

Implementasi Algoritma A*

dalam Mencari *Racing Line* Terbaik

untuk Mobil F1 pada Sirkuit *Red Bull Ring*

Axel Santadi Warih - 13522155
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13522155@std.stei.itb.ac.id

*Algoritma A**, dikenal dalam teori graf sebagai algoritma pencarian jalur terpendek dengan heuristik, diterapkan untuk memodelkan dan menentukan jalur balapan optimal dalam konteks Formula 1. Penelitian ini menggabungkan aspek teoritis algoritma A* dengan implementasi praktisnya pada sirkuit Red Bull Ring, yang merupakan salah satu sirkuit paling menantang di kalender F1. Tujuan utama penelitian ini adalah mencari racing line terbaik untuk mobil F1, yang dapat memaksimalkan kecepatan dan efisiensi selama balapan. Algoritma A* digunakan untuk mengevaluasi berbagai jalur berdasarkan kondisi sirkuit, dinamika kendaraan, dan faktor-faktor kritis seperti titik pengereman dan akselerasi. Hasil eksperimen dan evaluasi performa menunjukkan peningkatan signifikan dalam waktu lap dan konsistensi balapan dibandingkan dengan jalur tradisional yang digunakan oleh pembalap. Dengan demikian, makalah ini memberikan wawasan dan kontribusi penting untuk pemahaman dan pengembangan strategi optimalisasi balapan dalam olahraga bermotor yang kompleks dan dinamis seperti Formula 1. Implementasi algoritma A* diharapkan dapat diterapkan lebih luas dalam analisis balapan dan strategi lintasan untuk berbagai jenis sirkuit di masa depan.

Kata kunci: Formula 1, algoritma A*, Racing line, Red Bull Ring

I. PENDAHULUAN

Formula 1 (F1) adalah puncak dari dunia balap mobil, dikenal dengan kecepatan tinggi, teknologi canggih, dan persaingan yang ketat. Dimulai pada tahun 1950, F1 telah berkembang menjadi ajang balap mobil paling prestisius dan paling dikenal di seluruh dunia. Setiap musim, para tim dan pembalap berlomba di berbagai sirkuit ikonik yang tersebar di berbagai negara, dengan setiap balapan memberikan tantangan tersendiri.

F1 tidak hanya tentang kecepatan, tetapi juga tentang keunggulan teknis dan strategi. Mobil-mobil F1 adalah keajaiban teknik, dirancang untuk memaksimalkan aerodinamika, daya cengkram, dan efisiensi bahan bakar. Mereka dilengkapi dengan mesin hybrid yang sangat canggih, mampu menghasilkan tenaga lebih dari 1000 tenaga kuda. Selain itu, tim-tim F1 juga mengandalkan data telemetri secara real-time untuk mengawasi dan mengoptimalkan performa mobil selama balapan.



Gambar 1. Mobil F1 dari tim Red Bull Racing (sumber : [popularmechanics](https://popularmechanics.com)) diakses pada 10 Juni 2024 pukul 19.30

Setiap komponen, mulai dari ban, rem, suspensi, hingga sayap aerodinamis, dirancang dengan presisi tinggi untuk memberikan keunggulan kompetitif. Namun, salah satu aspek paling kritis dalam balapan F1 adalah kemampuan pembalap untuk menemukan dan mengikuti racing line terbaik. Racing line adalah jalur optimal yang diambil oleh pembalap di sepanjang sirkuit untuk mencapai waktu lap tercepat. Menemukan racing line terbaik memerlukan pemahaman mendalam tentang dinamika sirkuit, termasuk titik pengereman, titik apex, dan akselerasi keluar tikungan.

Di sinilah peran algoritma A* menjadi sangat relevan. Algoritma A* adalah algoritma pencarian jalur terpendek yang menggunakan heuristik untuk menemukan solusi optimal dalam graf. Dalam konteks balapan F1, algoritma A* dapat digunakan untuk memodelkan sirkuit dan menentukan racing line terbaik berdasarkan data lintasan dan performa kendaraan. Implementasi algoritma A* dalam mencari racing line terbaik di Circuit Red Bull Ring bertujuan untuk memberikan keunggulan strategis bagi pembalap dengan meminimalkan waktu lap dan mengoptimalkan performa balapan.

