

**Pratikum Artificial Intelligence
Rangkuman Pembelajaran
Materi 1-9**



Disusun Oleh :

21346016 Nafisa Aura Dwiyanti

Dosen Pengampu :

Widya Darwin S.kom., M.Pd.T

**PROGRAM STUDI INFORMATIKA
JURUSAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG**

2023

RINGKASAN MATERI PERTEMUAN 1 – 9

A. PENGANTAR KECERDASAN BUATAN

Apa itu kecerdasan ?

Kemampuan untuk belajar atau mengerti dari pengalaman, dan memahami pesan yang kontradiktif dan ambigu, juga menanggapi dengan cepat dan baik atas

situasi yang baru, dan menggunakan penalaran dalam memecahkan masalah serta menyelesaikannya dengan efektif.

Kategori definisi AI dikelompokkan 4 macam :

- System that think like humans
- System that act like humans
- System that think rationality
- System that act rationality

Dari Sudut Pandang Penelitian, Kecerdasan buatan adalah studi bagaimana membuat komputer dapat melakukan sesuatu sebaik yang dilakukan manusia.

Ada dua bagian utama dari AI yaitu :

- Basis pengetahuan (knowledge base) yaitu berisi fakta-fakta, teori, pemikiran dan hubungan komponen satu dengan yang lainnya.
- Motor inferensi (inference engine) yaitu Kemampuan menarik kesimpulan berdasar pengalaman. Berkaitan dengan representasi dan duplikasi proses tersebut melalui mesin (misalnya, komputer dan robot).

Konsep kecerdasan buatan ada 5 yaitu :

- Turing test
Metode Pengujian Kecerdasan (Alan Turing). Proses uji ini melibatkan seorang penanya (manusia) dan dua obyek yang ditanyai.
- Pemrosesan Simbolik
Sifat penting dari AI adalah bahwa AI merupakan bagian dari ilmu komputer yang melakukan proses secara simbolik dan non-algoritmik dalam penyelesaian masalah.
- Heuristic
Suatu strategi untuk melakukan proses pencarian (*search*) ruang problem secara efektif, yang memandu proses pencarian yang kita lakukan di sepanjang jalur yang memiliki kemungkinan sukses paling besar.
- Inferensi (penarikan Kesimpulan)
AI mencoba membuat mesin memiliki kemampuan berpikir atau mempertimbangkan (*reasoning*), termasuk didalamnya proses (*inferencing*) berdasarkan fakta-fakta dan aturan dengan menggunakan metode heuristik, dll.
- Pencocokan pola (pattern Matching)
Berusaha untuk menjelaskan obyek, kejadian (*events*) atau proses, dalam hubungan logik atau komputasional.

Tujuan kecerdasan buatan yaitu membuat computer lebih cerdas dan mengerti tentang kecerdasan juga membuat mesin lebih berguna. Kecerdasan buatan dan kecerdasan alami memiliki perbedaan yaitu kecerdasan buatan memiliki tingkat kecerdasan yang lebih permanen dan dapat memudahkan duplikasi dan penyebaran serta konsisten dan menyeluruh. Sedangkan kecerdasan alami bersifat lebih kreatif dan dapat melakukan proses pembelajaran secara langsung dan menggunakan focus yang luas sebagai referensi untuk pengambilan keputusan.

B. MASALAH, RUANG DAN PENCARIAN

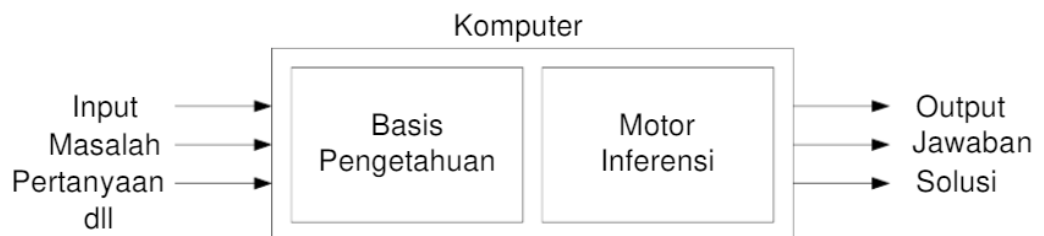
Untuk membangun system yang mampu menyelesaikan masalah, perlu dipertimbangkan 4 hal :

- Mendefenisikan masalah dengan tepat, dengan cara memahami spesifikasi yang tepat mengenai keadaan awaldan memberikan solusi yang diharapkan.
- Menganalisis masalah serta mencari beberapa Teknik penyelesaian masalah yang sesuai.
- Mempresentasikan pengetahuan yang perlu untuk menyelesaikan masalah.
- Memilih Teknik penyelesaian masalah yang terbaik.

