Aplikasi Algoritma Teori Graf Pada Robot Pemecah Labirin Cerdas

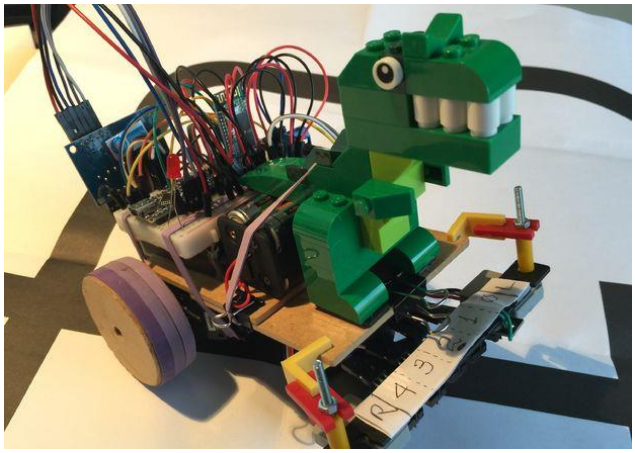
Dimas Aditia Pratikto - 13516153
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516153@std.stei.itb.ac.id

***Abstract*— Salah satu tantangan yang terpenting bagi robot pemecah labirin adalah seberapa cepat dan mangkus robot tersebut untuk menemukan jalur terpendek dari titik awal menuju tujuan akhir yang diinginkan. Makalah ini mengulas tentang robot pemecah labirin berdasarkan pengolahan citra dan algoritma teori graf yang telah penulis pelajari. Graf sangat membantu robot pemecah labirin dalam menentukan jalur terpendeknya pada garis labirin. Gambar garis labirin akan ditangkap oleh kamera dan dikirim ke komputer untuk dianalisis dan diproses oleh program yang dikembangkan menggunakan *library* Visual C++ dan OpenCV berdasarkan pada algoritma teori graf. Program yang dikembangkan untuk memecahkan labirin yaitu memeriksa semua kemungkinan jalur yang ada di labirin tersebut agar bisa membawa robot ke titik tujuan yang diinginkan. Setelah itu, jalur terpendek yang terbaik ditentukan dan kemudian petunjuk akan membimbing robot mencapai titik tujuan yang diinginkan dikirim ke robot melalui bluetooth. Robot mengikuti jalur panduan yang diterima untuk mencapai tujuannya. Selain itu, aplikasi teori graf juga memungkinkan robot pemecahan labirin untuk menghindari jebakan dan jatuh dalam lingkaran tanpa batas. Aplikasi sistem pemecahan labirin mencakup kontrol lalu lintas cerdas yang membantu ambulans, pemadam kebakaran, atau menyelamatkan robot untuk menemukan jalur terpendek ke tempat tujuan mereka.**

***Kata Kunci*—Robot, Graf, Pemecah Labirin, Pengolahan Citra, Aplikasi Graf**

I. PENDAHULUAN

Pesatnya pertumbuhan teknologi di zaman kita lebih besar dari sebelumnya. Robot merupakan alat mekanik yang dapat melakukan tugas fisik baik menggunakan pengawasan dan kontrol manusia maupun menggunakan program yang telah di definisikan terlebih dahulu. Banyak perangkat robot diciptakan agar menjadikan hidup lebih mudah. Robot biasanya digunakan untuk menyelesaikan tugas yang berat, berbahaya, pekerjaan yang berulang, dan kotor. Robot bisa ditempatkan di industri untuk transportasi suku cadang yang akurat dan cepat untuk berpindah dari satu terminal ke terminal lainnya. Robot juga telah digunakan untuk menyelamatkan nyawa-nyawa manusia dengan menjangkau daerah-daerah berbahaya yang manusia tidak dapat mencapai daerah berbahaya tersebut. Lalu robot juga dapat diimplementasikan sebagai penyedot debu yang perlu dinavigasikan sendiri di rumah dan dibersihkan pada waktu bersamaan. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air.



