

Algoritma Runut-Balik pada Robot Pemadam Api

Rakhmatullah Yoga Sutrisna (13512053)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

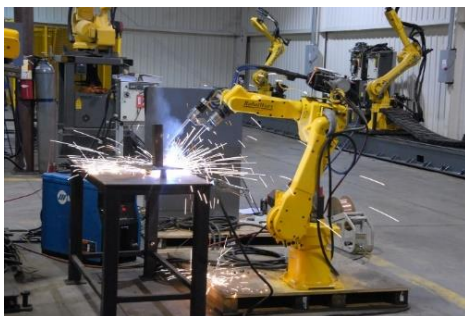
13512053@std.stei.itb.ac.id

Abstract—Robot dapat dimanfaatkan manusia untuk mempermudah pekerjaan manusia yang tergolong berat atau membahayakan. Salah satu contoh pemanfaatannya adalah robot pemadam api yang dapat menggantikan tugas manusia dalam pekerjaan pemadam kebakaran. Robot pemadam api bertugas mencari titik api dalam sebuah rumah kemudian memadamkannya. Untuk mengembangkan robot pemadam api, diselenggarakan kontes robot pemadam api agar pekerjaan tersebut dapat dibuat miniaturnya. Untuk dapat memadamkan api, robot harus menerapkan program yang efisien untuk mencari dan memadamkan api agar api tidak cepat membesar diakibatkan proses pencarian yang lama. Algoritma runut-balik dapat diimplementasikan untuk pencarian titik api yang terpusat di dalam suatu ruangan dalam sebuah rumah. Makalah ini akan membahas penggunaan algoritma runut-balik pada program robot pemadam api agar proses pencarian api menjadi lebih efisien.

Kata Kunci—Robot, Pemadam Api, KRPAI, Algoritma, Runut-Balik.

I. PENDAHULUAN

Kata robot diambil dari kata yang berasal dari kata robota, yang mempunyai arti pekerja, dipopulerkan oleh Isaac Asimov pada tahun 1950 dalam sebuah karya fiksinya. Robot sebagai salah satu produk teknologi terkini sering kali dimanfaatkan oleh manusia untuk menunjang kegiatannya sehari-hari, atau digunakan dalam aktivitas perindustrian misalnya produksi di dalam sebuah pabrik. Robot biasanya digunakan untuk tugas berat, bahaya, pekerjaan berulang dan kotor. Penggunaan lainnya termasuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, cari dan tolong, dan pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, penyedot debu, dan pendeteksi kebocoran gas [1].



Gambar 1. Salah satu robot yang digunakan untuk keperluan industri (sumber : <http://www.robots.com>)

Robotika dapat dimaknai sebagai ilmu yang mematerikan kecerdasan/intelegensia terhadap energi, artinya pengendalian secara cerdas terhadap gerakan yang terkoordinasi secara nyata. Istilah ini diperkenalkan oleh Isaac Asimov untuk pertama kalinya dalam cerita pendeknya yang berjudul Runaround yang terbit tahun 1942 [2].

Di zaman yang semakin maju ini, robot tidak hanya digunakan untuk mempermudah pekerjaan manusia. Di sisi lain, robot juga dapat digunakan sebagai ajang kreativitas kaum pelajar. Para pelajar biasanya tiap tahun dipertemukan dalam berbagai ajang perlombaan robot baik skala nasional maupun internasional. Mereka unjuk kebolehan dalam membuat dan merancang robot yang mempunyai berbagai tujuan dan mekanisme berbeda yang dipertandingkan.

II. KONTES ROBOT PEMADAM API INDONESIA

Di Indonesia, kontes robot sudah mulai diselenggarakan sejak tahun 1990. Kontes ini diberi nama Kontes Robot Indonesia. Ratusan mahasiswa dari berbagai perguruan tinggi di Indonesia telah meramaikan kontes ini dan beberapa diantaranya meraih gelar juara pada kontes ini. Perlombaan ini terdiri dari beberapa kategori robot yang dipertandingkan, antara lain Kontes Robot ABU Indonesia, Kontes Robot Sepakbola Indonesia, Kontes Robot Seni Indonesia, dan Kontes Robot Pemadam Api Indonesia. Kompetisi ini diadakan secara berjenjang dengan 2 tahap, yakni tahap regional dan tahap nasional.

