



## *Application of the Best First Search (BFS) Algorithm for Determining Parking Slots*

### **Penerapan Algoritma Best First Search (BFS) untuk Penentuan Slot Parkir**

**Adi Suswiantara<sup>1\*</sup>, Naufal Dzakhwan<sup>2</sup>, Asraf Fahrudin<sup>3</sup>,  
Muhammad Naufal Dzaky<sup>4</sup>, Auliana<sup>5</sup>, Tedy Setiadi<sup>6</sup>**

<sup>1,2</sup>Program Studi Informatika, Fakultas Teknologi Industri,  
Universitas Ahmad Dahlan, Yogyakarta, Indonesia

E-Mail: <sup>1</sup>2200018058@webmail.uad.ac.id, <sup>2</sup>2200018025@webmail.uad.ac.id,  
<sup>3</sup>2200018046@webmail.uad.ac.id, <sup>4</sup>2200018055@webmail.uad.ac.id,  
<sup>5</sup>2200018029@webmail.uad.ac.id, <sup>6</sup>tedy.setiadi@tif.uad.ac.id

*Received Jul 15th 2024; Revised Aug 25th 2024; Accepted Sept 25th 2024*  
*Corresponding Author: Adi Suswiantara*

#### **Abstract**

*Campus X Motorcycle Parking is one of the parking areas provided by Campus X for its students. This parking area often experiences congestion, especially during peak class hours, making it difficult for users to find available parking slots. The disorganized placement of motorcycles can also make it challenging for users to locate their vehicles. Therefore, a system is needed to efficiently manage and determine parking slots. The objective of this research is to develop a system using the Best First Search (BFS) algorithm that can automatically determine parking slots. The BFS algorithm uses a search approach that prioritizes the most promising node based on a heuristic function at each step. The implementation of the BFS algorithm results in a system that can place motorcycles in available slots based on proximity to the parking entrance, thereby making it easier for students to find and park their vehicles in a more organized manner, and assisting campus parking attendants in managing parking at Campus X Motorcycle Parking. The research methodology taken based on best first search, geographic information system and tree structure.*

*Keywords: BFS, Kampus X, Parking, Placement, Slot*

#### **Abstrak**

Parkir Motor Kampus X merupakan salah satu area parkir yang disediakan oleh Kampus X untuk mahasiswanya. Area parkir ini sering kali mengalami kepadatan, terutama pada jam-jam sibuk perkuliahan, sehingga menyebabkan kesulitan bagi pengguna dalam menemukan slot parkir yang tersedia. Penempatan motor yang tidak teratur dapat membuat pengguna kesulitan menemukan kembali kendaraannya. Oleh karena itu, diperlukan sebuah sistem yang mampu mengatur dan menentukan slot parkir secara efisien. Tujuan dari penelitian ini adalah mengembangkan sistem dengan algoritma Best First Search (BFS) yang dapat menentukan slot parkir secara otomatis. Algoritma BFS adalah algoritma yang menggunakan pendekatan pencarian dengan mengutamakan node yang paling menjanjikan berdasarkan fungsi heuristik pada setiap langkahnya. Implementasi algoritma BFS menghasilkan sistem yang mampu menempatkan motor pada slot yang kosong berdasarkan kedekatan dengan pintu masuk parkir, sehingga memudahkan mahasiswa dalam mencari dan menempatkan kendaraan mereka dengan lebih teratur serta membantu karyawan (juru parkir kampus) dalam mengatur parkir di Parkir Motor Kampus X. Metodologi penelitian diambil berdasarkan best first search, sistem informasi geografis dan struktur tree.

Kata Kunci: BFS, Kampus X, Parkir, Penempatan, Slot

#### **1. PENDAHULUAN**

Parkir motor merupakan salah satu fasilitas penting yang harus disediakan dalam setiap kampus [1]. Dengan semakin bertambahnya Mahasiswa di setiap tahunnya pastinya akan membuat kendaraan di setiap tahun akan meningkat, sehingga pencarian data letak kendaraan menjadi lebih lama dan kurang efisien, terlebih jika pencarian data letak kendaraan tersebut belum Tersistemasi secara maksimal sehingga memiliki tingkat efisiensi rendah, yang pada akhirnya mempersulit pengguna untuk menemukan kembali kendaraannya. Melihat

kondisi tersebut diatas, maka diperlukan adanya sebuah Tersistemasi yang mampu melakukan pencarian data letak kendaraan secara efisien [2]. Area parkir di Kampus X total 10 slot parkir yang terbagi ke dalam satu petak tempat parkir. Meskipun telah disediakan slot yang memadai, pengaturan dan penempatan motor yang efektif masih menjadi tantangan utama.

