

BAB 2

TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan merupakan kecerdasan yang dibuat oleh manusia dan biasanya ditanamkan pada komputer. Kecerdasan buatan dibuat dengan meniru proses yang ada dalam kehidupan. Terdapat banyak bidang yang termasuk di dalam kecerdasan buatan diantaranya algoritma genetik yang dibuat dengan meniru proses seleksi, persilangan, dan mutasi gen, *evolutionary algorithm* yang meniru proses seleksi makhluk hidup, *neural network* yang meniru cara kerja jaringan syaraf pada manusia, dan *fuzzy logic* yang meniru cara berpikir manusia di mana terdapat nilai kemungkinan dalam pengambilan keputusan.

Menurut Russell & Norvig (2009), pengertian kecerdasan buatan dapat dikelompokkan menjadi 4 kelompok yaitu:

1. *Thinking humanly* (berpikir seperti manusia)
2. *Thinking rationally* (berpikir secara rasional)
3. *Acting humanly* (bertindak seperti manusia)
4. *Acting rationally* (bertindak secara rasional)

Menurut Suyanto (2007), definisi berpikir dan bertindak seperti manusia memiliki definisi yang terlalu luas karena sampai saat ini, proses berpikir manusia sendiri masih belum dapat dipetakan sehingga masih kurang cocok untuk mendeskripsikan kecerdasan buatan yang ada pada saat ini. Definisi bertindak secara rasional, menurut Suyanto (2007) merupakan definisi yang paling tepat

karena sudah melalui proses berpikir secara rasional dan tindakan yang dilakukan sesuai dengan hasil pemikiran yang rasional tersebut.

Kecerdasan NPC pada penelitian ini menggunakan definisi bahwa kecerdasan buatan berarti melakukan tindakan secara rasional. NPC yang dibuat akan berusaha untuk memenangkan pertandingan dengan cara bertindak secara rasional berdasarkan pemikiran yang rasional setelah menerima input dari kondisi lingkungan sekitarnya.

Pada masa “*The rise of artificial intelligence: the era of great expectations*” yang terjadi pada tahun 1956 sampai akhir 1960-an, peneliti mencoba untuk menemukan metode umum untuk menyelesaikan semua permasalahan dan mengalami kegagalan. Pada masa ini juga para peneliti mengklaim akan membuat mesin cerdas untuk segala tujuan berdasarkan pada pengetahuan manusia pada tahun 1980 dan melebihi kemampuan manusia pada tahun 2000, dan hal ini tidak terbukti pada saat ini (tahun 2012) di mana masih belum ada mesin cerdas yang memiliki kemampuan seperti manusia (Negnevitsky, 2004).

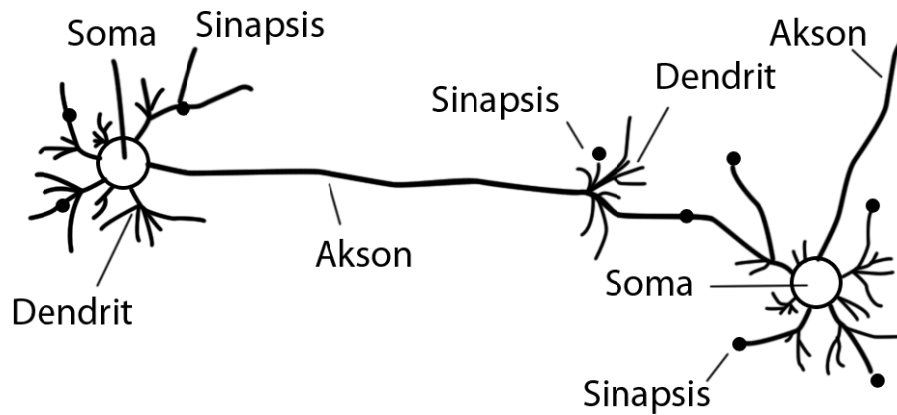
Klaim yang terlalu optimis di atas terjadi karena pada awalnya percobaan yang menerapkan kecerdasan buatan diterapkan pada masalah yang sederhana yang sebenarnya dapat diselesaikan dalam *polynomial time*, akan tetapi terdapat banyak permasalahan yang sebenarnya jauh lebih rumit dan tidak bisa diselesaikan dalam *polynomial time*. Cook (1971) dan Karp (1972) mengemukakan teori mengenai NP-Complete berupa permasalahan yang membutuhkan waktu *non-polynomial* untuk diselesaikan. Untuk menghindari masalah klaim yang terlalu optimis terhadap metode *novelty search*, maka pada tesis ini dilakukan penelitian untuk melakukan pengujian terhadap *novelty search*

pada masalah yang bersifat lebih dinamis dibandingkan dengan permasalahan *maze navigation* dan *biped locomotion*.