Salah satu divisi yang dipertandingkan di Kontes Robot Indonesia adalah Robot Pemadam Api. Misi dari robot yang berkategori pemadam api ini adalah berlomba secepat mungkin untuk memadamkan api yang bersumber dari sebuah lilin yang terletak secara acak di dalam arena pertandingan yang menyerupai labirin (maze), kemudian kembali lagi pada tempat asal peletakan robot yang disebut home. Pada awalnya, robot akan diletakkan pada home yang terdiri dari arbitrary home dan non-arbitrary home. Selanjutnya robot akan mencari api ke setiap ruangan di dalam lapangan pertandingan. Pencarian api ini dilakukan dengan memanfaatkan sensor-sensor yang dipasang pada robot untuk mendeteksi jalan yang akan dilalui robot dan untuk mendeteksi keberadaan api di sekitar robot. Saat api telah ditemukan, robot akan memadamkan api tersebut dengan menggunakan alat yang diperbolehkan, misalnya dengan kipas atau extinguisher yang menyemprotkan

cairan yang tidak berbahaya.

Kontes Robot Pemadam Api Indonesia terdiri dari dua divisi, yaitu divisi beroda dan berkaki. Aspek yang membedakan kedua divisi tersebut adalah jenis aktuator atau penggerak robot tersebut dan durasi pertandingannya.

A. Spesifikasi Robot

Dimensi robot (panjang x lebar x tinggi) maksimum adalah:

Divisi Beroda: 31 cm x 31 cm x 27 cm
Divisi Berkaki: 46 cm x 31 cm x 27 cm

Bagian apapun dari robot dilarang melebihi dimensi tersebut pada kondisi apapun, baik waktu berhenti, berjalan, bermanuver, maupun pada saat meniup lilin. Pada divisi beroda robot menggunakan roda sebagai alat geraknya untuk berpindah tempat, sedangkan pada divisi berkaki, definisi secara detail mengenai kaki robot adalah sebagai berikut:

1. Yang dimaksud dengan kaki adalah suatu bagian robot yang bila bergerak dengan pola dan urutan tertentu bersama-sama dengan kaki-kaki lainnya, dapat menggerakkan dan memindahkan badan robot.
2. Hanya bagian dari kaki yang diperkenankan menempel di lantai ketika robot telah aktif dan ketika robot bergerak atau berjalan. Tidak ada bagian dari badan yang tidak masuk kedalam definisi kaki diperkenankan menempel di lantai misalnya penopang badan, caster dan sejenisnya.
3. Setiap kaki memiliki minimal dua derajat kebebasan dengan kata lain memiliki minimal dua sendi atau tegasnya setiap kaki memiliki minimal dua motor/aktuator.
4. Jumlah kaki minimal dua.
5. Satu kaki adalah independen satu sama lainnya, artinya, tidak ada 2 kaki atau lebih yang digerakkan oleh satu motor/aktuator.
6. Kaki tidak diperkenankan melakukan putaran 360 derajat (seperti prinsip roda berputar) untuk memindahkan badan.



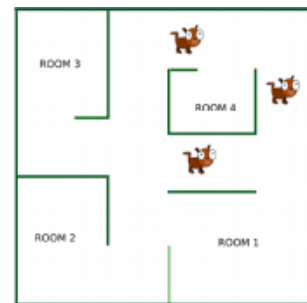
Gambar 2. Contoh bentuk robot divisi Berkaki (kiri) dan robot divisi Beroda (kanan)
(sumber : Panduan KRPAI 2013 Dikti)

Robot memiliki beberapa sensor seperti sensor suhu yang digunakan untuk mendeteksi keberadaan api, sensor cahaya yang digunakan untuk mendeteksi dinding-dinding lapangan pertandingan, dan sensor suara untuk mendeteksi sirine sebagai tanda dimulainya pertandingan. Selain itu robot juga memiliki sebuah sistem “minicomputer” yang dapat bekerja layaknya komputer biasa, sehingga robot dapat diprogram dengan bahasa pemrograman tertentu untuk melakukan berbagai pekerjaan sesuai input yang diterima oleh sensor pada robot.

B. Spesifikasi Lapangan Pertandingan

Lapangan/arena mensimulasikan interior dari sebuah rumah dengan 4 ruangan. Lapangan terbuat dari papan multipleks dengan ketebalan 1,8 s.d. 2 cm dan berukuran 248 cm x 248 cm x 30 cm. Di dalam lapangan terdapat 4 ruangan dengan posisi tetap namun dua diantaranya (ruang 1 dan 4, lihat gambar) memiliki pintu yang dapat digeser posisinya [3].

