

# Implementasi Algoritma Best First Search Untuk Melakukan Penyelesaian Game Sudoku

Dwi Resty Ananda<sup>1</sup>, Tommy<sup>2</sup>, Calvin Chiuloto<sup>3</sup>

<sup>1</sup>Universitas Sumatera Utara

<sup>2,3</sup>Program Studi Teknik Informatika Universitas Harapan Medan Indonesia

<sup>1</sup>dwirestyananda00@gmail.com, <sup>2</sup>tomshirakawa@gmail.com, <sup>3</sup>kalvin.chiuloto@yahoo.com

## Abstract

*The problem in this research is that the Sudoku puzzle game is difficult to solve because it is included in the NP-complete problem, so it cannot be solved at the same time. Until now, many programmers are looking for the right algorithm to solve this puzzle. One way to solve this game is to use the best first search algorithm. This algorithm is an improvement of the Brute Force algorithm, where solutions can be found with fewer searches and can find solutions to problems more efficiently because there is no need to check all possible solutions. Only searches that lead to solutions need to be considered. The results of this study are the application of the best first search method in the sudoku game and determine the performance of the best first search method in the sudoku game. This research was built using Unity 3D software with the C# programming language and then using the unified modeling language as the flow of the diagram in the Sudoku game.*

**Keywords:** Sudoku, algorithm, BFS

## 1. PENDAHULUAN

Perkembangan komputer memasuki beberapa tahap. Beberapa bidang ilmu berkembang dan komputer lahir [1][2][3]. Salah satunya adalah kecerdasan buatan. Ilmu komputer adalah bidang ilmu komputer yang membahas cara memprogram komputer menjadi mesin untuk melakukan apa yang ingin diprogram melalui instruksi[4][5]. Salah satu yang menarik dari bidang kecerdasan buatan ini adalah komputer dapat tampil cerdas[6][7]. Ungkapan kecerdasan buatan adalah permainan-permainan[8]. Kecerdasan buatan memungkinkan Anda untuk menyelesaikan permainan tertentu dengan bantuan komputer. Game yang bisa diklasifikasikan sebagai game kecerdasan buatan adalah Sudoku. Sudoku adalah permainan puzzle berbasis logika dengan angka. Secara umum, teka-teki Sudoku terdiri dari 81 kotak yang disusun dalam 9 baris, 9 kolom, dan 9 subbagian. Tujuan utama dari permainan ini adalah untuk mengisi semua kotak dengan angka dari 1 sampai 9 [9][10]. Permainan Sudoku membutuhkan pemikiran logis dari para pemainnya sesuai dengan aturan permainan. Permainan Sudoku tidak hanya populer tetapi juga salah satu masalah tersulit dalam ilmu komputer.

