

# Pemodelan Pencarian Tim SAR dengan Flood Fill BFS Multi-agent

Christopher Jeffrey – 13520055<sup>1</sup>

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13520055@std.stei.itb.ac.id

**Abstract**—Flood fill is an algorithm that can be use in many cases, including in SAR operation. Flood fill can be implemented using BFS. SAR operation would be faster in finding target if multi-agent is used. The program can be implemented and visualized using javascript, with p5.js library.

**Keywords**—component; flood fill; SAR; BFS; multi-agent; p5.js

## I. PENDAHULUAN

### A. Search and Rescue

Search and Rescue (SAR) adalah kegiatan dan usaha mencari, menolong, dan menyelamatkan jiwa manusia yang hilang atau dikhawatirkan hilang akibat musibah kecelakaan. Istilah SAR telah digunakan secara internasional, tidak terkecuali di Indonesia.

Operasi SAR seharusnya dilakukan oleh personal yang memiliki keterampilan dan teknik untuk tidak membahayakan tim penolongnya sendiri maupun korbannya. Operasi SAR dilaksanakan terhadap musibah penerbangan seperti pesawat jatuh, mendarat darurat dan lain – lain, sementara pada musibah pelayaran bila terjadi kapal tenggelam, terbakar, tabrakan, kandas dan lain – lain. Demikian juga terhadap adanya musibah lainnya seperti kebakaran, gedung runtuh, kecelakaan kereta api dan lain – lain.

Dalam penyelenggaraan operasi SAR, sistem SAR dapat dipisah menjadi beberapa *stage*, yaitu :

1. Awareness Stage (Tahap Kekhawatiran)
2. Initial Action Stage (Tahap Kesiagaan / Preliminary Mode) adalah tahap seleksi informasi yang diterima, untuk segera dianalisis dan ditetapkan bahwa berdasarkan informasi tersebut, maka keadaan darurat saat itu diklasifikasikan sebagai :
  - a. INCERFA (Uncertainty Phase / Fase meragukan)
  - b. ALERFA (Alert Phase / Fase Mengkhawatirkan / Siaga)
  - c. DITRESFA (Distress Phase / Fase Darurat Bahaya)
3. Planning Stage (Tahap Perencanaan / Confinement Mode) yaitu tindakan sebagai tanggapan terhadap keadaan sebelumnya, antara lain :

- a. Search Planning event (Tahap Perencanaan Pencarian).
  - b. Search Planning Sequence (Urutan Perencanaan Pencarian)
  - c. Degree of Search Planning (Tingkatan Perencanaan Pencarian).
  - d. Search Planning Computing (Perhitungan Perencanaan Pencarian)
4. Operation Stage Detection Mode / Tracking Mode and Evacuation Mode, yaitu operasi pencarian dan pertolongan serta penyelamatan korban secara fisik. Tahap operasi meliputi :
    - a. Fasilitas SAR bergerak ke lokasi kejadian.
    - b. Melakukan pencarian dan mendeteksi tanda – tanda yang ditemui yang diperkirakan ditinggalkan survivor (Detection mode).
    - c. Mengikuti jejak atau tanda – tanda yang ditinggalkan survivor (Tracking Mode).
    - d. Menolong / menyelamatkan dan mengevakuasi korban (Evacuation Mode), dalam hal ini memberi perawatan gawat darurat pada korban yang membutuhkannya dan membawa korban yang cedera kepada perawatan yang memuaskan (evakuasi).
    - e. Mengadakan briefing kepada SRU.
    - f. Mengirim / memberangkatkan fasilitas SAR.
    - g. Melaksanakan operasi SAR di lokasi kejadian.
    - h. Melakukan penggantian / penjadwalan SRU di lokasi kejadian.
  5. Mission Conclusion Stage (Tahap Akhir Misi) merupakan tahap akhir operasi SAR, meliputi penarikan kembali SRU dari lapangan ke posko, penyiagaan kembali tim SAR untuk menghadapi musibah selanjutnya yang sewaktu – waktu dapat terjadi, evaluasi hasil kegiatan, mengadakan pemberitaan (Press Release) dan menyerahkan jenazah korban, survivor kepada yang berhak serta mengembalikan SRU pada instansi induk masing – masing dan pada kelompok masyarakat.

