Aplikasi Algoritma Backtracking dalam Manajemen Sumber Daya Manusia pada Event Organizer Mahasiswa

Annisa Sekar Ayuningtyas - 13516044

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13516044@std.stei.itb.ac.id

*Abstrak*—Salah satu hal yang menjadi penentu keberhasilan suatu *event organizer* adalah kuantitas dan kualitas SDM yang bekerja di dalamnya. Namun pada pelaksanaannya, jumlah SDM yang hadir tidak seperti yang sudah direncanakan, dan ada pula SDM dari bagian yang lain yang hadir untuk sukarela membantu. Jika pada *event organizer* tersebut terdapat sistem pendataan, maka SDM yang sukarela membantu bisa dialokasikan ke bagian yang membutuhkan. Supaya pengalokasiannya merata, dapat digunakan algoritma pencarian *backtracking* untuk mendapatkan bagian mana yang paling membutuhkan SDM. Algoritma *backtracking* adalah algoritma pencarian berbasis pencarian mendalam yang merupakan bentuk mangkus dari algoritma *brute force.*

*Kata kunci*—*event organizer,* SDM, *backtrack*.

# I. Pendahuluan

Dalam suatu *event organizer* dalam skala besar, pastinya jumlah orang yang berpartisipasi di dalamnya tidak sedikit. Dalam pelaksanaannya juga dibutuhkan massa yang banyak. Namun, pada *event organizer* mahasiswa ini terdapat beberapa hal yang menjadi kendala dalam menyelenggarakan suatu acara, diantaranya beratnya beban akademik, konflik antar panitia, dan jumlah partisipan yang kurang. Seringkali jumlah massa yang hadir tidak sesuai dengan yang diharapkan. Selain itu ada juga yang hadir namun pada saat itu tidak memiliki tugas yang harus dikerjakan.

Salah satu hal yang sekiranya bisa menjadi solusi dari permasalahan manajemen SDM saat pelaksanaan acara yaitu dengan pendataan jumlah kebutuhan SDM serta pengalokasian SDM yang tepat. Supaya pengalokasian tersebut tepat sasaran, maka dapat diaplikasikan algoritma *backtracking.*

# II. Landasan Teori

1. *Event Organizer* pada Mahasiswa

*Event Organizer* atau yang sering disebut EO adalah penyedia jasa profesional pengelola suatu kegiatan (Pengorganisir Acara). Setiap kegiatan yang di selenggarakan bertujuan untuk memperoleh keuntungan di kedua belah pihak, baik penyelenggara maupun yang hadir pada saat kegiatan berlangsung. Keuntungan ini tidak harus bersifat material namun juga bisa bersifat non material.

Jenis-jenis *event organizer* adalah sebagai berikut :

1. Private Party : EO ini bergerak untuk penyelenggaraan pesta pribadi terutama bagi kalangan atas.
2. Penyelenggara Ulang Tahun : dari namanya bisa diketahui bahwa EO ini ahlinya membuat pesta ulang tahun termasuk untuk anak-anak.
3. Wedding Organizer : Ini adalah EO yang menyelenggarakan acara pesta pernikahan.
4. Musik dan Hiburan : EO yang menyelenggarakan acara di bidang hiburan terutama musik.
5. Brand Activation : EO yang secara spesifik membantu klien mempromosikan *brand* dalam rangka peningkatan penjualan.
6. MICE : Singkatan dari *meeting, incentive, convention dan exhibition* yang khusus bergerak di bidang penyelenggara acara berbentuk pertemuan.
7. One Stop Service Agency : EO yang mampu menyelenggarakan berbagai jenis acara besar hingga berskala internasional.

Jenis-jenis tersebut adalah jenis-jenis *event organizer* pada umumnya. Dan kebanyakan dari *event organizer* tersebut bergerak untuk bisnis mereka.