Gamabar 1.1 Maze Solving Robot

Bagi kebanyakan aplikasi robot mobile, perencanaan jalur merupakan tugas penting. Setiap struktur dari beberapa jalur dapat disebut labirin. Jalur antara dua lokasi spesifik yang ada di labirin harus direncanakan dengan baik dan jalur yang direncanakan harus bebas dari tabrakan dan salah satu yang paling pendek. Aplikasi sistem pemecahan labirin mencakup kontrol lalu lintas cerdas yang membantu ambulans, pemadam kebakaran, atau robot penyelamatan untuk menemukan jalur terpendek secara akurat ke tujuan mereka. Teknik tradisional yang digunakan labirin pemecahan robot didasarkan pada trial and error dimana robot harus mencoba setiap jalur yang ada di labirin sampai menemukan titik tujuannya. Teknik tradisional ini memakan waktu yang lama dan bisa menjebak robot dalam lingkaran tak terbatas. Apalagi teknik tradisional tidak selalu memberikan jalan terpendek. Pendekatan menggunakan teori graf untuk pemecahan labirin berguna demi menghindari masalah yang disebutkan di atas. Ini adalah teknik cerdas berdasarkan pengolahan citra dan algoritma kecerdasan buatan yang dapat mengurangi waktu pemecahan labirin dan menemukan jalur terpendek. Gambaran keseluruhan labirin ditangkap oleh kamera dan dikirim ke komputer untuk dianalisis dan diproses oleh program yang dikembangkan menggunakan perpustakaan Visual C++ dan OpenCV. Program yang dikembangkan didasarkan pada algoritma teori grafik sebagai salah satu algoritma cerdas buatan cerdas (AI) yang dapat menganalisis labirin dan menemukan jalur terpendek ke tujuan tujuan yang tersedia di labirin.

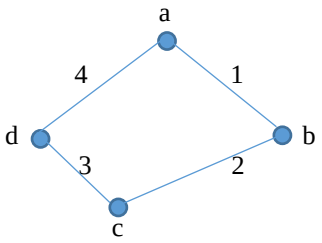
Ada banyak jenis algoritma teori graf yang bisa digunakan untuk pemecahan labirin. Untuk menemukan algoritma terbaik yang dapat memecahkan citra labirin yang ditangkap, tiga jenis algoritma teori graf diimplementasikan dan diuji yaitu algoritma Breadth First Search, Best First Search dan algoritma A*. Algoritma ini diuji pada banyak labirin dengan variasi kompleksitas dan hasilnya dibandingkan berdasarkan waktu penyelesaian yang diperlukan dan panjang jalur yang dihasilkan dari masing-masing metode. Berdasarkan data yang diuji, algoritma Breadth First Search adalah yang paling akurat yang menemukan jalur terpendek untuk setiap labirin yang diuji. Oleh karena itu algoritma Breadth First Search diimplementasikan pada program pemecahan labirin.

Makalah ini disusun sebagai berikut. Bagian 2 membahas dasar teori graf. Bagian 3 menjelaskan implementasi pada robot pemecah labirin. Bagian 4 menyajikan dan membahas hasil beberapa eksperimen. Bagian 5 membahas seputar kendala umum. Bagian 6 merangkum kesimpulan dari makalah ini. Terakhir, bagian 7 berisi tentang ucapan terima kasih.

II. Dasar Teori

A. Definisi Graf

Graf G didefinisikan sebagai pasangan himpunan *(V, E)*, ditulis dengan notasi *G = (V, E)*, yang dalam hal ini *V* adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan *E* adalah himpunan sisi (*edge* atau *arcs*) yang menghubungkan sepasang simpul. Definisi tersebut menyatakan bahwa *V* tidak boleh kosong, tetapi *E* boleh kosong. Jadi, sebuah graf dimungkinkan tidak memiliki sisi satu buah pun, tetapi minimal harus ada satu simpul. Berikut adalah contoh sebuah graf sederhana.



Gambar 2.1 Graf Sederhana

Gambar 2.1 memperlihatkan graf dengan himpunan simpul *V* dan himpunan sisi *E*, dimana

V = {a, b, c, d}
E = {1, 2, 3, 4}

B. Terminologi Graf

Di bawah ini merupakan beberapa terminologi yang akan sering digunakan ketika membahas masalah graf:

1. Bertetangga (Adjacent) Dua buah simpul pada graf tak-berarah dikatakan bertetangga, jika keduanya terhubung langsung melalui suatu sisi.
2. Bersisian (Incident) Untuk sembarang sisi *E* = (*V*, *U*), sisi *E* dikatakan bersisian dengan simpul *V* dan simpul *U*.
3. Graf Kosong (Null Graph) Graf yang himpunan sisinya kosong, namun himpunan simpulnya tidak.
4. Derajat (Degree) Pada graf tak-berarah, derajat suatu simpul merupakan banyak sisi yang bersisian dengan simpul tersebut. Sedangkan pada graf berarah, derajatnya terbagi dua, yaitu derajat masuk, dan derajat keluar, dimana derajat masuk adalah banyak busur yang masuk ke simpul tersebut, sedangkan derajat keluar adalah banyak busur yang keluar dari simpul tersebut.
5. Lintasan (Path) Lintasan yang panjangnya n dari simpul awal V_0 ke simpul tujuan vn di dalam graf G adalah barisan berselang-seling dari simpul-simpul dan sisi-sisi berbentuk V_0, E_1, V_1, E_2, ..., E_n, V_n sehingga banyaknya sisi yang terlewati adalah n, dan E_1, E_2, ..., E_n merupakan sisi-sisi dari graf G.
6. Terhubung (Connected) Graf tak-berarah G disebut graf terhubung jika untuk setiap pasang V_i dan V_j yang merupakan simpul di G, terdapat lintasan yang menghubungkan keduanya. Jika ada yang tidak terhubung maka graf tersebut disebut graf takterhubung.
7. Graf Berbobot (Weighted Graph) Graf berbobot adalah graf yang setiap sisinya mempunyai sebuah nilai (bobot).

