PENERAPAN ALGORITMA BACKTRACK DALAM MEMBANGKITKAN ELEMENT AWAL PERMAINAN SUDOKU

SKRIPSI

Di Susun Oleh:

Hermawan

NIM : 2011140955

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PAMULANG (UNPAM)

TANGERANG SELATAN

2015 / 2016

LEMBAR PENYATAAN

Yang bertanda tangan dibawah ini:

Nama : Hermawan

NIM : 2011140955

Program Studi : Teknik Informatika

Fakultas : Teknik

Jenjang Pendidikan : Strata 1

Menyatakan bahwa skripsi yang saya buat dengan judul:

"PENERAPAN ALGORITMA BACKTRACK DALAM MEMBANGKITKAN ELEMENT AWAL PERMAINAN SUDOKU"

1. Merupakan hasil karya tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya ijinkan untuk dikelola oleh Universitas Pamulang sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila dikemudian hari penyataan ini tidak benar.

Pamulang, September 2016

(Hermawan)

LEMBAR PERSETUJUAN

NIM	: 2011140955
Nama	: HERMAWAN
Program Studi	: TEKNIK INFORMATIKA
Fakultas	: TEKNIK
JenjangPendidikan	: STRATA 1
JudulSkripsi	: "PENERAPAN ALGORITMA BACKTRACK DALAM MEMBANGKITKAN ELEMEN AWAL PERMAINAN SUDOKU".

Skripsi ini telah diperiksa dan disetujui.

Pamulang, 5 Agustus 2016

Achmad Fikri Zulfikar, M.M

Pembimbing

Mengetahui,

Achmad Hindasyah, S.Si.,M.Si.

Kaprodi Teknik Informatika

LEMBAR PENGESAHAN

NIM	: 2011140955
Nama	: HERMAWAN
Program Studi	: TEKNIK INFORMATIKA
Fakultas	: TEKNIK
Jenjang Pendidikan	: STRATA 1
Judu lSkripsi	: "PENERAPAN ALGORITMA BACKTRACK DALAM MEMBANGKITKAN ELEMEN AWAL PERMAINAN SUDOKU ".

Skripsi ini telah dipertahankan dihadapan dewan penguji ujian skripsi fakultas Teknik, program studi Teknik Informatika dan dinyatakan LULUS.

Pamulang, 13 September 2016

Dede Supriyadi, S.Kom,M.Kom

Normalisa, S.Kom,M.Kom

Achmad Fikri Zulfikar,M.M
Pembimbing

Mengetahui,

Achmad Hindasyah, S.Si.,M.Si
KaProdi Teknik Informatika

KATA PENGANTAR

Dengan mengucapkan segala puji bagi kehadirat Allah SWT , yang Maha Pengasih lagi Maha Penyayang. Alhamdulillah berkat taufiq dan hidayah-Nya , penulis telah dapat menyelesaikan penyusunan proposal skripsi yang berjudul "**Penerapan Algoritma Backtrack Dalam Membangkitkan Element Awal Permainan Sudoku"**. Penulis menyadari sepenuhnya bahwa penyusunan proposal ini jauh dari kesempurnaan.

Penulis mengucapkan terima kasih dan penghargaan yang setinggi-tinginya kepada :

1. Bapak Drs. Dayat Hidayat M.M selaku Rektor Universitas Pamulang beserta seluruh staff.
2. Bapak Ir. Dadang Kurnia, M.M selaku Dekan Fakultas Teknik Informatika Universitas Pamulang.
3. Bapak Achmad Hindasyah, S.Si, M.Si selaku ketua program studi Teknik Informatika beserta seluruh Dosen Teknik Informatika Universitas Pamulang yang telah memberikan ilmu dan bimbingannya selama penulis menempuh pendidikan.
4. Bapak Achmad Fikri Zulfikar,M.M selaku dosen pembimbing yang telah sabar membimbing dan memberikan motivasi serta petunjuk kepada penulis dalam menyelesaikan Tugas Akhir ini.
5. Orang tua tercinta dan seluruh keluarga yang telah banyak memberikan bantuan moril, material, arahan dan selalu mendoakan keberhasilan dan keselamatan selama menempuh perkuliahan.
6. Rekan-rekan mahasiswa Program studi Teknik Informatika angkatan 2011 yang telah banyak memberikan masukan kepada penulis baik dalam mengikuti perkuliahan maupun dalam penulisan skripsi ini.
7. Dan semua pihak yang tidak dapat penulis sebut satu persatu yang telah membantu dalam penyelesaian penulisan skripsi ini.