2.2 Artificial Neural Network

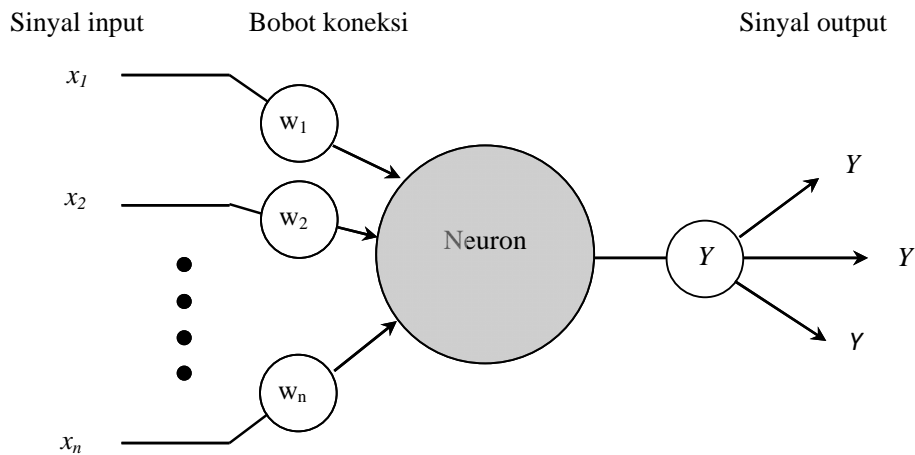
Neural network (NN) merupakan model berpikir yang didasarkan pada cara kerja otak manusia. Otak terdiri atas hampir 10 juta neuron / *node* dan 60 miliar sinapsis / koneksi antar neuron (Shepherd & Koch, 1990). Neuron pada otak manusia memiliki struktur yang sangat sederhana yang terdiri atas soma, dendrit, dan akson. Sinyal yang masuk ke otak manusia akan diteruskan dari suatu neuron ke neuron lain melalui reaksi elektrokimia. Sinyal yang diteruskan dari suatu sinapsis akan menyebabkan perubahan potensial pada soma. Ketika perubahan potensial ini melewati ambang batas, maka potensial listrik akan dikirimkan melalui akson yang akan melewati sinapsis lain dan masuk ke soma neuron lain dan menyebabkan perubahan potensial.

NN memiliki sifat plastisitas di mana struktur dari NN dapat berubah berdasarkan stimulasi yang diterimanya. Sinapsis yang menghubungkan suatu neuron dengan neuron lain dapat putus, membuat koneksi baru, ataupun berubah kekuatan koneksinya sesuai dengan stimulasi oleh potensial listrik yang melaluinya.



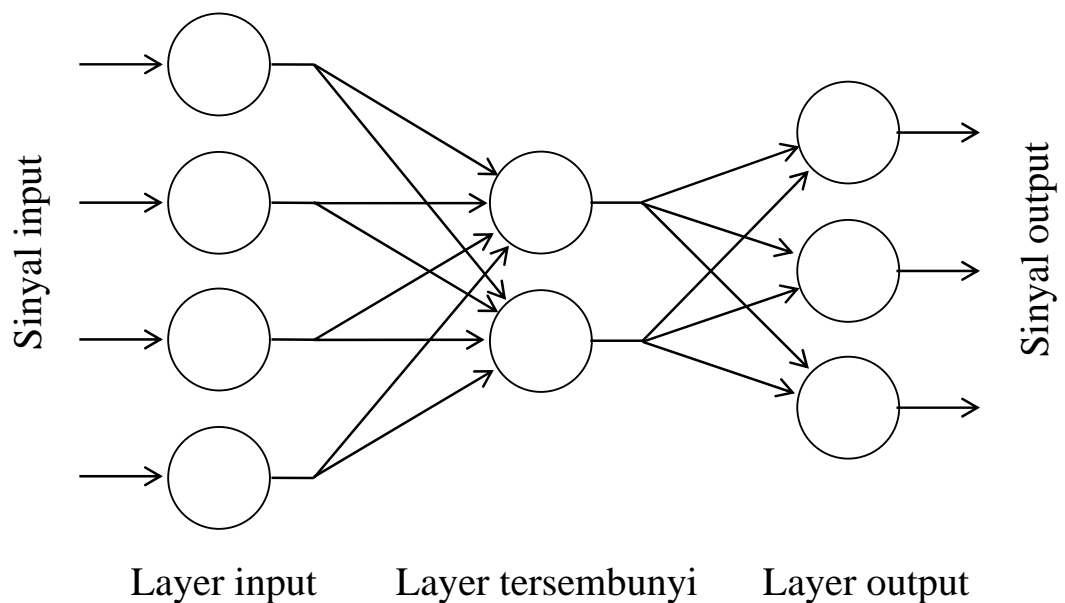
Gambar 1: Neural Network (Negnevitsky, 2004)

ANN memodelkan *neural network* pada otak manusia dan menerapkannya pada pembuatan kecerdasan buatan. Neuron terkoneksi dengan neuron lain melalui suatu koneksi yang memiliki bobot. Neuron menerima sinyal dari sejumlah sinyal input kemudian melakukan perhitungan berdasarkan sinyal input yang diterimanya dan menghasilkan sinyal output. Sinyal output keluar melalui koneksi keluar neuron dan mentransfer sinyal yang sama ke semua koneksi masuk neuron lain yang berhubungan dengan koneksi keluar neuron tersebut.



