

Penyelesaian Permainan Lights Out dengan Algoritma Brute Force

Sa'ad Abdul Hakim - 13522092

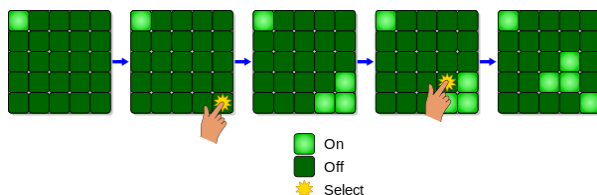
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13522092@std.stei.itb.ac.id

Abstract— Lights Out adalah permainan puzzle dimana pemain harus mematikan semua lampu pada papan grid dengan menekan beberapa lampu yang mempengaruhi lampu di sekitarnya. Penyelesaian permainan ini dapat dicapai dengan berbagai metode, namun penelitian ini memfokuskan pada penggunaan algoritma brute force untuk menemukan solusi. Algoritma brute force mencoba setiap kemungkinan kombinasi penekanan lampu hingga menemukan Solusi yang mematikan semua lampu di papan. Meskipun metode ini tidak efisien dalam hal waktu komputasi karena kompleksitas eksponensialnya, tetapi algoritma ini menjamin dihasilkannya solusi, jika ada. Makalah ini menjelaskan langkah-langkah implementasi algoritma brute force untuk permainan Lights Out dengan ukuran papan tertentu, analisis performa, serta keuntungan dan kerugiannya.

Kata Kunci—Lights Out, brute force, puzzle, penyelesaian

I. PENDAHULUAN

Permainan Lights Out adalah representasi klasik dari permasalahan teka-teki, di mana pemain dihadapkan pada tugas untuk mematikan semua lampu pada papan. Setiap lampu memiliki dua status, yaitu menyala atau mati. Tujuan utama dalam permainan ini adalah mengubah status setiap lampu sedemikian rupa sehingga tidak ada yang menyala pada akhir permainan.



Gambar 1.1 Permainan Lights Out

Sumber:

https://en.wikipedia.org/wiki/Lights_Out_%28game%29

Penyelesaian dari permainan Lights Out menjadi topik menarik dalam bidang komputasi dan teori kompleksitas. Berbagai pendekatan telah diusulkan untuk menyelesaikan permainan ini, mulai dari pendekatan heuristik hingga pendekatan yang lebih eksak. Salah satu pendekatan eksak yang paling mendasar adalah menggunakan algoritma brute force.

Makalah ini bertujuan untuk mengulas penyelesaian permainan Lights Out dengan menggunakan algoritma brute force. Makalah ini akan menyajikan konsep dasar dari permainan Lights Out, membahas secara rinci algoritma brute force, serta menganalisis kompleksitas waktu yang terkait dengan pendekatan ini. Selain itu, makalah ini juga akan menyajikan implementasi algoritma brute force dalam bahasa pemrograman Python, beserta hasil eksperimen yang menunjukkan kinerja algoritma tersebut dalam menyelesaikan permainan Lights Out.

Melalui makalah ini, penulis bertujuan untuk memberikan pemahaman yang mendalam tentang penyelesaian permainan Lights Out dengan menggunakan algoritma brute force, serta untuk mengidentifikasi potensi dan keterbatasan dari pendekatan ini dalam menangani permasalahan teka-teki yang kompleks. Penulis berharap makalah ini dapat menjadi sumber referensi yang bermanfaat bagi para peneliti dan praktisi dalam memahami dan menerapkan algoritma brute force dalam konteks permainan teka-teki atau puzzle.

II. DASAR TEORI

A. Brute Force

Algoritma Brute Force adalah pendekatan yang menekankan penyelesaian masalah dengan cara yang lurus dan sederhana. Berakar pada pernyataan dari persoalan yang dihadapi serta definisi serta konsep yang terlibat, pendekatan ini menawarkan solusi dengan cara yang terstruktur dan langsung. Namun demikian, karakteristik utama algoritma ini menunjukkan bahwa ketidakefisiennya dalam hal kompleksitas komputasi. Algoritma Brute Force jarang mengedepankan kecerdasan atau kreativitas, sebaliknya, algoritma ini lebih mengandalkan "tenaga", yang kadang-kadang diidentifikasi sebagai algoritma naif. Meskipun demikian, kelebihan utama algoritma brute force adalah kemampuannya untuk menyelesaikan hampir semua jenis persoalan, terutama saat ukuran masukan masih relatif kecil.