Tangerang Selatan , 10 September 2016

Penulis

DAFTAR ISI

ABSTRAC

Sudoku is a number puzzle game based on logic. Generally, the game consists of a 9x9 sized grid divided into 3x3 sized called minigrid. The goal of this game is filling the empty cell boxes with a number between 1 to 9, with rules that there must be no repeated number on one row, column and minigrid. Backtrack algorithm is an improved algorithm from the brute-force algorithm that does not explore all possible solutions but only the one that leads to the solution are considered. Through the discussion on this paper, backtrack algorithm is used to generate solutions for Sudoku game. The generated solution elements eliminated so that there is only few elements left with random position, thus initial elements used for initial clue for player to finish the game acquired. The amount of initial element displayed depend on the level that is selected by player in the beginning of the game. This application will be developed using Netbean 8.1 and Java 8 update 101 as a programming language.

Keywords: game, Sudoku, Backtrack, *Java 8 update 101*

ABSTRAK

Sudoku adalah permainan teka-teki angka berbasis logika. Pada umumnya, permainan ini terdiri dari grid berukuran 9x9 yang terbagi menjadi grid berukuran 3x3 yang disebut dengan minigrid. Tujuan dari permainan ini adalah mengisikan sel-sel kotak yang kosong dengan angka dari 1 sampai dengan 9, dengan aturan dalam satu baris, satu kolom dan satu minigrid tidak ada angka yang berulang. Algoritma Backtrack merupakan algoritma perbaikan dari algoritma brute-force yang tidak menelusuri seluruh kemungkinan solusi tetapi hanya pencarian yang mengarah kepada solusi saja yang dipertimbangkan. Melalui pembahasan pada tulisan ini, algoritma Backtrack digunakan untuk membangkitkan solusi permainan Sudoku. Elemen-elemen dari solusi yang dihasilkan dieliminasi hingga hanya tersisa beberapa elemen dengan posisi yang acak, sehingga diperoleh elemen-elemen awal yang digunakan sebagai petunjuk awal bagi pemain untuk menyelesaikan permainan Sudoku. Banyaknya elemen awal yang ditampilkan tergantung dari level yang dipilih pemain pada awal permainan. Aplikasi permainan ini akan dikembangkan dengan menggunakan Netbean 8.1 dan Java 8 update 101 sebagai bahasa pemrograman.

Kata kunci: permainan, Sudoku, Backtrack, *Java 8 update 101*

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Puzzle game merupakan permainan yang tidak hanya berfungsi sebagai hiburan, tetapi juga dapat melatih kemampuan otak. Salah satu puzzle game yang populer adalah Sudoku. Berdasarkan penelitian seorang ahli saraf bernama Ian Robertson, Sudoku dapat meningkatkan kemampuan mental. Selain itu, permainan ini juga dapat mencegah penyakit Alzheimerdan hilang ingatan (Baras, 2010).

Sudoku merupakan permainan teka-teki angka berbasis logika. Aturan permainannya cukup sederhana, akan tetapi untuk menyelesaikannya cukup rumit. Pada umumnya, permainan ini terdiri dari Grid berukuran 9x9 yang terbagi menjadi Grid berukuran 3x3 yang disebut dengan minigrid. Tujuan dari permainan ini adalah mengisikan sel-sel kotak yang kosong dengan angka dari1 sampai dengan 9, dengan aturan dalam satu baris, satu kolom dan satu mini grid tidak ada angka yang berulang.

Permainan Sudoku diciptakan oleh seorang arsitek, Howard Garns.Pada tahun 1979, Sudoku pertama kali diterbitkan oleh majalah Dell,dengan nama Number Place. Pada tahun 1984, permainan ini diterbitkan oleh Nikoli, sebuah perusahaan penerbitan di Jepang. Masyarakat Jepang menamakannya dengan "Suuji wa dokushin ni kagiru" (数字は独身に限る), yang kemudian disingkat menjadi Sudoku. Dalam bahasa Jepang, Sudoku diambil dari kata “su” yang artinya angka dan “doku” berarti tunggal. Jadi, Sudoku berarti angka-angkanya harus tetap tunggal.Tahun 2004, permainan ini mulai dikenalkan di Inggris oleh Wayne Gould,dan diterbitkan pertama kali pada surat kabar The Times, 12 November 2004, dengan tetap menggunakan nama Sudoku. Hanya dalam waktu beberapa bulan, surat kabar lain di Inggris juga ikut mempublikasikan permainan Sudoku ini.Sejak saat itulah, Sudoku mulai populer di berbagai belahan dunia (Jussien, 2007).