Sedangkan untuk pelajar, baik mahasiswa, SMA, dan yang lainnya tidak kalah uniknya dalam mempersiapkan suatu acara. Banyak sekolah yang melatih murid-muridnya untuk bisa mengadakan suatu acara besar. Dan tentu saja di perguruan tinggi dapat ditemukan kepanitiaan dimana-mana. Perbedaan dengan *event organizer* biasanya adalah, dalam skala mahasiswa atau tingkat dibawahnya, tujuan mereka mengikuti kepanitiaan adalah untuk melatih *softskill* mereka dalam menjalin hubungan dengan orang lain, manajemen diri, dan masih banyak alasan lainnya. Dan mayoritas *event organizer* dalam skala mahasiswa tidak mendapatkan bayaran.

Kegiatan yang diselenggarakan oleh mahasiswa tentunya berbeda dengan jenis *event organizer* pada umumnya seperti yang telah dijelaskan. Kegiatan tersebut biasanya bersifat hiburan, edukasi, atau pelayanan. Kegiatan yang bersifat hiburan misalnya mengadakan pentas seni, pagelaran budaya, atau konser dengan mengundang beberapa tokoh idola. Kegiatan edukasi misalnya mengadakan seminar dengan pembicara dari tokoh-tokoh yang sudah ahli di suatu bidang, *talkshow, training,* atau kegiatan pengabdian sosial dengan mengajar ke sekolah-sekolah di pelosok. Sedangkan kegiatan yang bersifat pelayanan lebih banyak dilakukan pada kepanitiaan yang berbasiskan keagamaan.

Pada *event organizer* mahasiswa, jenis kepanitiaan sering juga dibagi menjadi dua, yaitu :

1. Panitia Non-Lapangan

Bagian ini adalah yang mendukung kesuksesan suatu acara, baik dari sisi keuangan, manajemen sumber daya manusia, kreatif, dan lain-lain.

1. Panitia Lapangan

Bagian ini yang turun langsung ke lapangan, berinteraksi dengan sasaran, dan berkaitan sekali dengan teknis.

1. Kendala-Kendala dalam *Event Organizer* Mahasiswa

Acara yang diselenggarakan oleh *event organizer* memang bermacam-macam dan prosesnya juga beragam. Namun dari seluruh proses tersebut dapat dibagi menjadi tiga tahap, yaitu :

1. Tahap perencanaan

Dalam membuat sebuah acara, tentu perencaan adalah hal yang utama. Merancang acara dengan *What, When, Where, Who, Why dan How*. Berdiskusi dengan anggota lainnya acara akan seperti apa, kapan dan dimana tempatnya, siapa saja yang menjadi target audiens, mengapa acara tersebut harus diselenggarakan dan bagaimana cara mengemas acara tersebut dengan baik. Pembuatan organogram dan pemilihan orang-orang yang bertanggung jawab pada suatu divisi atau bidang juga termasuk dalam fase ini, namun urutan pelaksanaannya berbeda-beda. Fase ini menghabiskan banyak waktu untuk berdiskusi dan mencari inspirasi, misalnya seperti bagaimana ide dan konsep dari acara, juga perencanaan biaya. Inti dari tahap ini adalah merancang acara tersebut agar sukses dan tercapai tujuannya.

1. Tahap persiapan

Tahap persiapan ini adalah melaksanakan segala hal sebelum hari penyelenggaraan untuk mencapai tujuan. Pada mahasiswa, tahapan ini seringkali tidak sesuai dengan perencanaan awal karena jadwal akademik yang padat. Namun disinilah awal perkembangan *softskill* mahasiswa dalam manajemen waktu. Pada tahap ini biasanya *event organizer* mencari sponsor, membuat desain untuk acara tersebut, dan melakukan persiapan lainnya.