C. Jenis-jenis Graf

Berdasarkan ada tidaknya gelang atau sisi-ganda pada suatu graf, maka secara umum graf dapat digolongkan menjadi dua jenis, graf sederhana (simple graph) dan graf tidak-sederhana (unsimple-graph). Graf sederhana merupakan graf yang tidak mempunyai sisi-ganda dan sisi gelang, sedangkan graf tidak-sederhana mempunyai sisi gelang atau sisi-ganda atau keduanya. Sisi pada sebuah graf dapat mempunyai orientasi arah.

Berdasarkan orientasi arahnya, graf dibedakan menjadi dua jenis, yaitu graf berarah (directed graph) dan graf tak-berarah (undirected graph). Graf berarah merupakan graf yang setiap sisinya mempunyai orientasi arah, yang biasa disebut busur. Untuk suatu busur (sisi berarah) yang menghubungkan simpul V_j ke simpul V_k, V_j disebut simpul asal, sedangkan V_k disebut simpul terminal. Pada graf berarah sisi gelang diperbolehkan, sedangkan sisi-ganda tidak diperbolehkan. Sementara itu, graf tak-berarah adalah graf yang sisinya tidak mempunyai orientasi arah

D. Robot Pemecah Labirin

Robot pemecah labirin adalah robot yang dirancang khusus untuk bekerja pada sebuah labirin. Tujuannya adalah berpindah dari titik awal yang ditentukan ke tujuan yang ditentukan menggunakan sensor warna tertentu pada lintasannya. Robot ini harus mencari jalur terpendek ke tujuan yang ditentukan. Algoritma pencarian pada robot ini dapat menggunakan algoritma Breadth First Search, Best First Search dan A*.

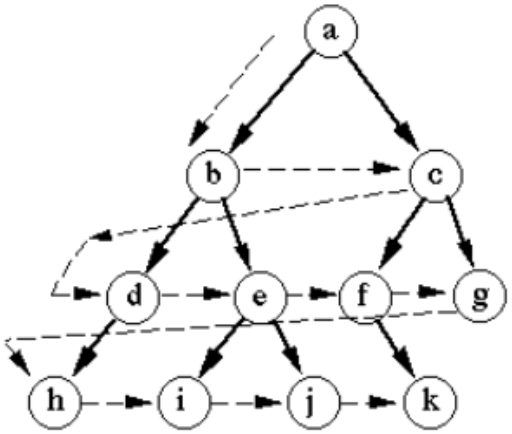
E. Breadth First Search (BFS)

Sekilas tentang Algoritma BFS. *Breadth-first search* adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi simpul secara preorder yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga

dengan simpul tersebut terlebih dahulu, selanjutnya simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya. Jika graf berbentuk pohon berakar, maka semua simpul pada aras *d* dikunjungi lebih dahulu sebelum simpul-simpul pad aras *d*+1. Berikut adalah algoritma *Breadth First Search* dalam pseudo code:

```
procedure BFS (input v:integer)
KAMUS

    w: integer
    q : antrian

    procedure BuatAntrian (input/output q:antrian)
    {membuat antrian kosong kepala (q) diisi 0}

    procedure MasukAntrian (input/output q:antrian, input
                            v:integer )
    {memasukkan v ke dalam antrian q pada posisi belakang}

    procedure HapusAntrian (input/output q:antrian, output
                            v:integer )
    {menghapus v dari kepala antrian q}

    function AntrianKosong (input q:antrian) → boolean
    {true jika antrian q kosong, false jika sebaliknya}

ALGORITMA
    BuatAntrian(q)
    write(v)
    dikunjungi(v) ← true

    MasukAntrian (q,v)

    while not AntrianKosong(q) do
        HapusAntrian(q,v)
        for tiap simpul w yang bertetangga dengan simpul v do
            if not dikunjungi(w) then
                write(w)
                MasukAntrian(q,v)
                dikunjungi(w) ← true
            end
        end
    end
```

Algoritma ini memerlukan sebuah antrian *q* untuk menyimpan simpul yang telah dikunjungi. Simpulsimpul ini diperlukan sebagai acuan untuk mengunjungi simpul-simpul yang bertetanggaan dengannya. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan tabel boolean untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali.