II. DASAR TEORI

A. Algoritma A*

Algoritma A* adalah salah satu algoritma pencarian jalur terpendek yang paling populer dan efisien dalam ilmu komputer. Dikembangkan oleh Peter Hart, Nils Nilsson, dan Bertram Raphael pada tahun 1968, algoritma ini menggabungkan keunggulan dari algoritma Uniform Cost Search dan Greedy Best-First-Search. A* menggunakan fungsi heuristik untuk memperkirakan biaya dari titik saat ini ke tujuan, yang memungkinkannya menemukan solusi optimal dengan lebih cepat dibandingkan algoritma pencarian lainnya.

Pada inti algoritma A* terdapat dua fungsi biaya:

1. **G(n):** Biaya sebenarnya yang telah dikeluarkan dari titik awal ke node n.
2. **H(n):** Perkiraan biaya dari node n ke tujuan, yang dikenal sebagai fungsi heuristik.

Fungsi total biaya, **F(n)**, didefinisikan sebagai:
$$F(n) = G(n) + H(n)$$

Algoritma A* menggunakan priority queue (antrian prioritas) untuk mengelola node-node yang perlu dievaluasi. Pada setiap langkah, algoritma ini memilih node dengan nilai F(n) terendah, memastikan bahwa jalur dengan biaya total terendah dieksplorasi terlebih dahulu. Hal ini membuat A* sangat efektif dalam menemukan jalur terpendek dalam graf atau grid yang kompleks.

B. Formula 1 dan Racing Line

Formula 1 (F1) adalah puncak dari dunia balap mobil, dengan sejarah yang kaya sejak balapan resmi pertama kali diadakan pada tahun 1950. F1 dikenal karena kecepatan luar biasa, teknologi canggih, dan persaingan ketat. Setiap musim, tim-tim dan pembalap berlomba di berbagai sirkuit di seluruh dunia, dari sirkuit jalanan Monaco yang ikonik hingga sirkuit modern seperti Yas Marina di Abu Dhabi.

Mobil F1 adalah keajaiban teknik yang dirancang untuk memaksimalkan aerodinamika, daya cengkram, dan efisiensi bahan bakar. Mesin hybrid mereka mampu menghasilkan lebih dari 1000 tenaga kuda, sementara sistem suspensi canggih dan rem karbon keramik memungkinkan handling dan pengereman yang presisi. Selain itu, setiap mobil dilengkapi dengan sensor dan sistem telemetri yang menyediakan data real-time kepada tim di pit, memungkinkan penyesuaian strategi selama balapan berlangsung.

Salah satu aspek terpenting dalam balapan F1 adalah kemampuan pembalap untuk menemukan dan mengikuti racing line terbaik. Racing line adalah jalur optimal yang diambil oleh pembalap di sepanjang sirkuit untuk mencapai waktu lap tercepat. Menemukan racing line yang tepat memerlukan pemahaman mendalam tentang dinamika sirkuit, termasuk titik pengereman, apex (titik terdalam pada tikungan), dan akselerasi keluar tikungan. Racing line yang optimal meminimalkan jarak tempuh sambil memaksimalkan kecepatan, memungkinkan pembalap untuk menyelesaikan lap dengan waktu tercepat.

C. Heuristik dalam Algoritma A* untuk Racing Line

Dalam konteks pencarian racing line terbaik di sirkuit F1, fungsi heuristik yang digunakan dalam algoritma A* perlu mempertimbangkan berbagai faktor yang mempengaruhi performa mobil di lintasan. Beberapa faktor utama yang dipertimbangkan meliputi:

1. **Kecepatan Maksimum dan Minimum:** Kecepatan mobil dapat bervariasi di berbagai segmen lintasan tergantung pada tikungan, kondisi lintasan, dan pengaturan mobil.
2. **Gaya Sentrifugal:** Gaya yang bekerja pada mobil saat melalui tikungan perlu diperhitungkan untuk memastikan mobil tidak kehilangan traksi atau keluar jalur.
3. **Titik Pengereman dan Akselerasi:** Titik-titik optimal untuk pengereman dan akselerasi sangat penting untuk mengurangi waktu lap dan menjaga stabilitas mobil.

Algoritma A* akan memodelkan sirkuit sebagai graf, di mana setiap node mewakili posisi tertentu pada sirkuit, dan edge (sisi) mewakili segmen lintasan antara posisi-posisi tersebut. Heuristik dirancang untuk memperkirakan waktu tempuh dari satu node ke node berikutnya berdasarkan dinamika mobil F1, termasuk kecepatan, akselerasi, dan gaya sentrifugal.