Contoh nya yaitu posisi awal pada papan catur. Aturan untuk melakukan Gerakan secara legal yaitu :

- Aturan sangat berguna untuk menentukan Gerakan suatu bidak
- Untuk mempermudah kita harus menggunakan huruf (a,b,c,d,,e,f,g,h) Horizontal dan menggunakan angka (1,2,3,4,5,6,7,8) vertical.

Tujuan yang ingin dicapai adalah posisi pada papan catur yang menunjukkan kemenangan seseorang terhadap lawannya. Dengan ditandai posisi raja yang sudah tidak dapat bergerak lagi. Keadaan yaitu terdiri dari node-node yang menunjukkan keadaan yaitu keadaan awal dan keadaan baru yang akan dicapai dengan menggunakan operator.



Untuk membangun system yang mampu menyelesaikan masalah menggunakan AI :

1. Mendefenisikan masalah dengan tepat, mencakup spesifikasi yang tepat mengenai keadaan awal dan solusi yang diharapkan.

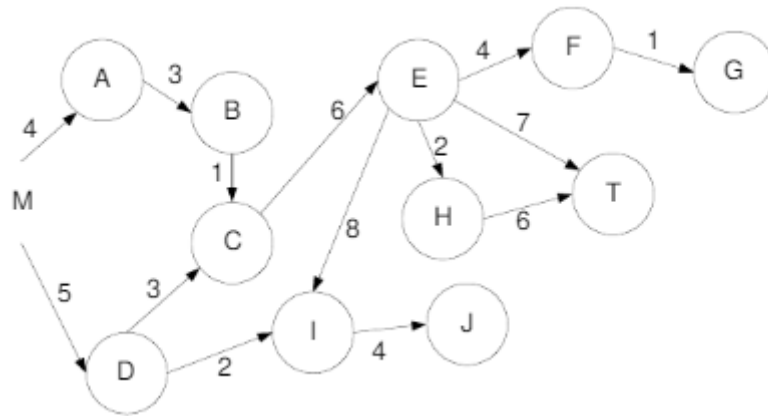
2. Menganalisis masalah tersebut dan mencari beberapa Teknik penyelesaian masalah yang sesuai.
3. Mereprantasikan pengetahuan yang perlu untuk menyelesaikan masalah tersebut.
4. Memilih Teknik penyelesaian masalah yang terbaik.

Untuk mendefenisikan suatu masalah kita perlu :

- Buat 'state space ' atau ruang masalah
- Tentukan keadaan awal (initial state)
- Tentukan keadaan akhir / tujuan (Goal state)
- Tentukan operatornya / aturannya

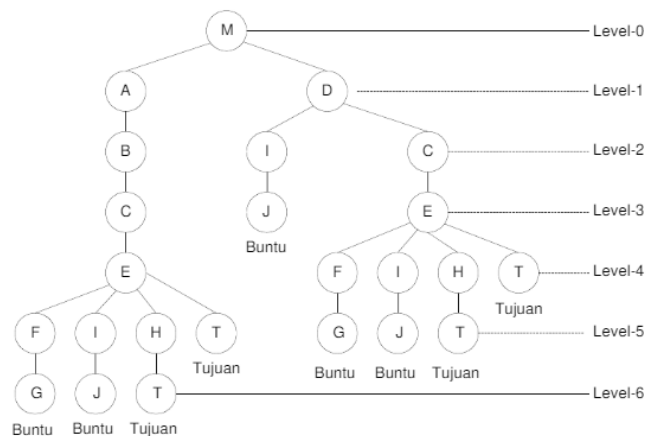
Beberapa cara mempresentasikan ruang masalah :

1. Graph keadaan

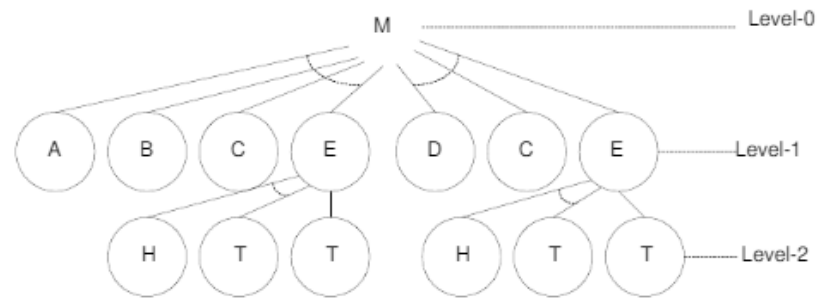


Graph berarah dengan satu tujuan (T)	Graph berarah yang menemui jalan buntu	Graph dengan siklus
<ul style="list-style-type: none"> • M-A-B-C-E-T • M-A-B-C-E-H-T • M-D-C-E-T • M-D-C-E-H-T 	<ul style="list-style-type: none"> • M-A-B-C-E-F-G • M-A-B-C-E-I-J • M-D-C-E-F-G • M-D-C-E-I-J • M-D-I-J 	<ul style="list-style-type: none"> • D-E-C-E-I-D