Keunggulan algoritma Brute Force terutama terletak pada kemampuannya yang luas untuk digunakan dalam berbagai jenis masalah. Algoritma ini dikenal sederhana dan mudah dipahami, sehingga sering dijadikan dasar untuk perbandingan dengan algoritma-algoritma lain yang lebih efisien. Namun, kendati memiliki aplikasi yang luas, algoritma Brute Force memiliki kelemahan yang signifikan. Salah satu kelemahan

utamanya adalah kinerjanya yang lambat, terutama ketika dihadapkan pada masukan dengan ukuran yang besar. Hal ini membuatnya kurang ideal untuk penyelesaian masalah skala besar atau waktu eksekusi yang terbatas.

Dalam konteks kekuatan dan kelemahan, algoritma Brute Force tetap relevan dan layak dipertimbangkan, terutama saat tidak ada solusi yang lebih efisien tersedia atau ketika kompleksitas masalah masih tergolong rendah. Meskipun jarang menghasilkan algoritma yang efisien, keberadaan algoritma Brute Force penting karena kemampuannya untuk menyelesaikan sebagian besar masalah, meskipun dengan tingkat efisiensi yang rendah. Oleh karena itu, sementara ada kelemahan yang perlu diperhatikan, algoritma Brute Force tetap menjadi instrumen penting dalam katalog algoritma komputer dan terus menjadi fokus penelitian dan pengembangan dalam upaya untuk meningkatkan kinerjanya.

B. Permainan Lights Out

Permainan Lights Out mengusung konsep sebuah grid dengan beberapa variasi ukuran yang terdiri dari lampu-lampu. Saat permainan dimulai, sejumlah lampu, baik itu jumlah yang dipilih secara acak maupun pola yang telah tersimpan sebelumnya, akan berada dalam keadaan menyala. Setiap kali seorang pemain menekan salah satu lampu, tidak hanya lampu yang ditekan yang akan berubah keadaannya, tetapi juga keadaan dari empat lampu tetangganya. Tujuan utama dari teka-teki ini adalah untuk mematikan semua lampu yang menyala, idealnya dengan menggunakan sejumlah tekanan tombol yang seefisien mungkin.

Dalam setiap iterasi permainan, pemain ditantang untuk menemukan solusi yang optimal untuk mematikan semua lampu. Penekanan tombol yang tidak hanya mempengaruhi lampu yang ditekan, tetapi juga lampu-lampu di sekitarnya, membuat pemain harus mempertimbangkan setiap langkah dengan cermat. Mereka harus mencari pola atau strategi yang memungkinkan untuk mematikan sebanyak mungkin lampu dalam satu tekanan tombol.

Permainan Lights Out menawarkan tantangan yang menarik dari segi pemikiran strategis dan analisis algoritmik. Pemain dihadapkan pada tantangan untuk merencanakan serangkaian langkah yang efisien untuk mencapai tujuan utama permainan. Dengan demikian, permainan ini bukan hanya menghibur, tetapi juga merangsang pemikiran analitis dan strategis, serta meningkatkan kemampuan pemain dalam merumuskan dan melaksanakan strategi yang efektif.

III. IMPLEMENTASI DAN PENGUJIAN

Dalam implementasi program penyelesaian permainan Lights Out dengan menggunakan algoritma brute force, beberapa langkah penting diperlukan untuk memastikan algoritma berfungsi dengan baik dan efisien. Algoritma brute force harus mampu mencoba semua kemungkinan kombinasi pemencetan tombol untuk menemukan solusi yang mematikan semua lampu di papan permainan. Implementasi algoritma brute force mencakup penerapan kombinasi penekanan pada

papan dan pemeriksaan apakah semua lampu mati setelah setiap kombinasi.

Selain itu, beberapa fungsi pendukung diperlukan untuk mendukung operasi algoritma brute force, seperti fungsi untuk mengubah status lampu (on/off), menekan lampu dan lampu-lampu tetangganya, serta memeriksa apakah semua lampu di papan sudah mati. Fungsi-fungsi ini penting untuk memastikan setiap langkah penekanan dilakukan dengan benar dan hasilnya diperiksa dengan akurat.