Pada awal permainan Sudoku, pemain akan diberikan grid Sudoku berukuran 9x9, yang beberapa elemennya diketahui pada beberapa sel, dinamakan elemen awal.Elemen awal tersebut merupakan bilangan bantuan yang bernilai 1 sampai dengan 9. Pemain diharuskan mengisi sel-sel yang kosong dengan angka dari 1 sampai dengan 9 sedemikian sehingga setiap baris, kolom dan minigrid tidak terdapat angka yang berulang atau tepat satu kali. Pemain dinyatakan menang jika seluruh sel pada grid Sudoku terisi penuh dan memenuhi aturan Sudoku.

Untuk membangkitkan elemen awal Permainan Sudoku, dibutuhkan suatu algoritma yang dapat menentukan solusi permainan Sudoku. Algoritma yang akan digunakan dalam menentukan solusi permainan adalah algoritma Backtracking. Beberapa elemen dari solusi yang diperoleh kemudian dieliminasi sedemikian sehingga diperoleh grid Sudoku yang berisi beberapa elemen awal yang diketahui sesuai dengan level yang dipilih pemain dengan posisi yang acak.

Algoritma Backtrack (Munir, 2004) adalah algoritma yang berbasis pada Depth First Search(DFS) untuk mencari solusi persoalan yang lebih efektif. Selain itu, algoritma ini merupakan perbaikan dari algoritma Brute Force, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada, namun hanya yang mengarah pada solusi saja yang dipertimbangkan. Dengan begitu, waktu pencarian dapat dihemat. Pada Skripsi ini akan dibangun sebuah aplikasi yang digunakan untuk membangkitkan elemen awal permainan Sudoku dengan menggunakan algoritma Backtrack. Aplikasi ini akan dikembangkan dengan menggunakan bahasa pemrograman Java pada NETBEAN 8.1

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan sebelumnya, permasalahan yang akan dibahas dalam Skripsi ini adalah sebagai berikut:

1. Apakah penerapan algoritma Backtrack dalam menentukan elemen-elemen awal permainan Sudoku dapat dilakukan?

2. Apakah penerapan algoritma Backtrack dalam menentukan elemen-elemen awal permainan kedalam bahasa pemrograman Java pada Netbean 8.1 dapat berjalan?

1.3 Tujuan Penelitian

Adapun tujuan dari Skripsi ini antara lain adalah

1. Mengetahui apakah penerapan algoritma Backtrack dalam menentukan elemen elemen awal permainan Sudoku dapat dilakukan?

2. Mengetahui apakah penerapan algoritma Backtrack dalam menentukan elemen-elemen awal permainan ke dalam bahasa pemrograman Java pada Netbean 8.1 dapat berjalan?

1.4 Batasan Masalah

Ruang lingkup permasalahan dalam merancangaplikasi permainan Sudoku ini dibatasi sebagai berikut :

a. Grid Sudoku yang di bahas berukuran 9x9 dengan minigrid berukuran 3x3.
b. Aplikasi ini akan dibangun dengan menggunakan bahasa pemrogram Java pada Netbean 8.1.
c. Teknik yang digunakan dalam membangkitkan elemen awal pada permainan Sudoku adalah dengan membangkitkan solusi permainan Sudoku, kemudian satu persatu dieliminasi hingga pada grid Sudoku hanya ditampilkan beberapa elemen awal sesuai dengan level yang dipilih pemain dengan posisi yang acak.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah sebagai referensi dari pengembangan mata kuliah Algoritma Pemrograman. Selain itu juga,

diharapkan dapat menjadi sebuah acuan dalam pembuatan aplikasi permainan Sudoku dengan menggunakan algoritma yang lebih efektif.

