

Analisis Penggunaan Algoritma Breadth First Search Dalam Konsep Artificial Intellegencia

Edi Wijaya
STMIK Time Medan
Jalan Merbabu No. 32 AA – BB
Telp 061. 456 1932, E-mail : wiwileosummer@gmail.com

Abstrak

Artificial Intellegencia merupakan suatu konsep pemetaan suatu bahasa pemrograman yang dapat membuat suatu kesimpulan berdasarkan pemetaan yang telah dilakukan didalam pemrograman. Dalam hal ini, banyak metode yang dapat digunakan dan dimanfaatkan untuk menyelesaikan permasalahan tersebut. Contohnya seperti masalah Teko Air. Permasalahan ini dapat diselesaikan dengan menerapkan konsep AI yaitu dengan bantuan pohon pelacakan dan menerapkan metode pencarian melebar pertama (*breadth-first search* / BFS). Pencarian solusi dimulai dari kondisi dimana kedua teko kosong (*node* akar dari pohon pelacakan). Perangkat lunak ini memiliki kemampuan untuk mencari solusi permasalahan teko air dengan 2 buah kendi dan menampilkan semua langkah – langkah yang dapat diambil untuk mendapatkan solusi atau volume air yang diinginkan. Perangkat lunak dapat mencari solusi terpendek dari permasalahan teko air karena menggunakan metode pencarian melebar pertama (*breadth-first search*). Pencarian tidak selalu menghasilkan solusi walaupun setelah semua *node* pada pohon pelacakan diperiksa dan dikembangkan.

Kata Kunci : AI, *breadth-first search* / BFS, *node* akar dari pohon pelacakan

1. PENDAHULUAN

Perkembangan pengetahuan mengenai sistem komputer semakin berkembang berdasarkan kebutuhan manusia akan sistem. Dalam konsep bahasa pemrograman, terutama didalam pemrograman yang berbasis sistem operasi semakin lama semakin dibutuhkan untuk menyelesaikan permasalahan. Pada masa kini, perkembangan teknologi semakin tinggi dengan ditemukannya suatu sistem yang dapat memberikan kontribusi kesimpulan kepada pengguna sistem. Sistem yang dimaksud adalah konsep AI.

Artificial Intellegencia merupakan suatu konsep pemetaan suatu bahasa pemrograman yang dapat membuat suatu kesimpulan berdasarkan pemetaan yang telah dilakukan didalam pemrograman. Dalam hal ini, banyak metode yang dapat digunakan dan dimanfaatkan untuk menyelesaikan permasalahan tersebut. Contohnya seperti masalah Teko Air.

Permasalahan ini dapat diselesaikan dengan menerapkan konsep AI yaitu dengan bantuan pohon pelacakan dan menerapkan metode pencarian melebar pertama (*breadth-first search* / BFS). Pencarian solusi dimulai dari kondisi dimana kedua teko kosong (*node* akar dari pohon pelacakan). Proses dilanjutkan dengan menggambarkan kondisi (*state*) berikutnya (dengan melakukan aksi terhadap *state* sebelumnya) hingga semua *state* diperiksa dan mendapatkan tujuan (*goal state*).

Berdasarkan uraian di atas bertujuan untuk mencari penyelesaian atau solusi terhadap permasalahan teko air dengan bantuan pohon pelacakan dan metode pencarian melebar pertama (*breadth-first search*).

2. Artificial Intelligence (AI)

2.1. Sejarah AI

Artificial Intelligence (AI) atau kecerdasan buatan termasuk bidang ilmu yang relatif muda. Pada tahun 1950-an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang bisa dikerjakan oleh manusia. Alan Turing, seorang matematikawan dari Inggris pertama kali mengusulkan adanya pengujian untuk melihat bisa tidaknya sebuah mesin dikatakan cerdas. Hasil pengujian tersebut kemudian dikenal dengan *Turing Test*, di mana mesin tersebut menyamar seolah-olah sebagai seseorang di dalam suatu permainan yang mampu memberikan respon terhadap serangkaian pertanyaan yang diajukan. Turing beranggapan bahwa, jika mesin dapat membuat seseorang percaya bahwa dirinya mampu

berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia).

Kecerdasan buatan atau *Artificial Intelligence (AI)* itu sendiri dimunculkan oleh seorang professor dari *Massachusetts Institute of Technology* yang bernama John McCarthy pada tahun 1956 pada *Dartmouth Conference* yang dihadiri oleh para peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan kelakuan manusia tersebut.