Kegiatan search and rescue dapat dibagi berdasarkan medan dan situasi pelaksanaan SAR menjadi

1. Ground (Lowland) search and rescue
2. Mountain rescue
3. Cave rescue
4. Urban search and rescue
5. Combat search and rescue
6. Maritime search and rescue

Saat tulisan ini dibuat, setidaknya ada 40 negara yang memiliki badan nasional SAR masing-masing.

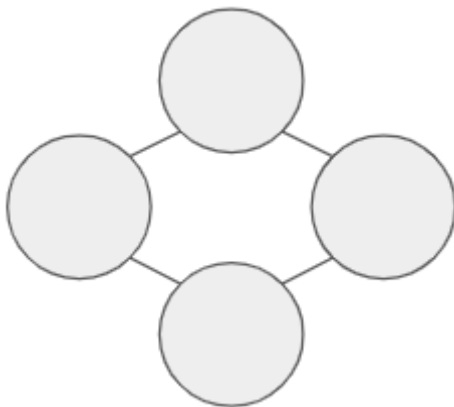
Badan nasional SAR Indonesia bernama Badan Nasional Pencarian dan Pertolongan, atau BNPP atau BASARNAS. Badan Nasional Pencarian dan Pertolongan adalah sebuah agen pemerintah yang bertanggung jawab untuk melaksanakan Search and Rescue secara nasional di Indonesia. Dalam kegiatannya, BASARNAS bisa dibantu oleh TNI, Brimob(Korps Brigade Mobil), dan Pemadam Kebakaran.

Berawal dari timbul pemikiran bahwa perlu diadakan suatu organisasi SAR Nasional yang mengkoordinir segala kegiatan – kegiatan SAR dibawah satu komando, mendorong terbentuknya cikal bakal SAR Indonesia hingga akhirnya menjadi BASARNAS

Tugas pokok Dalam Peraturan Menteri Perhubungan Nomor KM.43 Tahun 2005 Tentang Organisasi dan tata kerja Departemen Perhubungan, Badan SAR Nasional mempunyai tugas pokok melaksanakan pembinaan, pengkoordinasian dan pengendalian potensi Search And Rescue ( SAR ) dalam kegiatan SAR terhadap orang dan material yang hilang atau dikhawatirkan hilang, atau menghadapi bahaya dalam pelayaran dan atau penerbangan, serta memberikan bantuan SAR dalam penanggulangan bencana dan musibah lainnya sesuai dengan peraturan SAR Nasional dan Internasional.

### B. Graph

Dalam lingkup matematika, graph adalah suatu cara memodelkan suatu data, dengan sekumpulan objek, dan keterkaitan antara objek-objek tersebut. Suatu objek disebut dengan node. Keterkaitan antara node tersebut digambarkan menggunakan sebuah garis, disebut dengan vertex (vertices untuk jamak). Jadi suatu graph, dapat didefinisikan sebagai sekumpulan nodes dan sekumpulan vertices.



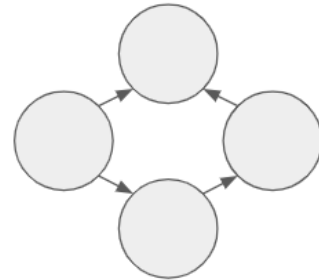
Gambar 1. Visualisasi graph

sumber: arsip penulis

Pada gambar x, terdapat empat buah node dan empat vertex.

Graph dapat dikategorikan menjadi directed graph dan undirected graph.

Pada directed graph, vertex mengimplikasikan arah. Biasanya dilambangkan dengan sebuah panah (dibandingkan dengan sebuah garis).



Gambar 2. Visualisasi directed graph  
sumber: arsip penulis

sedangkan pada undirected graph, vertex tidak memiliki arah. Biasanya hanya sebuah garis polos. Contohnya pada gambar 1.

### C. Penelusuran Graph

Penelusuran graph adalah sebuah tindakan memproses semua node yang terhubung (menggunakan vertex) dengan node awal (root node), dan memproses node yang terhubung dengan node-node yang baru saja di proses, begitu seterusnya hingga semua node yang terhubung berhasil di proses. Dua buah algoritma yang sering digunakan adalah depth first search(DFS) dan Breadth first Search(BFS).

### D. BFS

Breadth First Search adalah sebuah algoritma penelusuran graph dengan karakteristik pencarian node dilakukan per-depth atau per-layer. Setelah suatu layer selesai diproses, barulah berpindah ke depth atau layer berikutnya. Biasanya menggunakan queue sebagai data struktur pembantu.