1. Tahap pelaksanaan dan setelahnya

Bagian ini adalah saat suatu acara mulai dilaksanakan sampai kepanitiaan dibubarkan, termasuk evaluasi didalamnya. Tahap ini juga sering tidak sesuai dengan perencanaan, ditambah lagi jika persiapan banyak terkendala. Tahap pelaksanaan bisa jadi membutuhkan SDM yang banyak, namun bisa juga tidak. Dan hal itu juga yang sering menjadi masalah dalam *event organizer* pada mahasiswa.

Berdasarkan pengalaman penulis, kendala yang sering dialami pada *event organizer* mahasiswa yaitu :

1. Beratnya beban akademik, bagian ini memang tidak bisa dihindari karena kewajiban utama mahasiswa adalah belajar.
2. Konflik antar anggota. Bagian ini juga bisa menghambat persiapan suatu acara, bahkan pelaksanaan acara
3. Kurangnya jumlah SDM saat pelaksanaan acara. Bagian ini juga yang menjadi hal yang krusial. Tingginya minat mahasiswa dalam berorganisasi menjadikan mereka mengikuti kepanitiaan di berbagai tempat, dan bisa jadi mereka tidak ada saat dibutuhkan karena kegiatan yang lain.
4. Algoritma Backtracking

Backtracking adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih mangkus. Algoritma ini merupakan perbaikan dari algoritma *brute force (exhaustive search).* Algoritma *exhaustive search* memeriksa seluruh kemungkinan yang ada, namun algoritma *backtracking* ini tidak. Pada algoritma ini, yang ditelusuri hanyalah pilihan yang mengarah kepada solusi sehingga penggunaan waktu lebih dihemat.

Sebagai salah satu algoritma yang banyak digunakan oleh para programmer. *Backtracking* merupakan sebuah alat yang penting untuk dapat menyelesaikan permasalahan pemenuhan terbatas, seperti teka – teki silang, aritmatika verbal, sudoku dan berbagai macam puzzle sejenisnya. Algoritma ini juga dapat digunakan untuk menyelesaikan permasalahan komputasional seperti memecahkan kata sandi atau password pada suatu program, membuat sistem kerja atau mekanisme kerja dari suatu video game, ataupun sistem dasar dari suatu simulasi komputer terhadap permasalahan di dunia nyata.

Atribut dari algoritma backtracking adalah :

1. Solusi persoalan.

Solusi dinyatakan sebagai vektor dengan n buah tuple:

X = (x1 , x2 , …, xn ), xi ∈ Si

Mungkin saja S1 = S2 = … = Sn

Contoh: Si = {0, 1}, xi = 0 atau 1

1. Fungsi pembangkit nilai xk

Dinyatakan sebagai predikat: T(k)

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

1. Fungsi pembatas

Dinyatakan sebagai predikat B(x1 , x2 , …, xk )

B bernilai true jika (x1 , x2 , …, xk ) mengarah ke solusi.

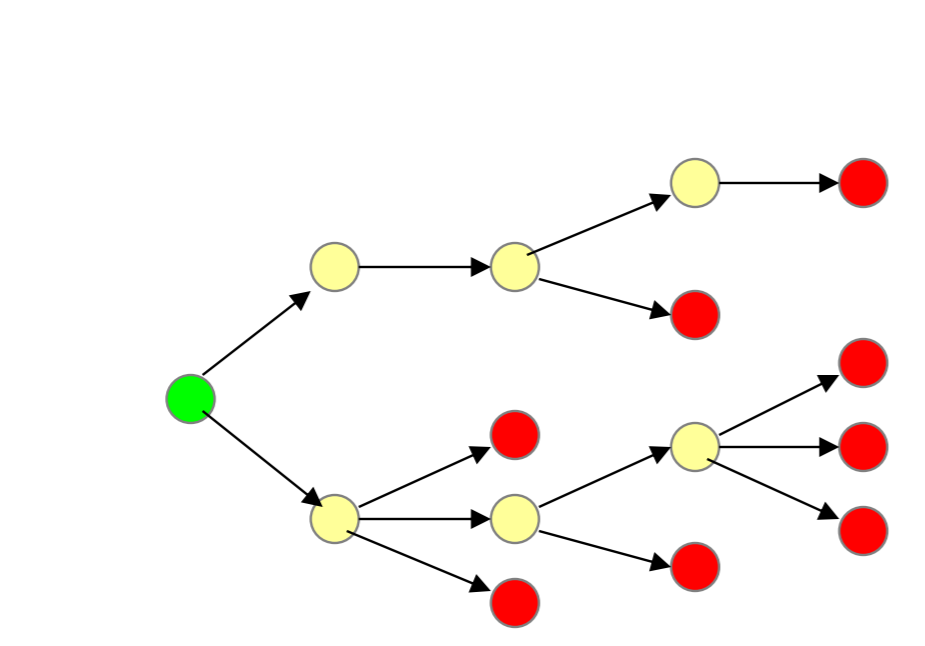
Jika true, maka pembangkitan nilai untuk xk+1 dilanjutkan, tetapi jika false, maka (x1 , x2 , …, xk ) dibuang.