2.2. Definisi AI

Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peranan komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu komputer diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan manusia.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian pula, dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus

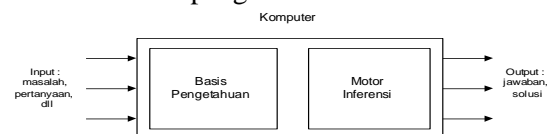
diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu pada AI akan mencoba untuk memberikan beberapa metoda untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin yang pintar.

Pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain :

1. Sudut pandang kecerdasan
Kecerdasan buatan akan membuat mesin menjadi 'cerdas' (mampu berbuat seperti apa yang dilakukan oleh manusia).
2. Sudut pandang penelitian
Kecerdasan buatan adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.
3. Sudut pandang bisnis
Kecerdasan buatan adalah kumpulan peralatan yang sangat *powerful* dan metodologis dalam menyelesaikan masalah-masalah bisnis.
4. Sudut pandang pemrograman
Kecerdasan buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).

Untuk membuat aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan, yaitu :

1. Basis Pengetahuan (*Knowledge Base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor Inferensi (*Inference Engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar 1 Penerapan konsep kecerdasan buatan di komputer Representasi

Representasi merupakan aturan untuk mendeskripsikan suatu himpunan objek. Jika masalah sudah dinyatakan dengan representasi yang tepat, maka masalah hampir selesai.

Susunan atau pola representasi suatu masalah antara lain :

1. Representasi Leksikal, merupakan simbol-simbol yang boleh ada.
2. Representasi Struktural, merupakan hubungan dan kombinasi yang boleh antar simbol.
3. Representasi Prosedural, merupakan prosedur-prosedur yang berkaitan.
4. Representasi Semantik, merupakan arti dari simbol-simbol yang ada.

Representasi dimaksudkan untuk menangkap sifat-sifat penting dari suatu masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah. Kecerdasan buatan lebih ditujukan untuk menangani pemecahan masalah kualitatif daripada kuantitatif yaitu lebih dititikberatkan pada sebab-akibat daripada penghitungan, dan lebih diarahkan pada pengaturan sejumlah besar pengetahuan yang bervariasi daripada diimplementasikan pada algoritma yang *well - defined*.

Representasi Leksikal

Representasi jenis ini menggunakan ekspresi-ekspresi dalam logika formal untuk merepresentasikan basis pengetahuan. Aturan-aturan inferensi dan prosedur pembuktian menerapkan pengetahuan ini pada masalah yang harus dipecahkan.

Contoh :

Premis Mayor : Jika mahasiswa mendapat nilai E, maka mahasiswa tidak lulus.

Premis Minor : Mahasiswa mendapat nilai E.

Konklusi : Mahasiswa tidak lulus.

Representasi Struktural

Bahasa-bahasa representasi terstruktur memperluas *network* dengan cara membuat setiap simpulnya menjadi sebuah struktur data kompleks yang berisi tempat-tempat bernama dengan nilai-nilai tertentu. Nilai-nilai ini dapat merupakan data numerik atau simbolik sederhana, pointer ke bingkai (*frame*) lain, atau merupakan prosedur untuk mengerjakan tugas tertentu.

Representasi Prosedural

Representasi prosedural menggambarkan pengetahuan sebagai sekumpulan instruksi untuk memecahkan suatu masalah. Ini berlawanan dengan representasi deklaratif yang ditangani oleh *network* logika dan semantik.

Dalam sistem berbasis aturan, aturan IF-THEN dapat ditafsirkan sebagai sebuah prosedur untuk mencapai tujuan pemecahan masalah, yakni memecahkan kesimpulan, memecahkan premis secara berurutan.

Representasi Semantik

Representasi ini menangkap pengetahuan sebagai sebuah graf dimana simpul-simpulnya menggambarkan obyek atau konsep dalam masalah yang dihadapi sedangkan lengkungan-lengkungannya menggambarkan hubungan atau asosiasi antar mereka. Contoh-contoh representasi *network* antara lain adalah jaringan semantik (*semantic network*) dan graf konseptual.

2.3. Kecerdasan Buatan dan Kecerdasan Alamiah

Kecerdasan alamiah adalah kecerdasan yang dimiliki oleh manusia. Jika dibandingkan dengan kecerdasan buatan, ada beberapa keuntungan kecerdasan buatan dibanding kecerdasan alamiah, yaitu:

1. Lebih permanen.

Kecerdasan alamiah akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer dan program tidak diubah.

2. Memberikan kemudahan dalam duplikasi dan penyebaran.

Mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama, dan keahlian itu juga tidak akan pernah dapat diduplikasi dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.

3. Relatif lebih murah dan kecerdasan alamiah. Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.

4. Konsisten dan teliti.

Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.

