentukan secara pasti apakah suatu konfigurasi permainan merupakan posisi menang (*winning position*) atau posisi kalah (*losing position*). Analisis ini berguna tidak hanya dalam teori permainan, tetapi juga memperkenalkan penerapan konkret dari aljabar Boolean dan manipulasi bit dalam konteks strategis.

Permainan NIM terdiri dari beberapa tumpukan (*pil*e) objek, biasanya batu atau stik. Misalnya, terdapat tiga tumpukan dengan masing-masing berisi 2, 3, dan 4 objek. Konfigurasi ini secara konvensional ditulis sebagai [2, 3, 4].

Langkah pertama dalam menganalisis permainan ini secara matematis adalah mengubah jumlah objek dalam setiap tumpukan menjadi representasi biner. Hal ini dilakukan karena operasi utama dalam strategi NIM adalah XOR, yang bekerja secara langsung pada bit-bit bilangan biner. Sebagai contoh, konfigurasi [2, 3, 4] direpresentasikan sebagai:

TABLE III. Representasi Biner Permainan NIM

Tumpukan	**Jumlah**	**Biner**
A	2	010
B	3	011
C	4	100

Setelah semua tumpukan dikonversi ke bentuk biner, dilakukan operasi XOR (exclusive OR) secara berurutan terhadap seluruh tumpukan. Hasil akhir dari proses ini disebut sebagai NIM-sum.

NIM-sum dirumuskan sebagai:

$$\text{NIM-sum} = T_1 \oplus T_2 \oplus T_3 \oplus \ldots \oplus T_n$$

di mana T_1 hingga T_n adalah jumlah objek dari setiap tumpukan dalam bentuk desimal atau biner.

Hasil dari NIM-sum digunakan untuk menentukan posisi strategis pemain, jika NIM-sum = 0, maka konfigurasi disebut sebagai posisi kalah, sebaliknya jika NIM-sum $\neq$ 0, maka konfigurasi disebut sebagai posisi menang. Berikut adalah beberapa contoh dari perhitungan NIM-sum dalam beberapa konfigurasi tumpukan:

1) *Konfigurasi [1,4,5]*

Konversi Biner:

a) 1 = 001

b) 4 = 100

c) 5 = 101

Langkah XOR:

$$001 \oplus 100 = 101$$
$$101 \oplus 101 = 000$$
$$\text{NIM–sum} = 0$$

Interpretasi: Posisi ini merupakan posisi kalah, karena hasil NIM-sum adalah 0. Pemain yang bermain dalam kondisi ini tidak dapat memaksa kemenangan jika lawan bermain dengan strategi optimal.

2) *Konfigurasi [2,3,4]*

Konversi Biner:

a) 2 = 010

b) 3 = 011

c) 4 = 100

Langkah XOR:

$$010 \oplus 011 = 001$$
$$001 \oplus 100 = 101$$
$$\text{NIM–sum} = 5$$

Interpretasi: Posisi ini merupakan posisi menang, karena hasil NIM-sum $\neq$ 0. Pemain dapat mengambil langkah untuk memaksa lawan masuk ke posisi kalah.

B. *Strategi Optimal Menggunakan XOR*

Dalam permainan NIM versi normal, di mana pemain yang mengambil objek terakhir dari tumpukan dinyatakan sebagai pemenang, strategi optimal dapat ditentukan secara matematis melalui hasil XOR (NIM-sum) dari konfigurasi tumpukan yang ada. Dengan menggunakan prinsip-prinsip aljabar Boolean, kita dapat mengklasifikasikan setiap posisi permainan menjadi posisi menang (*winning position*) atau posisi kalah (*losing position*). Posisi kalah terjadi ketika hasil NIM-sum dari seluruh tumpukan adalah nol (0). Jika seorang pemain menghadapi posisi kalah, maka pemain tersebut tidak dapat menang jika lawan bermain optimal. Posisi menang terjadi jika NIM-sum tidak sama dengan nol. Dalam posisi ini, pemain memiliki langkah strategis untuk memindahkan objek dari satu tumpukan sehingga konfigurasi baru menjadi posisi kalah untuk lawan.