1. Pengorganisasian Solusi

Cara untuk mendaptkan solusi dengan menggunakan atribut-atribut yang telah disebutkan sebelumnya adalah :

1. Semua kemungkinan solusi dari persoalan disebut ruang solusi (solution space). Secara formal dapat dinyatakan, bahwa jika xi ∈ Si maka S1 x S2 x … x Sn disebut ruang solusi. Jumlah anggota didalam ruang solusi adalah |S1| . |S2|. … . |Sn|
2. Sebagai gambaran, tinjau persoalan Knapsack 0/1 untuk n = 3.
3. Solusi persoalan dinyatakan sebagai (x1 , x2 , x3 ) dengan xi ∈ {0,1}.
4. 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) }
5. Ruang solusi diorganisasikan ke dalam struktur pohon.
6. Tiap simpul pohon menyatakan status (state) persoalan, sedangkan sisi (cabang) dilabeli dengan nilai-nilai xi.
7. Lintasan dari akar ke daun menyatakan solusi yang mungkin.
8. Seluruh lintasan dari akar ke daun membentuk ruang solusi.
9. Pengorganisasian pohon ruang solusi diacu sebagai pohon ruang status (state space tree).

Sebuah pohon adalah sekumpulan simpul dan busur yang tidak mempunyai sirkuit.