Berikut adalah implementasi lengkap program tersebut dalam bahasa python:

```
program.py
import itertools

# Fungsi untuk mengubah status lampu (on/off)
def toggle(board, x, y):
    if 0 <= x < len(board) and 0 <= y < len(board[0]):
        board[x][y] = 1 - board[x][y]

# Fungsi untuk menekan lampu dan lampu-lampu tetangganya
def press(board, x, y):
    toggle(board, x, y)
    toggle(board, x+1, y)
    toggle(board, x-1, y)
    toggle(board, x, y+1)
    toggle(board, x, y-1)

# Fungsi untuk menerapkan kombinasi penekanan pada papan
def apply_combination(board, combination, steps):
    for i in range(len(combination)):
        if combination[i]:
            x = i // len(board[0])
            y = i % len(board[0])
            press(board, x, y)
            steps.append((x, y))

# Fungsi untuk memeriksa apakah semua lampu mati
def all_off(board):
    return all(cell == 0 for row in board for cell in row)
```

```

# Fungsi utama brute force untuk menemukan solusi
def brute_force_lights_out(board):
    n = len(board)
    m = len(board[0])
    steps = []
    for combination in itertools.product([False, True],
repeat=n*m):
        temp_board = [row[:] for row in board]
        apply_combination(temp_board, combination, steps)
        if all_off(temp_board):
            return steps
        steps.clear() # Kosongkan langkah jika tidak
menemukan solusi
    return None

# Fungsi untuk membaca kondisi awal papan dari pengguna
def read_initial_board():
    print("Masukkan jumlah baris dan kolom papan lampu (n
x m):")
    n = int(input("Masukkan jumlah baris: "))
    m = int(input("Masukkan jumlah kolom: "))
    print("Masukkan kondisi awal papan tiap baris (gunakan
0 untuk lampu mati dan 1 untuk lampu hidup):")
    board = []
    for i in range(n):
        row = list(map(int, input().split()))
        if len(row) != m:
            print("Jumlah kolom tidak sesuai!")
            return read_initial_board()
        board.append(row)
    return board

# Fungsi untuk mencetak kondisi papan
def print_board(board):
    print("Kondisi papan saat ini:")
    for row in board:
        print(" ".join(map(str, row)))

```

Program utama harus mampu menerima masukan kondisi awal papan dari pengguna, menjalankan algoritma brute force untuk menemukan solusi, dan mencetak urutan pemencetan tombol yang menghasilkan solusi. Dalam hal ini, program utama akan meminta pengguna untuk memasukkan kondisi awal papan permainan yang terdiri dari matriks nilai biner, di

mana 1 mewakili lampu yang menyala dan 0 mewakili lampu yang mati. Setelah menerima masukan, program akan menjalankan algoritma brute force untuk mencari solusi optimal yang dapat mematikan semua lampu dengan jumlah penekanan tombol yang minimal. Selain itu, program utama juga harus mencatat dan melaporkan waktu eksekusi untuk memberikan gambaran tentang efisiensi algoritma. Hal tersebut penting untuk mengevaluasi performa algoritma brute force, mengingat kompleksitas komputasinya yang tinggi terutama pada papan permainan berukuran besar. Dengan mencatat waktu eksekusi, kita dapat mengidentifikasi batasan kinerja algoritma dan mempertimbangkan alternatif atau optimasi jika diperlukan.

Berikut adalah implementasi program utama dalam bahasa python:

```

main.py
from program import *

    print("Selamat datang di program penyelesaian permainan
Lights Out dengan algorithm brute force!")
    print("Silakan masukkan kondisi awal papan lampu:")
    # Baca kondisi awal papan dari pengguna
    initial_board = read_initial_board()

    # Cetak kondisi awal papan
    print_board(initial_board)

    # Temukan solusi dengan brute force
    start_time = time.time()
    solution = brute_force_lights_out(initial_board)
    end_time = time.time()

    # Cetak solusi jika ditemukan
    if solution:
        print("\nSolusi ditemukan! Urutan pemencetan tombol:")
        for step in solution:
            print(f"Tekan lampu di {step}")
        else:
            print("\nTidak ada solusi yang ditemukan.")
        print(f"Waktu eksekusi: {end_time - start_time} detik")

```

Pengujian dilakukan untuk memastikan bahwa program yang diimplementasikan berfungsi dengan benar dan efisien dalam menyelesaikan permainan Lights Out menggunakan algoritma brute force. Pengujian ini melibatkan berbagai skenario untuk menguji keandalan dan kinerja program, baik

dalam kondisi normal maupun kondisi ekstrem. Tujuan dari pengujian ini adalah untuk mengidentifikasi kekuatan dan kelemahan algoritma brute force dalam konteks permainan Lights Out, serta untuk memberikan wawasan mengenai efisiensi dan skalabilitas algoritma tersebut.