5. Dapat didokumentasi.
Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
6. Dapat mengerjakan beberapa *task* dengan lebih cepat dan lebih baik dibanding manusia.

Sedangkan, keuntungan kecerdasan alamiah dibanding kecerdasan buatan:

1. Bersifat lebih kreatif.
Kemampuan untuk menambah ataupun memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.
2. Dapat melakukan proses pembelajaran secara langsung, sementara AI harus mendapatkan masukan berupa simbol dan representasi.
3. Fokus yang luas sebagai referensi untuk pengambilan keputusan, sebaiknya AI menggunakan fokus yang sempit.

Komputer dapat digunakan untuk mengumpulkan informasi tentang obyek, kegiatan (*events*), proses dan dapat memproses sejumlah besar informasi dengan lebih efisien dari yang dapat dikerjakan manusia. Namun di sisi lain, manusia dengan menggunakan insting dapat melakukan hal yang sulit diprogram pada komputer, yaitu kemampuan mengenali (*recognize*) hubungan antara hal-hal tersebut, menilai kualitas dan menemukan pola yang menjelaskan hubungan tersebut.

2.4. Lingkup Kecerdasan Buatan pada Aplikasi Komersial

Dewasa ini, kecerdasan buatan juga memberikan kontribusi yang cukup besar di bidang manajemen. Adanya sistem pendukung keputusan dan Sistem Informasi Manajemen juga tidak terlepas dari andil AI. Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran AI. Karakteristik 'cerdas' sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. AI tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang lain.

Irisan antara psikologi dan AI melahirkan sebuah area yang dikenal dengan nama *cognition & psycholinguistics*. Irisan antara teknik elektro dengan AI melahirkan berbagai ilmu, seperti: pengolahan citra, teori kendali, pengenalan pola dan robotika.

Adanya irisan penggunaan AI di berbagai disiplin ilmu tersebut menyebabkan cukup rumitnya untuk mengklasifikasikan AI menurut disiplin ilmu yang menggunakannya. Untuk memudahkan hal tersebut, maka pengklasifikasian lingkup AI didasarkan pada *output* yang diberikan, yaitu pada aplikasi komersial (meskipun sebenarnya AI itu sendiri bukan merupakan medan komersial).

Lingkup utama dalam kecerdasan buatan adalah:

1. Sistem Pakar (*Expert System*).
Disini, komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian, komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.
2. Pengolahan Bahasa Alami (*Natural Language Processing*).
Dengan pengolahan bahasa alami ini diharapkan *user* dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan Ucapan (*Speech Recognition*).
Melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer menggunakan suara.
4. Robotika & Sistem Sensor (*Robotics & Sensory Systems*).
5. *Computer Vision*, mencoba untuk dapat menginterpretasikan gambar atau objek-objek tampak melalui komputer.
6. *Intelligent Computer-aided Instruction*.
Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. *Game Playing*.

Seiring dengan perkembangan teknologi, muncul beberapa teknologi yang juga bertujuan untuk membuat agar komputer menjadi cerdas sehingga dapat menirukan kerja manusia sehari-hari. Teknologi ini juga mampu mengakomodasi adanya ketidakpastian dan ketidaktepatan data *input*. Dengan didasari pada teori himpunan, maka pada tahun 1965 muncul Logika Fuzzy. Kemudian pada tahun 1975, John Holland mengatakan bahwa setiap *problem* berbentuk adaptasi (alami maupun

buatan) secara umum dapat diformulasikan dalam terminologi genetika. Algoritma genetika ini merupakan simulasi proses evolusi Darwin dan operasi genetika atas kromosom.

2.5. Masalah Ruang Keadaan

Secara umum, untuk membangun suatu sistem yang mampu menyelesaikan masalah, perlu dipertimbangkan 4 hal:

1. Mendefinisikan masalah dengan tepat. Pendefinisian ini mencakup spesifikasi yang tepat mengenai keadaan awal dan solusi yang diharapkan.
2. Menganalisis masalah tersebut serta mencari beberapa teknik penyelesaian masalah yang sesuai.
3. Merepresentasikan pengetahuan yang perlu untuk menyelesaikan masalah tersebut.
4. Memilih teknik penyelesaian masalah yang terbaik.