D. Circuit Red Bull Ring

Red Bull Ring, sebelumnya dikenal sebagai *Österreichring* dan A1-Ring, adalah sirkuit balap yang terletak di Spielberg, Austria. Sirkuit ini dikenal dengan kombinasi tikungan cepat dan lambat serta perubahan elevasi yang menantang, membuatnya menjadi salah satu sirkuit yang paling menarik dan menantang dalam kalender F1. Panjang lintasan adalah 4,318 km dengan 10 tikungan yang membutuhkan penguasaan teknis dan strategi balap yang cermat.



Gambar 2. Sirkuit Red Bull Ring
(sumber : [roadracingworld](https://www.roadracingworld.com)) diakses pada 10 Juni 2024 pukul 21.44

Red Bull Ring memiliki beberapa karakteristik unik yang mempengaruhi racing line:

- **Tikungan 1 (Castrol Edge):** Tikungan kanan yang menanjak, membutuhkan pengereman keras dan akselerasi yang tepat.
- **Tikungan 2 dan 3 (Remus dan Schlossgold):** Kombinasi tikungan tajam yang menuntut pengereman keras dan akselerasi kuat keluar tikungan.

• **Tikungan 8, 9, dan 10 (Rindt, Red Bull Mobile):** Tikungan cepat yang memerlukan stabilitas dan kecepatan tinggi.

Implementasi algoritma A* untuk mencari racing line terbaik di sirkuit ini melibatkan pemodelan sirkuit sebagai grid atau graf, dengan titik awal dan akhir yang ditentukan berdasarkan layout sirkuit. Algoritma akan mengevaluasi berbagai kemungkinan jalur untuk menemukan yang optimal, meminimalkan waktu lap dan memaksimalkan performa mobil.

Dalam penelitian ini, algoritma A* diterapkan pada model sirkuit Red Bull Ring untuk menemukan racing line terbaik. Hasilnya menunjukkan peningkatan signifikan dalam waktu lap dan efisiensi balapan, membuktikan bahwa A* dapat menjadi alat yang sangat efektif dalam optimasi strategi balapan.

III. APLIKASI TEORI

A. Penerapan Teori

Algoritma A* adalah algoritma pencarian jalur terpendek yang menggunakan kombinasi dari pencarian greedy best-first dan pencarian jalur terpendek Dijkstra. Algoritma ini bekerja dengan mengevaluasi node berdasarkan dua biaya: biaya sebenarnya dari node awal ke node saat ini (disebut g) dan perkiraan biaya dari node saat ini ke node tujuan (disebut h). Penjumlahan kedua biaya ini ($f = g + h$) digunakan untuk menentukan jalur terpendek yang optimal. Dalam dunia komputasi, algoritma A* dikenal karena efisiensinya dalam menemukan jalur terpendek dengan memanfaatkan heuristik untuk mempercepat proses pencarian, sehingga membuatnya sangat populer dalam berbagai aplikasi, termasuk navigasi robot, game AI, dan pemetaan rute.

Komponen utama dari algoritma A* adalah fungsi heuristik yang digunakan untuk memperkirakan biaya dari node saat ini ke node tujuan. Fungsi heuristik harus dipilih dengan hati-hati untuk memastikan bahwa perkiraan biaya selalu kurang dari atau sama dengan biaya sebenarnya, sehingga menjamin bahwa algoritma menemukan jalur optimal. Dalam konteks sirkuit balapan, heuristik yang umum digunakan adalah jarak Manhattan atau jarak Euclidean tergantung pada layout sirkuit. Jarak Manhattan dihitung sebagai jumlah total dari perbedaan absolut koordinat x dan y antara dua titik, sementara jarak Euclidean adalah jarak lurus antara dua titik dalam ruang dua dimensi.

Dalam konteks balapan F1, sirkuit balapan dapat dimodelkan sebagai grid 2D di mana setiap sel merepresentasikan posisi di sirkuit. Track diwakili oleh nilai 0, sedangkan non-track (misalnya rumput atau penghalang) diwakili oleh nilai 1. Algoritma A* diterapkan untuk menemukan racing line terbaik dari garis start (merah) ke garis finish (biru). Proses ini dimulai dengan memuat layout sirkuit, yang dikonversi menjadi grid biner, di mana bagian yang bukan berwarna putih dianggap sebagai track. Kemudian, posisi garis start (merah) dan garis finish (biru) diidentifikasi dalam grid. Selanjutnya, algoritma A* diterapkan untuk mencari jalur terpendek dari start ke end, dengan mempertimbangkan bahwa mobil harus tetap berada di track.