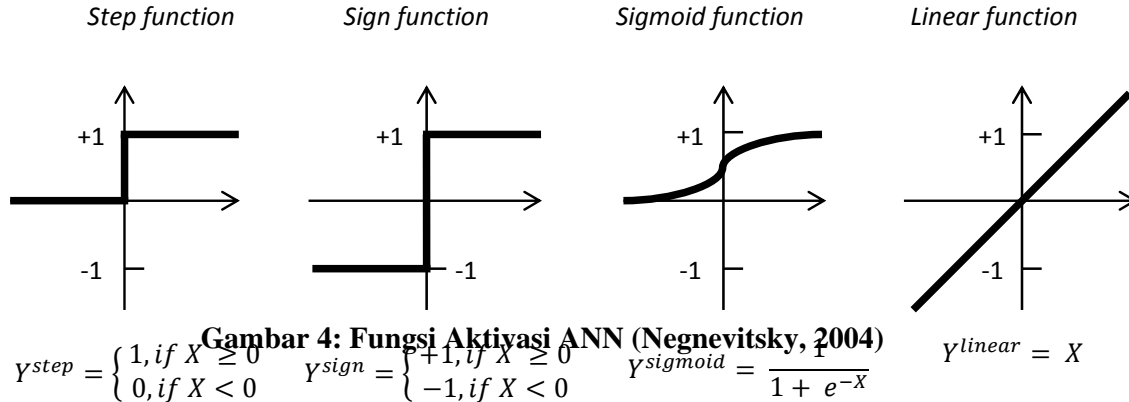
Gambar 2: Artificial Neural Network (Negnevitsky, 2004)

ANN yang dipakai pada saat ini biasanya dibagi menjadi 3 layer yang berupa 1 layer input, beberapa layer tersembunyi, dan 1 layer output. Layer input akan menerima sinyal yang berasal dari lingkungan sekitarnya, kemudian memproses sinyal masuk menjadi sinyal keluar yang diteruskan ke beberapa neuron pada layer tersembunyi. Layer tersembunyi akan memproses sinyal input yang berasal dari layer input dan diteruskan ke layer tersembunyi berikutnya atau layer output sesuai dengan ANN yang dibuat. Layer output merupakan layer terakhir yang digunakan untuk menentukan perilaku dari sistem yang menggunakan model ANN yang telah dibuat.



Gambar 3: Layer pada ANN (Negnevitsky, 2004)

Pemrosesan sinyal input menjadi sinyal output pada setiap neuron dilakukan melalui suatu fungsi aktivasi. Beberapa fungsi aktivasi yang umum dipakai adalah sebagai berikut:



2.3 Algoritma Genetik

Algoritma genetik merupakan salah satu cabang *evolutionary computation*. *Evolutionary computation* merupakan metode untuk mensimulasikan evolusi pada kehidupan di dalam komputer. Evolusi sendiri pada kehidupan melakukan optimisasi untuk mempertahankan dan menaikkan kemampuan populasi suatu spesies untuk bertahan hidup dan bereproduksi. *Evolutionary computation* memiliki 3 cabang utama, yaitu algoritma genetik, strategi evolusi, dan pemrograman genetik, akan tetapi strategi evolusi dan pemrograman genetik sebenarnya juga merupakan penurunan dari algoritma genetik.

Algoritma genetik merupakan algoritma pencarian stokastik yang berdasarkan pada evolusi biologis (Negnevitsky, 2004). Algoritma genetik terdiri atas 3 proses utama, yaitu:

1. Seleksi

Seleksi merupakan proses pemilihan kromosom-kromosom pada populasi berdasarkan nilai *fitness* masing-masing kromosom. Terdapat berbagai cara

untuk melakukan seleksi pada populasi seperti *roulette wheel selection* / *fitness proportionate selection*, *rank selection*, dan *tournament selection*.