Misalkan permasalahan yang dihadapi adalah "Permainan Catur", maka harus ditentukan:

1. Posisi awal pada papan catur;
Posisi awal setiap permainan catur selalu sama, yaitu semua bidak diletakkan di atas papan catur dalam 2 sisi, yaitu kubu putih dan kubu hitam.
2. Aturan-aturan untuk melakukan gerakan secara legal;
Aturan-aturan ini sangat berguna untuk menentukan gerakan suatu bidak, yaitu melangkah dari satu keadaan ke keadaan lain. Misalkan untuk mempermudah menunjukkan posisi bidak, setiap kotak harus ditunjukkan dalam huruf (a, b, c, d, e, f, g, h) pada arah horizontal dan angka (1, 2, 3, 4, 5, 6, 7, 8) pada arah vertikal. Suatu aturan untuk menggerakkan bidak dari posisi (e,2) ke (e,4) dapat ditunjukkan dengan aturan:
IF Bidak putih pada Kotak(e,2), And Kotak(e,3) Kosong, And Kotak(e,4) Kosong, Then Gerakkan bidak dari (e,2) ke (e,4)
3. Tujuan (*goal*);
Tujuan yang ingin dicapai adalah posisi pada papan catur yang menunjukkan kemenangan seseorang terhadap lawannya. Kemenangan ini ditandai dengan posisi Raja yang sudah tidak dapat bergerak lagi.

Contoh di atas menunjukkan representasi masalah dalam Ruang Keadaan (*State Space*), yaitu suatu ruang yang berisi semua keadaan yang mungkin. Kita dapat memulai bermain catur dengan menempatkan diri pada keadaan awal, kemudian bergerak dari satu keadaan ke keadaan yang lain sesuai dengan aturan yang ada, dan mengakhiri permainan jika salah satu pemain telah mencapai tujuan.

Sehingga secara umum, untuk mendeskripsikan masalah dengan baik, harus:

1. Mendefinisikan suatu ruang keadaan.
2. Menetapkan satu atau lebih keadaan awal.
3. Menetapkan satu atau lebih tujuan.
4. Menetapkan kumpulan aturan.

3. Metode Pencarian dan Pelacakan

Konsep utama dalam menentukan keberhasilan sistem yang berlandaskan AI adalah kesuksesan dalam melakukan dan mengembangkan pencarian. Pencarian atau pelacakan merupakan salah satu teknik untuk menyelesaikan permasalahan AI. Keberhasilan suatu sistem, salah satunya ditentukan oleh kesuksesan dalam pencarian dan pencocokan. Teknik dasar pencarian memberikan suatu kunci bagi banyak sejarah penyelesaian yang penting dalam bidang AI. Ada beberapa aplikasi yang menggunakan teknik pencarian ini, yaitu:

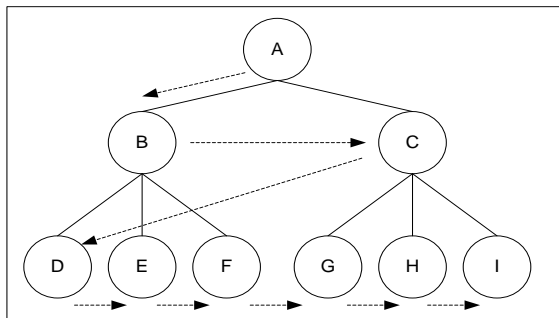
1. Papan *game* dan *puzzle* (tic-tac-toe, catur)
2. Penjadwalan dan masalah *routing* (*travelling salesman problem*)
3. *Parsing* bahasa dan interpretasinya (pencarian struktur dan arti)
4. Logika pemrograman (pencarian fakta dan implikasinya)
5. *Computer vision* dan pengenalan pola

Konsep pencarian untuk suatu solusi dalam ruang keadaan (*state space*) merupakan pusat AI yang menjadikan AI lebih unggul dalam bidang ilmu komputer dibandingkan dengan yang lainnya, dan prinsip kontribusi AI untuk ilmu pengetahuan dari pencarian ini merupakan konsep basis pengetahuan (*knowledge based*) heuristik untuk pembatasan dan pencarian berarah (*directing search*). Pada dasarnya, ada dua teknik pencarian dan pelacakan yaitu:

1. Pencarian buta (*blind search*), terdiri atas:
 - a. Pencarian melebar pertama (*Breadth-First Search*)
 - b. Pencarian mendalam pertama (*Depth-First Search*)
2. Pencarian heuristik (*heuristic search*).

3.1. Pencarian Melebar Pertama (Breadth-First Search)

Pada metode pencarian ini, semua *node* pada level *n* akan dikunjungi terlebih dahulu sebelum mengunjungi *node-node* pada level *n+1*. Pencarian dimulai dari *node* akar terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya. Demikian seterusnya hingga ditemukannya solusi. Gambaran pencarian BFS dapat dilihat pada gambar 2.