Selain pintu, terdapat pula halangan berupa boneka anjing (dog obstacle) yang bisa diletakkan di sembarang tempat sesuai undian. Boneka anjing ini memiliki bobot 500 gram, menutupi sekitar 50% hingga 75% lorong pada lapangan. Robot pemadam api tidak diperkenankan menerobos boneka anjing tersebut, jika robot melewati boneka anjing, maka dinyatakan gagal trial.



Gambar 3. Kemungkinan peletakan dog obstacle
(sumber : Panduan KRPAI 2013 Dikti)

Home yang menjadi tempat awal robot melakukan pencarian juga dapat diletakkan di berbagai tempat dalam lapangan. Secara garis besar jenis peletakan home dispesifikasi menjadi dua jenis, yaitu arbitrary home dan non-arbitrary home. Arbitrary home adalah home yang terletak di suatu ruangan yang bisa diibaratkan dengan sebuah ujung dari labirin yang buntu, sedangkan non-arbitrary home adalah home yang terletak di tengah sebuah lorong pada labirin sehingga robot mempunyai dua pilihan untuk menentukan arah awal pencarian api.

Kemungkinan orientasi robot di Home ada 6, ditandai dengan angka 1, 2, 3, 4, 5, dan 6 searah jarum jam yang merepresentasikan sudut 0°, 60°, 120°, 180°, 240° dan 300°. Home diletakkan di lapangan dengan cara sebagai berikut: ambil acuan garis putus-putus yang terdapat di tengah Home. Letakkan Home sedemikian sehingga garis putus-putus tersebut menghadap kearah “Utara” lapangan. Arah “Utara” yang dimaksud adalah arah yang sejajar dengan arah ruangan 2 ke ruangan 3. Sedangkan arah “1”

adalah arah yang digeser 15° terhadap arah “Utara” lapangan tersebut.



Gambar 4. Lapangan pertandingan KRP AI
(sumber : Panduan KRP AI 2013 Dikti)

Selain itu, lapangan pertandingan juga dilengkapi dengan halangan dan rintangan lain seperti uneven floor (lantai yang tidak rata), furniture seperti tiang, cermin yang digunakan untuk menguji sensor cahaya robot. Halangan dan rintangan tersebut tentunya akan memperlambat kinerja robot pemadam api, karena robot akan terhambat dalam perjalanan menelusuri lapangan pertandingan.

C. Peraturan Pertandingan KRP AI

1. Sesi

Satu Sesi (Trial) untuk divisi Beroda dan Berkaki adalah satu tahap pertandingan. Diberikan waktu maksimal 3 menit untuk divisi beroda dan 5 menit untuk divisi berkaki untuk bergerak dan bernavigasi di lorong atau ruangan dalam rangka mencari posisi lilin dan memadamkannya secepat cepatnya. Setelah memadamkan api, robot diberikan waktu 1 menit untuk divisi beroda dan 2 menit untuk divisi berkaki untuk kembali ke Home yang terhitung sejak api padam.

2. Retry

Retry adalah suatu upaya pengulangan Start didalam suatu SESI. Dalam setiap Sesi hanya diijinkan satu kali Retry.

Retry hanya boleh diajukan ke Jury bila robot gagal berfungsi misalnya: robot tertahan di dinding, robot terguling, robot “hang” (berputar terus, berjalan bolak-balik, dan lain-lain). Retry tidak boleh diajukan pada kondisi robot salah jalan atau pada kondisi tidak berhasilnya robot memadamkan api. Ketika Retry diajukan, peserta wajib menunggu ijin/keputusan Jury. Bila Retry diijinkan maka robot akan dibawa kembali ke Home tetapi stopwatch tidak dihentikan. Saat Retry peserta tidak diperkenankan menyentuh robotnya kecuali seijin Jury. Aktivasi robot saat Retry dilakukan oleh juri.

3. Pass

Pass adalah upaya pemberhentian Sesi oleh peserta. Pass dapat diajukan kapan saja. Pass bertujuan:

- Menyelamatkan robot dari kerusakan.
- Menghemat waktu pertandingan.
- Menjadi strategi peserta.