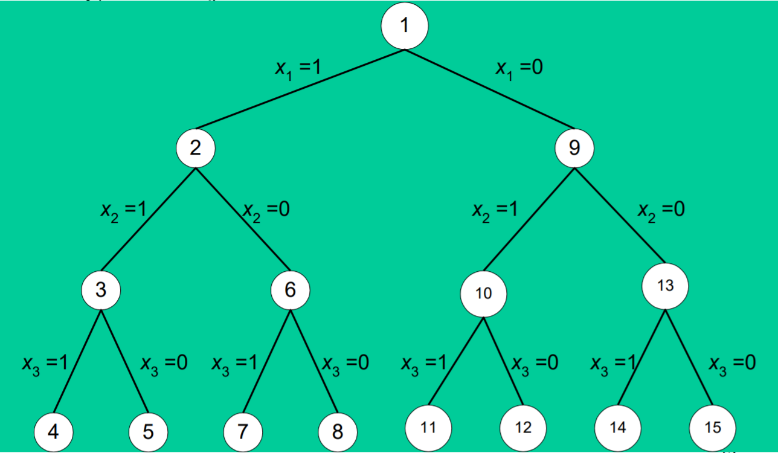


Gambar 1 Ilustrasi Pohon

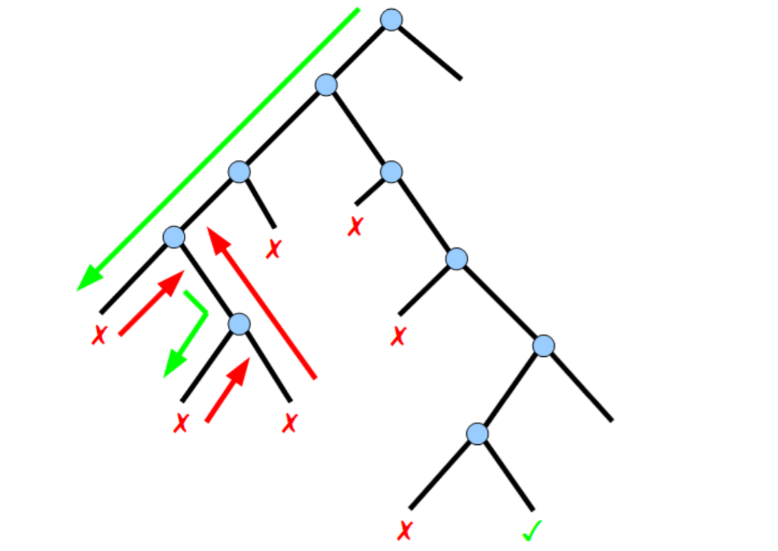
Ada tiga macam simpul pada pohon, yaitu :

1. Simpul akar, yaitu yang berwarna hijau. Satu pohon hanya memiliki satu simpul akar.
2. Simpul dalam, yaitu yang berwarna kuning. Simpul dalam adalah simpul yang terletak ditengah, ia mempunyai anak dan orang tua.
3. Simpul daun, yaitu yang berwarna merah. Simpul daun adalah simpul yang terletak di ujung. Ia adalah anak, karena hanya mempunyai orang tua.

Backtracking dapat dipandang sebagai pencarian di dalam pohon menuju simpul daun (goal) tertentu.

Dengan meninjau persoalan knapsack 1/0 untuk persoalan n = 3, maka didapatkan ruang solusi sebagai berikut :

Gambar 2 Ruang Solusi

1. Prinsip Pencarian Solusi dengan Algoritma *Backtracking*
2. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti aturan pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan **simpul hidup** (live node). Simpul hidup yang sedang diperluas dinamakan **simpul-E** (Expand node). Simpul dinomori dari atas ke bawah sesuai dengan urutan kelahirannya.
3. Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang.
4. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut “dibunuh” sehingga menjadi **simpul mati** (dead node).
5. Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan **fungsi pembatas** (bounding function).
6. Simpul yang sudah mati tidak akan pernah diperluas lagi.
7. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian backtrack ke simpul aras diatasnya.
8. Lalu, teruskan dengan membangkitkan simpul anak yang lainnya.
9. Selanjutnya simpul ini menjadi simpul-E yang baru.
10. Pencarian dihentikan bila kita telah sampai pada goal node atau tidak ada lagi simpul hidup yang bisa digunakan untuk *backtracking.*

Gambar 3 Ilustrasi Kerja Algoritma Backtracking

1. Skema Umum Algoritma *Backtracking*

Dibawah ini adalah skema umum algoritma *backtracking* dengan dua versi, yaitu rekursif dan iterative. Dua-duanya menghasilkan solusi yang sama, namun algoritma *backtracking* lebih tepat diterapkan dengan rekursif.

1. 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)) dan 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

1. Versi Iteratif

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])

}

Deklarasi :

K : integer

Algoritma :

K 🡨 1

while k > 0 do :

if (x[k] belum dicoba sedemikian sehingga

(x[k] 🡨 T(k)) dan B(x[1], x[2], ... ,x[k])=

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}

Prosedur CetakSolusi adalah sebagai berikut :

procedure CetakSolusi(input x : TabelInt)

{Mencetak solusi persoalan

Masukan : x[1], x[2], … , x[n]

Keluaran : nilai x[1], x[2], … , x[n] tercetak ke

layar}

Deklarasi :

k : integer

Algoritma :

for k🡨1 to n do

write(x[k])

endfor

Setiap simpul dalam pohon ruang status berasosiasi dengan sebuah pemanggilan rekursif. Jika jumlah simpul dalam pohon ruang status adalah **2n** atau **n!**, maka untuk kasus terburuk, algoritma runut-balik membutuhkan waktu dalam **O(p(n)2n)** atau **O(q(n)n!)**, dengan p(n) dan q(n) adalah polinom derajat n yang menyatakan waktu komputasi setiap simpul.