Gambar 2 Pencarian melebar pertama (*Breadth-First Search*)

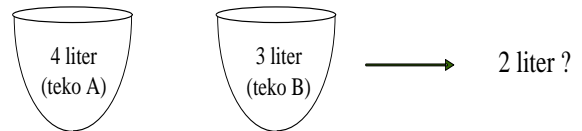
Pada gambar 2, terlihat bahwa pencarian dimulai dari keadaan awal (*node A*). Dari *node A* dikembangkan dua *node* baru yang menjadi anaknya, yaitu *node B* dan *node C*. Penelusuran dilanjutkan ke *node B* (menghasilkan *node D, E, F*) dan *node C* (menghasilkan *node G, H, I*). Demikian seterusnya hingga ditemukan solusi.

Masalah Teko Air

Masalah Teko Air (*Water Jug Problem*) dapat diilustrasikan seperti berikut, terdapat 2 buah teko air X dan Y, masing – masing berukuran A dan B liter. Salah satu teko tersebut hendak diisi air sebanyak N liter, dengan menggunakan asumsi bahwa sumber air tidak terbatas. Aksi – aksi yang dapat dilakukan, antara lain mengisi teko dengan air hingga penuh, mengosongkan teko dan menuangkan isi teko ke teko yang lain.

Sebagai contoh, terdapat 2 buah teko masing-masing berkapasitas 4 liter (teko A) dan 3 liter (teko B). Tidak ada tanda yang

menunjukkan batas ukuran pada kedua teko tersebut. Ada sebuah pompa air yang akan digunakan untuk mengisi air pada kedua teko tersebut. Permasalahannya: bagaimanakah kita dapat mengisi tepat 2 liter air ke dalam salah satu teko yang ada?



Gambar 3 Ilustrasi Masalah Teko Air
Penyelesaian:

A. Identifikasi ruang keadaan.

Permasalahan ini dapat direpresentasikan dengan 2 bilangan integer, yaitu *x* dan *y*:

- *x* = air yang diisi pada teko 4 liter (teko A);
- *y* = air yang diisi pada teko 3 liter (teko B);

Ruang keadaan: (*x,y*) sedemikian sehingga *x* ∈ {0,1,2,3,4} dan *y* ∈ {0,1,2,3}

B. Keadaan awal dan tujuan.

- Keadaan awal, kedua teko dalam keadaan kosong (0, 0);
- Keadaan tujuan, keadaan dimana terdapat 2 liter air pada salah satu teko, teko A (2,*n*) atau teko B (*n*,2).

C. Aturan-aturan.

Aturan-aturan dapat digambarkan pada tabel 1 berikut:

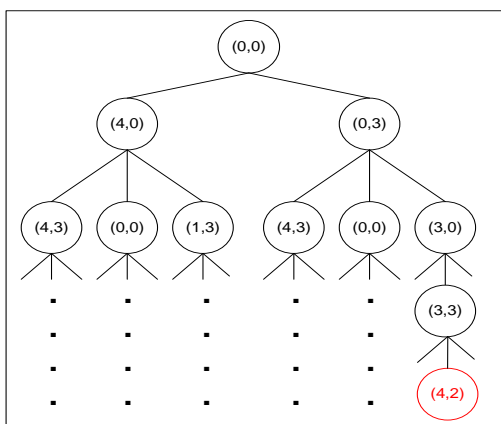
Tabel 1 Aturan-aturan masalah teko air

Aturan ke -	Jika	Maka
1	(<i>x,y</i>), <i>x</i> < 4	(4, <i>y</i>). Isi teko A.
2	(<i>x,y</i>), <i>y</i> < 3	(<i>x</i> ,3). Isi teko B.
3	(<i>x,y</i>), <i>x</i> > 0	(0, <i>y</i>). Kosongkan teko A dengan membuang airnya.
4	(<i>x,y</i>), <i>y</i> > 0	(<i>x</i> ,0). Kosongkan teko B dengan membuang airnya.
5	(<i>x,y</i>), <i>x</i> + <i>y</i> ≥ 4 dan <i>y</i> > 0	(4, <i>y</i> -(4- <i>x</i>)). Tuangkan air dari teko B ke teko A sampai teko A penuh.
6	(<i>x,y</i>), <i>x</i> + <i>y</i> ≥ 3 dan <i>x</i> > 0	(<i>x</i> -(3- <i>y</i>), <i>y</i>).Tuangkan air dari teko A ke teko B sampai teko B penuh.
7	(<i>x,y</i>), <i>x</i> + <i>y</i> ≤ 4 dan <i>y</i> > 0	(<i>x</i> + <i>y</i> ,0). Tuangkan seluruh air dari teko B ke teko A.
8	(<i>x,y</i>), <i>x</i> + <i>y</i> ≤ 3 dan <i>x</i> > 0	(0, <i>x</i> + <i>y</i>). Tuangkan seluruh air dari teko A ke teko B.

