

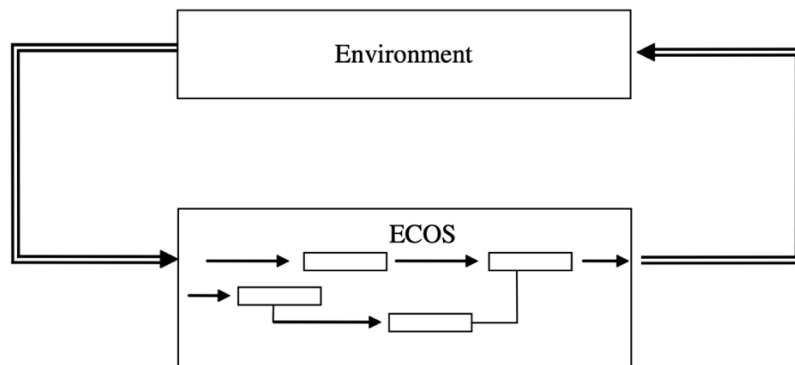
BAB II

TINJAUAN PUSTAKA

II.1 *Evolving Connectionist System (ECoS)*

Evolving Connectionist System (ECoS) merupakan salah satu bentuk dari *Evolving Intelligence Systems (EIS)*. ECoS adalah sebuah metode pembelajaran yang adaptif dan bertahap yang mengembangkan struktur dan fungsinya sendiri. Inti dari system *Evolving Connectionist System (ECoS)* tersebut adalah arsitektur *connectionist* yang terdiri dari *neuron* (unit pengolah informasi) dan hubungan antar-*neuron*. *Evolving Connectionist System (ECoS)* juga merupakan sistem komputasi cerdas yang berbasis neural networks, namun menggunakan teknik lain dari komputasi cerdas yang beroperasi secara kontinu dan mengadaptasikan struktur dan fungsinya melalui interaksi lanjutan terhadap lingkungan dan dengan sistem lainnya (Al-Khowarizmi, 2021).

(Kasabov, 2007) menjelaskan *Evolving Connectionist System (ECoS)* adalah sistem representasi pengetahuan dan pembelajaran adaptif dan inkremental yang mengembangkan struktur dan fungsionalitasnya, dimana inti dari sistem tersebut adalah arsitektur *connectionist* yang terdiri dari *neuron* (unit pemrosesan informasi) dan koneksi di antara mereka. Adapun ilustrasi dari *Evolving Connectionist System (ECoS)* dapat dilihat pada Gambar II.1 sebagai berikut.

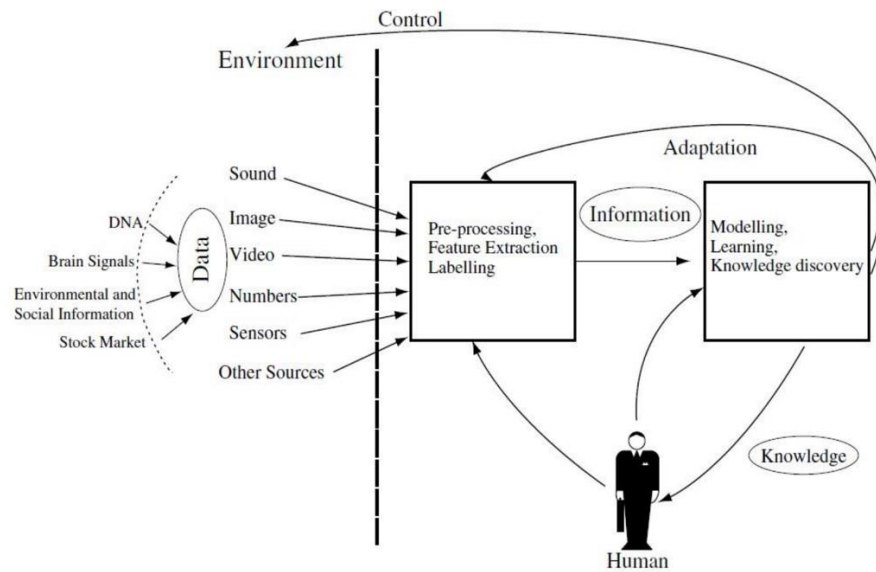


Gambar II.1 Ilustrasi *Evolving Connectionist System* (ECoS)
(Kasabov, 2007)

Pada Metode *Evolving Connectionist System* (ECoS) yang disajikan dalam buku dapat digunakan sebagai teknik *computational intelligence* (CI) tetapi *Evolving Connectionist System* (ECoS) memiliki beberapa karakteristik khusus yang membuatnya dapat diterapkan pada masalah yang lebih kompleks:

1. *Evolving Connectionist System* (ECoS) dapat berkembang dalam proses, dimana proses tersebut dapat berubah.
2. *Evolving Connectionist System* (ECoS) belajar melalui pembelajaran inkremental, mungkin dalam mode online.
3. *Evolving Connectionist System* (ECoS) dapat belajar secara terus-menerus dalam mode pembelajaran seumur hidup.
4. *Evolving Connectionist System* (ECoS) belajar baik sebagai sistem individu maupun sebagai populasi evolusi dari sistem tersebut.