Gambar 2.2 Ilustrasi Algoritma Breadth First Search

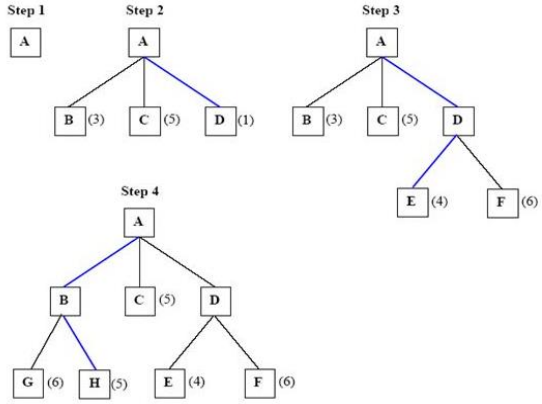
F. Best First Search

Best-First Search merupakan sebuah metode yang membangkitkan simpul dari simpul sebelumnya. Best-first search memilih simpul baru yang memiliki biaya terkecil diantara semua *leaf nodes* (simpul-simpul pada level terdalam) yang pernah dibangkitkan. Penentuan simpul terbaik dilakukan dengan menggunakan sebuah fungsi yang disebut fungsi evaluasi *f(n).* Fungsi evaluasi best-first search dapat berupa biaya perkiraan dari suatu simpul menuju ke goal atau gabungan antara biaya sebenarnya dan biaya perkiraan tersebut.

Pada setiap langkah proses pencarian terbaik pertama, kita memilih node-node dengan menerapkan fungsi heuristik yang memadai pada setiap node/simpul yang kita pilih dengan menggunakan aturan-aturan tertentu untuk menghasilkan penggantinya. Fungsi heuristic merupakan suatu strategi untuk melakukan proses pencarian ruang keadaan suatu problema secara selektif, yang memandu proses pencarian yang kita lakukan sepanjang jalur yang memiliki kemungkinan sukses paling besar.

Ada beberapa istilah yang sering digunakan pada metode best-first search, yaitu:

1. Start node adalah sebuah terminologi untuk posisi awal sebuah pencarian
2. Current node adalah simpul yang sedang dijalankan dalam algoritma pencarian jalan terpendek
3. Suksesor adalah simpul-simpul yang yang akan diperiksa setelah current node
4. Simpul (node) merupakan representasi dari area pencarian
5. Open list adalah tempat menyimpan data simpul yang mungkin diakses dari starting node maupun simpul yang sedang dijalankan
6. Closed list adalah tempat menyimpan data simpul yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan
7. Goal node yaitu simpul tujuan
8. Parent adalah curret node dari suksesor.



Gambar 2.3 Ilustrasi Algoritma Best First Search

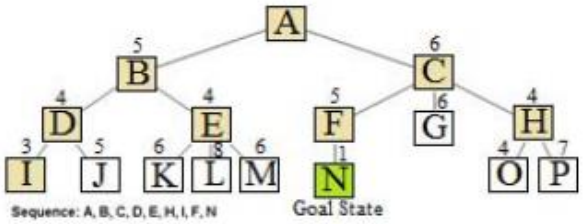
G. Algoritma A-star (A)*

Algoritma A-Star (A)*, ditemukan pertama kali oleh Peter Hart, Nils Nilsson, dan Bertram Raphael pada tahun 1968 adalah algoritma pencarian rute terpendek (shortest path) yang merupakan perbaikan dari Algoritme BFS dengan memodifikasi fungsi heuristiknya untuk memberikan hasil yang optimal. Dimana menggabungkan fungsi heuristik *h(n)* dan jarak sesungguhnya/cost *g(n)*. Notasinya adalah sebagai berikut:

$$f(n) = g(n) + h(n)$$

Keterangan:

1. ***f(n)*** adalah jumlah dari *g(n)* dan *h(n)*. Ini adalah perkiraan jalur terpendek sementara. maka *f(n)* adalah jalur terpendek yang sebenarnya yang tidak ditelusuri sampai Algoritma A-Star (A*) diselesaikan.
2. ***g(n)/Geographical Cost*** adalah total jarak yang didapat dari verteks awal ke verteks sekarang (halangan).
3. ***h(n)/Heuristic Cost*** adalah perkiran jarak dari vertek sekarang (yang sedang dikunjungi) ke vertek tujuan. sebuah fungsi *heuristic* digunakan untuk membuat perkiraan seberapa jauh lintasan yang akan diamnbil ke vertek tujuan.