Keterangan: x = ukuran kendi A, y = ukuran kendi B.

D. Representasi ruang keadaan dengan pohon pelacakan.

Pencarian suatu solusi dapat dilukiskan dengan menggunakan pohon. Tiap-tiap *node* menunjukkan satu keadaan. Jalur dari *parent* ke *child* menunjukkan satu operasi. Tiap-tiap *node* pada pohon pelacakan ini memiliki *node-node child* yang menunjukkan keadaan yang dapat dicapai oleh *parent*.



Gambar 4 Representasi Ruang Keadaan pada Masalah Teko Air

Salah satu solusi yang dapat ditemukan terlihat pada tabel 2 berikut,

Tabel 2 Contoh solusi pada masalah teko air

Teko A (gallon)	Teko B (gallon)	Aturan yang dipakai
0	0	2
0	3	7
3	0	2
3	3	5
4	2 (solusi)	-

4. HASIL DAN PEMBAHASAN

4.1. Hasil

Implementasi sistem program ini mencakup spesifikasi kebutuhan perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*).

Spesifikasi Perangkat Keras dan Perangkat Lunak

Program ini sebaiknya dijalankan dengan menggunakan perangkat keras (*hardware*) yang mempunyai spesifikasi minimal sebagai berikut:

1. Prosesor Intel Pentium II 233 Mhz.
2. Memory 64 MB.
3. Harddisk dengan *free space* minimal 300 MB.
4. Monitor dengan resolusi 800×600 *pixel* dengan VGA *card* 1 MB.
5. Keyboard dan Mouse

Adapun sistem operasi (*operating system*) yang digunakan untuk menjalankan aplikasi ini adalah lingkungan sistem operasi *Windows 98, Me, 2000 dan XP*.

Pengujian Program

Sebagai contoh, *input* data sebagai berikut:

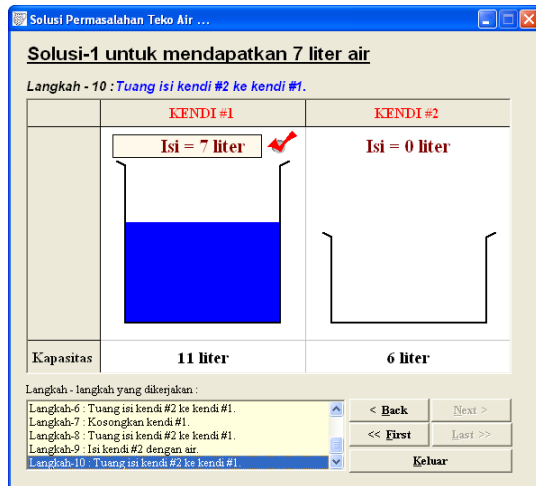
Contoh 1 :

1. Kapasitas kendi-1: **11 liter**
2. Kapasitas kendi-2: **6 liter**
3. Volume air yang diinginkan : **7 liter**
4. Solusi permasalahan teko air yang didapatkan :



Gambar 5 Tampilan Program

List Solusi permasalahan teko air dengan kapasitas kendi-1 = 11 liter, kapasitas kendi-2 = 6 liter dan volume air yang diinginkan = 7 liter



Gambar 6 Tampilan Program

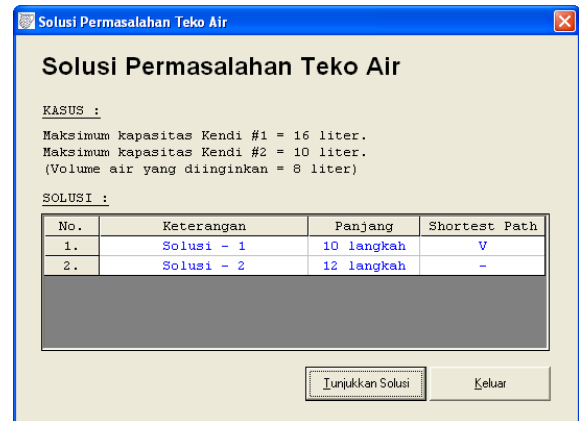
Solusi permasalahan teko air dengan kapasitas kendi-1 =11 liter, kapasitas kendi-2 = 6 liter dan volume air yang diinginkan = 7 liter

Langkah – langkah yang dikerjakan pada solusi:

- Langkah-0 : Kondisi Awal (Kendi kosong). (***Kendi-1 = 0 liter, Kendi-2 = 0 liter***)
- Langkah-1 : Isi kendi #2 dengan air. (***Kendi-1 = 0 liter, Kendi-2 = 6 liter***)
- Langkah-2 : Tuang isi kendi #2 ke kendi #1. (***Kendi-1 = 6 liter, Kendi-2 = 0 liter***)
- Langkah-3 : Isi kendi #2 dengan air. (***Kendi-1 = 6 liter, Kendi-2 = 6 liter***)
- Langkah-4 : Tuang isi kendi #2 ke kendi #1. (***Kendi-1 = 11 liter, Kendi-2 = 1 liter***)
- Langkah-5 : Kosongkan kendi #1. (***Kendi-1 = 0 liter, Kendi-2 = 1 liter***)
- Langkah-6 : Tuang isi kendi #2 ke kendi #1. (***Kendi-1 = 1 liter, Kendi-2 = 0 liter***)
- Langkah-7 : Isi kendi #2 dengan air. (***Kendi-1 = 1 liter, Kendi-2 = 6 liter***)
- Langkah-8 : Tuang isi kendi #2 ke kendi #1. (***Kendi-1 = 7 liter, Kendi-2 = 0 liter***)

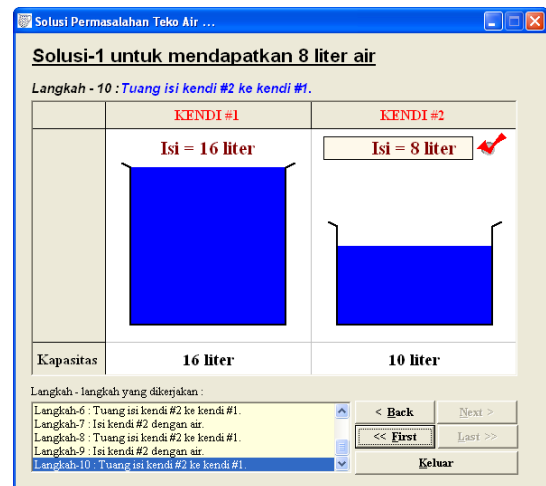
Contoh – 2 :

- Kapasitas kendi-1 : **16 liter**
- Kapasitas kendi-2 : **10 liter**
- Volume air yang diinginkan : **8 liter**
- Solusi permasalahan teko air yang didapatkan:



Gambar 7 Tampilan Program

List Solusi permasalahan teko air dengan kapasitas kendi-1 =16 liter, kapasitas kendi-2 = 10 liter dan volume air yang diinginkan = 8 liter



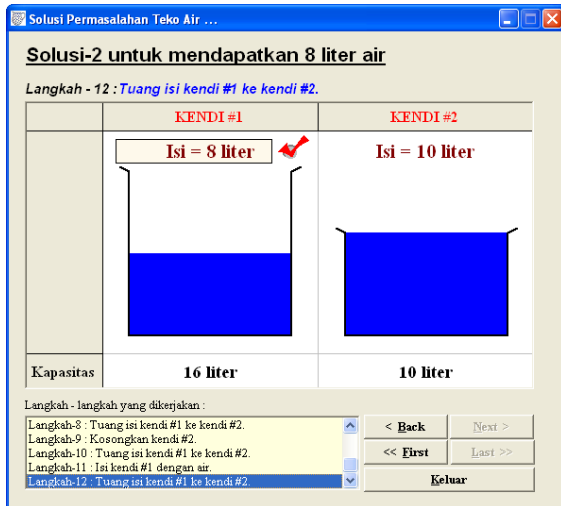
Gambar 8 Tampilan Program

Solusi-1 permasalahan teko air dengan kapasitas kendi-1 =16 liter, kapasitas kendi-2 = 10 liter dan volume air yang diinginkan = 8 liter

Langkah – langkah yang dikerjakan pada solusi-1:

- Langkah-0 : Kondisi Awal (Kendi kosong). (***Kendi-1 = 0 liter, Kendi-2 = 0 liter***)
- Langkah-1 : Isi kendi #2 dengan air. (***Kendi-1 = 0 liter, Kendi-2 = 10 liter***)
- Langkah-2 : Tuang isi kendi #2 ke kendi #1. (***Kendi-1 = 10 liter, Kendi-2 = 0 liter***)
- Langkah-3 : Isi kendi #2 dengan air. (***Kendi-1 = 10 liter, Kendi-2 = 10 liter***)
- Langkah-4 : Tuang isi kendi #2 ke kendi #1. (***Kendi-1 = 16 liter, Kendi-2 = 4 liter***)
- Langkah-5 : Kosongkan kendi #1. (***Kendi-1 = 0 liter, Kendi-2 = 4 liter***)
- Langkah-6 : Tuang isi kendi #2 ke kendi #1. (***Kendi-1 = 4 liter, Kendi-2 = 0 liter***)