2. Pohon Pelacakan



3. Pohon AND / OR

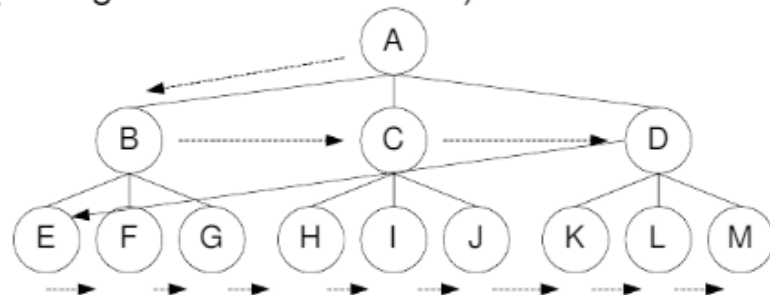


Pencarian Parsial (Blind Search)

1. Breadth-First Search

Pada metode ini semua node pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi node-node pada level $n+1$.

Pencarian akan dimulai dari node akar terus ke level 1 dari kiri ke kanan, kemudian berpindah ke level berikutnya. Demikian pula dari kiri ke kanan hingga ditemukan nya solusi.



Keuntungan :

- Tidak akan menemui jalan buntu
- Jika ada solusi, maka Breadth-First Search akan menemukannya. Dan jika ada lebih dari satu solusi, maka solusi minimum akan ditemukan.

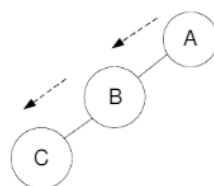
Kelemahan nya :

- Membutuhkan memori yang cukup banyak, karena menyimpan semua node dalam satu pohon.
- Membutuhkan waktu yang mencakup lama, karena akan menguji n level untuk mendapatkan solusi pada level yang ke $(n+1)$.

2. Depth-first Search

Pada tahap ini akan dilakukan proses pencarian pada semua anaknya sebelum dilakukan pencarian ke node-node yang selevel.

Pencarian dimulai dari akar ke level yang lebih tinggi. Proses ini akan diulangi terus hingga ditemukannya solusi.



Keuntungan nya :

- Membutuhkan memori yang relative kecil, karen hanya node-node pada perlintasan yang aktif saja yang disimpan.
- Secara kebetulan,metode ini akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan.

Kelemahan nya :

- Memungkinkan tidak ditemukannya tujuan yang diharapkan.
- Hanya akan menemukan solusi pada setiap pencarian.

3.Pencarian Heuristik (Heuristic Search)

Heuristic adalah sebuah Teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan.

Fungsi heuristic digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

C. REASONING

Penalaran secara literal Bahasa Inggris adalah *reasoning*. Berasal dari kata *reason*, yang secara literal berarti alasan. Berarti *reasoning* atau *to reason* adalah memberikan/memikirkan alasan. Mungkin beberapa dari kita masih belum memahami betul apa arti penalaran. Apakah orang yang salah nalar berarti orang bodoh? Tidak, orang salah nalar bisa terjadi karena *strategem* (kecoh yang bertujuan tertentu), salah nalar (*reasoning fallacy*), atau salah nalar karena aspek kemanusiaan. Jadi, bedakan antara penalaran dan kebodohan.

Penalaran dari aspek teoritis dapat didefinisikan sebagai proses berpikir logis dan sistematis untuk membentuk dan mengevaluasi suatu keyakinan terhadap pernyataan atau asersi. Tujuan dari penalaran adalah untuk menentukan secara logis dan objektif, apakah suatu pernyataan valid (benar atau salah) sehingga pantas untuk diyakini atau dianut. Dari definisi dan tujuan, dapat dilihat bahwa penalaran digunakan untuk mengevaluasi apakah suatu pernyataan itu dapat diyakini atau dianut atau kembali secara literal, kita melihat alasan (reason) dibalik suatu pernyataan.

kita akan membahas teknik reasoning (penalaran), yakni teknik penyelesaian masalah dengan cara merepresentasikan masalah kedalam basis pengetahuan (*knowledge base*) menggunakan logika atau bahasa formal (bahasa yang di pahami komputer). Disini, kami memfokuskan bahasan pada *propositional logic* (logika proposisi) dimana didalamnya terdapat tata bahasa pada proposition logic, semantic pada proposition logic, aturan inferensi pada proposition logic, masalah dunia wumpus (contoh penggunaan propositional logic), dan terakhir *knowledge base systems*, dan kami akan membahas secara singkat tentang *first order logic* atau *predicate calculus* (kalkulus predikat), dan *fuzzy logic* (logika samar).