Gambar 2.4 Ilustrasi Algoritma A*

III. IMPLEMENTASI PADA ROBOT PEMECAH LABIRIN

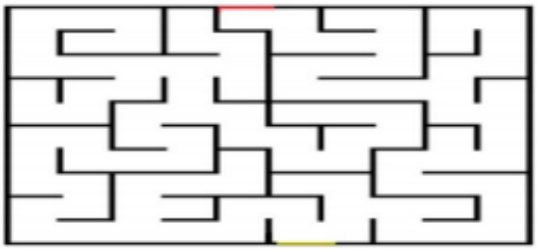
A. Menangkap Gambar Labirin

Proses pemecahan labirin dimulai dengan menangkap tampilan atas labirin menggunakan kamera dan gambar labirin yang tertangkap disimpan di folder yang diinginkan di komputer untuk diproses lebih lanjut. Kamera harus dikalibrasi untuk mendapatkan parameter intrinsik dan ekstrinsiknya. Kalibrasi kamera merupakan langkah penting dalam penglihatan komputer yang memberikan kemampuan untuk membangun hubungan antara koordinat piksel 2D dan koordinat dunia nyata.

Gambar 3.1 Tujuan dari Sistem Robot Pemecah Labirin

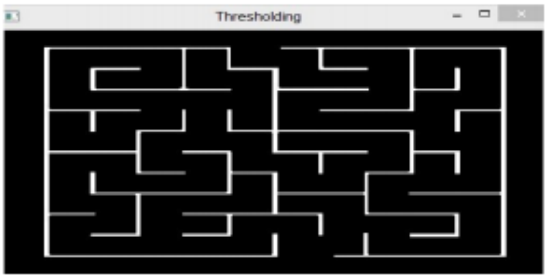
B. Memproses Ulang Gambar Labirin

Ini adalah langkah penting yang melibatkan preprocessing gambar untuk mempersiapkan gambar labirin yang akan dianalisis dengan menggunakan salah satu algoritma teori graf. Langkah ini mencakup penyaringan dan pendeteksian gambar serta titik akhir labirin. Kode dikembangkan menggunakan *library* Visual C ++ dan OpenCV. Pertama, kode mengimport gambar labirin yang ditangkap dan menerapkan beberapa filter untuk peningkatan gambar. Setelah itu dari gambar labirin berwarna, lokasi awal dan lokasi akhir labirin terdeteksi. Garis berwarna merah mewakili lokasi awal atau pintu masuk labirin sedangkan garis berwarna kuning mewakili lokasi akhir labirin atau gerbang keluar labirin seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3.2 Labirin Original

Kemudian setelah pendeteksi lokasi masuk dan keluar labirin dan berdasarkan intensitas warna ambang batas, gambar berwarna diubah menjadi gambar biner seperti yang ditunjukkan pada Gambar 3.2 dan menjadi siap untuk menerapkan salah satu algoritma teori graf di dalamnya seperti yang dijelaskan pada langkah selanjutnya.



Gambar 3.3 Labirin Setelah Dikonversi

C. Menemukan Jalur Terpendek Menggunakan Teori Graf

Berikut adalah algoritma umum suatu robot pemecah labirin dalam sebuah bahasa pemrograman:

```
/*L = left turn
R = right turn
S = going straight past a turn
B = turning around*/

void JalurTerpendek()
{
    int ShortDone = 0;

    if(path[path_length-3]=='L' && path[path_length-1]=='R')
    {
        path_length -= 3;
        path[path_length] = 'B';
        ShortDone = 1;
    }
    if(path[path_length-3]=='L' && path[path_length-1]=='S'
      && ShortDone==0)
    {
        path_length -= 3;
        path[path_length] = 'R';
        ShortDone = 1;
    }
    if(path[path_length-3]=='R' && path[path_length-1]=='L'
      && ShortDone==0)
    {
        path_length -= 3;
        path[path_length] = 'B';
        ShortDone = 1;
    }
}
```

Untuk memecahkan labirin dan menemukan jalur terpendek antara titik awal dan akhir, tiga jenis algoritma teori graf diimplementasikan dan hasilnya dibandingkan untuk memilih yang terbaik. Pencarian Breadth First Search, Best First Search dan A * diimplementasikan dan diuji pada labirin sederhana dan kompleks yang berbeda dan hasilnya dibandingkan berdasarkan waktu penyelesaian yang diperlukan dan panjang jalur yang dihasilkan dari masing-masing metode. Tujuan dari perbandingan ini adalah untuk memilih algoritma terbaik yang dapat memberikan jalur terpendek dengan waktu paling sedikit.