Intuisi di balik pembagian ini dapat dijelaskan sebagai berikut: XOR adalah operasi yang membatalkan dirinya sendiri. Jika semua nilai objek dari tumpukan di-XOR-kan dan hasilnya adalah nol, maka tidak ada langkah satu kali yang dapat dilakukan untuk mengubah hasil itu menjadi nol kembali, karena semua tumpukan sudah "seimbang". Namun, jika hasilnya bukan nol, berarti terdapat "ketimpangan" yang dapat dimanfaatkan oleh pemain aktif untuk mengatur langkah.

Strategi optimal dalam permainan NIM adalah mengubah posisi permainan menjadi posisi kalah bagi lawan. Secara umum, strategi ini mengikuti langkah-langkah berikut:

1) *Hitung NIM-sum dari konfigurasi saat ini.*
2) *Jika NIM-sum = 0, pemain sedang berada dalam posisi kalah. Pilihan langkah apa pun hanya akan memperkuat posisi lawan jika mereka bermain optimal.*
3) *Jika NIM-sum ≠ 0, pemain dapat menentukan langkah strategis sebagai berikut:*
   a) *Pilih salah satu tumpukan, misalnya tumpukan T_i, sehingga:*
   $$T_i \oplus \text{NIM−sum} < T_i$$
   b) *Kurangi jumlah objek di tumpukan T_i menjadi:*
   $$T_i' = T_i \oplus \text{NIM−sum}$$
   c) *Hasil konfigurasi baru akan memiliki NIM-sum = 0 (posisi kalah untuk lawan).*

Berikut adalah contoh dari penggunaan strategi ini:
Misalkan terdapat konfigurasi [2,3,4]

1) *Konversi Biner:*
   a) 2 = 010
   b) 3 = 011
   c) 4 = 100
2) *Hitung NIM-sum:*
   $$010 \oplus 011 = 001$$
   $$001 \oplus 100 = 101$$
   $$\text{NIM−sum} = 5$$
   Interpretasi: Posisi ini merupakan posisi menang, karena hasil NIM-sum adalah 5.
3) *Cari tumpukan yang dapat dikurangi sehingga hasil akhir XOR menjadi 0.*
   a) *Coba tumpukan 4:*
   $$4 \oplus 5 = 1 < 4$$
   Ubah tumpukan 4 menjadi 1 sehingga konfigurasi yang baru menjadi [2,3,1].
4) *Cek NIM-sum baru:*
   $$2 \oplus 3 \oplus 1 = 010 \oplus 011 \oplus 001 = 000$$
   $$\text{NIM−sum} = 0$$
   Interpretasi: Posisi ini merupakan posisi kalah, karena hasil NIM-sum adalah 0.

Dengan melakukan langkah tersebut, pemain berhasil memaksa lawan berada dalam posisi kalah. Jika pemain terus mempertahankan strategi ini setiap kali mendapat giliran dalam posisi menang, ia dapat memenangkan permainan.

Prinsip kunci dari strategi ini adalah bahwa operasi XOR memiliki sifat reversibel dan bersifat mengatur keseimbangan. NIM-sum yang tidak nol berarti terdapat setidaknya satu bit "ganjil" dalam representasi biner tumpukan, dan hanya pemain aktif yang memiliki kesempatan untuk menyeimbangkannya. Begitu ia berhasil mengubah konfigurasi menjadi "seimbang" (NIM-sum = 0), maka lawan akan terperangkap.

C. *Contoh Kasus*

Untuk memahami penerapan strategi optimal dalam permainan NIM secara menyeluruh, kita akan membahas sebuah studi kasus dari awal permainan hingga akhir, dengan menjelaskan langkah-langkah, perhitungan NIM-sum, dan alasan di balik setiap keputusan.