Berbeda dengan *searching* yang merepresentasikan kedalam state dan ruang masalah serta menggunakan strategi pencarian untuk menemukan solusi, *reasoning* merepresentasikan masalah kedalam basis pengetahuan dan merupakan proses penalaran untuk menemukan solusi. Pada teknik *searching*, dua masalah utama yang dihadapi adalah: kesulitan dalam menentukan apakah aturan produksi (*operator*) sudah lengkap atau belum. Jika masalah yang dihadapi cukup kompleks, maka akan terjadi ketidakefisienan representasi keadaan (*state*). Misalkan untuk masalah 8-*puzzle* representasi suatu state adalah: delapan posisi angka, satu *parent*, dan maksimum 4 suksesor.

Bagaimana jika masalah yang dihadapi adalah permainan catur satu state harus direpresentasikan sebagai 32 posisi buah catur dan 1 state bisa memiliki sangat banyak suksesor (faktor percabangan sangat besar) jika satu buah catur rata-rata bisa melakukan 1 langkah, maka satu state bisa memiliki 16 suksesor. Dengan demikian ruang masalahnya akan sangat besar. Untuk solusi dilevel 10, maka BFS harus membangkitkan dan menyimpan 16^{10} state! Cara representasi ini tidak mungkin digunakan untuk prosesor dan kapasitas memory komputer saat ini. Bagaimana solusinya, solusi yang mungkin adalah menggunakan representasi yang jauh lebih sederhana yaitu menggunakan logic. Lima jenis logic untuk merepresentasikan basis pengetahuan dan untuk melakukan penalaran ditunjukkan pada Tabel 1.

Jenis Logic	Apa yang ada di dunia nyata	Apa yang dipercaya agent tentang fakta
Proposotional Logic	Fakta	Benar/salah/tidak diketahui
First-order logic	Fakta, objek, relasi	Benar/salah/tidak diketahui
Temporal logic	Fakta, objek, relasi, waktu	Benar/salah/tidak diketahui
Probability theory	Fakta	Derajat kepercayaan [0,1]
Fuzzy Logic	Derajat kebenaran	Derajat kepercayaan [0,1]

Logic yang paling sederhana adalah *proporsition logic* dimana suatu simbol menyatakan 1 proposisi (fakta) yang bisa bernilai benar atau salah. Simbol proposisi bisa dihubungkan dengan *Boolean Connectives* sehingga membentuk suatu kalimat. Bagaimanapun, untuk masalah yang kompleks, *propositional logic* tidak bisa digunakan secara praktis karena kita harus membangun banyak sekali fakta untuk merepresentasikan keadaan sederhana.

First-order logic merepresentasikan fakta dan aturan di dunia nyata dengan cara yang lebih baik yaitu menggunakan objek, predikat (relasi), dan *connectives* serta *quantifier* sehingga beberapa fakta sederhana dapat direpresentasikan kedalam suatu kalimat logika. Pada *first-order logic* semua relasi bersifat tetap pada waktu kapanpun. Untuk masalah yang kompleks dimana suatu relasi bisa berubah sesuai dengan perubahan waktu, maka diperlukan *logic* bentuk lain yang disebut *temporal logic*.

Pada *temporal logic*, dunia nyata direpresentasikan dengan sederetan interval waktu dan penalaran dilakukan berdasarkan fakta-fakta yang nilai kebenarannya bergantung pada deretan waktu tersebut. Pada masalah yang lain, nilai kebenaran tidak harus benar, salah atau tidak diketahui. Mungkin saja terdapat nilai-nilai lain diantaranya. *Probability theory* adalah *logic* yang merepresentasikan fakta dengan derajat kepercayaan yang bernilai antara 0 (tidak dipercaya sama sekali atau salah) sampai 1 (dipercaya sepenuhnya atau benar). Yang terakhir adalah *fuzzy logic* yang menggunakan derajat kebenaran dengan tingkat kepercayaan antara 0 dan 1. Misalkan, 'Bandung adalah kota yang nyaman' mungkin benar tingkat kepercayaan 0.4.

D. PLANNING

Planning berbeda dengan Search-Based Problem Solving dalam hal representasi goals, states, dan actions, juga berbeda dalam representasi dan pembangunan urutan-urutan action. Planning berusaha untuk mengatasi kesulitan-kesulitan yang dialami dalam Search-Based Problem Solving.

Planning adalah suatu metode penyelesaian masalah dengan cara memecah masalah ke dalam sub-sub masalah yang lebih kecil, menyelesaikan sub-sub masalah satu demi satu, kemudian menggabungkan solusi-solusi dari sub-sub masalah tersebut menjadisebuah solusi lengkap dengan tetap mengingat dan menangani interaksi yang ada antar sub masalah.