4. Penghentian Sesi

Sesi dihentikan bila:

- Robot telah menyelesaikan misi dan kembali ke Home.
- Tim pass.
- Robot tidak bergerak selama 30 detik, dan hak retry telah digunakan.
- Robot gagal padamkan api dalam waktu yang ditentukan.
- Juri lapangan menghendaknya dikarenakan adanya situasi penting dan mendesak.

III. ALGORITMA RUNUT-BALIK

Algoritma Runut Balik (*backtracking*) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih mangkus. Runut-balik, yang merupakan perbaikan dari algoritma brute-force, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Dengan metode ini, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan. Akibatnya, waktu pencarian dapat dihemat. Runut-balik lebih alami dinyatakan dalam algoritma rekursif. Kadang-kadang disebutkan pula bahwa runut-balik merupakan bentuk tipikal dari algoritma rekursif.

A. Properti Umum Algoritma Runut-balik

Untuk menerapkan metode runut-balik, properti berikut didefinisikan:

1. Solusi persoalan

Solusi dinyatakan sebagai vektor dengan n -tuple:

$$X = (x_1, x_2, \dots, x_n), x_i \in S_i.$$

Mungkin saja $S_1 = S_2 = \dots = S_n$.

Contoh: $S_i = \{0, 1\}$, $x_i = 0$ atau 1

2. Fungsi pembangkit nilai x_k

Dinyatakan sebagai predikat:

$$T(k)$$

$T(k)$ membangkitkan nilai untuk x_k , yang merupakan komponen vektor solusi.

3. Fungsi pembatas (pada beberapa persoalan fungsi ini dinamakan fungsi kriteria)

Dinyatakan sebagai predikat

$$B(x_1, x_2, \dots, x_k)$$

Fungsi pembatas menentukan apakah (x_1, x_2, \dots, x_n) mengarah ke solusi. Jika ya, aka pembangkitan nilai untuk x_{k+1} dilanjutkan, tetapi jika tidak, maka (x_1, x_2, \dots, x_n) dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

Fungsi pembatas tidak selalu dinyatakan sebagai fungsi matematis. Ia dapat dinyatakan sebagai predikat yang bernilai true atau false, atau dalam bentuk lain yang ekuivalen.

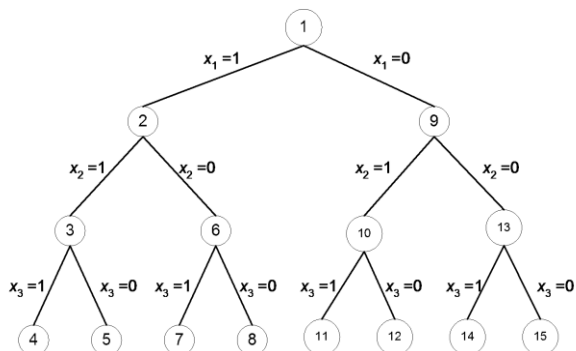
B. Pengorganisasian Solusi

Semua kemungkinan solusi dari persoalan disebut ruang solusi (solution space). Sebagai gambaran, tinjau persoalan

Knapsack 0/1 untuk $n = 3$. Solusi persoalan dinyatakan sebagai (x_1, x_2, x_3) dengan $x_i \in \{0,1\}$. Ruang solusinya adalah:

$$\{(0, 0, 0), (0, 1, 0), (0, 0, 1), (1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)\}$$

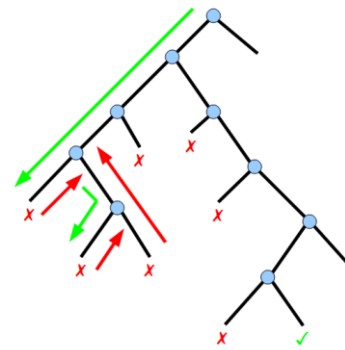
Algoritma runut-balik memperbaiki pencarian solusi secara exhaustive search dengan mencari solusi secara sistematis. Untuk memfasilitasi pencarian ini, maka ruang solusi diorganisasikan ke dalam struktur pohon. Tiap simpul pohon menyatakan status (state) persoalan, sedangkan sisi (cabang) dilabeli dengan nilai-nilai x_i . lintasan dari akar ke daun menyatakan solusi yang mungkin. Seluruh lintasan dari akar ke daun membentuk ruang solusi. Pengorganisasian pohon ruang solusi diacu sebagai pohon ruang status (state space tree). Tinjau kembali persoalan Knapsack 1/0 untuk $n = 3$. Ruang solusinya diorganisasikan sebagai pohon ruang status pada Gambar. Lintasan dari 1 sampai 4 misalnya, menyatakan slusi $X = (1,1,1)$. Perhatikanlah bahwa simpul-simpul diberi urutan nomor sesuai dengan prinsip pencarian DFS. Metode Runut-balik menerapkan DFS dalam pencarian solusi.