Pengujian dilakukan dengan beberapa pendekatan, termasuk pengujian kasus uji standar, pengujian kasus uji khusus, dan pengujian kinerja. Pengujian kasus uji standar melibatkan papan permainan berukuran 5x5 dengan kondisi awal di mana semua lampu menyala, untuk memastikan bahwa program dapat menemukan solusi yang benar. Pengujian kasus uji khusus melibatkan kondisi papan yang lebih kompleks dan beberapa variasi untuk mengevaluasi kemampuan program dalam menyelesaikan masalah dengan berbagai konfigurasi awal. Terakhir, pengujian kinerja dilakukan untuk mengukur waktu eksekusi program dengan berbagai ukuran papan dan kondisi awal, guna mengevaluasi efisiensi algoritma brute force yang diimplementasikan.

Berikut merupakan hasil pengujian:

```
Kondisi papan saat ini:
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1

Solusi ditemukan! Urutan pemencetan tombol:
Tekan lampu di (0, 3)
Tekan lampu di (0, 4)
Tekan lampu di (1, 0)
Tekan lampu di (1, 1)
Tekan lampu di (1, 3)
Tekan lampu di (1, 4)
Tekan lampu di (2, 0)
Tekan lampu di (2, 1)
Tekan lampu di (2, 2)
Tekan lampu di (3, 1)
Tekan lampu di (3, 2)
Tekan lampu di (3, 3)
Tekan lampu di (4, 0)
Tekan lampu di (4, 2)
Tekan lampu di (4, 3)
waktu eksekusi: 41.282124519348145 detik
```

Gambar 3.1 Uji standar dengan ukuran 5x5 dan kondisi semua lampu menyala
Sumber: penulis

```
Kondisi papan saat ini:
1 1 1 0 0
1 1 0 1 1
1 1 0 0 0
0 0 1 0 1
1 0 1 1 0

Solusi ditemukan! Urutan pemencetan tombol:
Tekan lampu di (0, 2)
Tekan lampu di (0, 3)
Tekan lampu di (0, 4)
Tekan lampu di (1, 0)
Tekan lampu di (1, 2)
Tekan lampu di (1, 3)
Tekan lampu di (2, 1)
Tekan lampu di (2, 2)
Tekan lampu di (2, 4)
Tekan lampu di (3, 0)
Tekan lampu di (3, 1)
Tekan lampu di (3, 2)
Tekan lampu di (3, 3)
Tekan lampu di (3, 4)
Tekan lampu di (4, 2)
Tekan lampu di (4, 3)
waktu eksekusi: 84.02945327758789 detik
```

Gambar 3.2 Uji khusus pertama dengan ukuran 5x5 dan kondisi nyala lampu acak
Sumber: penulis

```
Kondisi papan saat ini:
0 1 0 1 1
0 1 1 0 1
1 1 0 1 0
0 0 1 1 0
0 0 0 0 0

Solusi ditemukan! Urutan pemencetan tombol:
Tekan lampu di (0, 2)
Tekan lampu di (0, 4)
Tekan lampu di (1, 2)
Tekan lampu di (1, 3)
Tekan lampu di (2, 4)
Tekan lampu di (3, 0)
Tekan lampu di (3, 1)
Tekan lampu di (3, 2)
Tekan lampu di (3, 3)
Tekan lampu di (3, 4)
Tekan lampu di (4, 1)
Tekan lampu di (4, 4)
waktu eksekusi: 55.86932158470154 detik
```

Gambar 3.3 Uji khusus kedua dengan ukuran 5x5 dan kondisi nyala lampu acak
Sumber: penulis

```

Kondisi papan saat ini:
1 0 0 0 0
1 0 0 1 0
1 0 1 0 1
1 1 1 1 0
1 0 0 1 1

Solusi ditemukan! Urutan pemencetan tombol:
Tekan lampu di (0, 2)
Tekan lampu di (0, 3)
Tekan lampu di (0, 4)
Tekan lampu di (1, 0)
Tekan lampu di (1, 1)
Tekan lampu di (1, 3)
Tekan lampu di (2, 0)
Tekan lampu di (2, 2)
Tekan lampu di (2, 3)
Tekan lampu di (3, 0)
Tekan lampu di (3, 1)
Tekan lampu di (3, 2)
Tekan lampu di (3, 3)
Tekan lampu di (4, 2)
Tekan lampu di (4, 4)
Waktu eksekusi: 86.4062237739563 detik

```

Gambar 3.4 Uji khusus ketiga dengan ukuran 5x5 dan kondisi nyala lampu acak
Sumber: penulis

```

Kondisi papan saat ini:
1 1 0
1 0 1
1 0 0

Solusi ditemukan! Urutan pemencetan tombol:
Tekan lampu di (0, 0)
Tekan lampu di (0, 1)
Tekan lampu di (1, 0)
Tekan lampu di (1, 1)
Tekan lampu di (1, 2)
Tekan lampu di (2, 2)
Waktu eksekusi: 0.0029790401458740234 detik