Pengujian keberfungsian suatu metode perencanaan dapat dilakukan pada suatu domain yang dinamakan Dunia Balok (Blocks-World). Dunia Balok dinilai cukup layak sebagai lahan pengujian karena tingkat kompleksitas permasalahan yang mungkin timbul di dalamnya. Di dalam Dunia Balok dikenal istilah initial-state dan goal-state yang masing-masing direpresentasikan oleh suatu komposisi dari sejumlah balok. Kemudian, ada satu set operator yang dapat diaplikasikan oleh sebuah tangan robot

untuk memanipulasi balok. Permasalahan yang ada di dalam Dunia Balok adalah: Rangkaian operator seperti apa yang dapat mengubah initial-state menjadi goal-state? Rangkaian operator tersebut biasa disebut sebagai Rencana Penyelesaian.

Dua metode perencanaan yang cukup populer dan sudah pernah diuji pada Dunia Balok adalah Goal-Stack-Planning (GSP) dan Constraint-Posting (CP). GSP dan CP memiliki kelemahan dan keunggulan masing-masing. Dari segi kemudahan implementasi dan biaya komputasi, GSP lebih unggul dibanding CP. Sedangkan, dari segi efisiensi Rencana Penyelesaian yang dihasilkan, CP pada umumnya lebih unggul dibanding GSP.

Tetapi, dari seluruh kemungkinan permasalahan yang timbul pada Dunia Balok, meskipun GSP dan CP mampu menghasilkan rencana-rencana penyelesaian, namun rencana-rencana penyelesaian yang dihasilkan pada umumnya tidak efisien [RIC91]. Padahal, rencana penyelesaian yang efisien adalah salah satu hal yang penting, terutama pada sistem-sistem raksasa.

Pada domain tertentu, pemecahan masalah menggunakan komputer perlu dilakukan dengan cara bekerja pada bagian-bagian kecil dari masalah secara terpisah kemudian menggabungkan solusi-solusi per bagian kecil tersebut menjadi sebuah solusi lengkap dari masalah. Karena, jika hal-hal tersebut tidak dilakukan, jumlah kombinasi state dari komponen masalah menjadi terlalu besar untuk dapat ditangani dipandang dari segi waktu yang tersedia.

Dua syarat untuk melakukan dekomposisi di atas adalah :

- Menghindari penghitungan ulang dari seluruh state masalah ketika terjadi perubahan dari suatu state ke state lainnya.
- Membagi masalah ke dalam beberapa sub masalah yang relatif lebih mudah untuk diselesaikan.

Penggunaan metode-metode yang terfokus pada cara mendekomposisi masalah ke dalam sub-sub masalah yang sesuai dan cara untuk mengingat dan menangani interaksi antar sub masalah ketika terjadi proses penyelesaian masalah tersebut diistilahkan dengan Perencanaan.

Pemecahan masalah dengan menggunakan perencanaan pada umumnya:

- Modus Goal-Directed, yaitu pencarian solusi dilakukan dari kondisi goal-state sampai ke kondisi initial-state yang dapat dicapai.
- Runut-Balik-Terpandu-Kebergantungan (Dependency-Directed-Backtracking) ketika menemukan jalan buntu.

Untuk berpindah dari satu state ke state lainnya, perencana diperkenankan untuk mengaplikasikan sejumlah operator. Sebuah sistem perencanaan pada umumnya perlu untuk mampu:

- Memilih operator
- Mengaplikasikan operator
- Mendeteksi ketika solusi telah tercapai
- Mendeteksi jalan-jalan buntu
- Mendeteksi ketika solusi yang hampir benar telah dicapai dan melakukan teknik khusus untuk membuat solusi tersebut menjadi benar

Pemilihan operator pada umumnya dilakukan dengan:

- Pertama, mengumpulkan kondisi pembeda goal-state dan current –state (current - state adalah suatu state yang menunjukkan kondisi saat ini).
- Kedua, mengidentifikasi operator yang dapat mengurangi perbedaan tersebut.

Pendeteksian bahwa solusi telah ditemukan dilakukan dengan cara menguji rangkaian operator yang telah dihasilkan. Jika rangkaian operator tersebut dapat mengubah kondisi initial-state menjadi goal-state, maka solusi telah ditemukan.

Pendeteksian jalan-jalan buntu dilakukan dengan: jika suatu rangkaian operator menghasilkan state yang diyakini bukan state antara goal-state dan initial-state atau jika kemajuan yang dicapai dianggap terlalu kecil atau menimbulkan permasalahan yang lebih sulit, maka rangkaian operator tersebut dianggap telah menemukan atau akan menuju jalan buntu.