1.6 Metodelogi Penelitian

Metode penulisan yang digunakan adalah studi literature berbasis kualitatif melalui beberapa buku, jurnal, skripsi, dan artikel dari internet yang berkaitan dengan permasalahan yang dihadapi. Pada pelaksanaanya untuk mendapatkan pengarahan dan pendalaman materi, penulis juga senantiasa berkonsultasi dengan dosen pembimbing

1.7 Sistematika Penulisan

Sistematika penulisan yang akan diuraikan dalam Skripsi ini terbagi menjadi 5 bab, yaitu:

BAB I PENDAHULUAN

Pada bab ini akan dibahas latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah serta sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini akan dibahas teori-teori dasar yang mendukung pemecahan masalah yang dihadapi, dimana sumbernya berasal dari buku sumber yang menunjang dalam penyusunan Skripsi ini.

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini menguraikan cara pemecahan masalah dengan menggunakan algoritma Backtrack serta menganalisis algoritma tersebut dalam menyelesaikan masalah. Selain itu akan diuraikan bagaimana merancang prosedur yang digunakan algoritma Backtrack dalam memecahkan masalah.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab ini berisi tentang implementasi setiap prosedur yang telah dirancang ke dalam bentuk aplikasi, kemudian akan dibahas mengenai hasil dan pengujian dari aplikasi yang dirancang

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dan saran yang diperoleh setelah melakukan penelitian,yang berguna untuk perbaikan sistem yang lebih baik.

BAB II LANDASAN TEORI

2.1 Permainan

Permainan (*games*) adalah setiap kontes antara pemain yang berinteraksi satu sama lain dengan mengikuti aturan-aturan tertentu untuk mencapai tujuan tertentu pula (Sadiman, 1993:75).

Jadi permainan adalah cara bermain dengan mengikuti aturan-aturan tertentu yang dapat dilakukan secara individu maupun berkelompok guna mencapai tujuan tertentu. Alat permainan adalah semua alat bermain yang dapat digunakan oleh peserta didik untuk memenuhi naluri bermainnya dan memiliki barbagai macam sifat, seperti bongkar pasang, mengelompokkan, memadukan, mencari padanannya, merangkai, membentuk, atau menyusun sesuai dengan bentuk aslinya.

2.2 Sudoku

Sudoku adalah permainan teka-teki angka yang terdiri dari 81 kotak (9 x 9) yang berisi angka-angka antara 1 sampai 9 yang harus diisi penuh pada setiap kotaknya. Tujuan permainan ini adalah refreshing dengan mengasah otak. Pemain sudoku diharuskan untuk mengisi semua kotak kosong yang tersedia sedemikian setiap angkanya akan unik pada baris, kolom, dan daerahnya (akan dijelaskan). Contoh sudoku dapat dilihat pada gambar 1. Saat pertama memainkan sudoku, beberapa kotak sudah diisi angka yang merupakan hints untuk mengisi kotak kosong lain hingga penuh. Pada gambar di bawah, dapat kita lihat bahwa adanya perbedaan tebal garis kotak. Garis tebal kotak menandakan batas daerah unik yang harus diisi angka yang unik dari 1-9. Daerah unik terdiri dari 9 kotak (3x3). Sebagai contoh, Daerah unik pada posisi terkiri-teratas terdiri dari angka 5,3,6,9, dan 8, untuk itu 4 kotak kosong lain harus diisi dengan angka unik selain angka sudah ada pada daerah itu yaitu 1,2,4, dan 7. Setiap angka yang diisi pada suatu kotak harus unik secara horizontal, vertikal, dan daerahnya. Kita harus

mencari semua nilai yang unik pada setiap kotak dengan strategi tertentu sehingga semua kotak kosong terisi oleh angka. (Krisna Dibyo Admojo.ITB,2012)

Sebagai permulaan, beberapa kotak telah diisi dengan angka-angka pembuka atau biasa disebut sebagai soal sudoku. Tugas dari setiap pemain nantinya harus bisa mengisi setiap kotak yang masih kosong sesuai aturan yang berlaku. Meskipun aturannya sederhana namun penyelesaian teka-teki ini tidak semudah aturannya. Tentu saja tingkat kesulitan tiap teka-teki dapat bervariasi.