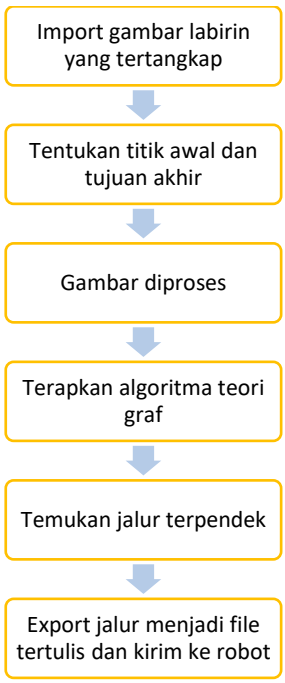
Breadth First Search adalah algoritma untuk mencari struktur data graf. Ini bisa digunakan untuk mencari sel labirin atau node. Tugas dari algoritma ini adalah untuk mencapai sel tujuan dari sel awal. Ini dimulai pada memulai sel labirin dan menjelajahi sel tetangga terlebih dahulu sebelum pindah ke tetangga tingkat berikutnya. Ini menggunakan penyimpanan biaya untuk menentukan urutan sel yang dikunjungi seperti yang ditunjukkan pada gambar 2.2 dan menyimpan catatan sel mana yang merupakan tetangga dari sel awal. Sel awal diberi label sebagai “a”. Algoritma ini memperluas sel agar jaraknya dari sel awal, menghasilkan satu tingkat pohon sekaligus. Ini melintasi pohon demi layer dengan membuat daftar simpul / sel untuk melintasi dan menambahkan anak-anak dari setiap simpul dalam daftar di akhir daftar yang dihasilkan. Sel ini berkembang terus sampai menemukan tujuan atau sel akhir. Dengan cara ini, BFS bisa menemukan jalur terpendek ke sel tujuan dan waktu yang dibutuhkan untuk menemukan jalur ini sebanding dengan jumlah sel yang dihasilkan karena masing-masing sel dapat dihasilkan dalam waktu konstan.

A * adalah algoritma pencarian yang bisa digunakan dalam pencarian jalan. A * mencari di antara semua jalur maze yang mungkin dari titik awal sampai titik akhir untuk menemukan jalur yang membutuhkan biaya terkecil sesuai dengan persamaan:

$$f(n) = g(n) + h(n) \quad (1)$$

Dimana n adalah simpul terakhir di jalan, h (n) adalah jarak ke setiap simpul dari simpul sasaran dan g (n) adalah biaya jalur dari simpul awal sampai n. A * dimulai dari simpul tertentu dari grafik dan membangun pohon jalur yang dimulai dari simpul tersebut. Ini terus memperluas jalan setapak satu per satu dengan mengambil simpul dengan nilai minimal fungsi evaluasi f (n), sampai salah satu jalurnya berakhir pada simpul tujuan yang telah ditentukan (lihat gambar 2.4).

Algoritma Best First Search juga bisa digunakan untuk menelusuri jalur labirin. Untuk menemukan jalur terpendek ke node tujuan target, pada setiap simpul dari grafik, ia mengevaluasi node mana yang akan dikembangkan selanjutnya berdasarkan biaya terendah dengan menggunakan fungsi evaluasi yang mengukur jarak dari simpul yang bersangkutan dengan sasaran seperti yang ditunjukkan pada gambar 2.3. Algoritma teori graf diterapkan pada citra biner labirin untuk menemukan jalur solusi terpendek antara titik awal dan titik akhir. Diagram alir kode Visual C ++ yang dikembangkan ditunjukkan pada gambar 3.4.