Gambar 5. Ruang solusi untuk persoalan Knapsack 0/1 dengan $n = 3$

C. Prinsip Pencarian Solusi Metode Runut-balik

Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti aturan *depth-first order* (DFS). Simpul-simpul yang sudah dilahirkan dinamakan **simpul hidup** (*live node*). Sedangkan simpul hidup yang sedang diperluas dinamakan **simpul-E** (*Expand-node*). Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut “dibunuh” sehingga menjadi **simpul mati** (*dead node*). Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan **fungsi pembatas** (*bounding function*). Simpul yang sudah mati tidak akan pernah diperluas lagi. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian *backtrack* ke simpul aras diatasnya. Lalu, teruskan dengan membangkitkan simpul anak yang lainnya. Selanjutnya simpul ini menjadi simpul-E yang baru. Pencarian dihentikan bila kita telah sampai pada *goal node*.



Gambar 6. Alur pencarian solusi dengan menggunakan algoritma runut-balik

D. Skema Umum Algoritma Runut-Balik

Berikut ini adalah skema umum algoritma runut-balik dalam dua versi, yakni versi rekursif dan versi iterative. Skema dalam versi rekursif lebih tepat karena algoritma runut-balik lebih alami dinyatakan dalam bentuk rekursi. Algoritma di bawah ini akan menghasilkan semua solusi.

(a) Versi rekursif

```

procedure RunutBalikR(input k:integer)
{Mencari semua solusi persoalan dengan metode runut-balik; skema rekursif
Masukan: k, yaitu indeks komponen vektor solusi, x[k]
Keluaran: solusi x = (x[1], x[2], ..., x[n])
}
Algoritma:
  for tiap x[k] yang belum dicoba sedemikian sehingga
    ( x[k]←T(k) and B(x[1], x[2], ... ,x[k]) = true) do
    if (x[1], x[2], ... ,x[k]) adalah lintasan dari akar ke daun then
      CetakSolusi(x)
    endif
    RunutBalikR(k+1) {tentukan nilai untuk x[k+1]}
  endfor

```

(b) Versi iteratif

```

procedure RunutBalikI(input n:integer)
{Mencari semua solusi persoalan dengan metode runut-balik; skema iteratif
Masukan: n, yaitu indeks komponen vektor solusi
Keluaran: solusi x = (x[1], x[2], ..., x[n])
}
Deklarasi:
  k : integer
Algoritma:
  k ← 1
  while k > 0 do
    if (x[k] belum dicoba sedemikian sehingga x[k]
      <- T(k) and (B(x[1], x[2], ..., x[n]) = true) then
      if (x[1], x[2], ..., x[k]) adalah lintasan dari akar ke daun then
        CetakSolusi(x)
      endif
      k ← k + 1 {indeks anggota tuple berikutnya}
    else {x[1], x[2], ..., x[k] tidak mengarah ke simpul solusi}
      k ← k - 1 {runut-balik ke anggota tuple sebelumnya}
    endif
  endwhile { k = 0 }

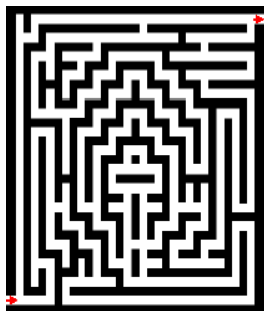
```

Setiap simpul dalam pohon ruang status berasosiasi dengan sebuah pemanggilan rekursif. Jika jumlah simpul dalam pohon ruang status adalah 2^n atau $n!$, maka untuk

kasus terburuk, algoritma runut-balik membutuhkan waktu dalam $O(p(n)2^n)$ atau $O(q(n)n!)$, dengan $p(n)$ dan $q(n)$ adalah polinom derajat n yang menyatakan waktu komputasi setiap simpul.

E. Aplikasi Algoritma Runut-balik

Algoritma runut-balik banyak digunakan pada program permainan (games). Salah satu permainan yang dibahas di dalam kuliah ini adalah mencari jalan keluar di dalam labirin (Maze Problem). Labirin adalah jaringan jalan yang ruwet dan berliku-liku. Persoalan labirin adalah menentukan lintasan (path) dari pintu masuk menuju pintu keluar labirin.