Secara umum, langkah algoritma BFS adalah sebagai berikut

1. Masukkan root node ke dalam queue.
2. Ambil satu node dari queue (dengan cara dequeue), kita sebut dia sebagai expand node, lalu dapatkan semua node yang terhubung(dengan vertex) ke expand node.
3. Proses setiap node yang terhubung dengan expand node, termasuk tandai sebagai node yang sudah pernah diproses, dan masukkan node ke dalam queue(dengan cara enqueue)
4. Selama queue belum kosong, kembali ke langkah dua.

### E. Flood Fill

Flood fill adalah sebuah algoritma untuk menentukan dan memproses area(atau node, atau cell) yang terhubung dengan suatu node yang telah kita tentukan, dalam suatu matriks multidimensi. Flood fill bertujuan untuk memproses seluruh node yang terhubung, sesuai namanya, flood yang artinya

banjir. Biasanya digunakan dalam permasalahan matematika seperti labirin(*maze*), atau graph.

Algoritma flood fill dapat diterapkan menggunakan Breadth First Search.

#### F. Multi-agent

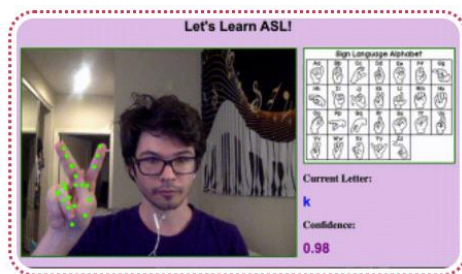
Agent adalah suatu objek yang bertindak sebagai pelaksana suatu algoritma. Multi-agent adalah sekumpulan agent yang bersama menyelesaikan suatu persoalan, dan dapat berbagi informasi satu sama lain. Salah satu keuntungan dari multi-agent adalah meningkatkan parameter keberhasilan penyelesaian suatu masalah, misalnya meningkatkan kecepatan suatu persoalan diselesaikan.

#### G. p5.js

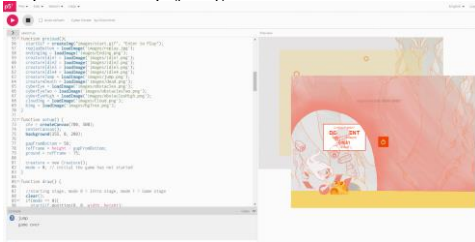


Gambar 3. Logo p5.js  
sumber : <https://p5js.org/>

p5.js adalah sebuah library javascript yang gratis dan open source, yang menawarkan utilitas untuk memvisualisasikan suatu hal. p5.js memudahkan kita untuk menampilkan suatu objek di browser, dapat berupa text, bentuk-bentuk dasar, video, suara. p5.js dapat digunakan untuk proyek yang sederhana, hingga yang rumit(misalnya dengan digunakan bersamaan dengan library ml5)



Gambar 3. Contoh proyek membaca ASL  
sumber: <https://showcase.p5js.org/#/2021-Advanced/The-ASL-Machine/>



Gambar 4. Contoh proyek dijalankan di online editor  
sumber: <https://showcase.p5js.org/#/2021-Beginner/Cyber-Forest/>

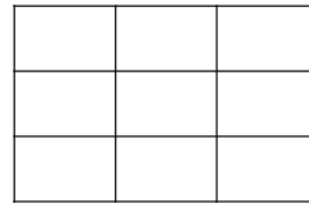
p5.js relatif sederhana untuk digunakan dan relatif mampu untuk melakukan banyak hal (*powerful, capable*). p5js dapat dijalankan melalui *online editor*.

p5.js sangatlah *community driven*, dengan komunitas yang melakukan development, dokumentasi, pengajaran, pembuatan project dan pendanaan.

Penulis menggunakan p5.js sebagai library untuk memvisualisasikan pemodelan pencarian SAR yang akan dijelaskan lebih lengkap dibawah.

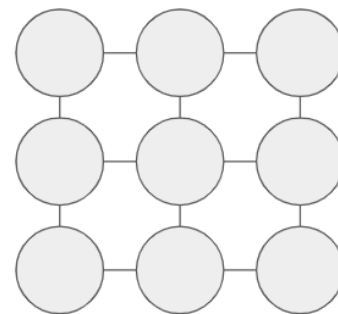
## II. MATRIKS DUA DIMENSI SEBAGAI GRAPH

Kita dapat memodelkan matriks dua dimensi menjadi sebuah graph.