Studi kasus: Konfigurasi Awal [2,3,4]
Terdapat tiga tumpukan objek, masing-masing berisi:

1) *Tumpukan A* = 2 objek
2) *Tumpukan B* = 3 objek
3) *Tumpukan C* = 4 objek

Kita asumsikan permainan dimulai oleh Pemain 1 dan keduanya menggunakan strategi optimal dalam bermain.

TABLE IV. STUDI KASUS PERMAINAN NIM

Langkah	**A**	**B**	**C**	**NIM-sum (sekarang)**	**Konfigurasi Perubahan**	**NIM-sum (kemudian)**
1 (P1)	2	3	4	5	[2,3,1]	0
2 (P2)	2	3	1	0	[0,3,1]	2
3 (P1)	0	3	1	2	[0,1,1]	0
4 (P2)	0	1	1	0	[0,1,0]	1
5 (P1)	0	1	0	1	[0,0,0]	0

Pemain 1 mengambil objek terakhir dari Tumpukan B yang menyebabkan semua tumpukan kosong, hasil akhir dari permainan ini dimenangkan oleh Pemain 1.

Dari studi kasus di atas dapat dilihat bahwa pemain yang berada dalam posisi menang (NIM-sum $\neq 0$) dapat selalu memaksa lawan masuk ke posisi kalah (NIM-sum $= 0$). Selain itu dengan mengikuti langkah optimal berbasis XOR, pemain pertama berhasil memenangkan permainan meskipun konfigurasi awal tidak terlihat “istimewa”.

IV. Implementasi Program dan Simulasi

A. *Metode Implementasi*

Untuk menguji validitas teori yang telah dibahas sebelumnya serta memahami strategi optimal permainan NIM secara lebih intuitif, penulis mengembangkan sebuah program simulasi menggunakan Bahasa pemrograman Python. Pemilihan Python didasarkan pada kelebihan sintaksnya yang sederhana dan mudah dipahami, serta banyaknya Pustaka (*library*) pendukung yang relevan untuk pemrograman matematis.

1) Tujuan Program

Tujuan utama dari program ini adalah untuk:

a) Menyimulasikan permainan NIM dengan konfigurasi tumpukan tertentu.

b) Menghitung nilai NIM-sum dari konfigurasi tersebut.

c) Mengidentifikasi apakah posisi saat ini merupakan posisi menang atau posisi kalah.

d) Menentukan langkah optimal bagi pemain yang sedang bermain (jika posisi menang).

2) Struktur Umum Program

Program dibagi menjadi beberapa komponen utama sebagai berikut:

a) Input konfigurasi: Pengguna memasukkan jumlah objek pada tiap tumpukan.

b) Fungsi NIM-sum: Fungsi yang menghitung XOR dari seluruh tumpukan.

c) Deteksi posisi menang/kalah: Berdasarkan hasil NIM-sum.

d) Fungsi saran langkah optimal: Jika posisi menang, program akan memberikan langkah optimal.

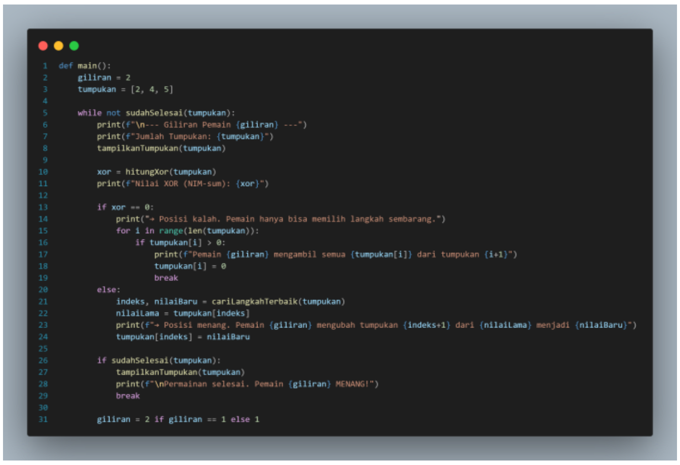
e) Loop permainan: Menampilkan jalannya permainan secara bergantian antara dua pemain simulasi (otomatis/manual).