Gambar 7. Sebuah labirin

Dengan algoritma runut-balik kita mencoba sebuah lintasan hingga menemui jalan buntu, lalu jejak (retrace) langkah sebelumnya sampai kita menemukan yang mengarah ke pintu keluar, atau mencoba semua lintasan dan memutuskan tidak terdapat solusinya.

Untuk menggambarkan algoritma runut-balik secara rinci, kita membagi lintasan menjadi sederetan langkah. Sebuah langkah terdiri dari pergerakan satu unit sel pada arah tertentu. Arah yang mungkin hanya empat yaitu ke atas (up), ke bawah (down), ke kiri (left), dan ke kanan (right). Jadi, algoritma runut-baliknya secara garis besar adalah:

```

while belum sampai pada tujuan do
  if terdapat arah yang benar sedemikian sehingga kita
  belum pernah berpindah ke sel pada arah tersebut
  then
    pindah satu langkah ke arah tersebut
  else
    backtrack langkah sampai terdapat arah seperti
    yang disebutkan di atas
  endif
endwhile

```

Masalahnya di sini, bagaimana kita tahu langkah yang mana yang perlu dijejaki kembali? Ada dua solusi untuk masalah ini, pertama, simpan semua langkah yang pernah dilakukan, atau kedua, gunakan rekursi (yang secara implisit menyimpan semua langkah). Rekursi adalah solusi yang lebih mudah.

```

function SolveMaze(input M : labirin)→boolean
{ true jika pilihan mengarah ke solusi }

Deklarasi
  arah : integer { up = 1, down, 2, left = 3,
                 right = 4 }

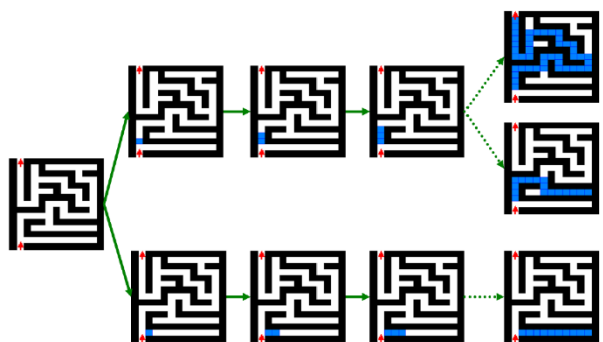
```

```

Algoritma:
  if pilihan arah merupakan solusi then
    return true
  else
    for tiap arah gerakan (lurus, kiri, kanan) do
      move(M, arah) { pindah satu langkah (satu sel) sesuai arah tersebut }
      if SolveMaze(M) then
        return true
      else
        unmove(M, arah) { backtrack }
      endif
    endfor
  return false { semua arah sudah dicoba, tetapi tetap buntu, maka kesimpulannya: bukan solusi }
endif

```

Jika kita menggambarkan sekuens pilihan yang kita lakukan, maka diagram akan berbentuk seperti pohon. Simpul daun merupakan titik runut-balik (backtrack) atau simpul goal. Pada titik *backtrack*, simpul tersebut menjadi mati (tidak bisa diekspansi lagi). Pembangkitan pohon ruang status ini menggunakan aturan DFS (Depth First Search).



Gambar 8. Contoh runut-balik pada sebuah labirin

IV. PENGGUNAAN ALGORITMA RUNUT-BALIK

Pada umumnya, robot KRPAI hanya mengandalkan hasil pengolahan sensor untuk menentukan jalan pencarian api dalam lapangan pertandingan KRPAI. Hal ini menyebabkan robot cenderung berjalan secara tidak teratur untuk mencari api, bahkan robot mempunyai kemungkinan untuk menelusuri beberapa jalur yang sama. Hal ini disebabkan karena robot tidak memanfaatkan mekanisme memorisasi (hafalan) terhadap langkah-langkah serta ruangan yang telah dilalui oleh robot tersebut. Mekanisme pencarian api seperti ini sangat tidak menguntungkan, terlebih apabila hal seperti ini terjadi pada kasus kebakaran pada kenyataannya, apabila pencarian sumber api berlangsung dalam waktu yang terlalu lama, maka dikhawatirkan api akan semakin membesar dan kerugian maupun korban yang diakibatkan oleh kebakaran ini akan semakin membesar.