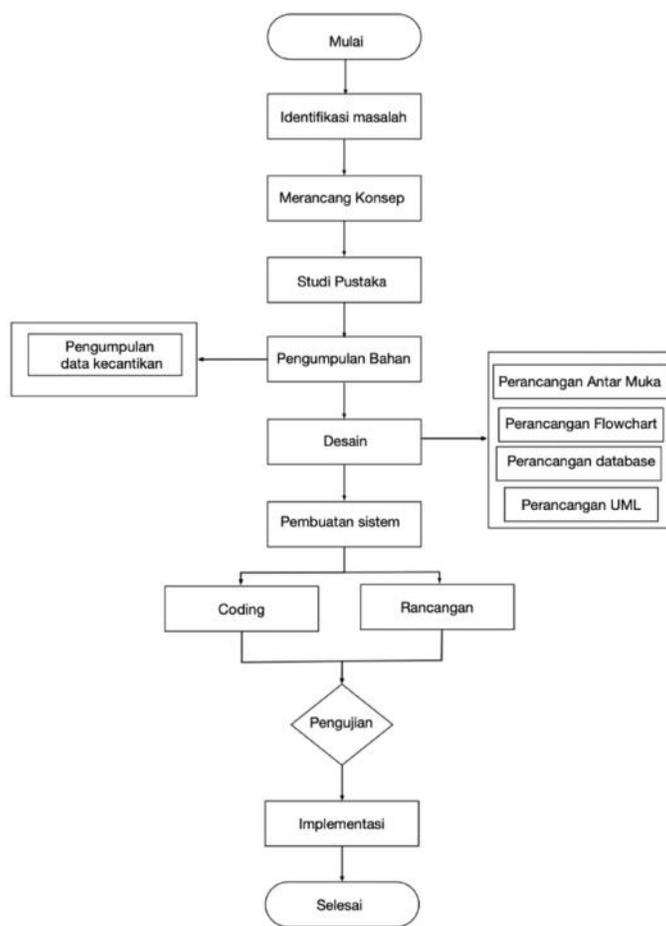
Permasalahan puzzle Sudoku ini sulit untuk dipecahkan karena termasuk dalam permasalahan NP-complete, sehingga tidak bisa diselesaikan dalam waktu yang sama. Hingga saat ini banyak programmer yang mencari algoritma yang tepat untuk menyelesaikan puzzle ini. Salah satu cara untuk menyelesaikan game ini yaitu dengan menggunakan algoritma best first search. Algoritma ini merupakan perbaikan dari algoritma Brute Force, dimana solusi dapat ditemukan dengan penelusuran yang lebih sedikit dan dapat mencari solusi permasalahan secara lebih mangkus karena tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang perlu dipertimbangkan. Penelitian lainnya yang dilakukan oleh [11] yang berjudul Penyelesaian Puzzle Sudoku Menggunakan Algoritma Brute Force dan Backtracking menyimpulkan bahwa Algoritma Brute force dan Backtracking dapat digunakan untuk menyelesaikan puzzle sudoku dengan baik. Waktu yang diperlukan untuk memecahkan puzzle tidak dipengaruhi oleh tingkat kesulitan dari puzzle itu sendiri. Waktu untuk memecahkan puzzle lebih dipengaruhi oleh spesifikasi komputer yang digunakan terutama prosessor. Semakin tinggi clock prosessor, semakin cepat waktu yang diperlukan untuk menyelesaikan puzzle, adapun penelitian lainnya yang dilakukan oleh [12] yang berjudul Aplikasi Pemecahan Soal Sudoku dengan Metode Backtracking menyimpulkan bahwa Aplikasi

sudoku dapat memecahkan soal sudoku pada tingkat yang sulit dalam hal ini berdasarkan website sumber soal yang diambil yaitu pada tingkat extreme yang juga merupakan tingkat tersulit pada website tersebut. Adapun akurasi dari pemecahan soal yang diuji adalah 100%. Berdasarkan latar belakang yang sudah dipaparkan diatas maka mempunyai ide untuk membahas langkah penyelesaian game sudoku, Adapun judul dari penelitian ini ialah Implementasi algoritma Best First Search untuk melakukan penyelesaian game sudoku

## 2. METODE PENELITIAN

### 2.1 Kerangka Penelitian

Pada rancangan penelitian sistem terdapat rancangan penelitian yang berfungsi sebagai tahapan-tahapan dalam membangun sistem game sudoku berbasis android. Berikut ini rancangan penelitian yang peneliti buat, seperti terlihat pada Gambar 1, berikut.



Gambar 1. Kerangka Penelitian

Pada Gambar 1, menjelaskan bahwa dalam kerangka kerja penelitian terdapat tahapan seperti berikut ini:

1. Identifikasi masalah yang merupakan tahapan dalam menganalisa masalah apa yang terdapat pada penelitian ini yaitu tidak pernah dilakukan implementasi algoritma *best first search*
2. Merancang konsep merupakan tahapan dalam game sudoku dengan merancang konsep game yang dihasilkan dapat berjalan efektif.

3. Studi pustaka merupakan langkah awal dalam metode pengumpulan data. Studi pustaka merupakan metode pengumpulan data yang diarahkan kepada pencarian data dan informasi melalui dokumen-dokumen, baik dokumen tertulis, foto-foto, gambar, maupun dokumen elektronik yang dapat mendukung dalam proses penulisan.
4. Pengumpulan bahan merupakan tahapan dalam mengumpulkan data-data game
5. Desain merupakan tahapan dalam merancang antar muka, *flowchart* dan alur diagram UML dari sistem
6. Pembuatan sistem merupakan tahapan dalam membuat game sudoku dengan algoritma best first search

## 2.2 Analisis Algoritma

Analisis Algoritma Seperti yang sudah dipaparkan dalam bab-bab sebelumnya, algoritma yang akan digunakan dalam membangun aplikasi ini adalah algoritma BFS. Penerapan algoritma BFS dalam menentukan elemen awal permainan Sudoku adalah sebagai berikut:

1. Algoritma dimulai dari sel kosong di baris pertama ( $x=0$ ) dan kolom pertama ( $y=0$ ) pada matriks Sudoku berukuran  $9 \times 9$ .
2. Pilih nilai num secara acak, dengan num  $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .
3. Jika nilai num yang dipilih memenuhi aturan sudoku (valid), maka isilah sel kosong tersebut dengan nilai num, dan lanjutkan pemeriksaan ke sel kosong berikutnya.
4. Jika nilai num yang dipilih tidak memenuhi aturan sudoku, uji kembali dengan nilai num acak lainnya yang belum terpilih.
5. Jika seluruh nilai num telah teruji, dan tidak ada nilai num yang memenuhi aturan sudoku, maka lakukan *backtrack* ke sel sebelumnya.
6. Sel ini akan diuji kembali dengan nilai num baru berdasarkan aturan sudoku.
7. Lakukan cara yang sama dengan poin nomor 3.
8. Proses di atas akan dilakukan terus menerus secara rekursif hingga seluruh sel kosong pada matriks berisi nilai-nilai num yang valid.
9. Pilih posisi sel secara acak, kemudian hapus nilai num pada sel tersebut.
10. Lakukan poin nomor 9 secara terus menerus, hingga diperoleh matriks Sudoku berukuran  $9 \times 9$  dengan beberapa nilai num ditampilkan. Banyaknya nilai num yang ditampilkan tergantung dari level permainan yang dipilih pemain, semakin sedikit jumlah elemen awal yang ditampilkan, maka tingkat kerumitan soal (level) permainan Sudoku akan semakin sulit. Nilai-nilai num yang ditampilkan ini merupakan elemen-elemen awal permainan Sudoku yang

## 2.3 Contoh Kasus

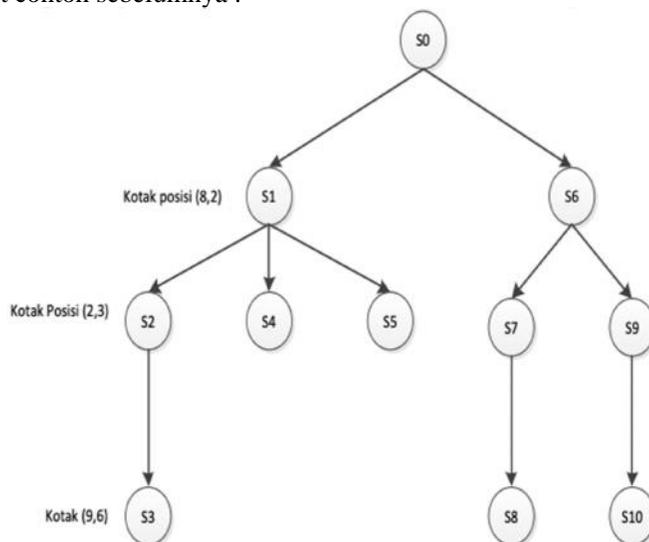
Pertama-tama peneliti akan memberikan indeks pada setiap kotak pada sudoku agar mudah untuk menunjuk posisi kotak tertentu. Posisi atau indeks kotak akan dinotasikan dengan (X,Y). Dengan X menyatakan posisi suatu kotak secara horizontal dan Y menyatakan posisi kotak secara vertikal. Pengindeksan kotak lengkap dapat dilihat pada Gambar 2 di bawah ini.

|     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1,1 | 2,1 | 3,1 | 4,1 | 5,1 | 6,1 | 7,1 | 8,1 | 9,1 |
| 1,2 | 2,2 | 3,2 | 4,2 | 5,2 | 6,2 | 7,2 | 8,2 | 9,2 |
| 1,3 | 2,3 | 3,3 | 4,3 | 5,3 | 6,3 | 7,3 | 8,3 | 9,3 |
| 1,4 | 2,4 | 3,4 | 4,4 | 5,4 | 6,4 | 7,4 | 8,4 | 9,4 |
| 1,5 | 2,5 | 3,5 | 4,5 | 5,5 | 6,5 | 7,5 | 8,5 | 9,5 |
| 1,6 | 2,6 | 3,6 | 4,6 | 5,6 | 6,6 | 7,6 | 8,6 | 9,6 |
| 1,7 | 2,7 | 3,7 | 4,7 | 5,7 | 6,7 | 7,7 | 8,7 | 9,7 |
| 1,8 | 2,8 | 3,8 | 4,8 | 5,8 | 6,8 | 7,8 | 8,8 | 9,8 |
| 1,9 | 2,9 | 3,9 | 4,9 | 5,9 | 6,9 | 7,9 | 8,9 | 9,9 |