Dalam rangka mengatasi permasalahan ini, penelitian ini bertujuan untuk mengembangkan sebuah sistem penempatan slot parkir otomatis dengan menggunakan algoritma Best First Search (BFS) [3]. Masalah penentuan rute biasanya diselesaikan dengan menggunakan metode Best-First Search [4]. Memilih rute dengan algoritma BFS, dimana pencarian rute boleh mengunjungi node pada level rendah jika node pada level tinggi memiliki nilai tidak baik, dapat digunakan untuk menyelesaikan masalah pencarian rute terbaik untuk tujuan tempat parkir [5][6]. Algoritma ini memilih node yang paling menjanjikan pada setiap tahap prosedur pencarian terbaik pertama [7]. Manfaat dari breadth first pertama dan depth first search digabungkan dalam pendekatan Best First Search [8, 9]. Setelah itu, secara langsung membangun node yang dipilih dengan menggunakan aturan untuk membuat penggantinya [10]. Dengan menggunakan pendekatan ini, diharapkan sistem yang dikembangkan dapat menentukan slot parkir yang kosong secara cepat dan efisien, serta menempatkan motor berdasarkan kedekatan dengan pintu masuk parkir. Pemilihan algoritma BFS untuk penentuan slot parkir dalam lingkungan kampus didasarkan pada kebutuhan untuk menemukan slot parkir terdekat dengan pintu masuk atau fasilitas utama. BFS memungkinkan pencarian dilakukan secara sistematis dengan menjelajahi slot parkir pada level yang sama sebelum beralih ke level berikutnya, memastikan bahwa slot terdekat yang tersedia akan ditemukan lebih dahulu. Selain itu, kesederhanaan dan keefektifan BFS dalam struktur graf statis menjadikannya pilihan yang sesuai untuk lingkungan parkir kampus yang relatif tidak dinamis.

Penelitian Yosdarso Afero (2021), menyoroti keunggulan Algoritma Best First Search dalam menemukan solusi optimal untuk mengukur jarak dari satu titik ke titik lain yang akan dikunjungi [11]. Algoritma ini sangat efektif dalam memudahkan pencarian solusi optimal. Pengambilan keputusan dapat dibantu dengan penerapan Best First Search dalam sistem informasi geografis [12]. Implementasi Best First Search dalam Sistem Informasi Geografis dapat berfungsi sebagai alat bantu dalam pengambilan keputusan sistem [13]. Penelitian Mardiana (2022), menjelaskan tentang pencarian jalur terbaik menuju lokasi pustaka untuk Sistem Navigasi AR, hasil dari penelitian ini diperoleh sistem navigasi augmented reality perpustakaan dengan berdasarkan 4 user story yang ditentukan, 9 backlog dan 10 task card. Sistem berhasil memberikan informasi mengenai informasi rak buku tujuan berdasarkan nomor katalog dan terdapat juga informasi navigasi seperti jarak, waktu serta rute untuk sampai ke rak buku tujuan. Hasil juga menunjukkan bahwa Algoritma A Star (A\*) memiliki total jalur yang tepat sebanyak 7 dengan persentase keakuratan 70%, Algoritma Dijkstra memiliki total jalur yang tepat sebanyak 9 dengan persentase keakuratan 90%, dan Algoritma BFS memiliki total jalur yang tepat 0 dengan persentase keakuratan 0% [14]. Penelitian Ryano Alderio (2023), menjelaskan mengenai pencarian Rute menggunakan metode Branch and Bound dengan perbandingan pendekatan Breadth First Search, Depth First Search dan Best First Search. Dan hasil perbandingan yang didapat adalah pendekatan Breadth First Search memerlukan perhitungan lebih banyak dengan 28 kali perhitungan, sedangkan untuk pendekatan Depth First Search membutuhkan 25 kali perhitungan, dan untuk pendekatan Best First Search membutuhkan 23 kali perhitungan [15].

Penelitian ini berbeda dengan penelitian sebelumnya karena mengaplikasikan algoritma BFS secara efektif dengan menggunakan pemodelan pohon. Pemodelan ini membantu dalam membimbing kendaraan menuju slot parkir yang tersedia berdasarkan kedekatan dengan pintu masuk, sehingga mengurangi manuver dan kemacetan yang tidak diperlukan.