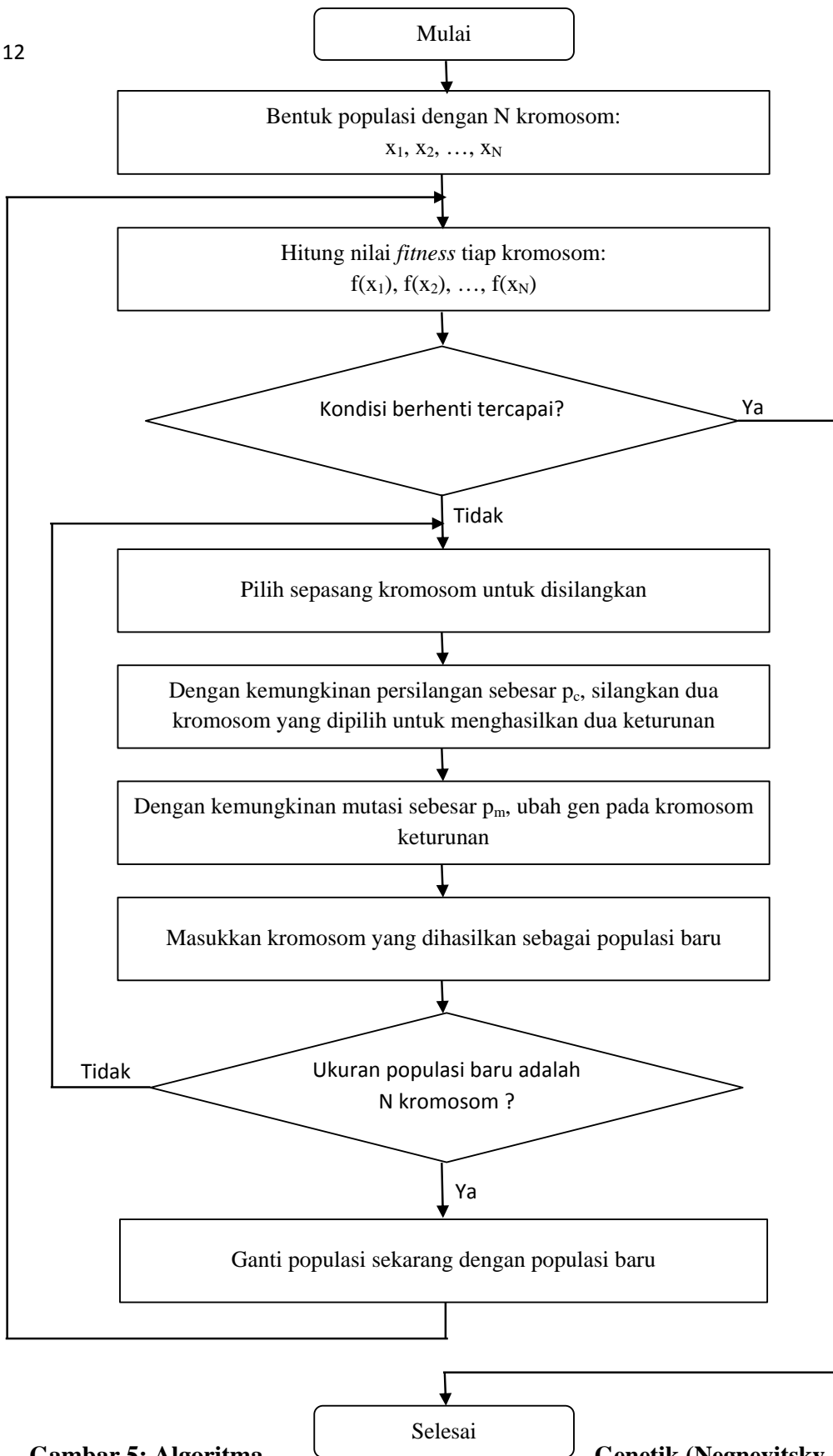
2. Persilangan

Persilangan merupakan proses menghasilkan keturunan berdasarkan hasil seleksi. Dua individu yang terpilih akan disilangkan dengan menggunakan operator persilangan. Kromosom dari individu yang berisi gen-gen akan dipecah menjadi dua bagian di mana masing-masing bagian pada suatu individu akan disilangkan dengan bagian lain dari individu lainnya.

3. Mutasi

Mutasi terjadi setelah proses persilangan selesai. Mutasi akan mengubah nilai suatu gen dalam kromosom. Seringkali mutasi akan mengurangi nilai *fitness* dari individu, akan tetapi mutasi diperlukan agar populasi yang dihasilkan dapat keluar dari solusi lokal optimal. Populasi terjebak dalam lokal optimal karena proses seleksi dan persilangan dari algoritma genetik menghasilkan populasi yang terlalu mirip antara satu dengan lainnya. Dengan adanya mutasi, keanekaragaman pada populasi dapat terjaga sehingga populasi tidak terjebak dalam lokal optimal.

Secara umum, proses pada algoritma genetik dapat digambarkan sebagai berikut:



Gambar 5: Algoritma

Genetik (Negnevitsky,

2004)

2.4 NEAT

NEAT merupakan penggabungan dari ANN dan algoritma genetik di mana jaringan pada ANN dievolusi dengan menggunakan algoritma genetik. Tujuan dari pembuatan metode *neuroevolution* adalah mengevolusi ANN sehingga dapat belajar selama masa hidupnya (Risi, Vanderbleek, Hughes, & Stanley, 2009).

Salah satu masalah menggunakan ANN adalah menentukan secara heuristik jumlah *node* pada ANN dan topologi yang cocok untuk suatu permasalahan. Jawaban untuk permasalahan ini tidak dapat ditentukan melalui pengalaman empiris maupun berdasarkan metode analisa tertentu karena sangat sedikit yang diketahui mengenai solusi yang diharapkan (Stanley & Miikkulainen, 2004).

Biasanya pemecahan masalah menggunakan ANN memerlukan penentuan topologi jaringan terlebih dahulu sebelum eksperimen dilakukan. Topologi jaringan mungkin harus diatur berkali-kali sebelum peneliti mendapatkan topologi jaringan neural yang pas untuk suatu permasalahan. Hal ini menyebabkan permasalahan pada ANN menjadi permasalahan optimisasi bobot koneksi pada jaringan untuk menghasilkan ANN yang dapat menyelesaikan suatu permasalahan. Akan tetapi, bobot koneksi pada jaringan bukan hanya aspek yang menentukan perilaku pada suatu ANN. Topologi ANN juga memiliki dampak terhadap perilaku ANN dalam menyelesaikan suatu permasalahan (Stanley & Miikkulainen, 2002).