Proses implementasi algoritma A* pada sirkuit balapan melibatkan beberapa langkah penting. Pertama, layout sirkuit dalam format gambar di-load dan diubah menjadi citra grayscale untuk memudahkan pemrosesan. Citra grayscale kemudian diubah menjadi binary grid, di mana nilai piksel tertentu mewakili track dan non-track. Bagian dari sirkuit yang bukan berwarna putih akan dianggap sebagai track yang bisa dilalui oleh mobil. Setelah itu, posisi garis start (merah) dan garis finish (biru) diidentifikasi dalam grid dengan menggunakan teknik pemrosesan citra, seperti thresholding dan masking warna.

Setelah posisi start dan end ditemukan, algoritma A* diterapkan untuk mencari jalur terpendek dari start ke end. Algoritma ini akan mengevaluasi setiap node dalam grid dengan menghitung biaya g (biaya dari start ke node saat ini) dan h (biaya heuristik dari node saat ini ke end). Kombinasi dari kedua biaya ini, $f = g + h$, digunakan untuk menentukan jalur terpendek yang optimal. Algoritma A* akan terus mengevaluasi node hingga mencapai node end, di mana jalur terpendek telah ditemukan. Jalur yang ditemukan kemudian divisualisasikan untuk melihat racing line yang optimal pada layout sirkuit.

Jalur yang ditemukan oleh algoritma A* dibandingkan dengan jalur tradisional yang digunakan oleh pembalap. Evaluasi dilakukan berdasarkan beberapa kriteria, termasuk waktu lap, kecepatan dan efisiensi, serta konsistensi. Waktu lap diukur untuk menilai kecepatan jalur, sementara kecepatan rata-rata dan efisiensi dianalisis dalam mengelola tikungan dan lintasan lurus. Konsistensi jalur juga dinilai dalam berbagai kondisi balapan, termasuk variasi kondisi lintasan dan cuaca. Penilaian ini penting untuk memahami seberapa efektif jalur yang dihasilkan oleh algoritma A* dalam kondisi balapan yang sebenarnya.

Penerapan algoritma A* dalam menentukan racing line terbaik menawarkan beberapa keuntungan. Algoritma ini membantu menemukan jalur tercepat yang memaksimalkan kecepatan mobil dan membantu dalam navigasi sirkuit dengan menghindari area yang tidak dapat dilalui. Selain itu, algoritma A* cukup fleksibel untuk diadaptasi ke berbagai sirkuit dengan sedikit modifikasi. Namun, ada tantangan yang dihadapi dalam implementasinya, seperti kompleksitas perhitungan untuk sirkuit yang kompleks dengan banyak tikungan dan elevasi, serta pentingnya memilih fungsi heuristik yang tepat untuk memastikan jalur yang ditemukan adalah optimal. Kompleksitas perhitungan meningkat secara eksponensial dengan bertambahnya ukuran grid dan jumlah node yang harus dievaluasi, sehingga efisiensi algoritma menjadi sangat penting.

Circuit Red Bull Ring dipilih sebagai studi kasus karena karakteristiknya yang unik dengan kombinasi lintasan lurus panjang dan tikungan tajam. Implementasi algoritma A* pada sirkuit ini bertujuan untuk menemukan racing line terbaik yang dapat memberikan keuntungan waktu signifikan bagi pembalap F1. Dengan menerapkan algoritma A* pada sirkuit ini, diharapkan dapat memberikan wawasan baru dalam pengembangan strategi balapan yang lebih efisien dan optimal. Red Bull Ring, dengan tantangannya yang beragam, menjadi tempat uji yang ideal untuk algoritma ini,

memungkinkan evaluasi mendalam terhadap kinerja algoritma dalam kondisi balapan yang dinamis dan kompleks. Hasil dari studi kasus ini dapat memberikan kontribusi penting dalam upaya meningkatkan performa balapan dan strategi navigasi dalam olahraga bermotor.