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

Gambar 2.1 Contoh Sudoku[sumber : Wikipedia]

2.3 Algoritma Runut Balik (*backtracking*)

Runut-balik (*backtracking*) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih mangkus. Runut-balik (*backtracking*) merupakan perbaikan dari algoritma *brute-force* yang secara sistematis akan melakukan pencarian solusi permasalahan 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 akan 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.

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. Lintasan dari akar ke daun menyatakan solusi yang mungkin. Seluruh lintasan dari dari akar ke daun membentuk ruang solusi.

Algoritma Backtracking merupakan perbaikan dari algoritma Bruteforce.Algoritma ini mempunyai property yang dinamakan fungsi pembatas, Fungsi pembatas menentukan apakah (x1,x2,x3,….,xk) mengarah ke suatu solusi jika ya, maka pembangkitan xk+1 dilanjutkan jika tidak maka backtrack ke komponen xk-1. Dalam hal permainan sudoku ini xk+1 dianggap sebagai elemen berikut nya pada matriks,dan xk-1 merupakan elemen sebelumnya dalam matriksWalaupun dalam kasus permainan sudoku ini kompleksitas asimptotik dari algoritma Backtracking,fungsi pembatas dapat melakukan pruning yang menyebabkan waktu komputasi berkurang secara signifikan (Rama Adhitia.ITB,2007)

Algoritma Backtracking membentuk sebuah pohon ruang status selama prosesnya Struktur pohon inilah, yang juga merupakan sebuah graf tak berarah, yang ditraversal dengan prinsip DFS (*Depth First Search*). Simpul-simpul pada pohon ruang status yang tidak mengarah ke solusi maka akan "dimatikan". Sedangkan simpul-simpul pohon ruang status yang masih mengarah ke solusi maka akan terus berkembang. Pematian simpul pohon ruang status yang tidak mengarah kepada solusi ini sering disebut dengan istilah *prunning*. Dengan demikian, seluruh lintasan dari akar ke daun yang melalui simpul-simpul yang tidak "dimatikan" akan membentuk sebuah ruang solusi. (Sibghatulah Mujaddid.ITB,2009)

Algoritma runut balik (*backtracking*) merupakan algoritma yang digunakan untuk mencari solusi persoalan secara lebih mangkus daripada menggunakan algoritma *brute force.* Algoritma ini akan mencari solusi berdasarkan ruang solusi yang ada secara sistematis namun tidak semua ruang solusi akan diperiksa, hanya

pencarian yang mengarah kepada solusi yang akan diproses. (Rinaldi Munir, Diktat Strategi Algoritmik, Teknik Informatika ITB. 2005).

2.3.1 Properti Umum Metode runut balik (Backtracking)

Untuk menerapkan metode runut-balik, properti berikut didefinisikan:

1. Solusi persoalan.

   Solusi dinyatakan sebagai vektor n-tuple:

   X=(x1, x2, ..., xn), xi anggota himpunan berhingga Si .

   Mungkin saja S1 = S2 = ... = Sn.

   Contoh: Si = {0,1}

   Si = 0 atau 1

2. Fungsi pembangkit nilai xk

   Dinyatakan sebagai:

   T(k)

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

3. Fungsi Pembatas (fungsi kriteria)

   Dinyatakan sebagai:

   B(x1, x2, ..., xk)

   Fungsi pembatas menentukan apakah (x1, x2, ..., xk) mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk xk+1 dilanjutkan, tetapi jika tidak, maka (x1, x2, ..., xk) dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