Terdapat beberapa penelitian yang dilakukan untuk mengatasi permasalahan dalam pembentukan topologi ANN. Angeline, Saunders, & Pollack

(1993) melakukan evolusi terhadap topologi ANN dan menemukan bahwa evolusi terhadap topologi ANN dapat menghasilkan perilaku baru yang tidak dapat dihasilkan oleh model ANN standar.

Perubahan terhadap topologi ANN dapat dikategorikan menjadi tiga cara, yaitu melakukan penambahan neuron dan koneksi dari struktur ANN yang paling sederhana (metode konstruktif), menghilangkan neuron dan koneksi dari struktur ANN yang besar (metode destruktif), ataupun gabungan keduanya. Permasalahan pada metode konstruktif adalah topologi ANN dapat menjadi sangat besar sehingga tidak efisien untuk digunakan (Giles et al., 1995). Sedangkan permasalahan pada metode destruktif adalah topologi ANN harus dimulai dari struktur acak yang kompleks dan tidak diketahui seberapa kompleks struktur yang sebenarnya dibutuhkan di awal. Pemangkasan terhadap topologi ANN sendiri dapat dilakukan dengan memberikan pengurangan terhadap nilai *fitness* pada ANN yang lebih kompleks dan memiliki hasil yang mirip dengan ANN yang lebih sederhana (Zhang & Mühlenbein, 1993).

Evolusi terhadap topologi bersamaan dengan bobot pada jaringan ANN dapat meningkatkan performa dari *neuroevolution* (Stanley & Miikkulainen, 2002). Dengan menggunakan NEAT, topologi ANN dievolusi dari topologi paling sederhana dan hanya akan ditambahkan kompleksitasnya apabila diperlukan. Dengan topologi yang sederhana, maka ANN dapat bekerja dengan cepat. Selain itu, dengan adanya proses evolusi topologi pada NEAT, maka struktur ANN akan berkembang sehingga dapat menyelesaikan permasalahan yang kompleks.

Evolusi terhadap topologi ANN memiliki beberapa tantangan secara teknikal, yaitu:

1. Apakah terdapat representasi genetik pada topologi ANN untuk dapat melakukan persilangan ANN dengan manfaat persilangan yang berarti?
2. Bagaimana cara memproteksi inoasi baru agar tidak hilang secara prematur sebelum memiliki manfaat dalam menyelesaikan permasalahan dapat diproteksi sehingga tidak hilang dari populasi secara prematur?
3. Bagaimana cara meminimalkan evolusi topologi ANN tanpa menggunakan fungsi yang secara khusus mengukur kompleksitas?

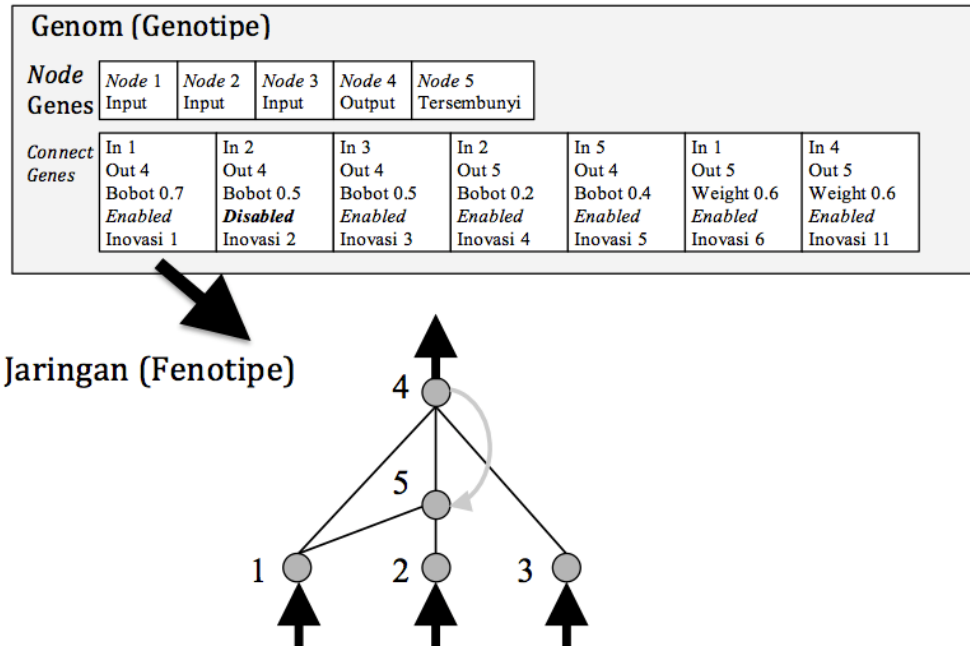
Untuk menyelesaikan permasalahan yang ada, Stanley & Miikkulainen, (2002) menggunakan pendekatan berupa:

1. *Historical marking*

Historical marking digunakan untuk menyelesaikan masalah dalam pembuatan topologi ANN agar topologi ANN memiliki representasi genetik yang memiliki manfaat persilangan yang berarti.