1. Jenis *Event Organizer* yang Sesuai dengan Penerapan Algoritma *Backtracking*

Seperti yang sudah dijelaskan sebelumnya, *event organizer* secara umum dan *event organizer* pada mahasiswa memiliki perbedaan. Dalam kegiatannya, *event organizer* pada mahasiswa terlihat lebih beragam, bahkan kegiatan malam keakraban pun ada *event organizer* nya walaupun tidak terlalu besar dan formal.

Berdasarkan penjelasan pada sebelum-sebelumnya, algoritma *backtracking* ini dapat membantu mengatasi masalah sumber daya manusia pada tahap pelaksanaan. Algoritma ini dapat digunakan dalam manajemen SDM untuk pemerataan alokasi SDM pada bagian yang membutuhkan. Sistem ini dapat diterapkan pada *event organizer* yang membutuhkan banyak SDM pada pelaksanaan acara. Selain itu bisa juga diterapkan pada *event organizer* yang membuat banyak acara dalam rentang waktu tertentu. Dan yang paling penting untuk menerapkan sistem ini adalah adanya pendataan setiap dilaksanakan suatu acara. Karena sistem ini tidak akan bisa berjalan tanpa pendataan yang jelas.

# III. Aplikasi Algoritma *Backtracking* dalam Manajemen Sumber Daya Manusia pada *Event Organizer Mahasiswa*

1. Sistem Pendataan

Sebelum menerapkan algoritma *backtracking,* perlu data yang jelas terlebih dahulu. Misalkan suatu panitia akan melaksanakan acara bazar bukudan bagi-bagi makanan dalam hari yang sama. Maka didapatkan data yang jelas, yaitu sebagai berikut :

Bazar buku :

1. Lapangan (1)

Butuh : 20

Hadir : 17

PJ : A (08xxxxxxxxxx)

1. Logistik (2)

Butuh : 10

Hadir : 4

PJ : B (08xxxxxxxxxx)

Bagi-bagi makanan :

1. Lapangan (3)

Butuh : 30

Hadir : 12

PJ : C (08xxxxxxxxxx)

1. Dokumentasi (4)

Butuh : 5

Hadir : 4

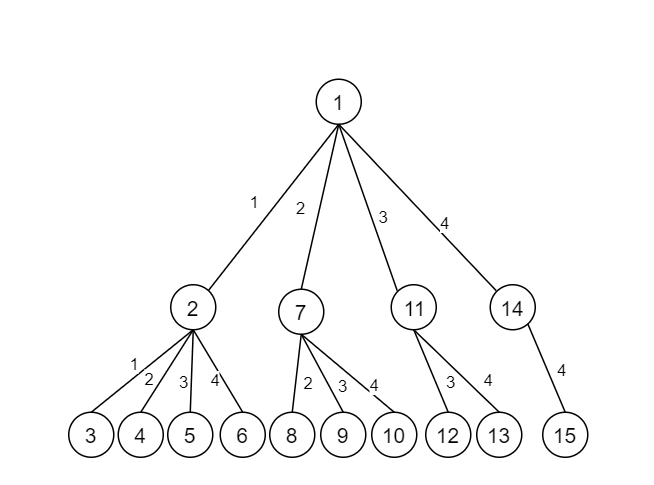
PJ : D (08xxxxxxxxxx)