Gambar 5. Matriks dua dimensi  
sumber: arsip penulis

Kita dapat melihat matriks sebagai sekumpulan *Cell*, yang masing-masing berada pada baris *i* dan kolom *j*. Tiap *Cell* akan menjadi node pada graph. Sebuah *Cell* dapat bertetangga dengan *Cell* lainnya, jika salah satu sisinya berhimpit. Jika ada sisi yang berhimpit, dibuat sebuah vertex yang menghubungkan kedua *Cell* (yang sudah menjadi node). Kita lakukan hal ini untuk setiap *Cell*, dan akan kita dapatkan representasi graph dari matriks tersebut.



Gambar 6. Representasi graph dari matriks dua dimensi  
sumber: arsip penulis

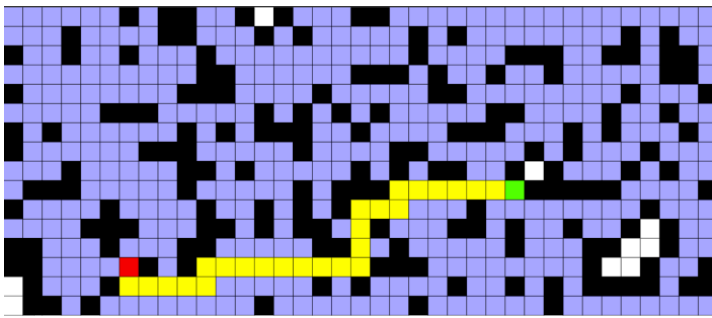
## III. MEMODELKAN PENCARIAN SAR DENGAN MATRIX DUA DIMENSI

Pemodelan ini akan mencakup tahap operation stage dari kegiatan Search and Rescue, terutama pada bagian a hingga d.

Tipe pencarian yang dibuat dapat dilakukan oleh lebih dari satu agent(multi-agent), dengan medan darat yang dapat memiliki area yang tidak terjangkau, seperti pada mountain rescue dan cave rescue.

Peta pencarian kita modelkan sebagai matriks dua dimensi, dengan masing-masing *Cell* mewakili satu satuan area di dunia nyata, misalnya satu *Cell* mewakili satu meter persegi.

Peta pencarian dapat kita visualisasikan sebagai sekumpulan persegi. Untuk memudahkan pengidentifikasian, dapat kita buat suatu konvensi. Untuk *Cell* biasa, kita beri warna putih. Untuk Agent, kita beri warna hijau. Target kita beri warna merah. *Cell* yang telah dieksplorasi kita beri warna ungu. *Cell* yang merupakan jalan dari Agent ke Target kita beri warna kuning(kita sebut sebagai *Path*). Untuk *Cell* yang tidak dapat dieksplor, kita beri warna hitam (kita sebut sebagai *Wall*)



Gambar 7. Peta pencarian dengan agent(hijau), target(merah), wall(hitam), path(kuning)  
sumber: arsip penulis

#### IV. PENERAPAN FLOOD FILL BFS MULTI-AGENT

Untuk membantu implementasi, program didekati dengan konsep Object Oriented. Dibuat beberapa kelas untuk meningkatkan abstraksi. Kelas-kelas yang dibuat adalah

##### 1. Kelas Cell

Memiliki atribut `isVisited`, `isWall`, `isPath`, `isTargetHere`, `isAgentHere`, `row`, `col`, dan `previousCell`. Memiliki method `constructor()`, `draw()`, `highlightAsAgent()`, `highlightAsTarget()`, dan `getUnvisitedNeighbors()`. Instansiasi dari kelas ini mewakili satu buah Cell pada peta, atau node pada graph.

##### 2. Kelas Agent

Memiliki atribut `row`, `col`, `queueOfCells`. Memiliki method `constructor()`, dan `doBFSNextLayer()`. Method `doBFSNextLayer()` lah yang menerapkan FloodFill dan BFS. Instansiasi dari kelas ini mewakili satu buah agent, yaitu sebuah grup tim SAR yang melakukan pencarian.