3) Cara Kerja Algoritma

Berikut adalah gambaran algoritma dasar dari simulasi permainan NIM yang penulis buat:

1. *Buat konfigurasi awal tumpukan dalam bentuk list (misalnya [2,3,4]) serta set giliran pemain dimulai dari pemain ke-berapa.*
2. *Selama permainan belum selesai (selama masih ada tumpukan yang tidak kosong program akan mencetak konfigurasi tumpukan saat ini serta menghitung NIM-sum dari seluruh tumpukan dengan operasi XOR.*
3. *Jika nimSum = 0:*
   a) Pemain dalam posisi kalah, sehingga tidak ada saran langkah optimal, pemain melakukan langkah sembarang.
4. *Jika nimSum ≠ 0:*
   a) Pemain dalam posisi menang, temukan tumpukan i dengan kondisi tumpukan[i] ⊕ nimSum < tumpukan[i].
   b) Gantilah nilai tumpukan tersebut dengan tumpukan[i] = tumpukan[i] ⊕ nimSum.
5. *Perbarui konfigurasi, ganti giliran pemain, ulangi proses hingga seluruh tumpukan bernilai nol.*

B. *Kode Program*



```python
def main():
    giliran = 2
    tumpukan = [2, 4, 5]

    while not sudahSelesai(tumpukan):
        print(f"\n--- Giliran Pemain {giliran} ---")
        print(f"Jumlah Tumpukan: {tumpukan}")
        tampilkanTumpukan(tumpukan)

        xor = hitungXor(tumpukan)
        print(f"Nilai XOR (NIM-sum): {xor}")

        if xor == 0:
            print("→ Posisi kalah. Pemain hanya bisa memilih langkah sembarang.")
            for i in range(len(tumpukan)):
                if tumpukan[i] > 0:
                    print(f"Pemain {giliran} mengambil semua {tumpukan[i]} dari tumpukan {i+1}")
                    tumpukan[i] = 0
                    break
        else:
            indeks, nilaiBaru = cariLangkahTerbaik(tumpukan)
            nilaiLama = tumpukan[indeks]
            print(f"→ Posisi menang. Pemain {giliran} mengubah tumpukan {indeks+1} dari {nilaiLama} menjadi {nilaiBaru}")
            tumpukan[indeks] = nilaiBaru

        if sudahSelesai(tumpukan):
            tampilkanTumpukan(tumpukan)
            print(f"\nPermainan selesai. Pemain {giliran} MENANG!")
            break

        giliran = 2 if giliran == 1 else 1
```

Fig. 3. Kode Program Fungsi Main

Fig. 3. Merupakan fungsi utama yang menjalankan permainan NIM secara bergiliran, yaitu menampilkan tumpukan dan nilai XOR, mengecek posisi menang atau kalah, mengubah tumpukan berdasarkan strategi, serta menentukan siapa pemenang jika semua tumpukan habis.

```python
def tampilkanTumpukan(tumpukan):
    print("\nTampilan Tumpukan:")
    for i, jumlah in enumerate(tumpukan):
        simbol = "|" * jumlah if jumlah > 0 else "-"
        print(f"Tumpukan {i+1}: {simbol} ({jumlah})")
```

Fig. 4. Kode Program Prosedur tampilkanTumpukan

Fig. 4. Adalah prosedur untuk menampilkan representasi visual dari setiap tumpukan. Tumpukan divisualisasikan dengan simbol *bars* (|) sesuai dengan banyaknya objek. Jika kosong, diganti dengan (-).

```python
def hitungXor(tumpukan):
    xor = 0
    for banyak in tumpukan:
        xor ^= banyak
    return xor
```

Fig. 5. Kode Program Fungsi hitungXor

Fig. 5. Adalah fungsi untuk menghitung NIM-sum dari seluruh tumpukan yang merupakan kunci strategi dari permainan NIM (jika XOR = 0, maka posisi kalah, sedangkan jika XOR ≠ 0, maka posisi menang dan bisa dikalkulasi langkah optimalnya).

```python
def cariLangkahTerbaik(tumpukan):
    xorSekarang = hitungXor(tumpukan)
    for i, banyak in enumerate(tumpukan):
        sisa = banyak ^ xorSekarang
        if sisa < banyak:
            return i, sisa
    return None, None
```