Gambar 2. Indeks Sudoku

Angka yang ada di dalam matriks merupakan indeks posisi kotak, indeks dimulai dari ujung kiri-atas

yaitu (1,1) sampai ujung kanan-bawah yaitu (9,9). Hal ini akan bermanfaat untuk proses implementasi selanjutnya. Peneliti akan mendefinisikan isi simpul dan kedalaman pohon terkait dengan proses pencarian solusi. Simpul akan didefinisikan sebagai nilai yang pada suatu kotak diposisi tertentu. Ketinggian pohon akan didefinisikan sebagai posisi kotak kosong yang akan diisi. Jadi misalkan *problem sheet* sudoku terdiri dari 30 kotak kosong yang harus diisi dari 81 total kotak sudoku, maka pohon pencarian harus mencapai kedalaman 30 untuk mencapai solusi. Urutan posisi kotak yang akan dicari nilai simpulnya adalah dari paling kiri-atas ke kanan bawah. Misalkan terdapat 3 kotak kosong yaitu (2,3),(8,2), dan (9,6), maka pada kedalaman 1 akan berisi simpul kotak posisi (8,2) , kedalaman 2 akan berisi simpul kotak posisi (2,3) , dan yang terakhir kedalaman 3 akan berisi simpul kotak posisi (9,6). Untuk mempermudah pemahaman mengenai definisi ini, pembaca dapat melihat penjelasan pada Gambar 3, terkait contoh sebelumnya :



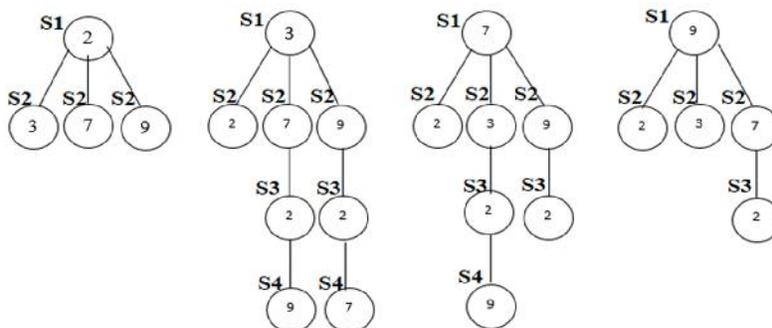
Gambar 3. Contoh Kasus Penelusuran 3 Kotak

Pada Gambar 3 di atas, kita dapat melihat contoh model penelusuran yang dilakukan pada contoh sebelumnya. Selanjutnya kita akan mendefinisikan fungsi kelayakan yaitu suatu nilai pada simpul dinyatakan layak jika pada saat kondisi dimana tidak ada nilai yang sama pada baris, kolom, dan daerahnya dengan nilai tersebut. Fungsi Solusi didefinisikan jika semua kotak kosong telah terisi dengan nilai yang sesuai (sesuai dengan fungsi kelayakan). Peneliti akan memberikan contoh kasus sudoku dan menyelesaikan dengan menggunakan pohon status BFS. Misalkan diberikan kasus sudoku seperti gambar di bawah ini :

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
|   |   | 4 |   | 8 | 1 |   |   |   |
|   | 1 | 5 | 9 | 6 | 7 |   |   | 3 |
| 8 | 6 |   |   | 2 | 3 |   | 1 | 5 |
|   |   | 1 |   |   |   | 7 |   | 6 |
|   |   |   | 3 |   |   |   | 5 | 2 |
|   |   |   |   |   |   |   |   | 8 |
|   |   |   | 8 | 5 | 2 | 6 |   |   |
|   | 5 | 8 | 7 |   | 6 |   |   |   |
|   |   | 6 |   |   |   | 5 |   |   |

Gambar 4. Kasus Sudoku

Langkah pertama adalah kita akan melakukan pencarian solusi pada kotak paling kiri-atas kotak S1 dan seterusnya seperti yg terlihat di Gambar 4.



Gambar 5. Penyelesaian Sudoku dengan Pohon Solusi BFS

Dari keterangan Gambar 5, di atas maka angka yang akan terisi di S1 (2,3,7,9) S2 (2,3,7,9) S3 (2) S4 (7,9).

### 3. HASIL DAN PEMBAHASAN

Implementasi dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman yang akan digunakan pada penelitian ini menggunakan bahasa pemrograman C#. Tujuan Implementasi adalah untuk mengkonfirmasi modul program perancangan dan melakukan implementasi *game* sudoku berbasis *android* dalam melakukan permainan sudoku yang mempunyai tingkat kesulitan sehingga perlu dikembangkan menggunakan *software* unity 3D.