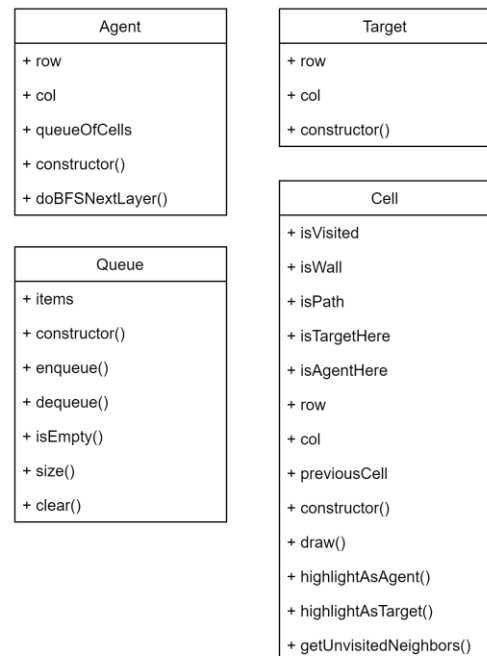
##### 3. Kelas Target

Memiliki atribut `row`, dan `col`. Memiliki method `constructor()`. Instansiasi dari kelas ini mewakili satu buah target, yaitu seseorang yang akan di evakuasi.

##### 4. Kelas Queue

Memiliki atribut `items`. Memiliki method `constructor()`, `enqueue()`, `dequeue()`, `isEmpty()`, `size()`, `clear()`. Kelas ini digunakan dalam penerapan BFS.

Berikut diagram kelas untuk masing-masing kelas



Gambar 8. Diagram kelas Agent, Target, Queue, Cell  
sumber: arsip penulis

Method utama yang digunakan untuk menerapkan Flood Fill adalah `doBFSNextLayer()` pada kelas `Agent()`. Flood Fill dilakukan dengan menjalankan method tersebut terus menerus, hingga nilai `isVisited` seluruh Cell yang terhubung dengan Agent menjadi true. Method akan dijalankan pada peta pencarian yang merupakan matriks dua dimensi, yang dapat kita modelkan sebagai graph.

Berikut Langkah `doBFSNextLayer`

1. Dequeue semua node yang ada di dalam queue agent dan masukkan kedalam suatu variabel. Seluruh node yang berada di dalam queue agent adalah cel-cel yang berada pada *depth* atau *layer* yang sama.
2. Proses setiap node di layer tersebut. Jadikan dia sebagai Expand Node.
3. Proses setiap Expand Node. Jika Expand Node belum dikunjungi, ubah `isVisited` menjadi true dan dapatkan semua Neighboring Node dari Expand Node.
4. Proses setiap Neighboring Node. Set Expand Node sebagai Previous Cell dari Neighboring Node tersebut. Tambahkan Neighboring Node tersebut ke dalam queue agent(Node ini berada pada *depth* atau *layer* selanjutnya, akan diproses sebagai Expand Node pada pemanggilan `doBFSNextLayer` selanjutnya)
5. Cek apakah Expand Node merupakan Target. Jika iya, set `isPath` sebagai true secara rekursif ke previous node hingga ke node pertama (node Agent)

Berikut kodenya

```
doBFSNextLayer() {
    let nodeInThisLayer = [];
```

```

// 1. get all the node in queue. semua node di dalam queue saat ini pasti berada di
layer atau depth yang sama.
while (this.queueOfCells.size() > 0) {
    nodeInThisLayer.push(this.queueOfCells.dequeue());
}

//2. untuk tiap node di layer ini, jadikan dia expand node.
for (let i = 0; i < nodeInThisLayer.length; i++) {
    let expandNode = nodeInThisLayer[i];
    if (!expandNode.isVisited) {
        // 3. untuk setiap expandNode, jika expandNode belum dikunjungi, set
isVisited menjadi true dan dapatkan neighboring node dari expandNode.
        expandNode.isVisited = true;
        let neighbors = expandNode.getUnisVisitedNeighbors();

        // 4. untuk setiap neighboring node, set expandNode sebagai previous cell, dan
tambahkan dia ke dalam queue agent(kita tambahkan layer selanjutnya ke dalam
queue)
        for (let i = 0; i < neighbors.length; i++) {
            neighbors[i].previousCell = expandNode;
            this.queueOfCells.enqueue(neighbors[i]);
        }

        //5. cek apakah expandNode merupakan target. jika iya, set atribut 'apakah cell
tersebut bagian dari path' pada setiap cell menjadi true
        if (expandNode.isTargetHere) {
            console.log("found target");

            // set cell as path
            let current = expandNode;
            while (current.previousCell) {
                current.isPath = true;
                current = current.previousCell;
            }
        }
    }
}