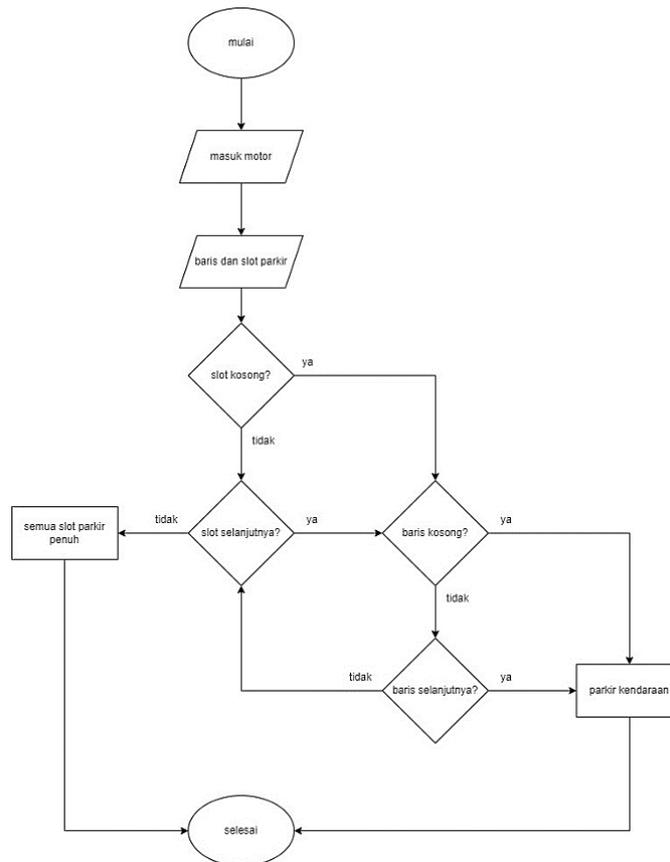
## **2. METODOLOGI PENELITIAN**

### **2.1. Best First Search (BFS)**

Prosedur pencarian pelebaran pertama adalah prosedur yang menjamin bahwa suatu solusi diperoleh jika solusi ada dengan jumlah cabang pohon yang terbatas. Jika adalah solusi, maka adalah jalur berhingga dari keadaan awal keadaan tujuan. Pencarian yang diperluas ini pertama-tama mencari semua jalur dengan panjang satu (yang panjangnya berhingga), kemudian mencari semua jalur dengan panjang dua (yang juga berhingga). Nilai pencarian heuristik digunakan oleh Algoritma Best First Search untuk membantu menemukan solusi dari permasalahan [16]. Proses ini berlanjut hingga semua lintasan teridentifikasi sehingga solusi dapat ditemukan. Prosedur ini memastikan tidak hanya menemukan solusi, tetapi juga solusi yang mengambil jalur terpendek menuju tujuan [17].

Untuk memvisualisasikan prosedur pencarian ini, sebuah flowchart digunakan untuk menggambarkan langkah-langkah sistematis yang diambil oleh algoritma BFS. Jika solusi ditemukan, flowchart pada gambar 1 menunjukkan bagaimana algoritma memastikan bahwa jalur terpendek dipilih.

Menggambarkan alur proses pencarian slot parkir yang dilakukan secara berurutan berdasarkan baris dan slot yang tersedia. Proses dimulai dengan memasukkan kendaraan ke area parkir, di mana sistem akan mengecek apakah terdapat slot kosong pada baris parkir saat ini. Jika slot kosong ditemukan, kendaraan akan langsung diparkir pada slot tersebut, dan proses selesai.



**Gambar 1.** Flowchart pencarian slot parkir

Namun, jika tidak ada slot kosong pada baris yang sedang diperiksa, sistem akan melanjutkan ke baris berikutnya untuk mengecek apakah ada slot kosong di sana. Proses ini terus berlanjut hingga semua baris dan slot diperiksa. Jika pada akhirnya semua slot parkir penuh, sistem akan menyatakan bahwa parkir penuh dan proses selesai. Flowchart ini sesuai dengan penerapan algoritma Best First Search (BFS) dalam pencarian slot parkir, yang dilakukan dengan mengecek secara berurutan baris demi baris hingga ditemukan slot yang sesuai atau hingga semua opsi diperiksa.

## 2.2. Sistem Informasi Geografis

Sistem Informasi Geografis, sering disebut juga Sistem Informasi Geografis (SIF), adalah teknologi pemetaan berupa sistem informasi komputer yang dirancang untuk bekerja dengan data yang berisi informasi spasial atau referensi spasial [18]. Teknologi pemetaan banyak digunakan untuk mempermudah pekerjaan masyarakat [19]. SIG sebagai sistem yang dapat mengakomodasi data spasial dengan data atribut dalam sebuah tabel yang mampu memberikan analisis spasial, dapat digunakan untuk memberikan informasi dalam perencanaan untuk diambil keputusan [20]. Sistem informasi geografis dapat mengambil manfaat dari penggunaan Best First Search sebagai sarana untuk mendukung proses pengambilan keputusan.

## 2.3. Struktur Tree

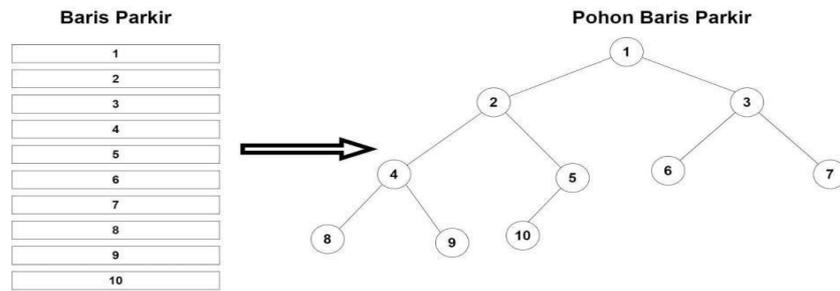
Struktur tree adalah struktur data yang mengandung aspek hierarki yang dibentuk melalui pengelompokan elemen atau node dalam tingkatan tertentu. Struktur ini terdiri dari satu node utama yang disebut root, dan node lainnya yang disebut child [21]. Setiap node, kecuali root, memiliki tepat satu parent dan dapat memiliki nol atau lebih child. Node yang tidak memiliki child disebut leaf atau daun. Struktur tree digunakan untuk merepresentasikan hubungan hierarki antara elemen-elemen, seperti dalam organisasi, silsilah keluarga, atau struktur folder dalam sistem file komputer. Selain itu, struktur tree juga sering digunakan dalam algoritma pencarian dan pengurutan, serta untuk mengelola data dalam database dan sistem manajemen informasi [22].