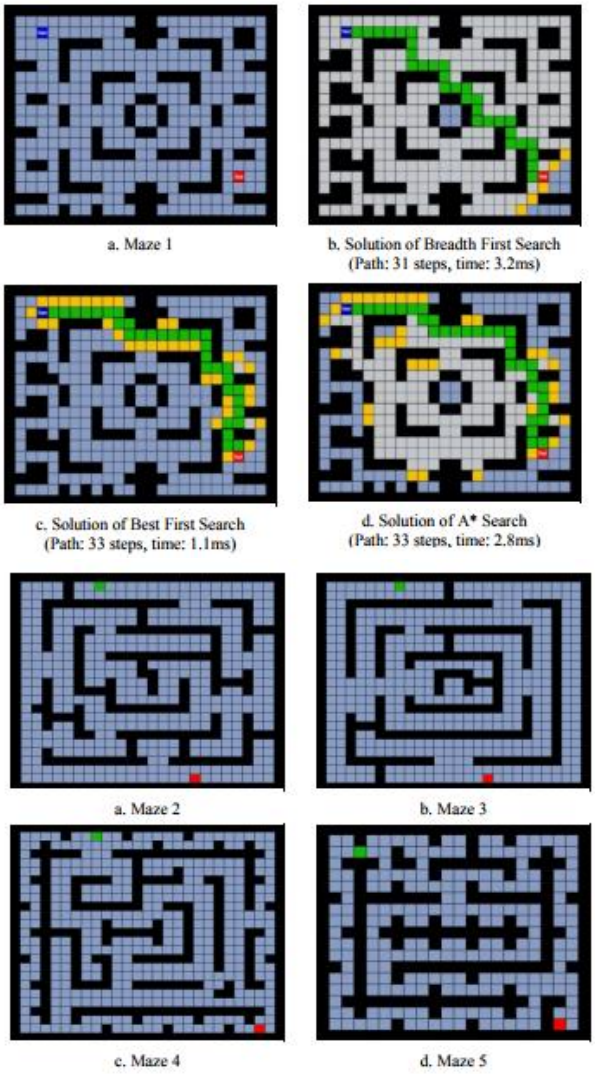
Gambar 3.4 Diagram Alir Program

D. Mengonversikan Jalur Terpendek dari Gambar yang ditangkap dan Dikirim ke Robot

Solusi jalur terpendek yang ditemukan dikonversi dari koordinat piksel (dalam piksel) ke koordinat dunia nyata (dalam cm) dengan menerapkan proses kalibrasi kamera menggunakan NI Vision Assistant Module yang didukung oleh LabVIEW. Jalur yang dibangun, diterjemahkan ke jalur panduan untuk dikirim ke robot melalui bluetooth. Jalur pemandu terdiri dari sekelompok instruksi urutan sebagai jarak terjemahan dalam skala cm dan gerakan (belok kanan, belok ke kiri, maju ke depan, dan gerakkan mundur). Akhirnya, petunjuk jalur panduan ini diexport ke teks file dan dikirim ke robot secara nirkabel melalui bluetooth untuk membimbingnya melalui labirin untuk mencapai tujuan akhir targetnya.

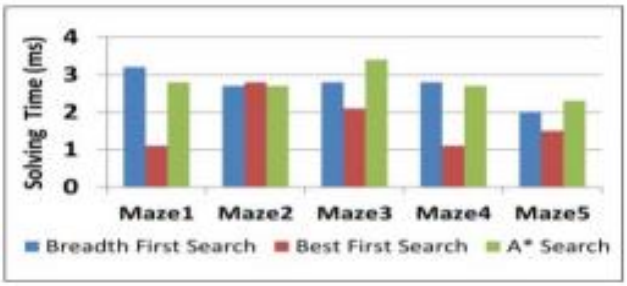
IV. EKSPERIMEN

Beberapa percobaan telah dilakukan untuk mengevaluasi keakuratan sistem robot pemecahan maze yang dikembangkan. Webcam Logitech C920 kelas enam konsumen kelas atas telah digunakan di sistem ini dan dipasang di atas labirin. Dalam percobaan yang dilakukan untuk penelitian ini robot Lego Mindstorms NXT2 digunakan.

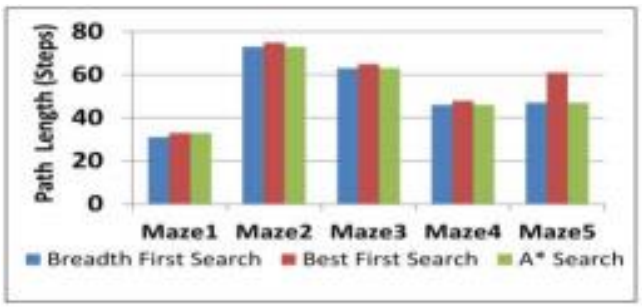


Gambar 4.1

Algoritma teori graf AI (Breadth First Search, Best First Search, A * search) diuji pada labirin berbeda seperti yang ditunjukkan pada gambar 4.1. Panjang jalur penyelesaian dan waktu penyelesaian dihitung dari titik awal sampai titik akhir setiap labirin untuk menentukan algoritma mana yang paling efisien dalam menemukan jalur terpendek. Gambar 4.1 mengilustrasikan hasil pengujian tiga algoritma pada labirin. Sel biru dan sel merah mewakili titik awal dan titik akhir; masing-masing. Sel hijau mewakili jalur solusi terpendek yang dihasilkan.



Gambar 4.2



Gambar 4.3

Berdasarkan gambar 4.2 dan 4.3, ditemukan bahwa Breadth First Search selalu memberikan jalan terpendek di semua labirin yang diuji dan pencarian Best First adalah yang tercepat. Akhirnya, karena faktor terpenting dalam sistem pemecahan labirin adalah menemukan jalur solusi terpendek, diputuskan untuk menerapkan algoritma Breadth First Search dalam program pemecahan labirin akhir karena ini adalah algoritma terbaik yang dapat menghasilkan jalur solusi terpendek dan memiliki trade-off yang bagus antara biaya waktu pemrosesan dan keandalan mencari solusi labirin.

Kode akhir ditulis berdasarkan algoritma Breadth First Search dan dikembangkan untuk dapat memproses dan menganalisis citra labirin yang ditangkap di dunia nyata. Kode yang dikembangkan diuji pada labirin nyata yang berbeda dengan kompleksitas varian, jalur solusi yang berbeda, dan bentuk dinding yang berbeda (dinding lurus dan dinding melengkung). Kode yang dikembangkan berhasil memecahkan labirin ini dengan tepat dan benar tanpa ada jalan yang hilang atau jalan yang lebih panjang. Ini selalu memberikan jalan solusi terpendek untuk setiap labirin yang diuji. Solusi jalur terpendek yang dihasilkan dikirim ke robot NXT2 Lego Mindstroms melalui bluetooth sebagai jalur pemandu bagi robot ke garis tujuan.