```

Gambar 3.5 Uji kinerja pertama dengan ukuran 3x3 dan kondisi nyala lampu acak
Sumber: penulis

```

Kondisi papan saat ini:
1 0 0 1 0
0 0 1 0 0
0 0 0 1 0
0 1 1 0 1
0 1 1 1 1

Solusi ditemukan! Urutan pemencetan tombol:
Tekan lampu di (0, 2)
Tekan lampu di (0, 3)
Tekan lampu di (0, 4)
Tekan lampu di (1, 0)
Tekan lampu di (1, 1)
Tekan lampu di (2, 2)
Tekan lampu di (2, 3)
Tekan lampu di (2, 4)
Tekan lampu di (3, 0)
Tekan lampu di (4, 0)
Tekan lampu di (4, 3)
Waktu eksekusi: 88.55374693870544 detik

```

Gambar 3.6 Uji kinerja kedua dengan ukuran 5x5 dan kondisi nyala lampu acak
Sumber: penulis

```

Kondisi papan saat ini:
0 0 1 1
0 1 0 1
1 0 0 1
1 1 1 0

Solusi ditemukan! Urutan pemencetan tombol:
Tekan lampu di (1, 2)
Tekan lampu di (1, 3)
Tekan lampu di (2, 3)
Tekan lampu di (3, 0)
Tekan lampu di (3, 3)
Waktu eksekusi: 0.004480600357055664 detik

```

Gambar 3.7 Uji kinerja ketiga dengan ukuran 4x4 dan kondisi nyala lampu acak
Sumber: penulis

IV. KESIMPULAN

Dalam makalah ini, telah dibahas penyelesaian permainan Lights Out menggunakan algoritma brute force. Algoritma brute force dipilih karena kesederhanaan dan kemampuannya untuk mencoba semua kemungkinan kombinasi dalam mencari solusi. Implementasi algoritma ini mencakup beberapa langkah penting, termasuk penerapan kombinasi penekanan pada papan permainan, pemeriksaan status lampu, dan pelaporan hasil.

Hasil pengujian menunjukkan bahwa algoritma brute force berhasil menemukan solusi yang mematikan semua lampu di papan permainan, meskipun dengan waktu eksekusi yang bervariasi tergantung pada kompleksitas papan. Algoritma ini

mampu menyelesaikan kasus standar dan beberapa kasus khusus dengan baik, namun kinerjanya menurun secara signifikan pada ukuran papan yang lebih besar atau konfigurasi awal yang lebih kompleks. Hal ini menunjukkan bahwa meskipun algoritma brute force dapat menyelesaikan hampir semua masalah, efisiensinya terbatas oleh jumlah kemungkinan yang harus diperiksa.

Secara keseluruhan, penggunaan algoritma brute force dalam penyelesaian permainan Lights Out membuktikan bahwa metode ini, meskipun tidak efisien untuk semua kasus, tetap merupakan pendekatan yang dapat diandalkan untuk masalah dengan ruang pencarian yang relatif kecil. Untuk ruang pencarian yang lebih besar, diperlukan optimasi atau penggunaan algoritma yang lebih cerdas untuk meningkatkan efisiensi dan mengurangi waktu eksekusi.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya yang melimpah dalam proses penulisan makalah ini sehingga dapat diselesaikan tepat waktu. Penulis juga mengucapkan terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi, S.T, M.Sc. sebagai dosen mata kuliah IF2211 Strategi Algoritma kelas K02, yang telah memberikan bimbingan dan ilmu pengetahuan yang berharga untuk menyelesaikan makalah ini. Penulis juga berterima kasih kepada orang tua yang telah memberikan dukungan moril dan teman-teman yang telah menjadi sumber motivasi selama proses penulisan makalah.

REFERENCES

- [1] R. Munir, 'IF2211 Strategi Algoritma · Semester II Tahun 2023/2024', [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf)
Diakses pada 12 Juni 2024
- [2] R. Munir, 'IF2211 Strategi Algoritma · Semester II Tahun 2023/2024', [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 12 Juni 2024
- [3] <https://www.lightsout.ir/help.html>
Diakses pada 12 Juni 2024
- [4] Repository Github penulis, 'Lights-Out-brute-force', <https://github.com/saadabha/Lights-Out-brute-force>
Diakses pada 12 Juni 2024

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, 12 Juni 2024



Sa'ad Abdul Hakim, 13522092