1. Aplikasi Algoritma Backtracking

Pada kali ini akan digunakan algoritma *backtracking* dalam mencari nilai terbesar. Nilai terbesar yang dimaksud adalah nilai terbesar dari n kali penjumlahan dari jumlah SDM yang belum terpenuhi. n adalah jumlah orang yang hadir dan belum memiliki tugas. Berdasarkan jumlah diatas, maka SDM yang dibutuhkan adalah :

1. 1 : 3
2. 2 : 6
3. 3 : 18
4. 4 : 1

Misalkan ada dua orang yang datang, maka ruang solusinya sebagai berikut :

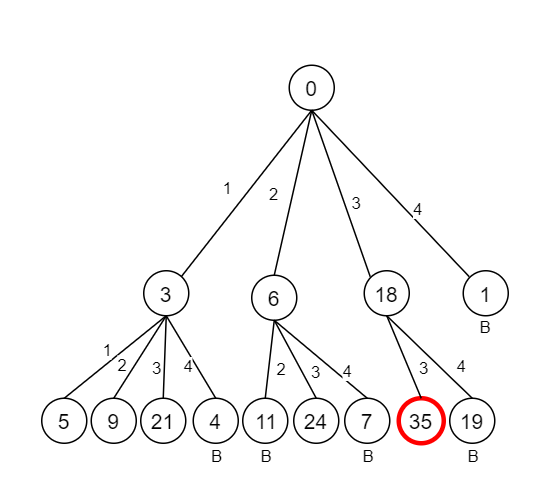


Gambar 4 Ruang Solusi

Angka di dalam lingkaran adalah urutan simpul yang di kunjungi. Lalu, lakukan algoritma *backtracking* dengan beberapa syarat, diantaranya :

1. Jika suatu bilangan sudah dimasukkan, kemudian meng-*expand* bilangan itu lagi, maka nilainya dikurangi 1. (Contoh : 1 (Lapangan Bazar Buku) + 1 (Lapangan Bazar Buku) = 3 + (3-1) = 5).
2. Nilai terbesar selalu disimpan.
3. Jika bilangan tersebut dijumlah dengan nilai terbanyak pada bilangan yang tersisa dan hasilnya tidak melebihi nilai terbesar, maka simpul tersebut langsung di *Bound.*
4. Jika setelah dijumlahkan, nilainya tidak lebih besar dari nilai terbesar, maka simpul tersebut di *Bound.*

Dengan syarat tersebut, maka didapatkan hasil sebagai berikut :



Gambar 5 Hasil penelusuran dengan Algoritma Backtracking

Penjelasan dari ilustrasi tersebut adalah :

1. Simpan bilangan terbesar berupa 0.
2. Bilangan 1 yang pertama kali dikunjungi, maka kita dapatkan nilai 3.
3. Kemudian jika kita mengunjungi bilangan 1 juga, maka kita jumlahkan nilainya, yaitu 3 + 2 = 5. Nilai 2 karena bilangan 1 sudah di kunjungi sebelumnya, karena itu dikurangi 1 menjadi 2. Lalu 5 disimpan sebagai nilai terbesar.
4. Jika kita mengunjungi bilangan 2 maka kita jumlahkan nilainya menjadi 3 + 6 = 9. Nilai terbesar diganti menjadi 9. Lalu Begitu pula jika bilangan yang lain dikunjungi. Sampai bilangan 4, hasil penjumlahannya adalah 4, lalu simpul itu di *Bound.*
5. Setelah itu *backtrack* dan kunjungi bilangan 2, lalu lakukan seperti prosedur sebelumnya. Jangan lupa untuk mengecek jika bilangan tersebut dijumlah dengan nilai terbanyak pada bilangan yang tersisa dan hasilnya tidak melebihi nilai terbesar, maka simpul tersebut langsung di *Bound.*
6. Pada saat *backtrack* dan mengunjungi bilangan 4, maka simpul tersebut langsung di *Bound,* karena sisanya hanya 4 dan nilai nya jika dijumlah (1 + 0 = 1) tidak melebihi nilai terbesar.