Membuat solusi yang hamper benar menjadi benar dilakukan jika teknik yang dipakai dalam pencarian solusi hanya menghasilkan rangkaian solusi yang hamper benar. Perencana kemudian akan mengaplikasikan suatu metode tertentu untuk membuat solusi yang hamper benar tersebut menjadi benar.

Untuk menghindari pencatatan kondisi keseluruhan di setiap titik persimpangan (node) dalam pencarian solusi, pengaplikasian operator memerlukan tiga daftar predikat untuk mendeskripsikan perubahan kondisi, yaitu:

- **PRECONDITION** : predikat - predikat yang harus bernilai benar sebelum pengaplikasian operator
- **ADD**: predikat-predikat yang bernilai benar setelah pengaplikasian suatu operator.
- **DELETE** : predikat-predikat yang bernilai salah setelah pengaplikasian suatu operator

Ketiga daftar predikat tersebut selanjutnya disebut dengan daftar-PAD

Dunia Balok

Dunia Balok adalah sebuah domain dengan karakteristik:

- Memiliki sebuah permukaan datar tempat menyimpan balok, umumnya disebut dengan meja.
- Memiliki sejumlah balok kotak yang berukuran sama.
- Memiliki sebuah tangan robot yang dapat memanipulasi balok. Untuk memudahkan pendefinisian kondisi balok pada suatu state dalam.

Dunia Balok digunakan predikat-predikat berikut :

- **ON (A, B)** – Balok A menempel di atas balok B.
- **ONTABLE (A)** – Balok A berada di permukaan meja.
- **CLEAR (A)** – Tidak ada balok yang sedang menempel di atas balok A. Sedangkan, untuk tangan robot digunakan :
- **HOLDING (A)** – Tangan robot sedang memegang balok A.
- **ARMEMPTY** – Tangan robot tidak sedang memegang balok.

Permasalahan yang akan dicari rencana penyelesaiannya adalah permasalahan yang memenuhi spesifikasi berikut:

- Jumlah balok sama.
- Balok yang berada pada initial-state juga berada pada goal-state.
- Kondisi terdefinisi dengan benar.
- Tangan robot tidak sedang memegang balok.

Goal-Stack-Planning (GSP)

Dalam menyelesaikan sebuah masalah, GSP menggunakan sebuah stack untuk menampung kondisi-kondisi (kondisi goal dan kondisi-kondisi yang mungkin terjadi ketika pencarian solusi) dan operator-operator yang telah diajukan untuk memenuhi kondisi-kondisi tersebut. Juga, tergantung pada sebuah basis data yang menggambarkan current –state dan satu set operator yang dideskripsikan sebagai daftar predikat PRECONDITION, ADD dan DELETE atau daftar-PAD.

Langkah – langkah menyelesaikan masalah

- Langkah pertama GSP dalam menyelesaikan sebuah masalah adalah menempatkan kondisi-kondisi goal-state pada stack. Kondisi-kondisi tersebut akan disimpan di dalam sebuah slot stack.
- Langkah kedua, mengacu pada current -state, kondisi-kondisi goal-state yang belum tercapai dimasukkan ke dalam stack, masing-masing menempati sebuah slot. GSP tidak memiliki aturan khusus yang mengatur urutan pemasukan ke dalam stack dari kondisi-kondisi yang belum tercapai tersebut
- Langkah ketiga, slot terisi yang berada paling atas pada stack akan diperiksa. Hal-hal yang akan dilakukan bergantung pada kondisi slot tersebut. Kondisi yang mungkin terjadi pada slot tersebut adalah sebagai berikut:

Kondisi 1

Jika slot berisi kondisi yang sudah memenuhi current -state, tetapi slot tidak terletak di dasar stack dan juga tidak terletak di atas slot yang berisi operator, maka isi slot akan di-pop dari stack dan pemeriksaan dilanjutkan pada slot berikutnya.

Kondisi 2

Jika slot berisi kondisi yang belum memenuhi current –state maka isi slot akan di-pop dari stack. Kemudian, sebuah operator yang sesuai untuk mencapai kondisi tersebut akan dimasukkan ke dalam stack. Setelah itu, serangkaian kondisi yang dibutuhkan agar operator itu bisa diaplikasikan akan dimasukkan ke dalam sebuah slot stack. Selanjutnya, setiap kondisi dari rangkaian kondisi yang dibutuhkan operator agar dapat diaplikasikan tersebut akan dimasukkan ke dalam sebuah slot secara terurut, dimana kondisi yang harus dicapai paling akhir dimasukkan pertama kali.