## 3. HASIL DAN PEMBAHASAN

### 3.1. Pengukuran

Pada studi kasus penentuan slot parkir di Kampus X, terdapat 10 titik objek lokasi blok parkir. Algoritma Best First Search (BFS) digunakan untuk menentukan slot parkir yang tersedia secara efisien. Dalam Gambar

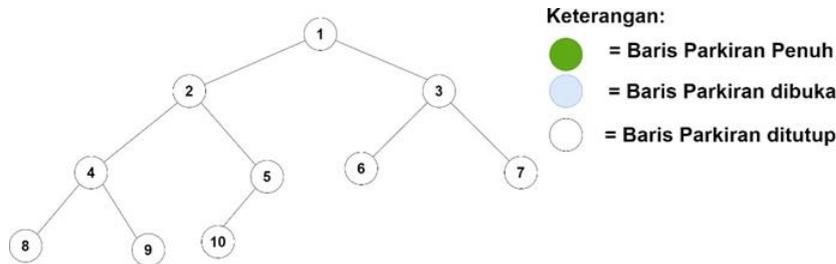
1 baris balok akan dikonversi menjadi pohon baris parkir di sampingnya. Struktur pohon ini terdiri dari 10 slot parkir yang diwakili oleh node angka dari angka 1 sampai dengan 10.



**Gambar 2.** Konversi Pemodelan Pohon Baris Parkir

Dalam pemodelan ini, struktur pohon biner dengan 10 slot parkir diwakili oleh node dengan angka 1 hingga 10. Warna hijau menandakan baris parkir penuh, biru muda menunjukkan baris parkir sedang dibuka atau dipertimbangkan, dan putih menandakan baris parkir yang ditutup atau belum dipertimbangkan. Proses dimulai dari node akar (node 1) yang diperiksa terlebih dahulu dan kemudian berubah menjadi hijau ketika penuh. Pergerakan berlanjut ke anak node 2 dan 3, mengikuti alur BFS yang mengeksplorasi setiap tetangga sebelum melanjutkan ke level berikutnya.

Algoritma BFS diterapkan untuk memastikan setiap slot parkir diperiksa secara sistematis, dimulai dari node akar dan melanjutkan ke anak node. Setiap node diperiksa untuk menentukan status ketersediaan slot parkir, memastikan setiap slot diperiksa dengan efisien. Proses ini membantu mengelola slot parkir dengan lebih efektif dan menunjukkan kecepatan serta efisiensi dalam menemukan slot parkir kosong dibandingkan dengan metode pencarian lainnya. Dari hasil penerapan algoritma ini, dapat diidentifikasi bagaimana setiap keputusan diambil oleh BFS, membantu dalam pengambilan keputusan yang lebih baik untuk manajemen parkir di Kampus X.



**Gambar 3.** Pemodelan Pohon Binary

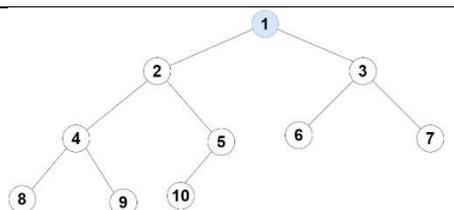
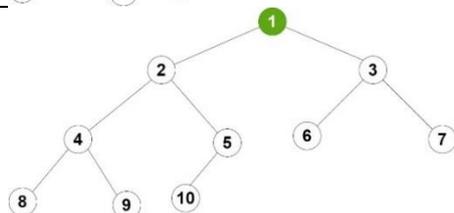
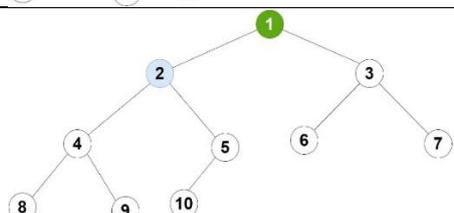
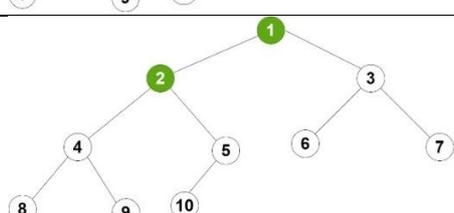
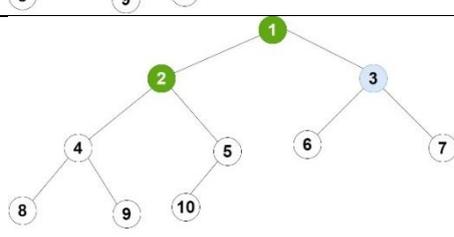
**Tabel 1.** Pemodelan dari gambar Pemodelan Pohon Binary