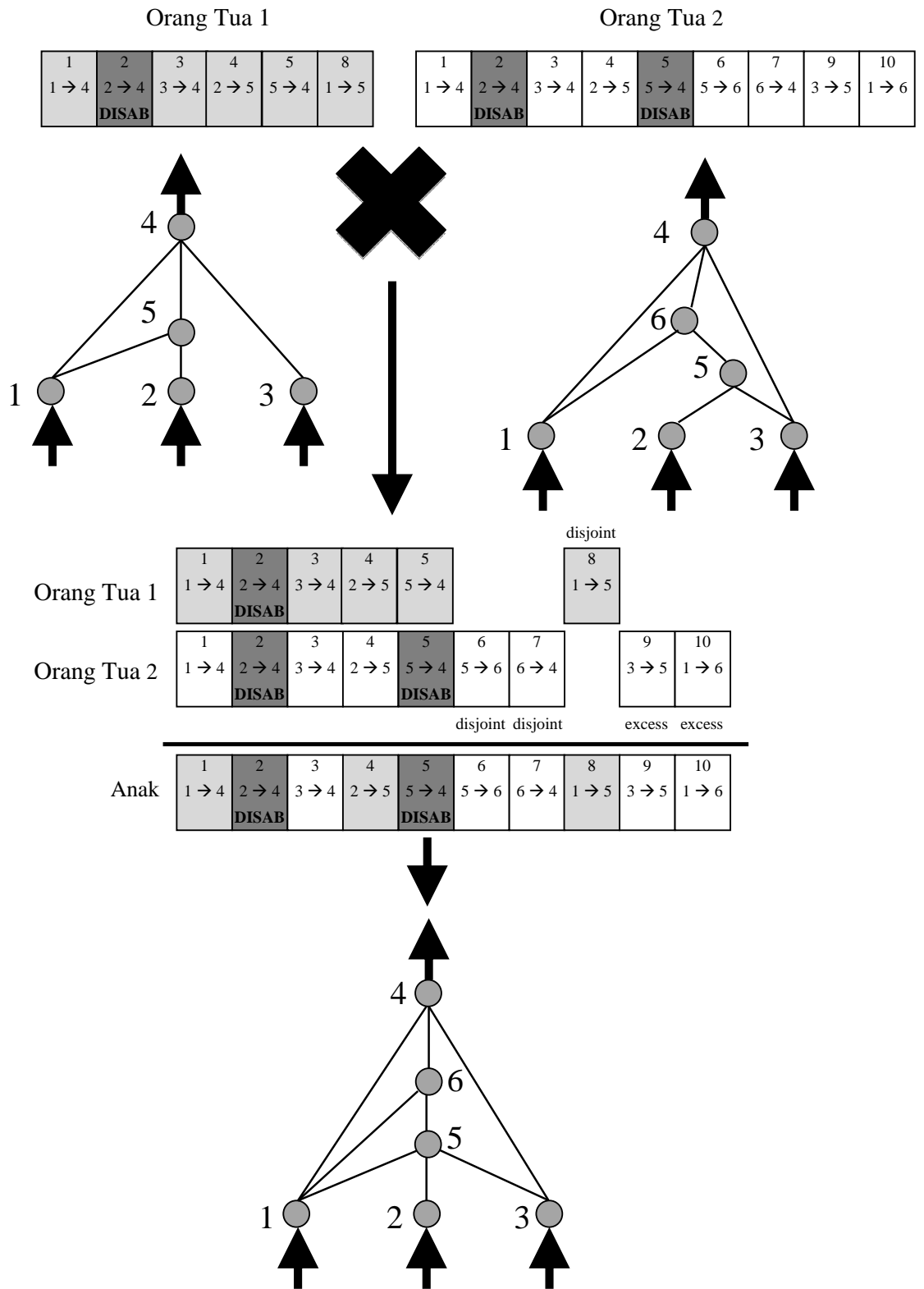
Permasalahan mengenai *historical marking* juga dihadapi oleh makhluk hidup. Pada makhluk hidup, terkadang genom dapat memiliki ukuran yang berbeda. Hal ini disebabkan karena adanya proses mutasi di mana terjadi penambahan suatu genom. Diperlukan suatu cara dalam melakukan persilangan antara makhluk hidup yang memiliki ukuran genom yang berbeda. Pada makhluk hidup, permasalahan ini diselesaikan dengan menggunakan pendekatan homologi, di mana 2 gen dikatakan homolog apabila merupakan alel dari suatu sifat yang sama. Pada *E. coli*, terdapat suatu proses yang

bernama *synapsis*, di mana suatu protein spesial bernama RecA akan menata gen yang homolog sebelum terjadi persilangan.



Gambar 6: Genotipe dan Fenotipe NEAT (Stanley, 2004)

Pada NEAT, permasalahan *historical marking* diselesaikan dengan cara membuat representasi gen yang memungkinkan terjadinya penambahan struktur tanpa kehilangan bentuk struktur yang sudah ada. Gen direpresentasikan dalam bentuk *array* yang memiliki beberapa informasi pada setiap elemennya. Setiap elemen akan memiliki nilai inovasi yang akan digunakan pada saat melakukan persilangan.