- h. Langkah-7 : Isi kendi #2 dengan air. (**Kendi-1 = 4 liter, Kendi-2 = 10 liter**)
- i. Langkah-8 : Tuang isi kendi #2 ke kendi #1. (**Kendi-1 = 14 liter, Kendi-2 = 0 liter**)
- j. Langkah-9 : Isi kendi #2 dengan air. (**Kendi-1 = 14 liter, Kendi-2 = 10 liter**)
- k. Langkah-10 : Tuang isi kendi #2 ke kendi #1. (**Kendi-1 = 16 liter, Kendi-2 = 8 liter**)



Gambar 9 Tampilan Program

Solusi-2 permasalahan teko air dengan kapasitas kendi-1 =16 liter, kapasitas kendi-2 = 10 liter dan volume air yang diinginkan = 8 liter

Langkah – langkah yang dikerjakan pada solusi 2 yaitu:

- a. Langkah-0 : Kondisi Awal (Kendi kosong). (**Kendi-1 = 0 liter, Kendi-2 = 0 liter**)
- b. Langkah-1 : Isi kendi #1 dengan air. (**Kendi-1 = 16 liter, Kendi-2 = 0 liter**)
- c. Langkah-2 : Tuang isi kendi #1 ke kendi #2. (**Kendi-1 = 6 liter, Kendi-2 = 10 liter**)
- d. Langkah-3 : Kosongkan kendi #2. (**Kendi-1 = 6 liter, Kendi-2 = 0 liter**)
- e. Langkah-4 : Tuang isi kendi #1 ke kendi #2. (**Kendi-1 = 0 liter, Kendi-2 = 6 liter**)
- f. Langkah-5 : Isi kendi #1 dengan air.. (**Kendi-1 = 16 liter, Kendi-2 = 6 liter**)
- g. Langkah-6 : Tuang isi kendi #1 ke kendi #2. (**Kendi-1 = 12 liter, Kendi-2 = 10 liter**)
- h. Langkah-7 : Kosongkan kendi #2. (**Kendi-1 = 12 liter, Kendi-2 = 0 liter**)
- i. Langkah-8 : Tuang isi kendi #1 ke kendi #2. (**Kendi-1 = 2 liter, Kendi-2 = 10 liter**)
- j. Langkah-9 : Kosongkan kendi #2. (**Kendi-1 = 2 liter, Kendi-2 = 0 liter**)
- k. Langkah-10 : Tuang isi kendi #1 ke kendi #2. (**Kendi-1 = 0 liter, Kendi-2 = 2 liter**)

- l. Langkah-11 : Isi kendi #1 dengan air. (**Kendi-1 = 16 liter, Kendi-2 = 2 liter**)
- m. Langkah-12 : Tuang isi kendi #1 ke kendi #2. (**Kendi-1 = 8 liter, Kendi-2 = 10 liter**)

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

- 1. Perangkat lunak ini memiliki kemampuan untuk mencari solusi permasalahan teko air dengan 2 buah kendi dan menampilkan semua langkah – langkah yang dapat diambil untuk mendapatkan solusi atau volume air yang diinginkan.
- 2. Perangkat lunak dapat mencari solusi terpendek dari permasalahan teko air karena menggunakan metode pencarian melebar pertama (*breadth-first search*).
- 3. Pencarian tidak selalu menghasilkan solusi walaupun setelah semua *node* pada pohon pelacakan diperiksa dan dikembangkan.

5.2. Saran

- 1. Perangkat lunak dapat dikembangkan untuk dapat memecahkan permasalahan teko air dengan jumlah kendi lebih besar dari 2 buah.
- 2. Perangkat lunak dapat dikembangkan dengan menambahkan pencarian solusi dengan metode pelacakan lainnya.

DAFTAR PUSTAKA

- Ario Suryokusumo, *Microsoft Visual Basic 6.0*, PT. Elex Media Komputindo, 2001.
- Djoko Pramono, *Mudah menguasai Visual Basic 6*, PT. Elex Media Komputindo, 2002.
- Sri Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*, Penerbit Graha Ilmu, Edisi 3, 2010.
- Anita Desiani & Muhammad Arhami, *Konsep Kecerdasan Buatan*, Penerbit Andi Yogyakarta, 2011.
- <http://www.stormloader.com/ajy/waterjug.html>
- www.cut-the-knot.org/ctk/Water.shtml
- www.ithaca.edu/tpfaff/pdf/jugnew.pdf