Node	Keterangan
1	Baris Pertama
2	Baris Kedua
3	Baris Ketiga
4	Baris Keempat
5	Baris Kelima
6	Baris Keenam
7	Baris Ketujuh
8	Baris Kedelapan
9	Baris Kesembilan
10	Baris Kesepuluh

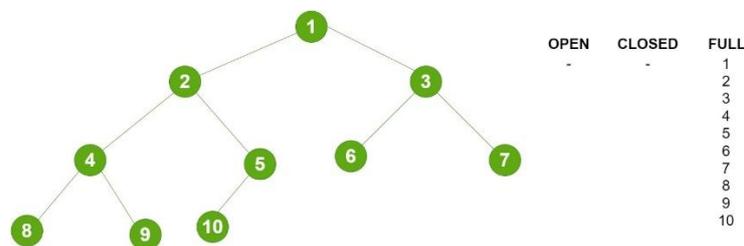
**3.2. Hasil Rancangan dan Implementasi**

Algoritma BFS adalah salah satu metode pencarian yang digunakan untuk menelusuri graf atau pohon secara sistematis, lapis demi lapis [23]. Dalam konteks penentuan slot parkir, algoritma ini sangat berguna untuk menemukan slot parkir yang terdekat dari pintu masuk atau lokasi awal kendaraan, dengan cara mengeksplorasi semua kemungkinan slot parkir di satu level sebelum berpindah ke level berikutnya. Proses Best First Search dari node 1 hingga 3 ditunjukkan pada tabel 2.

**Tabel 2.** Proses Best First Search dari node 1 hingga 3

Langkah Pertama		<table border="1"> <thead> <tr> <th>OPEN</th> <th>CLOSED</th> <th>FULL</th> </tr> </thead> <tbody> <tr><td>1</td><td>2</td><td>-</td></tr> <tr><td></td><td>3</td><td></td></tr> <tr><td></td><td>4</td><td></td></tr> <tr><td></td><td>5</td><td></td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td>7</td><td></td></tr> <tr><td></td><td>8</td><td></td></tr> <tr><td></td><td>9</td><td></td></tr> <tr><td></td><td>10</td><td></td></tr> </tbody> </table>	OPEN	CLOSED	FULL	1	2	-		3			4			5			6			7			8			9			10		<p>Proses dimulai dari node 1 yang merupakan baris parkir Terdekat dengan pintu masuk. Pada Tahap ini, semua node lain dianggap Belum dikunjungi (Baris lain di tutup)</p>
OPEN	CLOSED	FULL																															
1	2	-																															
	3																																
	4																																
	5																																
	6																																
	7																																
	8																																
	9																																
	10																																
Langkah Kedua		<table border="1"> <thead> <tr> <th>OPEN</th> <th>CLOSED</th> <th>FULL</th> </tr> </thead> <tbody> <tr><td>-</td><td>2</td><td>1</td></tr> <tr><td></td><td>3</td><td></td></tr> <tr><td></td><td>4</td><td></td></tr> <tr><td></td><td>5</td><td></td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td>7</td><td></td></tr> <tr><td></td><td>8</td><td></td></tr> <tr><td></td><td>9</td><td></td></tr> <tr><td></td><td>10</td><td></td></tr> </tbody> </table>	OPEN	CLOSED	FULL	-	2	1		3			4			5			6			7			8			9			10		<p>Setelah node 1 terisi penuh, node ini berubah warna menjadi hijau yang menandakan bahwa slot di baris pertama sudah penuh.</p>
OPEN	CLOSED	FULL																															
-	2	1																															
	3																																
	4																																
	5																																
	6																																
	7																																
	8																																
	9																																
	10																																
Langkah Ketiga		<table border="1"> <thead> <tr> <th>OPEN</th> <th>CLOSED</th> <th>FULL</th> </tr> </thead> <tbody> <tr><td>2</td><td>3</td><td>1</td></tr> <tr><td></td><td>4</td><td></td></tr> <tr><td></td><td>5</td><td></td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td>7</td><td></td></tr> <tr><td></td><td>8</td><td></td></tr> <tr><td></td><td>9</td><td></td></tr> <tr><td></td><td>10</td><td></td></tr> </tbody> </table>	OPEN	CLOSED	FULL	2	3	1		4			5			6			7			8			9			10		<p>Setelah node 1 terisi penuh atau Baris pertama penuh, sistem membuka baris parkir pada level berikutnya (node 2), yang ditandai dengan node berwarna biru, node 2 dipilih untuk diisi, karena merupakan node berikutnya dalam urutan BFS</p>			
OPEN	CLOSED	FULL																															
2	3	1																															
	4																																
	5																																
	6																																
	7																																
	8																																
	9																																
	10																																
Langkah Keempat		<table border="1"> <thead> <tr> <th>OPEN</th> <th>CLOSED</th> <th>FULL</th> </tr> </thead> <tbody> <tr><td>-</td><td>3</td><td>1</td></tr> <tr><td></td><td>4</td><td>2</td></tr> <tr><td></td><td>5</td><td></td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td>7</td><td></td></tr> <tr><td></td><td>8</td><td></td></tr> <tr><td></td><td>9</td><td></td></tr> <tr><td></td><td>10</td><td></td></tr> </tbody> </table>	OPEN	CLOSED	FULL	-	3	1		4	2		5			6			7			8			9			10		<p>Node 2 kemudian berubah warna menjadi hijau, menunjukkan bahwa baris parkir ini telah terisi penuh.</p>			
OPEN	CLOSED	FULL																															
-	3	1																															
	4	2																															
	5																																
	6																																
	7																																
	8																																
	9																																
	10																																
Langkah Kelima		<table border="1"> <thead> <tr> <th>OPEN</th> <th>CLOSED</th> <th>FULL</th> </tr> </thead> <tbody> <tr><td>3</td><td>4</td><td>1</td></tr> <tr><td></td><td>5</td><td>2</td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td>7</td><td></td></tr> <tr><td></td><td>8</td><td></td></tr> <tr><td></td><td>9</td><td></td></tr> <tr><td></td><td>10</td><td></td></tr> </tbody> </table>	OPEN	CLOSED	FULL	3	4	1		5	2		6			7			8			9			10		<p>Setelah node 2 penuh, proses berlanjut ke node 3. Node 3 dipilih dan diisi. Node ini akan berubah warna menjadi hijau ketika sudah terisi penuh dan begitu pula node node berikutnya sehingga terisi secara bertahap</p>						
OPEN	CLOSED	FULL																															
3	4	1																															
	5	2																															
	6																																
	7																																
	8																																
	9																																
	10																																