2.3.2 Pengorganisasian Solusi

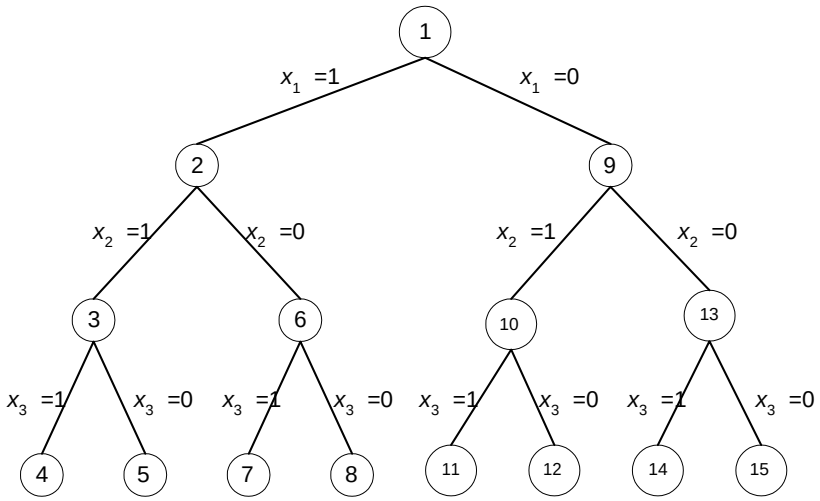
Semua kemungkinan solusi dari persoalan disebut **ruang solusi** (*solution space*). Jika $x_i \in S_i$, maka $S_1 \times S_2 \times \ldots \times S_n$ disebut ruang solusi. Jumlah anggota di dalam ruang solusi adalah $|S_1| \cdot |S_2| \cdot \ldots \cdot |S_n|$. Tinjau persoalan *Knapsack* 0/1 untuk $n = 3$. Solusi persoalan dinyatakan sebagai vektor (x_1, x_2, x_3) dengan $x_i \in \{0,1\}$. Ruang solusinya adalah

$\{0,1\} \times \{0,1\} \times \{0,1\} = \{(0, 0, 0), (0, 1, 0), (0, 0, 1), (1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)\}$.

Pada persoalan *Knapsack* 0/1 dengan $n = 3$ terdapat $2^n = 2^3 = 8$ kemungkinan solusi, yaitu:

(0, 0, 0), (0, 1, 0), (0, 0, 1), (1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1), dan (1, 1, 1).

Penyelesaian secara *exhaustive search* adalah dengan menguji setiap kemungkinan solusi. 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:



Gambar 2.2 Ruang solusi untuk persoalan *Knapsack* 0/1 dengan $n = 3$

2.3.3 Prinsip Pencarian Solusi dengan Metode Runut Balik

Langkah-langkah pencarian solusi dengan metode runut balik adalah sebagai berikut:

1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan yang dipakai adalah mengikuti metode pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan simpul hidupm dan simpul hidup yang sedang diperluas dinamakan simpul-E. Simpul dinomori dari atas ke bawah sesuai dengan kelahirannya.

2. Jika lintasan yang diperluas yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut "dibunuh" sehingga menjadi simpul mati (*dead node*). Simpul yang sudah mati ini tidak akan diperluas lagi.

3. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangkitkan simpul anak lainnya. Bila tidak ada lagi simpul anak yang dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut-balik (*backtracking*) ke simpul hidup terdekat. Selanjutnya simpul ini menjadi simpul-E yang terbaru.

4. Pencarian dihentikan bila telah ditemukan solusi atau tidak ada lagi simpul hidup untuk runut balik (*backtracking*). (Sibghatulah Mujaddid.ITB,2009)

   Tinjau persoalan *Knapsack* 0/1 dengan instansiasi:

   $n = 3$

   $(w_1, w_2, w_3) = (35, 32, 25)$

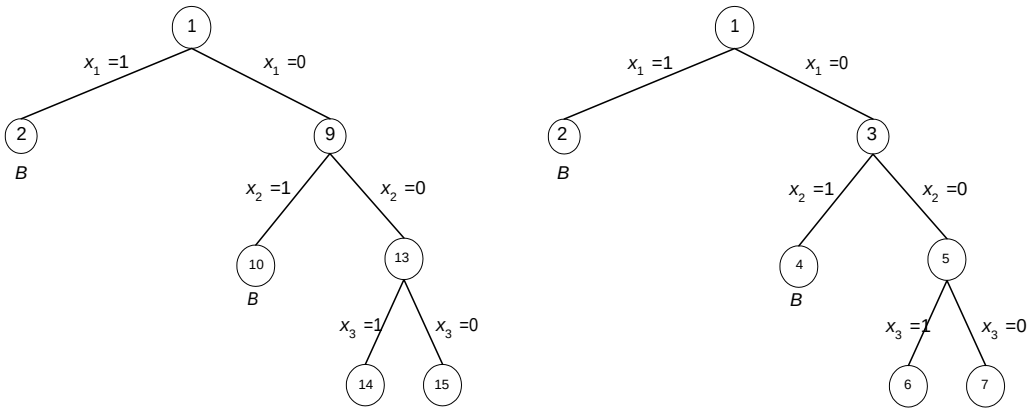
   $(p_1, p_2, p_3) = (40, 25, 50)$

   $M = 30$

   Solusi dinyatakan sebagai $X = (x_1, x_2, x_3)$, $x_i \in \{0, 1\}$.