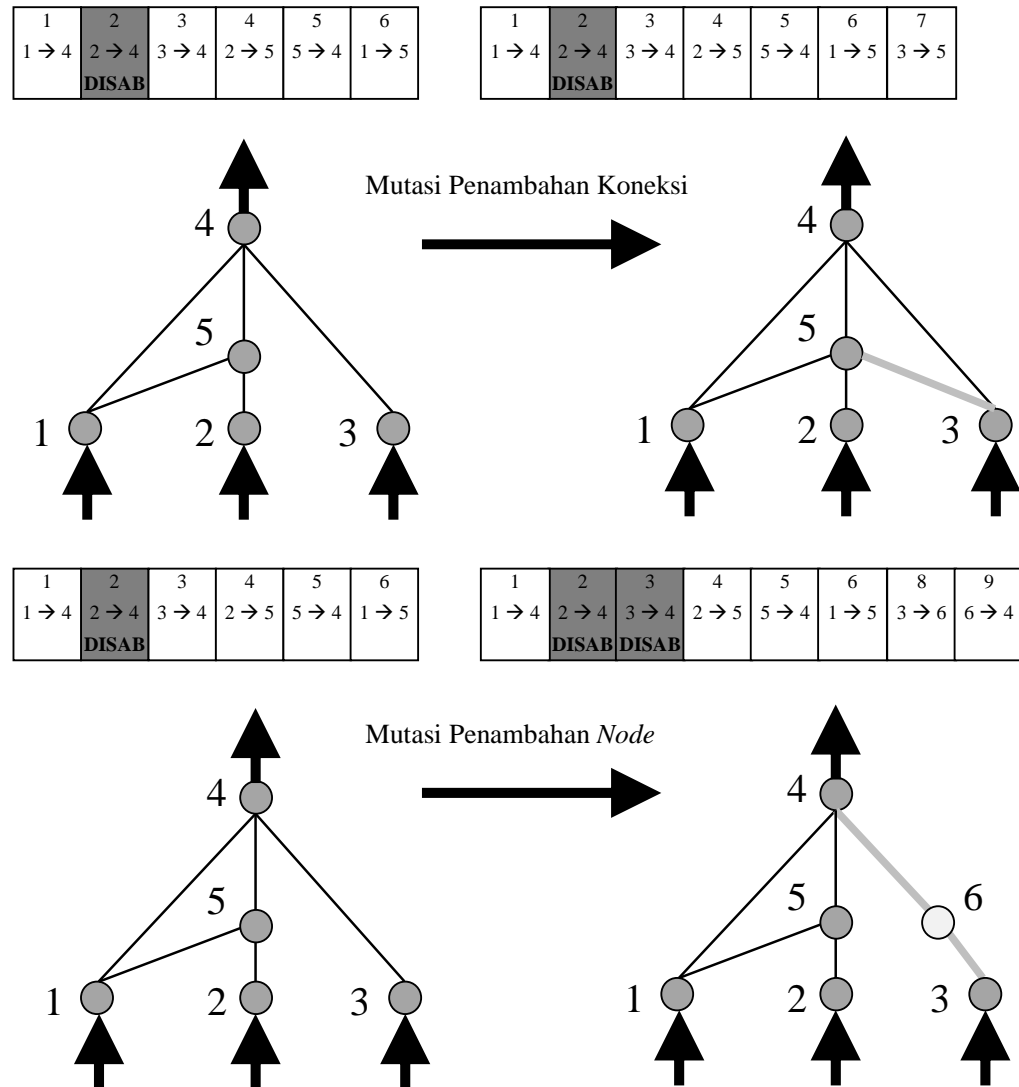
Gambar 7: Persilangan NEAT (Stanley, 2004)

Pada saat melakukan persilangan, 2 individu yang memiliki topologi yang mirip akan dipilih untuk menjadi orang tua dalam proses persilangan. Kemiripan topologi ANN diukur dengan menggunakan perbedaan bobot gen pada gen yang memiliki nomor inovasi yang sama. Untuk setiap gen yang memiliki nomor inovasi yang sama, maka persilangan dapat langsung dilakukan, akan tetapi apabila terdapat kelebihan gen pada suatu individu dibandingkan dengan individu lainnya pada orang tua, maka kelebihan gen itu akan dimasukkan pada gen anak.

2. *Speciation*

Speciation digunakan untuk menyelesaikan masalah dalam memproteksi inovasi baru agar tidak hilang dari populasi secara prematur.

Inovasi pada NEAT terjadi pada saat dilakukan penambahan struktur baru pada ANN melalui mutasi. Mutasi pada NEAT dilakukan dengan 2 cara, yaitu melalui penambahan *node* dan melalui penambahan koneksi. Mutasi dengan cara menambahkan koneksi dilakukan dengan cara menambahkan elemen pada *array* koneksi dengan nomor inovasi baru yang berisi 2 buah *node* yang akan disambungkan. Mutasi dengan cara menambahkan *node* dilakukan dengan mencari suatu elemen lama pada *array* koneksi yang berisi koneksi antara 2 buah *node*, kemudian menghilangkan elemen tersebut. Setelah itu, berdasarkan koneksi tersebut, ditambahkan 1 buah elemen baru pada *array node* yang berisi 1 buah *node* baru dan 2 buah nomor inovasi baru pada *array* koneksi yang menyambungkan 2 buah *node* pada koneksi yang baru dihilangkan dengan *node* yang baru ditambahkan.