Kondisi 3

Jika slot berisi kondisi atau rangkaian kondisi dan slot tersebut berada di atas slot yang berisi operator, maka isi slot teratas dari stack tersebut akan di-pop. Kemudian, operator pada slot berikutnya akan di-pop dan dimasukkan ke dalam antrian operator dalam rencana penyelesaian dan current –state di-update dengan mengaplikasikan operator tersebut pada current –state berdasarkan daftar-PAD.

Kondisi 4

Jika slot yang diperiksa adalah slot terdasar maka akan diuji kesamaan antara current - state dan goal-state. Jika sama (berarti goal-state telah tercapai) maka isi slot akan di-pop dan pencarian rencana penyelesaian dihentikan. Jika berbeda (goal-state belum tercapai) maka langkah ke dua diulangi. Jika kondisi yang terjadi bukan kondisi 4, setelah rangkaian tindakan yang bersesuaian dilakukan, langkah ketiga diulangi

GSP mungkin menemui jalan buntu yang disadari (meningkatkan biaya komputasi) atau pun yang tidak disadari (membuat rencana penyelesaian tidak efisien). Jalan buntu yang tidak disadari disebabkan karena GSP tidak memiliki aturan untuk mengurutkan pemasukan kondisi goal-state yang belum tercapai ke alam stack (langkah ke-2).

Constraint Posting (CP)

Berbeda dengan GSP yang berusaha mencapai kondisi goal-state dengan secara terurut berusaha mencapai sebuah kondisi pada suatu saat dengan operator-operator yang dibangkitkan pada rencana penyelesaian yang dihasilkan terurut untuk menyelesaikan kondisi pertama yang ingin dicapai, kedua dan seterusnya (linear-planning), CP bekerja secara paralel untuk mencapai kondisi-kondisi pada goal-state (non-linear-planning), berdasarkan kepada keyakinan bahwa: pada kebanyakan masalah, sub-sub masalah perlu untuk dikerjakan secara simultan [RIC91]. CP membangun rencana penyelesaian dengan secara bertahap menganalisa operator-operator, secara parsial mengurutkan operator-operator dan melekatkan variabel antar operator. Sehingga, pada suatu saat dalam proses penyelesaian masalah, bisa didapatkan sekumpulan operator meskipun cara yang jelas untuk mengurutkan operator-operator tersebut mungkin tidak dimiliki.

E. JARINGAN SYARAF TIRUAN

Jaringan saraf tiruan (Artificial Neural Network) merupakan salah satu sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Jaringan saraf tiruan mampu melakukan pengenalan kegiatan berbasis data masa lalu.

Data masa lalu akan dipelajari oleh jaringan syaraf tiruan sehingga mempunyai kemampuan untuk memberikan keputusan terhadap data yang belum pernah dipelajari. Sejak ditemukan pertamakali oleh McCulloch dan Pitts sistem jaringan syaraf tiruan berkembang pesat dan banyak digunakan oleh banyak aplikasi, jaringan syaraf tiruan (Artificial Neural Network) adalah suatu jaringan untuk memodelkan cara kerja sistem syaraf manusia (otak) dalam melaksanakan tugas tertentu.

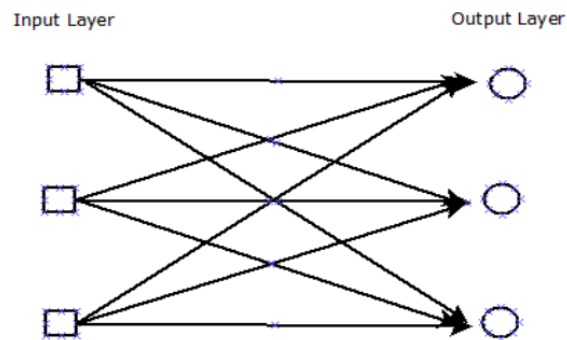
Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasi sel – sel penyusunan (neuron), sehingga memiliki kemampuan untuk melaksanakan tugas – tugas tertentu khususnya pengenalan pola dengan Efektifitas jaringan tertiggi. (Suyanto, 2013)

Sebagai sistem yang mampu menirukan perilaku manusia, umumnya sistem mempunyai ciri khas yang mampu menunjukkan kemampuan dalam hal :