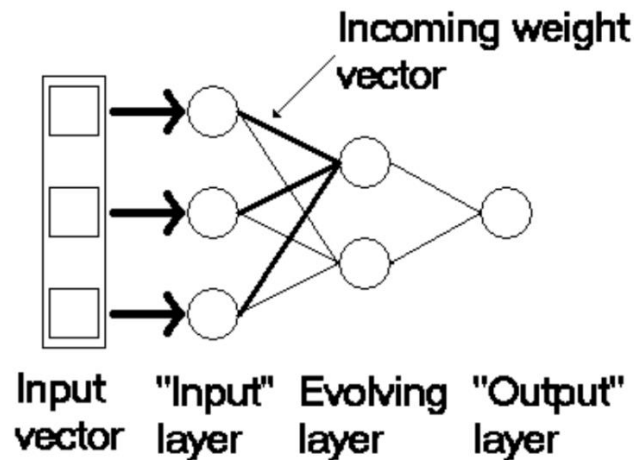
Adapun Proses Interaksik pada metode *Evolving Connectionist System* (ECoS) dapat dilihat pada Gambar II.2 sebagai berikut.



Gambar II.2 Proses Interaksi *Evolving Connectionist System* (ECoS)

II.1.1 Arsitektur dan Pembelajaran ECoS

Jaringan ECoS pertama adalah EFuNN (Subseksi II-B), dari mana arsitektur jaringan saraf tiruan konstruktif umum dan algoritma pelatihan berasal. Jaringan ECoS adalah jaringan saraf tiruan konstruktif dengan beberapa lapisan neuron. Jaringan ECoS selalu memiliki setidaknya satu lapisan neuron "evolving". Ini adalah lapisan konstruktif, lapisan yang akan tumbuh dan menyesuaikan diri dengan data masukan, dan merupakan lapisan yang paling berhubungan dengan algoritma pembelajaran. Adapun gambaran umum arsitektur ECoS dapat dilihat sebagai berikut.



Gambar II.3 Arsitektur umum ECoS

Makna dari koneksi yang mengarah ke lapisan ini, aktivasi neuron lapisan ini, dan algoritma propagasi maju lapisan yang berkembang semuanya berbeda dari sistem koneksi klasik seperti MLP. Untuk tujuan makalah ini, istilah *"layer input"* mengacu pada lapisan neuron yang langsung mendahului lapisan berkembang, sedangkan istilah *"layer output"* mengacu pada lapisan neuron yang segera mengikuti lapisan berkembang. Hal ini tidak tergantung pada apakah lapisan-lapisan ini merupakan layer *input* atau *output* aktual dari jaringan yang sesuai. Misalnya, pada Gambar II.8 yang menunjukkan struktur ECoS generik, *"layer input"* bisa menjadi layer input sebenarnya dari jaringan (seperti pada jaringan SECoS) atau bisa menjadi lapisan neuron yang memproses nilai input sebenarnya untuk disajikan kepada lapisan berkembang (seperti pada jaringan EFuNN). Demikian pula, *"layer output"* bisa menjadi *layer output* aktual dari jaringan (seperti pada SECoS) atau lapisan neuron yang memproses output dari lapisan berkembang (seperti pada EFuNN). Lapisan koneksi dari layer *neuron input* ke layer neuron berkembang dan dari layer

berkembang ke layer *neuron output*, terhubung sepenuhnya. Aktivasi neuron lapisan berkembang ditentukan oleh Persamaan II.1.

$$A_n = 1 - D_n \dots \dots \dots (II.1)$$

dimana D_n adalah jarak antara vektor input dan vektor bobot masukan untuk neuron tersebut. Karena jaringan ECoS terhubung sepenuhnya, mungkin untuk mengukur jarak antara vektor input saat ini dan vektor bobot masukan dari setiap neuron lapisan berkembang. Meskipun jarak dapat diukur dengan cara apa pun yang sesuai untuk input, fungsi jarak ini harus menghasilkan nilai dalam rentang nol hingga satu. Untuk alasan ini, sebagian besar algoritma ECoS mengasumsikan bahwa data input akan dinormalisasi, karena lebih mudah merumuskan fungsi jarak yang menghasilkan output dalam rentang yang diinginkan jika data dinormalisasi ke rentang nol hingga satu.

Sementara sebagian besar ANN mengirimkan aktivasi setiap neuron dari satu lapisan ke lapisan berikutnya, lapisan berkembang pada ECoS mengirimkan aktivasinya dengan salah satu dari dua strategi alternatif. Strategi pertama, yang disebut propagasi *One Of N*, hanya mengirimkan aktivasi neuron yang paling tinggi sebagai pemenang. Strategi kedua, propagasi *Many Of N*, mengirimkan nilai aktivasi dari neuron-neuron dengan nilai aktivasi yang lebih tinggi dari ambang batas aktivasi A_{thr} . Untuk setiap vektor input I dan vektor output yang diinginkan O_d do, lakukan hal berikut:

Propagasikan I melalui jaringan. Temukan neuron lapisan berkembang yang paling teraktivasi, neuron j , dan aktivasinya, A_j .