```

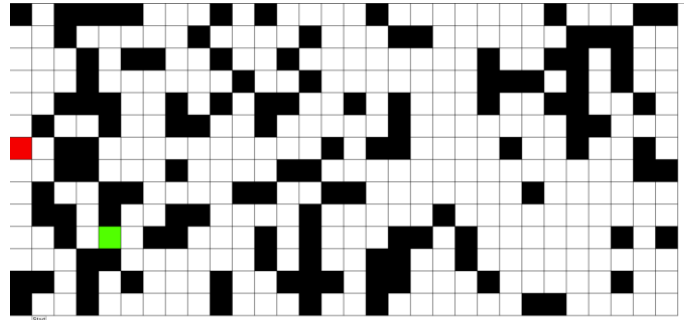
Algoritma ini menjamin target akan ditemukan oleh agent terdekat, dengan langkah yang paling minimum(layer atau depth minimum).

## V. MENJALANKAN PENCARIAN

Program diterapkan menggunakan p5js, sehingga kita perlu mengikuti konvensi dari p5js. File utama yang dari program ini bernama sketch. Pada file sketch, terdapat dua buah fungsi utama, yaitu setup dan draw. Fungsi setup akan dipanggil saat pertama kali program di-load ke dalam browser. Fungsi draw akan dipanggil terus menerus setiap kali layar browser diperbarui.

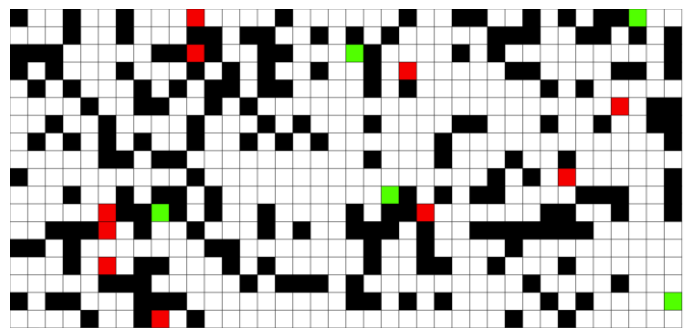
Pada fungsi setup, canvas tempat kita dapat meletakkan objek-objek akan dibuat. Akan dibuat Cell sesuai ukuran peta pencarian yang kita inginkan.

Pada fungsi draw, Cell akan ditampilkan ke layar. Warna dari Cell dibuat sesuai konvensi yang telah dijelaskan sebelumnya.



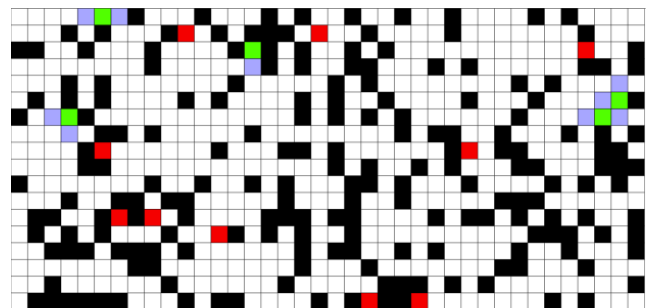
Gambar 9. Contoh peta dengan 1 Agent, 1 Target, dengan 30% Wall.  
sumber: arsip penulis

Akan dibuat juga Agent, dan Target dengan jumlah yang dapat diubah. Untuk tambahan, program juga memiliki tombol start, sebagai pemicu dijalkannya algoritma untuk melakukan Flood Fill. Peta akan di generate secara acak untuk memvariasikan pengetestan, dan tembok akan di letakkan secara acak juga, dengan persentase wall terhadap seluruh Cell dapat kita tentukan (misalnya terdapat 1 wall setiap 10 Cell).



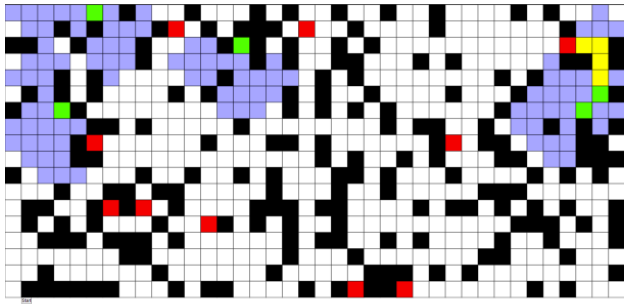
Gambar 10. Contoh peta dengan 5 Agent, 10 Target, dengan 30% Wall.  
sumber: arsip penulis

Jika tombol start telah di klik, maka setiap kali draw dipanggil oleh browser, akan dijalankan method doBFSNextLayer() untuk setiap Agent. Artinya akan dijalankan Flood Fill pada layer selanjutnya.