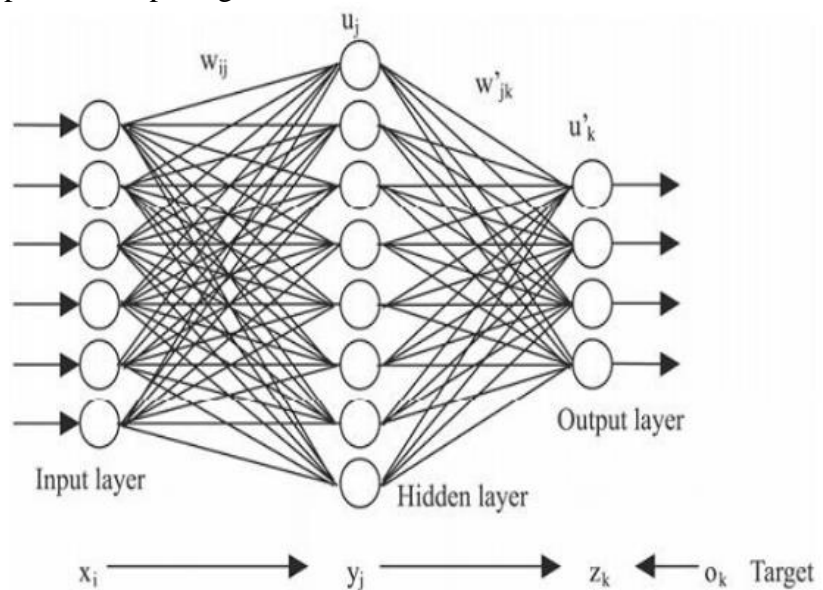
1. Menyimpan informasi,
2. Menggunakan informasi yang dimiliki untuk melakukan pekerjaan dan menarik kesimpulan,
3. Beradaptasi dengan keadaan baru,
4. Berkomunikasi dengan penggunanya.

Berdasarkan jumlah Layer arsitektur jaringan Artificial Neural Network (ANN) dapat di klasifikasikan menjadi dua kelas yang berbeda yaitu :

1. Jaringan layer tunggal (single layer network) merupakan semua unit input dalam jaringan ini dihubungkan dalam semua unit output, meskipun dengan bobot yang berbeda - beda, Berikut contoh jaringan single layer dapat di lihat pada gambar.



2. Jaringan layer jamak (multi layer network) adalah jaringan layer jamak yang merupakan perluasan dari layer tunggal jaringan layer jamak ini memperkenalkan satu atau layer tersembunyi (hidden layer) yang mempunyai simpun yang disebut neuron tersembunyi (hidden layer), berikut contoh jaringan layer jamak dapat di lihat pada gambar.



Berdasarkan arah sinyal masukan, arsitektur Artificial Neural Network (ANN) dapat di klasifikasikan menjadi dua kelas yang berbeda yaitu :

1. Unit Input ke unit output dalam arah maju (feed forward network) dalam jaringan umpan maju sinyal .
2. Jaringan syaraf tiruan dengan umpan balik (Recurrent network) pada jaringan ini terdapat neuron output yang member sinyal pada unit input.

Secara sederhana sistem jaringan syaraf tiruan adalah sebuah alat pemodelan data statis non-linier, selain itu juga dapat di gunakan untuk memodelkan hubungan yang kompleks antar input dan output untuk mengumpulkan data – data.

Suatu sistem jaringan syaraf tiruan memproses sejumlah besar informasi secara paralel dan terdistribusi hal ini terinspirasi oleh kinerja oleh model kerja otak sungguhan, berikut definisi dari Hecht-Nielsen mendefinisikan sistem saraf buatan sebagai berikut:“ Suatu neural network (NN), adalah suatu struktur pemroses informasi yang terdistribusi dan bekerja secara paralel, yang terdiri atas elemen pemroses (yang memiliki memori lokal dan beroperasi dengan informasi lokal) yang dikoneksi bersama dengan alur sinyal searah yang disebut koneksi.

Setiap elemen pemroses memiliki koneksi keluaran tunggal yang bercabang (fan out) ke sejumlah koneksi kolateral yang diinginkan (setiap koneksi membawa sinyal yang sama dari keluaran elemen pemrosesan tersebut). Keluaran 13 dari elemen proses tersebut dapat merupakan sebuah jenis persamaan matematis yang diinginkan. Seluruh proses yang berlangsung pada setiap elemen proses harus benar - benar dilakukan secara lokal, yaitu keluaran hanya bergantung pada nilai masukan pada saat itu yang diperoleh melalui koneksi dan nilai yang tersimpan dalam memori lokal".

Secara umum sistem jaringan syaraf juga memiliki keunggulan yang di gunakan untuk tugas atau pekerjaan yang kurang praktis jika di kerjakan secara manual, keunggulan antara lain :

1. Perkiraan Fungsi, atau Analisis Regresi, termasuk prediksi time series dan modeling.
2. Klasifikasi, termasuk pengenalan pola dan pengenalan urutan, serta pengambil keputusan dalam pengurutan.
3. Pengolahan data, termasuk penyaringan, pengelompokan, dan kompresi.
4. Robotik.

Suatu jaringan saraf tiruan memproses sejumlah besar informasi secara paralel dan terdistribusi, hal ini terinspirasi oleh model kerja otak biologis.