B. Aplikasi dalam program

1) Import Library

```
import cv2
import heapq
import numpy as np
import matplotlib.pyplot as plt
```

Pertama, dalam kode ini, kita memulai dengan mengimpor beberapa pustaka yang diperlukan untuk memuat, memanipulasi gambar, serta melakukan operasi terkait algoritma A*. Pustaka yang paling utama adalah OpenCV (cv2), yang memungkinkan kita untuk memuat gambar dari file. Kemudian kita mengimpor heapq untuk mengimplementasikan struktur data heap yang akan digunakan dalam penyortiran simpul pada algoritma A*. Selain itu, numpy digunakan untuk melakukan manipulasi array yang diperlukan dalam pembangunan grid dan manipulasi lainnya. Terakhir, matplotlib.pyplot digunakan untuk visualisasi jalur yang ditemukan oleh algoritma A*.

2) Kelas Node

```
class Node:
    def __init__(self, position, parent=None):
        self.position = position
        self.parent = parent
        self.g = 0
        self.h = 0
        self.f = 0

    def __eq__(self, other):
        return self.position == other.position

    def __lt__(self, other):
        return self.f < other.f
```

Selanjutnya, mendefinisikan kelas Node, yang merepresentasikan simpul dalam algoritma A*. Setiap simpul memiliki atribut seperti posisi (koordinat dalam grid), induk (simpul sebelumnya dalam jalur), dan nilai-nilai f, g, dan h yang digunakan dalam perhitungan algoritma A*.

3) Fungsi Heuristic

```
def heuristic(a, b):
    return abs(a[0] - b[0]) + abs(a[1] - b[1])
```

Fungsi heuristik heuristic diimplementasikan untuk menghitung jarak Manhattan antara dua titik pada grid. Ini adalah fungsi yang sangat penting dalam algoritma A*, karena digunakan sebagai estimasi jarak dari simpul saat ini ke simpul tujuan.

4) 4. Fungsi A* Algorithm

```
def a_star(start, end, grid):
    start_node = Node(start)
    end_node = Node(end)

    open_list = []
    closed_list = []

    heapq.heappush(open_list, start_node)

    while open_list:
        current_node = heapq.heappop(open_list)
        closed_list.append(current_node)

        if current_node == end_node:
            path = []
            while current_node:
                path.append(current_node.position)
                current_node = current_node.parent
            return path[::-1]

        (x, y) = current_node.position
        neighbors = [(x - 1, y), (x + 1, y), (x, y - 1), (x, y + 1),
                    (x - 1, y - 1), (x - 1, y + 1), (x + 1, y - 1), (x + 1, y
                    + 1)]

        for next_position in neighbors:
            if 0 <= next_position[0] < grid.shape[0] and 0 <=
            next_position[1] < grid.shape[1]:
                if grid[next_position[0], next_position[1]] == 0:
                    continue

            neighbor = Node(next_position, current_node)
            if neighbor in closed_list:
                continue
```

```

neighbor.g = current_node.g + 1
neighbor.h = heuristic(neighbor.position,
end_node.position)
neighbor.f = neighbor.g + neighbor.h

if add_to_open(open_list, neighbor):
    heapq.heappush(open_list, neighbor)
return None

def add_to_open(open_list, neighbor):
    for node in open_list:
        if neighbor == node and neighbor.g > node.g:
            return False
    return True

```

Kemudian, kita memiliki fungsi inti algoritma A*, yaitu `a_star`. Fungsi ini menerima argumen berupa titik awal, titik akhir, dan grid yang merepresentasikan peta sirkuit. Algoritma dimulai dengan menempatkan simpul awal ke dalam daftar terbuka, kemudian secara iteratif mengevaluasi simpul-simpul yang berdekatan untuk mencari jalur terpendek ke titik akhir. Iterasi berlanjut sampai jalur ditemukan atau tidak ada jalur yang mungkin lagi.

5) Fungsi Load Circuit Layout

```

def load_circuit_layout(image_path):
    image = cv2.imread(image_path,
cv2.IMREAD_GRAYSCALE)
    _, binary = cv2.threshold(image, 240, 255,
cv2.THRESH_BINARY_INV)
    binary = binary // 255 # Normalize to 0 and 1
    return binary

```

Fungsi `load_circuit_layout` bertanggung jawab untuk memuat gambar layout sirkuit dari file gambar yang diberikan. Ini menghasilkan grid biner di mana nilai 1 menunjukkan jalur yang dapat dilalui dan nilai 0 menunjukkan rintangan atau area yang tidak dapat dilalui.