Pada hasil akhir dari penerapan Algoritma BFS yang ditunjukkan dalam gambar, seluruh node pada pohon graf telah terisi. Setiap node dalam graf ini mewakili slot parkir, dan proses pengisian slot parkir telah dilakukan secara bertahap sesuai dengan level dalam graf.



**Gambar 4.** Hasil Akhir BFS jika semua baris parkir terisi

Hasil ini menunjukkan bahwa kendaraan-kendaraan yang ingin parkir diarahkan untuk mengisi slot parkir yang terdekat terlebih dahulu sebelum mengisi slot yang berikutnya. Proses BFS memastikan bahwa seluruh slot pada level yang lebih rendah terisi penuh sebelum membuka akses ke slot pada level yang lebih tinggi. Ini menghasilkan distribusi slot parkir yang optimal, di mana kendaraan dapat diparkir di tempat yang paling efisien sesuai dengan jarak terdekat dari pintu masuk.

Pseudo code yang disajikan menggambarkan proses sistematis untuk menentukan slot parkir yang tersedia menggunakan metode antrian yang diadaptasi dari algoritma BFS. Berikut diberikan prosedur dari algoritma BFS.

## Pseudocode Pencarian Slot Parkir Menggunakan Pendekatan Best First Search (BFS)

```

Mulai
  Inisialisasi antrian = semua_baris_parkir
  Inisialisasi slot_ditemukan = FALSE

  WHILE (kendaraan_masuk) DO
    WHILE (antrian tidak kosong DAN slot_ditemukan adalah FALSE) DO
      baris_saat_ini = dequeue(antrian)

      FOR setiap slot_parkir dalam baris_saat_ini DO
        IF (slot_parkir kosong) THEN
          Parkirkan kendaraan pada slot_parkir
          slot_ditemukan = TRUE
          BREAK
        END IF
      END FOR

      IF (slot_ditemukan adalah FALSE) THEN
        IF (baris_saat_ini penuh) THEN
          CONTINUE
        ELSE
          enqueue(antrian, baris_saat_ini)
        END IF
      END IF
    END WHILE

    IF (slot_ditemukan adalah FALSE) THEN
      Tampilkan "Parkir penuh"
      BREAK
    END IF
  END WHILE

Selesai

```

Proses dimulai dengan inisialisasi antrian yang berisi semua baris parkir dan variabel slot ditemukan untuk melacak apakah slot parkir kosong telah ditemukan. Selanjutnya, dalam loop utama, sistem akan terus mencari slot parkir selama ada kendaraan yang masuk dan belum ditemukan slot kosong. Pada setiap iterasi, sistem akan mengambil baris parkir dari antrian dan memeriksa setiap slot parkir dalam baris tersebut. Jika ditemukan slot kosong, kendaraan akan diparkir, dan loop berhenti.