Pendekatan cerdas ini lebih praktis dibandingkan dengan metode pemecahan labirin tradisional (teknik wall-follower berdasarkan peraturan tangan kiri). Ditemukan bahwa

pendekatan cerdas yang diajukan berdasarkan algoritma Breadth First Search sejauh ini lebih baik daripada metode tradisional dalam hal panjang lintasan (jarak tempuh) dan waktu yang dikonsumsi untuk mencapai tujuan seperti yang diilustrasikan pada Tabel I.

Metode	Jarak menuju tujuan akhir (cm)	Waktu tempuh (s)
•Wall-follower •Acuan Teori Graf (BFS)	•387 •213	•50.2 •29.6

Tabel I Perbandingan Metode Tradisional (*wall-follower*) dan *Breadth First Search* Pada Salah Satu Labirin

Dari Tabel I, jelas bahwa robot dapat mencapai tujuannya dengan menggunakan pendekatan yang diusulkan dalam waktu yang lebih singkat dan jarak tempuh lebih dekat dibandingkan dengan metode tradisional. Itu karena metode tradisional mendorong robot mobile untuk bergerak melalui sel demi sel hingga menemukan titik tujuannya. Pendekatan teori graf memberikan sistem robot kesempatan untuk mempersiapkan jalur keliling dan menghindari perangkap dan terjatuh dalam lingkaran tanpa batas yang mungkin terjadi dengan teknik tradisional.

V. Kendala Umum

Kendala umum yang kerap kali dihadapi dalam penelusuran menggunakan robot ini adalah:

1. Pencahayaan yang kadang mempersulit robot mendeteksi seluruh labirin.
2. Terkadang terdapat masalah pada lantai yang berupa medan magnet yang acak.
3. Lantai yang tidak dijamin kebersihannya dari debu dapat mempengaruhi pembacaan sensor garis

VI. Kesimpulan

Graf dapat diimplementasi di berbagai bidang, salah satunya adalah bidang robotika. Graf dalam kasus ini membantu manusia untuk membuat strategi merancang algoritma pada robot sehingga robot pemecah labirin dapat menganalisis jalur terpendek. Untuk menemukan jalur terpendek, seluruh gambar labirin ditangkap oleh kamera kemudian diproses dan dianalisis oleh program yang dikembangkan berdasarkan algoritma Breadth First Search dengan acuan teori graf. Data jalur yang terpendek dikirim ke robot melalui bluetooth sebagai panduan jalan untuk mencapai tujuan sasarannya. Dengan teori graf, manusia dapat menciptakan robot cerdas yang mampu menyelesaikan persoalan - persoalan yang ada di sekitarnya dengan lebih mangkus.

VII. Ucapan Terima Kasih

Saya mengucapkan terima kasih kepada Allah SWT atas berkat rahmat-Nya lah saya dapat menyelesaikan makalah ini dengan sebaik mungkin. Kemudian terima kasih kepada Dr. Judhi Santoso M.Sc., Dra. Harlili S., M.Sc. dan Dr. Ir. Rinaldi Munir, M.T. selaku dosen mata kuliah IF 2120 Matematika Diskrit yang telah membimbing dan memberi materi kepada penulis selama proses pengajaran mata perkuliahan Matematika Diskrit serta telah memberikan tugas ini sebagai pemacu saya untuk dapat memberikan sesuatu yang bermanfaat bagi sesama umat manusia mengenai keilmuan informatika.

Terakhir, saya ucapkan terima kasih kepada ayah, ibu, dan teman-teman saya yang senantiasa menyemangati saya dalam menyelesaikan makalah ini serta telah memberikan masukkan kepada saya seputar makalah ini. Saya harap makalah ini dapat memanjakan mata para penikmat bacaan.

Referensi

[1] R. Munir, Matematika Diskrit, 3rd ed. Bandung: Penerbit INFORMATIKA Bandung, 2010, ch. 10.
[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
[4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.
[5] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.
[6] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.
[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.d
[8] http://belajarbikinrobot.weebly.com/1-mengenal-apa-iturobot.html diakses pada 1 Desember 2017 pada pukul 13.00
[9] http://www.instructables.com/id/Maze-Solver-Robot-Using-Artificial-Intelligence-Wi/ diakses pada 1 Desember 2017 pada pukul 14.00
[10] https://onbuble.wordpress.com/2011/05/26/6/ diakses pada 2 Desember 2017 pada pukul 17.00
[11] http://deisyamalia.blogspot.co.id/2012/03/best-first-search.html diakses pada 2 Desember 2017 pada pukul 08.00

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, 3 Desember 2017

Dimas Aditia Pratikto
13516153