#### 3.1 Tampilan Game pada Unity

Pada tampilan implementasi game sudoku berbasis *android* pada unity yang berfungsi sebagai tampilan yang digunakan oleh *user* untuk melakukan permainan *Game* sudoku berbasis *Android*.

Berikut ini adalah tampilan yang terdapat pada sistem *user*:

##### 3.1.1 Tampilan Utama game sudoku

Pada tampilan game sudoku berbasis *android* dibuat menggunakan *software* unity 3D. berikut ini tampilan menu utama game sudoku berbasis *android*:



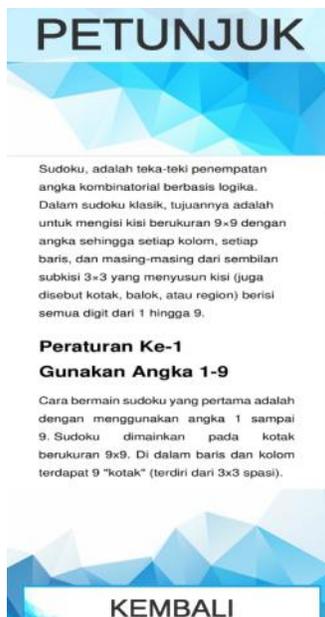
Gambar 6. Tampilan *Game* sudoku

Keterangan Gambar 6 dapat dijelaskan sebagai berikut:

1. Pada menu *start* merupakan menu yang akan menampilkan permainan sudoku berbasis *android*
2. Pada menu *petunjuk* merupakan menu yang akan menampilkan informasi dari petunjuk penggunaan *game* sudoku
3. Pada menu *pengembang* merupakan menu yang akan menampilkan informasi dari pengembang aplikasi

### 3.2 Tampilan menu petunjuk penggunaan game

Pada tampilan *menu* petunjuk penggunaan *game* akan menampilkan informasi dari penggunaan *game* dan tata cara bermain *game* sudoku. Seperti pada gambar berikut ini



Gambar 7. Tampilan Informasi Petunjuk

### 3.3 Tampilan menu pengembang

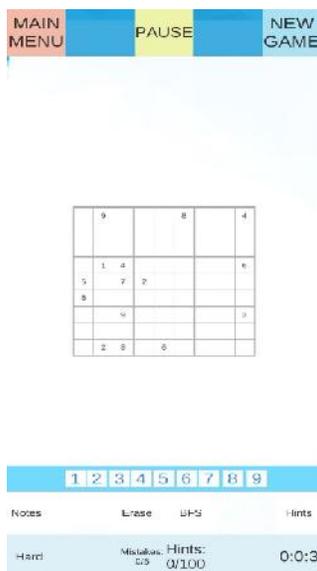
Pada tampilan menu pengembang akan menampilkan informasi dari pengembang aplikasi *game* sudoku berbasis *android*. Seperti pada gambar berikut ini:



Gambar 8. Tampilan Menu Pengembang

### 3.4 Tampilan game sudoku

Pada tampilan *game* sudoku akan menampilkan permainan dari *game* sudoku berbasis *android*. Seperti pada Gambar 9, berikut ini:

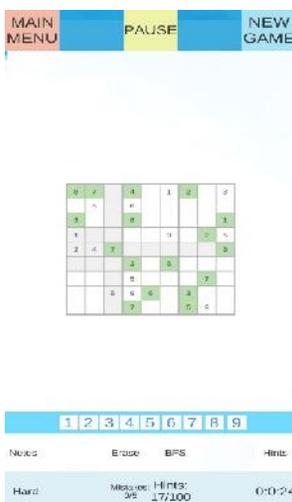


Gambar 9. *Game* sudoku

### 3.5 Pengujian Sistem

Pengujian dilakukan dengan cara melakukan pencarian solusi yang dapat dilakukan sebagai salah satu penerapan algoritma BFS, dalam aplikasi ini algoritma BFS diterapkan sebagai pencarian otomatis dalam menyelesaikan *game* dan mencari baris atau kolom yang paling cepat menyelesaikan masalah seperti pada Gambar 10, berikut ini:

1. Pengisian secara otomatis menggunakan algoritma BFS.



Gambar 10. Algoritma BFS Menyelesaikan Permainan Secara Otomatis

2. Mencari langkah tercepat dengan BFS

Pada permainan sudoko yang dikembangkan peneliti membuat tombol *hint* sebagai jalur terbaik dalam menyelesaikan permainan seperti pada Gambar 11, berikut ini:



Gambar 11. Algoritma BFS dalam mencari jalur terbaik

#### 4. KESIMPULAN

Dalam uraian rangkaian mulai dari proses pembuatan Game Sudoku berbasis Android yang dibuat menggunakan software unity 3D, dapat ditarik beberapa kesimpulan penting antara lain:

1. Pada game sudoku ini memiliki informasi petunjuk penggunaan dan memiliki button bantuan yang memudahkan pengguna dalam bermain game sudoku
2. Game ini dibangun dan diimplementasikan dengan *software unity 3D* sehingga dihasilkan Game yang dapat meningkatkan daya pikir
3. Algoritma BFS dapat menyelesaikan permainan sudoku setelah dilakukan pengujian sebanyak 5 kali dan menghasilkan kecepatan rata rata 7.1 detik.

#### BAHAN REFERENSI

- [1] Geraldly Tambajong, “Bauran Pemasaran Pengaruhnya Terhadap Penjualan Sepeda Motor Yamaha di PT. Sarana Niaga Megah Kerta Manado,” *J. EMBA*, vol. 1, no. 3, pp. 1291–1301, 2013.
- [2] “Latent Semantic Indexing (LSI) and Hierarchical Dirichlet Process (HDP) Models on News Data,” in *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*, 2022, pp. 314–319.
- [3] A. R. Lubis, M. K. M. Nasution, O. S. Sitompul, and E. M. Zamzami, “A Framework of Utilizing Big Data of Social Media to Find Out the Habits of Users Using Keyword,” 2020, pp. 140–144.
- [4] A. F. Pakpahan *et al.*, *Pengembangan media pembelajaran*. Yayasan Kita Menulis, 2020.
- [5] A. R. Lubis, S. Prayudani, M. Lubis, and O. Nugroho, “Sentiment Analysis on Online Learning During the Covid-19 Pandemic Based on Opinions on Twitter using KNN Method,” in *2022 1st International Conference on Information System & Information Technology (ICISIT)*, 2022, pp. 106–111.
- [6] C. Halim and H. Prasetyo, “Penerapan Artificial Intelligence dalam Computer Aided Instructure (CAI),” *J. Sist. Cerdas*, vol. 1, no. 1, pp. 50–57, 2018.
- [7] A. R. Lubis, S. Prayudani, and M. Lubis, “Analysis of the Markov Chain Approach to Detect Blood Sugar Level,” in *Journal of Physics: Conference Series*, 2019, vol. 1361, no. 1, p. 12052.
- [8] T. F. Iskandar, M. Lubis, T. F. Kusumasari, and A. R. Lubis, “Comparison between client-side and

- server-side rendering in the web development,” in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 801, no. 1, p. 12136.
- [9] M. Lubis, R. Fauzi, A. R. Lubis, and R. Fauzi, “Analysis of project integration on smart parking system in Telkom University,” in *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, 2018, pp. 1–6.
- [10] O. Nugroho, “Implementation of Marker Based Tracking Method in the Interactive Media of Traditional Clothes Knowledge-Based on Augmented Reality 360,” *J. Comput. Sci. Inf. Technol. Telecommun. Eng.*, vol. 1, no. 2, pp. 37–43, 2020.
- [11] A. Yusuf and H. Hendra, “Penyelesaian Puzzle Sudoku Menggunakan Algoritma Brute Force dan Backtracking,” *Techno Nusa Mandiri J. Comput. Inf. Technol.*, vol. 10, no. 2, pp. 203–208, 2019.
- [12] C. Danuputri and N. Santosa, “Aplikasi Pemecahan Soal Sudoku dengan Metode Backtracking,” *J. Inform. Univ. Pamulang*, vol. 6, no. 3, pp. 506–511, 2021.
- [13] Nasution, Muhammad Irwan Padli, 2014, Keunggulan Kompetitif dengan Teknologi Informasi. Jurnal Elektronik
- [14] Nasution, Muhammad Irwan Padli, 2016, Aplikasi Pembelajaran Berbasis Mobile Untuk Tuna Aksara. *MATICS: Journal Of Computer Science and Information Technology*, 8 (1). pp. 11-16. ISSN 2477-2550, <http://dx.doi.org/10.18860/mat.v8i1.3475>