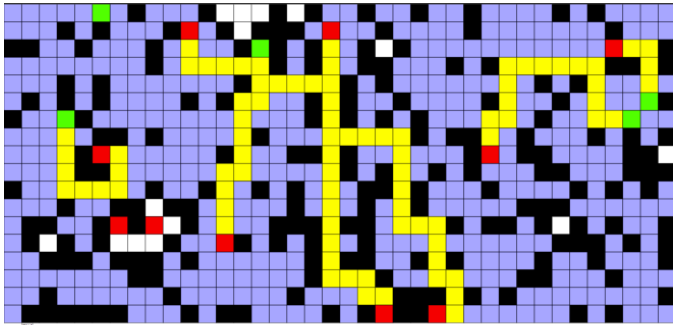
Gambar 11. Contoh flood fill pada layer pertama  
sumber: arsip penulis

Contoh flood fill pada layer pertama



Gambar 12. Contoh target berhasil ditemukan oleh Agent  
sumber: arsip penulis

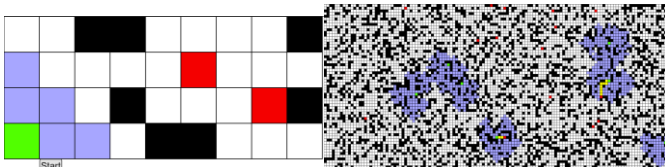
Flood Fill akan selesai ketika semua Cell yang terhubung ke seluruh Agent(sebagai root node) telah di proses (nilai isVisited menjadi true)



Gambar 13. Contoh flood fill selesai  
sumber: arsip penulis

Pada gambar 13, dapat dilihat terdapat dua buah target yang tidak terjangkau oleh Agent. Sesuai dengan realita, terkadang Target tidak terjangkau oleh Agent karena medan peta pencarian yang memang tidak memungkinkan (terhalang oleh Wall sepenuhnya)

Program dapat menerima peta pencarian dengan berbagai ukuran, disesuaikan dengan keinginan.



Gambar 14. Contoh ukuran peta pencarian dengan jumlah cell relatif sedikit, dan banyak  
sumber: arsip penulis

## VI. KESIMPULAN

Pemodelan dapat berguna untuk menyederhanakan sebuah masalah, membantu kita untuk fokus ke bagian yang ingin kita dalami lebih lanjut. Algoritma flood fill dapat dibuat menggunakan algoritma BFS, p5.js dapat digunakan untuk memvisualisasikan peta pencarian dengan relatif mudah.

## LINK VIDEO

Sebagai pendukung makalah, dibuat sebuah video yang dapat diakses pada link berikut [https://youtu.be/VgiBZfaK\\_0w](https://youtu.be/VgiBZfaK_0w)

## LINK GITHUB

Program implementasi yang dibuat penulis dibuat menggunakan javascript dengan library p5.js, dan ditulis menggunakan typescript dapat diakses pada link repository berikut <https://github.com/christojeffrey/FloodFill-BFS-p5js>

## UCAP SYUKUR

Puji syukur kepada Tuhan Yang Maha Esa, atas berkat dan rahmat-Nya membimbing penulis menyelesaikan makalah ini. Penulis berterima kasih secara pribadi kepada dosen-dosen pengampu mata kuliah IF2211 Strategi Algoritma, Dr. Masayu Leylia Khodra, ST., MT., Dr. Nur Ulfa Maulidevi, ST., M.Sc., dan Dr. Ir. Rinaldi Munir, MT yang telah membagikan ilmunya yang menjadi landasan penyusunan makalah ini.

## REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/stima21-22.htm#SlideKuliah>. Diakses pada 15 Mei 2022.
- [2] <https://p5js.org/get-started/>. Diakses pada 14 Mei 2022.
- [3] <https://basarnas.go.id/tugas-dan-fungsi>. Diakses pada 20 Mei 2022.
- [4] <https://www.geeksforgeeks.org/flood-fill-algorithm/>. Diakses pada 14 Mei 2022.
- [5] <https://thecodingtrain.com/beginners/p5js/>. Diakses pada 14 Mei 2022.

## 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, 21 Mei 2022

Christopher Jeffrey  
13520055