Gambar 8: Mutasi NEAT (Stanley, 2004)

Permasalahan pada inovasi baru adalah terjadinya penurunan nilai *fitness* pada saat terjadi inovasi yang mengakibatkan inovasi baru akan hilang karena nilai *fitness* yang lebih rendah dibandingkan individu lainnya. Diperlukan waktu selama beberapa generasi untuk melakukan optimisasi terhadap struktur baru ini.

Pada makhluk hidup, permasalahan ini tidak terjadi karena sifat alami makhluk hidup di mana biasanya spesies yang berbeda biasanya akan

berkompetisi pada lingkungan yang berbeda untuk mendapatkan makanan yang berbeda. Hal ini menyebabkan inovasi baru tidak mengalami masalah karena terjadi kompetisi di bidang yang berbeda antar spesies.

Pada NEAT, permasalahan ini diselesaikan dengan melakukan *explicit fitness sharing* di mana tiap individual dengan gen yang mirip akan disatukan pada kelompok yang sama. Dengan demikian, inovasi pada NEAT akan terproteksi pada kelompok spesies yang ada.

3. *Incremental Growth*

Incremental growth digunakan untuk menyelesaikan permasalahan minimalisasi struktur ANN.

Pada NEAT, topologi ANN dimulai dari topologi yang paling sederhana di mana hanya terdapat layer input dan layer output. Seiring dengan berjalannya generasi, maka NEAT akan menambahkan struktur hanya apabila penambahan struktur membawa NEAT untuk menemukan solusi yang lebih baik. Dengan memulai dari topologi ANN yang paling sederhana, maka NEAT memastikan bahwa suatu solusi akan ditemukan terlebih dahulu pada topologi ANN yang lebih sederhana. Dengan menggunakan topologi ANN yang sederhana, maka pemrosesan informasi pada ANN dapat dilakukan lebih cepat dibandingkan dengan pemrosesan informasi pada ANN yang kompleks.

2.5 *Novelty Search*

Novelty search merupakan salah satu bentuk dari *non-objective search*. Pada algoritma genetik, fungsi *fitness* biasanya diukur berdasarkan kedekatan antara kondisi yang telah dicapai pada saat ini dibandingkan dengan kondisi tujuan yang ingin dicapai. Pendekatan ini terkadang dapat menghalangi agen

untuk mencapai tujuan yang diinginkan. Hal ini terjadi karena adanya jebakan dalam bentuk jalan yang harus ditempuh untuk mencapai tujuan harus melewati kondisi di mana nilai *fitness* justru mengalami penurunan. Adanya jebakan di mana fungsi *fitness* akan membawa proses pencarian ke arah lokal optimal, namun menjauhi global optimal merupakan salah satu permasalahan yang paling besar yang dihadapi oleh *evolutionary computation* (Kistemaker & Whiteson, 2011).

Pendekatan dalam menghindari jebakan telah banyak diteliti. Untuk menghindari konvergensi ke lokal optimal yang terlalu cepat biasanya dilakukan teknik untuk menjaga keanekaragaman populasi. Untuk menjaga keanekaragaman populasi, biasanya terdapat metode yang bernama spesiasi di mana sekelompok kromosom yang dianggap masih dalam satu spesies akan saling berkompetisi dalam spesies itu sendiri.

Hu et al. (2005) melakukan spesiasi dengan cara membagi populasi ke dalam level-level berdasarkan nilai *fitness*. Pembagian ini mengikuti prinsip kompetisi yang adil di mana kompetisi dilakukan hanya antara individu yang memiliki tingkatan yang sebanding. Contoh tingkatan ini terdapat pada sistem edukasi, di mana siswa SD hanya akan berkompetisi dengan siswa SD, siswa SMP hanya akan berkompetisi dengan siswa SMP, dan seterusnya.

Meskipun sudah banyak pendekatan yang dilakukan untuk menghindari jebakan, Lehman & Stanley (2011) mengemukakan bahwa masalah utama pada *objective search* sendiri tidak terselesaikan karena fungsi objektif dapat secara aktif mengarahkan pencarian ke arah buntu (lokal optimal). Mereka mengajukan

pendekatan untuk mengatasi masalah terjadinya jebakan tersebut dengan melakukan pencarian berdasarkan keterbaruan. Keterbaruan di sini memiliki arti perbedaan perilaku antara generasi baru yang dihasilkan dengan generasi yang menghasilkan generasi baru tersebut. Model ini mengikuti proses yang tidak ada akhirnya seperti pada evolusi di mana tidak ada objektif akhir dalam evolusi makhluk hidup. (Lehman & Stanley, 2011).

Pada beberapa permasalahan, *novelty search* ternyata memiliki hasil yang lebih baik dibandingkan dengan *objective search*. Hal ini memperlihatkan bahwa ada kemungkinan di mana pencarian berdasarkan objektif yang selama ini diterapkan pada *evolutionary computation* dan *machine learning* justru memberikan batasan terhadap apa yang bisa dicapai oleh kedua hal tersebut. Dengan melakukan pencarian yang tidak berdasarkan pada objektif, kemungkinan yang dapat dicapai melalui metode evolusi seperti pada algoritma genetik dan pemrograman genetik mungkin dapat diperluas. (Lehman & Stanley, 2011).