Jika pada iterasi tersebut tidak ditemukan slot kosong, dan baris parkir penuh, sistem akan melanjutkan ke baris berikutnya. Namun, jika baris tidak penuh, baris tersebut akan dimasukkan kembali ke antrian untuk diperiksa nanti. Apabila seluruh baris telah diperiksa dan tidak ada slot kosong yang ditemukan, sistem akan menampilkan pesan "Parkir penuh" dan menghentikan proses pencarian. Pseudo code ini menggambarkan alur logis yang efisien dalam menentukan slot parkir yang tersedia, dan sesuai dengan alur flowchart yang telah dihasilkan sebelumnya. Simulasi gambar slot parkir ditunjukkan pada gambar 5 dan 6.



**Gambar 5.** Tampilan utama aplikasi penentuan slot parkir



**Gambar 6.** Tampilan dari salah satu baris parkir

Berdasarkan Gambar 5 dan 6, menunjukkan bahwa dari gambar pengaplikasian UI saling terhubung. Yang mana Gambar 5 adalah tampilan utama aplikasi penentuan slot parkir yang sudah tersedia didalam aplikasi. Sedangkan Gambar 6 adalah hasil dari salah satu slot parkir yang di klik dalam aplikasi tersebut. Dengan menampilkan hasil dari total slot parkir sebanyak 50 slot, terisi 36 slot, dan yang tersedia 14 slot.

Proses ini mengilustrasikan bagaimana algoritma BFS dapat diterapkan untuk mengelola slot parkir dengan lebih efisien. Dengan memulai dari node akar dan mengevaluasi setiap anak node, algoritma ini dapat mengidentifikasi slot parkir yang penuh dan yang masih tersedia secara sistematis. Hasil dari penerapan algoritma ini menunjukkan efisiensi dalam menemukan dan mengelola slot parkir, membantu memastikan penggunaan ruang parkir yang optimal di Kampus X.

#### 4. KESIMPULAN

Penelitian ini mengembangkan sistem penentuan slot parkir otomatis menggunakan algoritma Best First Search (BFS) di area parkir motor Kampus X. Algoritma BFS dipilih karena kemampuannya dalam melakukan pencarian rute optimal berdasarkan fungsi heuristik. Hasil implementasi menunjukkan bahwa sistem dapat secara efisien menentukan slot parkir yang tersedia, memulai dari node akar dan mengevaluasi setiap anak node secara sistematis. Hal ini meningkatkan efisiensi dalam menemukan slot parkir kosong dan mengelola penempatan kendaraan. Secara keseluruhan, sistem ini membantu pengguna dan petugas parkir dalam mengatur area parkir dengan lebih efektif, mengurangi kemacetan, dan meningkatkan keteraturan penempatan kendaraan. Oleh karenanya, peneliti menyarankan untuk mengintegrasikan sistem ini dengan teknologi sensor atau kamera untuk lebih meningkatkan akurasi dan efisiensi dalam penentuan slot parkir untuk penelitian selanjutnya.