Fig. 6. Kode Program Fungsi cariLangkahTerbaik

Fig. 6. Merupakan fungsi untuk mencari tumpukan yang bisa dikurangi sehingga NIM-sum menjadi nol (0). Logika dari fungsi ini adalah mencari indeks *i* dan nilai *sisa* sehingga *jumlah ^ nim_sum < jumlah*.

```python
def sudahSelesai(tumpukan):
    return all(b == 0 for b in tumpukan)
```

Fig. 7. Kode Program Fungsi cariLangkahTerbaik

Fig. 7. Merupakan fungsi untuk memeriksa apakah semua tumpukan sudah kosong atau belum.

C. Hasil Simulasi dan Analisis

Untuk menguji kebenaran strategi optimal permainan NIM menggunakan operasi XOR, dilakukan simulasi menggunakan program Python sebagaimana dijelaskan pada bagian sebelumnya. Simulasi ini bertujuan untuk memverifikasi konsep posisi menang dan kalah berdasarkan perhitungan *NIM-sum*, serta mengilustrasikan bagaimana algoritma dapat secara otomatis menentukan langkah optimal.

Simulasi dijalankan dengan konfigurasi awal tiga tumpukan, masing-masing berisi 2,3, dan 4 objek. Output dari program adalah sebagai berikut:

```
--- Giliran Pemain 1 ---
Jumlah Tumpukan: [2, 3, 4]

Tampilan Tumpukan:
Tumpukan 1: || (2)
Tumpukan 2: ||| (3)
Tumpukan 3: |||| (4)
Nilai XOR (NIM-sum): 5
→ Posisi menang. Pemain 1 mengubah tumpukan 3 dari 4 menjadi 1

--- Giliran Pemain 2 ---
Jumlah Tumpukan: [2, 3, 1]

Tampilan Tumpukan:
Tumpukan 1: || (2)
Tumpukan 2: ||| (3)
Tumpukan 3: | (1)
Nilai XOR (NIM-sum): 0
→ Posisi kalah. Pemain hanya bisa memilih langkah sembarang.
Pemain 2 mengambil semua 2 dari tumpukan 1

--- Giliran Pemain 1 ---
Jumlah Tumpukan: [0, 3, 1]

Tampilan Tumpukan:
Tumpukan 1: - (0)
Tumpukan 2: ||| (3)
Tumpukan 3: | (1)
Nilai XOR (NIM-sum): 2
→ Posisi menang. Pemain 1 mengubah tumpukan 2 dari 3 menjadi 1

--- Giliran Pemain 2 ---
Jumlah Tumpukan: [0, 1, 1]
Tampilan Tumpukan:
Tumpukan 1: - (0)
Tumpukan 2: | (1)
Tumpukan 3: | (1)
Nilai XOR (NIM-sum): 0
→ Posisi kalah. Pemain hanya bisa memilih langkah sembarang.
Pemain 2 mengambil semua 1 dari tumpukan 2

--- Giliran Pemain 1 ---
Jumlah Tumpukan: [0, 0, 1]

Tampilan Tumpukan:
Tumpukan 1: - (0)
Tumpukan 2: - (0)
Tumpukan 3: | (1)
Nilai XOR (NIM-sum): 1
→ Posisi menang. Pemain 1 mengubah tumpukan 3 dari 1 menjadi 0

Tampilan Tumpukan:
Tumpukan 1: - (0)
Tumpukan 2: - (0)
Tumpukan 3: - (0)

Permainan selesai. Pemain 1 MENANG!
```

Fig. 8. Output Program pada Terminal

Hasil simulasi di atas menunjukkan bahwa pada giliran pertama, Pemain 1 berada di posisi menang karena NIM-sum ≠ 0. Pemain 1 mampu mengubah posisi permainan menjadi posisi kalah bagi lawan dengan langkah strategis, sedangkan Pemain 2 berada di posisi kalah (NIM-sum = 0) dan tidak bisa mengambil langah yang mengarah ke kemenangan. Pola ini berulang dan terus dikontrol oleh Pemain 1 yang secara konsisten menjaga posisi lawan tetap kalah setelah setiap gilirannya.