Nilai yang harus di masukkan saat awal yaitu jumlah orang yang datang (n). Maka dari itu dihasilkan nilai terbesar berupa 35 dengan himpunan bilangan {3, 3}. Maka dua orang yang datang tersebut sebaiknya membantu di bilangan 3, yaitu Lapangan Bagi-Bagi Makanan.

1. Analisis Penerapan Algoritma *Backtracking*

Penerapan dengan algoritma *backtracking* memiliki beberapa kekurangan, yaitu orang yang mendata harus meng*update* data setiap kali acara tersebut sudah selesai. Data setiap harinya juga perlu di ubah. Hal tersebut mungkin sedikit menambah beban kerja. Alangkah baiknya jika dimodifikasi dengan beberapa masukan berupa waktu mulai dan waktu selesai, dan data tersebut otomatis di *update* setiap waktunya sudah berakhir. Dan juga ditambah fitur lainnya jika sekiranya dibutuhkan.

# V. Kesimpulan

Kesimpulan yang bisa diambil dari pembuatan makalah ini adalah algoritma yang telah dipelajari sebenarnya bisa digunakan dalam berbagai bidang, tergantung kreativitas masing-masing. Algoritma *backtracking* bisa menjadi salah satu alat bantu dalam manajemen SDM pada *event organizer* mahasiswa, yaitu untuk pengalokasian SDM saat tahap pelaksanaan acara.

# VII. Ucapan Terimakasih

Pertama-tama penulis mengucap syukur kepada Allah SWT, Tuhan Yang Maha Esa, karena dengan rahmat dan izin-Nya penulis dapat menyelesaikan makalah ini dengan baik. Tidak lupa penulis berterimakasih kepada orang tua penulis yang selalu mendukung apa yang menjadi jalan hidup penulis serta mendengar keluh kesah penulis dalam dinamika kehidupan penulis. Selanjutnya juga kepada Dr. Nur Ulfa Maulidevi, S.T., M.Sc., Dr. Mesayu Lelya Khodra, ST., MT., dan Dr. Ir. Rinaldi Munir, M.T. selaku dosen mata kuliah IF 2211 Strategi Algoritma yang telah membimbing dan memberi materi kepada penulis selama proses pengajaran mata perkuliahan Strategi Algoritma.

# Referensi

1. Munir, Rinaldi. Diktat Kuliah IF2211, Strategi Algoritma, Program Studi Teknik Informatika, STEI, ITB, 2009
2. https://id.wikipedia.org/wiki/Penyelenggara\_acara diakses pada 13 Mei 2018 pukul 19.13 WIB
3. https://dosenit.com/kuliah-it/pemrograman/algoritma-backtracking diakses pada 13 Mei 2018 pukul 19.15 WIB
4. https://squarefightstar.wordpress.com/2011/02/23/apa-itu-event-organizer/ diakses pada 13 Mei 2018 pukul 19.27 WIB
5. http://evioproductions.com/jenis-jenis-event-organizer/ diakses pada 13 Mei 2018 pukul 20.00 WIB
6. http://kampusupdate.com/3-fase-penting-dalam-membuat-event-2/ diakses pada 13 Mei 2018 pukul 20.00 WIB
7. http://dinus.ac.id/repository/docs/ajar/p.09\_dimensi\_warna\_munsell\_dan\_brewster.pdf diakses pada 2 Desember 2017 pukul 20.00 WIB
8. http://www.tahupedia.com/content/show/497/Kenali-Teori-Warna-Dasar

-Untuk-Mengetahui-Apa-Yang-Terlihat-Bagus diakses pada 3 Desember 2017 pukul 8.00 WIB

1. http://faculty.missouri.edu/leongl/Courses/InstructionalMaterial/ColorTh

eory.pdf diakses pada 3 Desember 2017 pukul 8.12 WIB

# 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, 14 Mei 2018



Annisa Sekar Ayuningtyas

13516044