   Fungsi pembatas:

$$\sum_{i=1}^{k} w_i x_i \leq M$$



(a) (b)

Gambar 2 3 (a) Pohon dinamis yang dibentuk selama pencarian untuk persoalan *Knapsack* 0/1 dengan n = 3,w = (35, 32, 25) dan p = (40, 25, 50)

Gambar 2 4 (b) Penomoran ulang simpul-simpul sesuai urutan pembangkitannya

Solusi optimumnya adalah X = (0, 0, 1) dan F = 50.

2.4 UML (Unified Modelling Language)

UML (*Unified Modelling Language*) merupakan kosakata umum berbasis objek dan diagram teknik yang cukup efektif untuk memodelkan setiap proyek pengembangan sistem mulai tahap analisis sampai tahap desain dan implementasi (Dennis et al, 2012:513).

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing* diagram) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa

model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. Intinya, UML merupakan alat komunikasi yang konsisten dalam *mensuport* para pengembang sistem saat ini. Diagram *Use Case*, Diagram Aktivitas (*Activity Diagram*), Diagram *Sequance*, dan Diagram *Class.*

2.4.1 Pengertian Perancangan

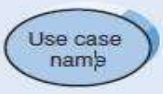
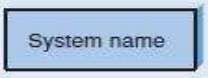
Beberapa *literature* menyebutkan bahwa UML menyediakan sembilan jenis diagram. Namun kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Diagram yang sering digunakan adalah Diagram *Use Case*, Diagram Aktivitas (*Activity Diagram*), Diagram *Sequance*, Diagram *Class*.

2.4.2 Use Case

Use Case Diagram merupakan suatu diagram yang menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi antara sistem dan lingkungannya. (Dennis *et al*, 2012:513)

Komponen pembentuk diagram *Use Case*, adalah :

Tabel 2.1 Notasi *Use Case* Diagram (Dennis *et al* :2012)

Term and Definition	Symbol
An actor ■ Is a person or system that derives benefit from and is external to the system. ■ Is labeled with its role. ■ Can be associated with other actors by a specialization/superclass association, denoted by an arrow with a hollow arrowhead. ■ Is placed outside the system boundary.	 Actor role name
A use case ■ Represents a major piece of system functionality. ■ Can extend another use case. ■ Can use another use case. ■ Is placed inside the system boundary. ■ Is labeled with a descriptive verb–noun phrase.	
A system boundary ■ Includes the name of the system inside or on top. ■ Represents the scope of the system.	
An association relationship ■ Links an actor with the use case(s) with which it interacts.	

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan disebuah system.
2. *Use Case*, aktifitas / sarana yang disiapkan oleh bisnis / sistem.
3. *System boundary*, adalah sebuah kotak yang mewakili sebuah sistem.

*Hubungan (*link), aktor mana saja yang terlibat dalam *Use case*, dan bagaimana hubungan *Use case* dengan *Use case* lain. Ada hubungan antar *Use case*. Digolongkan menjadi 2 : yaitu *extend* digambarkan dengan keterangan <<*extend*>>, dan *include* digambarkan dengan keterangan <<*include*>>, berikut perbedaanya :

2.4.3 Activity Diagram

Pengertian Diagram *Activity* adalah yang menggambarkan alur kerja bisnis independent dari *class*, aliran kegiatan dalam *Use Case*, atau desain rinci sebuah metode. (Dennis *et al* 2012:516)

Tabel 2.2 Notasi Pemodelan *Activity Diagram* (Dennis *et al*, 2012:516)

Actor ▪ Digunakan untuk melakukan tindakan .	
Activity ▪ Digunakan untuk mewakili serangkaian tindakan.	
Object Node ▪ Digunakan untuk mewakili suatu objek yang terhubung ke satu set Arus Obyek.	
Control Flow ▪ Menunjukkan urutan eksekusi.	
Object Flow ▪ Menunjukkan arus dari sebuah objek dari satu kegiatan (atau tindakan) untuk kegiatan lain (atau tindakan).	---------->
Initial Node ▪ Menggambarkan awal dari serangkaian tindakan atau kegiatan