Robot yang digunakan dalam perlombaan KRPAI sebenarnya memiliki kemampuan untuk menyimpan data tempat-tempat yang terdapat pada lapangan pertandingan serta langkah-langkah yang telah dilaluinya. Selain itu bentuk lapangan pertandingan KRPAI yang menyerupai sebuah labirin (maze), memungkinkan aplikasi algoritma runut-balik untuk permainan maze dapat diterapkan untuk

pencarian api pada lapangan pertandingan KRPAI. Robot dapat menerapkan algoritma runut-balik dengan alternatif gerak berupa maju, belok kanan, belok kiri, atau mundur hingga ditemukan ruangan tempat api dari lilin menyala.

Berikut adalah gambaran algoritma runut-balik pencarian api oleh robot KRPAI yang diberikan dalam notasi pseudo code.

```
procedure MencariApi(input M : labirin, input/output
found : boolean)
{ nilai found awal false karena api belum ditemukan }

Deklarasi
arah : integer { lurus = 1, belok kanan = 2,
                belok kiri = 3 }

Algoritma:
telusuri lorong
if menemukan titik api then
  found <- true
  padamkan api
else
  while masuk tiap arah and not found do
    maju(M, arah) { pindah satu langkah sesuai
                  arah tersebut }
    MencariApi(M, found)
  endwhile
endif
mundur(M, arah)
```

Mula-mula saat robot diletakkan di posisi home, robot akan mencatat posisi homenya sebagai akar dari pohon ruang status pencarian api. Selanjutnya robot akan bergerak menelusuri lorong-lorong yang terdapat pada lapangan pertandingan hingga robot menemui persimpangan lorong. Saat berada di persimpangan lorong, robot akan mengambil keputusan untuk memasuki salah satu lorong dan akan mengulang kembali proses pemilihan lorong secara iteratif untuk tiap pilihan lorong yang akan dimasuki jika api belum ditemukan. Jika robot sampai pada sebuah ruang yang buntu dan tidak terdapat api atau saat menemukan dog obstacle, robot akan mengembalikan perintah mundur menuju simpul persimpangan lorong sebelumnya dan fungsi rekursi akan berakhir dan selanjutnya kembali ke iterasi pilihan lorong. Ketika robot menemukan api di dalam suatu ruangan, robot akan mengubah nilai penemuan api menjadi true, kemudian melakukan pemadaman api hingga api tidak terdeteksi lagi dalam ruangan tersebut. Setelah itu semua fungsi rekursif dan iteratif akan berakhir sehingga robot akan bergerak mundur secara berulang-ulang hingga robot kembali pada tempat semulanya yaitu di home.

Pada perlombaan KRPAI, durasi waktu yang diberikan sangat terbatas untuk pencarian api dan perjalanan robot kembali ke home. Penggunaan algoritma ini jelas sangat efisien dibandingkan dengan mekanisme pencarian tanpa menggunakan memorisasi. Robot yang tidak memanfaatkan strategi memorisasi ini sering kali mengalami kegagalan dalam misi memadamkan api.

V. SIMPULAN

Algoritma runut-balik dapat diterapkan untuk strategi perlombaan pada pertandingan KRPAI. Keunggulan algoritma ini adalah robot tidak perlu memakan waktu

lama karena harus mencoba seluruh kemungkinan jalan secara tidak teratur. Algoritma runut-balik juga akan memangkas waktu perjalanan robot untuk kembali ke home karena hanya perlu memanfaatkan algoritma pergerakan yang dilakukan secara rekursif.

REFERENSI

- [1] <http://www.eyuana.com/2012/08/sejarah-robot-dan-pengertian-tentang.html>, diakses pada tanggal 12 Mei 2014 pukul 21.27 WIB.
- [2] <http://astridazarine.blogspot.com/2013/07/definisi-robotrobotikrobotics.html>, diakses pada tanggal 17 Mei 2014 pukul 08.50 WIB.
- [3] Direktorat Penelitian dan Pengabdian kepada Masyarakat – Dikti. Panduan Kontes Robot Pemadam Api Indonesia (KRPAI) Beroda dan Berkaki 2013.
- [4] Munir, Rinaldi. Diktat Kuliah IF3051 Strategi Algoritma. Informatika Bandung, 2009, hal. 125-149.

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, 19 Mei 2014



Rakhmatullah Yoga Sutrisna
13512053