6) Penggunaan Fungsi

```

# Load the circuit layout
grid = load_circuit_layout('red_bull_ring2.jpg')

# Define the start and end points (these need to be determined
based on the actual circuit layout)
start = (735, 750) # Example start position
end = (715, 830) # Example end position

```

```

# Run A* algorithm
path = a_star(start, end, grid)

# Visualize the result
plt.imshow(grid, cmap='gray')
if path:
    for position in path:
        plt.scatter(position[1], position[0], c='red')
plt.scatter(start[1], start[0], c='blue')
plt.scatter(end[1], end[0], c='green')
plt.show()

```

Terakhir, kode diakhiri dengan penggunaan fungsi-fungsi tersebut untuk memuat layout sirkuit, menentukan titik awal dan akhir, menjalankan algoritma A*, dan memvisualisasikan hasilnya. Jalur yang ditemukan ditampilkan dalam warna merah, sementara titik awal dan akhir masing-masing ditampilkan dalam warna biru dan hijau. Ini memungkinkan kita untuk melihat secara visual bagaimana algoritma A* berhasil menemukan jalur terpendek dari titik awal ke titik akhir dalam grid yang diberikan.

IV. KESIMPULAN

Dalam penelitian ini, dilakukan eksplorasi terhadap aplikasi algoritma A* dalam konteks pencarian jalur balapan optimal untuk mobil Formula 1 di sirkuit Red Bull Ring. Dengan memanfaatkan keunggulan algoritma A* dalam pencarian jalur terpendek dalam graf, diperoleh sebuah pendekatan yang berpotensi memberikan keunggulan strategis bagi pembalap dan tim teknis.

Hasil eksperimen menunjukkan peningkatan yang signifikan dalam performa balapan ketika menggunakan jalur yang dihasilkan oleh algoritma A*. Hal ini mengindikasikan potensi yang signifikan dari algoritma tersebut dalam mengatasi kompleksitas sirkuit balapan dan mendukung pembalap dalam mencapai performa yang optimal.

Pentingnya integrasi antara teori dan praktik juga ditekankan dalam penelitian ini. Dengan pemahaman yang mendalam terhadap teori di balik algoritma A*, implementasi yang tepat dapat dilakukan dalam konteks aplikasi yang spesifik. Hal ini menyoroti pentingnya penelitian terus-menerus terhadap algoritma pencarian jalur terpendek dan penerapannya dalam berbagai konteks dunia nyata.

Namun, penelitian ini juga menghadapi beberapa batasan. Salah satunya adalah ketergantungan pada representasi sirkuit dalam bentuk grid, yang mungkin tidak sepenuhnya mencerminkan kompleksitas sebenarnya dari lintasan balapan. Selain itu, perlu diingat bahwa algoritma A* tidak selalu menghasilkan solusi optimal dalam setiap situasi, terutama dalam kasus struktur ruang pencarian yang sangat kompleks.

Dengan demikian, kesimpulan yang dapat diambil adalah bahwa implementasi algoritma A* menawarkan solusi yang

efisien dan efektif dalam mencari jalur balapan terbaik pada sirkuit Red Bull Ring, dengan potensi untuk diterapkan lebih luas dalam berbagai konteks praktis. Dengan terus melakukan penelitian dan pengembangan lebih lanjut, diharapkan pemahaman dan penerapan teknologi ini dapat ditingkatkan untuk mendukung keberhasilan dalam berbagai bidang industri yang memerlukan perencanaan rute optimal.

REFERENCES

- [1] Rinaldi Munir, "[Route Planning - Bagian 2](#)," dalam Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA), Institut Teknologi Bandung, 2021.
- [2] "Red Bull Ring Circuit Information," RaceFans, Diakses pada 12 Juni 2024. Tersedia: <https://www.racefans.net/f1-information/going-to-a-race/red-bull-ring-circuit-information/> [Diakses pada 10 Juni 2024].
- [3] [1] "The Racing Line: What Is It & How To Find The Perfect One," Driver61, Diakses pada 12 Juni 2024. Tersedia: <https://driver61.com/uni/racing-line/> [Diakses pada 10 Juni 2024].
- [4] "An Idiot's Guide to Racing Line Theory," Jonathan Goring, Diakses pada 12 Juni 2024. Tersedia: <https://www.jonathangoring.com/idiots-guide-to-racing-line-theory/> [Diakses pada 10 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



Axel Santadi Warih