Dari simulasi, terlihat bahwa hasil perhitungan NIM-sum konsisten dengan posisi menang atau kalah pemain, dan langkah-langkah yang diambil program sesuai dengan strategi optimal berdasarkan operasi XOR. Simulasi berhasil memverifikasi teori strategi optimal pada permainan NIM. Hasil simulasi yang telah dilakukan menunjukkan bahwa strategi optimal dalam permainan NIM dapat diimplementasikan dengan sangat efektif menggunakan pendekatan aljabar Boolean, khususnya melalui operasi XOR.

V. KESIMPULAN

Permainan NIM merupakan permainan strategi dua pemain yang dapat dianalisis secara matematis menggunakan aljabar Boolean, khususnya operasi XOR. Melalui konsep NIM-sum, pemain dapat menentukan apakah ia berada dalam posisi menang atau kalah, dan memilih langkah yang optimal berdasarkan hasil operasi XOR dari seluruh tumpukan. Strategi ini membuktikan bahwa permainan sederhana pun dapat mengandung struktur logika yang mendalam.

Simulasi program menggunakan Python memperkuat pemahaman terhadap teori tersebut. Hasilnya menunjukkan bahwa strategi berbasis XOR mampu mengarahkan pemain menuju kemenangan jika diterapkan dengan tepat. Temuan ini memperlihatkan bahwa konsep dalam matematika diskrit tidak hanya bersifat teoritis, tetapi juga dapat diterapkan dalam praktik nyata, terutama dalam konteks teori permainan dan pengambilan keputusan strategis.

LINK VIDEO

Berikut adalah link video pembahasan dari makalah ini: Presentasi Matematika Diskrit | 13524038

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada dosen mata kuliah Matematika Diskrit yang telah memberikan bimbingan, materi, dan arahan dalam penyusunan makalah ini. Ucapan terima kasih juga disampaikan kepada rekan-rekan mahasiswa yang telah memberikan masukan serta semangat selama proses penyusunan berlangsung.

Tidak lupa, penulis juga mengapresiasi berbagai sumber bacaan dan referensi yang telah memperkaya pemahaman mengenai permainan NIM dan konsep aljabar Boolean. Semoga makalah ini dapat memberikan manfaat dan menjadi dasar pembelajaran lebih lanjut bagi siapa pun yang ingin memahami penerapan matematika diskrit dalam strategi permainan.

REFERENSI

[1] C. Bouton, "Nim, a Game with a Complete Mathematical Theory," *Annals of Mathematics*, vol. 3, no. 1/4, pp. 35–39, 1901. [diakses 15 Juni 2025].

[2] M. Gardner, *Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game "Life"*, Scientific American, 1970.[diakses 15 Juni 2025].

[3] L. Tehuayo, Z. A. Leleury, dan F. Kondolembang, "Penerapan Teori Kongruensi dalam Permainan NIM," *Barekeng: Jurnal Ilmu Matematika dan Terapan*, vol. 11, no. 2, pp. 151–162, Desember 2017.[diakses 18 Juni 2025].

[4] Wikipedia, "Nim," *Wikipedia, The Free Encyclopedia*, 13 Juni 2025. [Online]. Tersedia: https://en.wikipedia.org/wiki/Nim [Diakses: 18 Juni 2025].

[5] Rinaldi Munir, "Aljabar Boolean (Bag. 1)", 12-Aljabar-Boolean-(2024)-bagian1.[diakses 18 Juni 2025].

[6] S. Matheus dan F. A. Filho, "A Strategy to Play and Win in Nim Game," *Research, Society and Development*, vol. 10, no. 10, pp. 1–10, 2021.[diakses 19 Juni 2025].

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, 20 Juni 2025

An-Dafa Anza Avansyah - 13524038