#### REFERENSI

- [1] Sudipta Giri, I. K., Wirasutama, C. P., Darma Yoga, I. W. G., & Gede Mahardika, I. B. (2023). Perencanaan Desain Parkir Pada Kawasan Objek Wisata Pantai Kedungu Tabanan. *Jurnal Ilmiah Kurva Teknik*, 12(2), 190–199. <https://doi.org/10.36733/jikt.v12i2.7903>
- [2] A. Triawan dan R. S. Sofyan, “Penerapan Algoritma Booyer Moore untuk efisiensi pencarian data letak kendaraan pada aplikasi Gudang Dealer SETIAJAYA TOYOTA,” *Jurnal Ilmiah Teknologi-Informasi dan Sains (TeknoIS)*, vol. 7, hlm. 27–36, 2017.
- [3] Fadhillah, M. Y. (2023). Penerapan Metode Breadth First Search (Bfs) Pada Rekomendasi Jurusan Berdasarkan Minat Berbasis Sistem Informasi Cerdas. *Jurnal Ekonomi Volume 18*, Nomor 1 Maret201, 2(1), 41–49
- [4] Reza pahlevi, U. (2023). Penerapan Algoritma Greedy Best-First Search untuk Memecahkan Permainan Tebak Kata Katla.
- [5] N. G. Ginasta dan S. Supriady, “Implementasi Pencarian Rute Terbaik untuk Mengetahui Lokasi Tempat Parkir pada Sistem E-Parking Menggunakan Algoritma Dijkstra dan Best First Search,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, hlm. 607–613, Mar 2024, doi: 10.57152/malcom.v4i2.1261.
- [6] M. Ichwan, Y. I. N, and A. Pahlevi, “Implementasi Metode Best First Search untuk Pembelajaran Matematika Berbasis Multimedia Interaktif,” *MIND J.*, vol. 1, no. 1, pp. 38–50, 2018, doi: 10.26760/mindjournal.v1i2.38-50.
- [7] Herfandi, H., Soleha, U., Susilo Yuda Irawan, A., Ahmad Baihaqi, K., & Maulana, R. (2022). Implementasi Algoritma Best First Search untuk Pencarian Rute Terpendek pada Aplikasi Cerdas Pendaftaran Santri Baru. *Syntax : Jurnal Informatika*, 11(01). <https://doi.org/10.35706/syji.v11i01.6398>
- [8] Sujaini, H., Perwitasari, A., & Januardi, T. (2023). Sistem Pembelajaran Algoritma Best First Search, Breadth First Search & Depth First Search. *Jurnal Teknik Indonesia*, 2(2). <https://doi.org/10.58860/jti.v2i2.15>
- [9] Herfandi, H., Soleha, U., Susilo Yuda Irawan, A., Ahmad Baihaqi, K., & Maulana, R. (2022). Implementasi Algoritma Best First Search untuk Pencarian Rute Terpendek pada Aplikasi Cerdas Pendaftaran Santri Baru. *Syntax : Jurnal Informatika*, 11(01). <https://doi.org/10.35706/syji.v11i01.6398>
- [10] Angriani, H., & Saharaeni, Y. (2020). Implementasi Algoritma Best First Search Dalam Sistem Pakar Pertolongan Pertama Pada Bayi dan Anak. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, 10(2). <https://doi.org/10.35585/inspir.v10i2.2575>
- [11] Bertarina, & Arianto, W. (2021). Analisis Kebutuhan Ruang Parkir (Studi Kasus pada Area Parkir ICT Universitas Teknokrat Indonesia). *Jurnal SENDI*, 02(02).
- [12] Y. Afero, “Algoritma Best First Search menentukan lintasan jalur terpendek pada Kota Wisata Bukittinggi,” *JOISIE Journal Of Information System And Informatics Engineering*, vol. 5, no. Desember, hlm. 138–145, 2021.
- [13] Muhardono, A. (2023). Penerapan Algoritma Breadth First Search dan Depth First Search pada Game Angka. *Jurnal Minfo Polgan*, 12(1). <https://doi.org/10.33395/jmp.v12i1.12340>
- [14] M. Mardiana, D. Despa, M. Ardhi Muhammad, T. Septiana, and T. A. Lorenza, “Sistem Navigasi Augmented Reality Dengan Pencarian Jalur Terbaik Menuju Lokasi Pustaka (Studi Kasus Pada Upt Perpustakaan Unila),” *Jurnal Profesi Insinyur Universitas Lampung*, vol. 3, no. 2, pp. 36–42, Nov. 2022, doi: 10.23960/jpi.v3n2.78.
- [15] R. Alderio and R. P. Sari, “Perbandingan Pendekatan Breadth First Search, Depth First Search dan Best

- First Search Pada Pencarian Rute Menggunakan Metode Branch and Bound,” *Jurnal Serambi Engineering*, vol. 8, pp. 6707–6714, 2023, Accessed: Aug. 23, 2024. [Online]. Available: <https://doi.org/10.32672/jse.v8i3.6525>
- [16] Bunaen, M. C., Pratiwi, H., & Riti, Y. F. (2022). Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terpendek Dari Pusat Kota Surabaya Ke Tempat Bersejarah. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 4(1).
- [17] H. D. Hutahaean, “Penerapan Metode Best First Search Pada Permainan Tic Tac Toe,” *Journal of Computer Networks*, vol. 1, no. 1, hlm. 10–15, 2019.
- [18] M. Tinambunan dan S. Sintaro, “Aplikasi Restfull Pada Sistem Informasi Geografi Pariwisata Kota Bandar Lampung,” *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 2, no. 3, hlm. 312–323, 2021, [Daring]. Tersedia pada: <http://jim.teknokrat.ac.id/index.php/informatika>
- [19] Manek, M. P. M. (2018). Visualisasi Rencana Detail Tata Ruang Kota Berbasis Mobile GIS (Studi Kasus: Kecamatan Lowokwaru Kota Malang). ITN malang
- [20] Kurniawati, U. F., Handayani, K. E., Nurlaela, S., Idajati, H., Firmansyah, F., Pratomoadmojo, N. A., & Septriadi, R. S. (2020). Pengolahan Data Berbasis Sistem Informasi Geografis (SIG) di Kecamatan Sukolilo. *Sewagati*, 4(3), 190. <https://doi.org/10.12962/j26139960.v4i3.8048>
- [21] H. Septian, I. Suhartini, I. Petrio, L. A. Jihad, N. Lia, dan T. Fadilah A, “Implementasi Struktur Data Tree Pada Game Pacman Dengan C,” *Jurnal Ilmiah Digital of Information Technology*, vol. 11, no. 2, hlm. 120– 129, 2021.
- [22] Wicaksono, A. W., & Setiadi, T. (2023). Penerapan Klasifikasi Decision Tree (C4.5) untuk Memprediksi Kelulusan Siswa Sekolah Dasar di Kecamatan Juai. *Format : Jurnal Ilmiah Teknik Informatika*, 12(2), 151. <https://doi.org/10.22441/format.2023.v12.i2.008>
- [23] M. Zahid Hassan and B. Corresponding Author, “Advanced Recursive Best-First Search (RBFS) based Routing Protocol for Multi-hop and Multi-Channel Cognitive Wireless Mesh Networks,” 2024, doi: 10.32996/jcsts.