Jika $A_j < S_{thr}$, tambahkan satu neuron.

Jika tidak, cari kesalahan antara O_d dan aktivasi output A_o .

Jika $|O_d - A_o| > E_{thr}$, tambahkan satu neuron.

Jika tidak, perbarui koneksi ke neuron lapisan berkembang pemenang j .

Jika tidak, lanjutkan ke langkah berikutnya. Selesai.

Ulangi langkah-langkah di atas untuk setiap vektor *input* dan *output* yang ada.

Algoritma pembelajaran ECoS didasarkan pada kemampuan menyesuaikan contoh pelatihan baru dalam lapisan berkembang, baik dengan memodifikasi nilai bobot koneksi yang terhubung ke neuron lapisan berkembang, maupun dengan menambahkan neuron baru ke lapisan tersebut. Algoritma yang digunakan dijelaskan dalam Gambar II.7. Penambahan neuron ke lapisan berkembang didorong oleh kebaruan contoh pelatihan saat ini: jika contoh saat ini sangat baru (tidak cukup direpresentasikan oleh neuron yang ada), maka neuron baru akan ditambahkan. Terdapat empat parameter yang terlibat dalam algoritma ini: *threshold sensitivitas* S_{thr} , *threshold error* E_{thr} , serta dua tingkat pembelajaran η_1 dan η_2 . *Threshold sensitivitas* dan *threshold error* mengontrol penambahan neuron, dan saat sebuah neuron ditambahkan, vektor bobot koneksi masuknya diatur menjadi vektor *input* I , dan vektor bobot keluarnya diatur menjadi vektor output yang diinginkan O_d . *Threshold sensitivitas* dan *threshold error* merupakan ukuran kebaruan contoh saat ini. Seperti yang terlihat pada Gambar II.8, jika contoh saat ini menyebabkan aktivasi rendah (yaitu, konteks yang baru dibandingkan dengan neuron yang ada), maka *threshold sensitivitas* akan menyebabkan penambahan neuron yang mewakili

contoh tersebut. Jika contoh tersebut tidak memicu penambahan neuron melalui *threshold* sensitivitas, tetapi output yang dihasilkan oleh contoh tersebut menghasilkan kesalahan *output* yang lebih besar dari *threshold error* (yaitu, output yang baru), maka *neuron* akan ditambahkan. Bobot koneksi dari setiap *input* i ke *neuron* pemenang j dimodifikasi sesuai dengan Persamaan II.2.

$$W_{i,j}(t+1) = W_{i,j}(t) + \eta_1(I_i - W_{i,j}(t)) \dots\dots\dots(\text{II.2})$$

dimana:

$W_{i,j}(t)$ adalah bobot koneksi dari input i ke j pada waktu t

I_i adalah komponen ke- i dari vektor input I

Bobot dari neuron j ke *output* o dimodifikasi sesuai dengan Persamaan II.3.

$$W_{j,o}(t+1) = W_{j,o}(t) + \eta_2 A_j E_o \dots\dots\dots(\text{II.3})$$

dimana:

$W_{j,o}(t)$ adalah bobot koneksi dari j ke output o pada waktu t

A_j adalah aktivasi dari neuron j E_o adalah kesalahan tanda pada *output* o ,

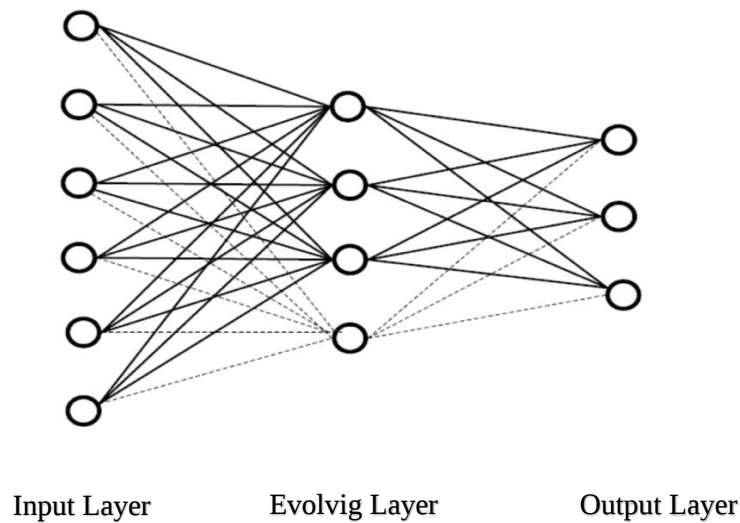
diukur sesuai dengan Persamaan II.4.

II.1.2 Simple Evolving Connectionist System (SECoS)

Simple Evolving Connectionist System (SECoS) adalah implementasi minimalis dari prinsip *Simple Evolving Connectionist System* (SECoS) dan versi yang disederhanakan dari *Evolving Fuzzy Neural Network* (EFuNN). Jaringan *Simple Evolving Connectionist System* (SECoS) mampu memodelkan jenis masalah yang sama dengan *Evolving Fuzzy Neural Network* (EFuNN), tetapi seringkali lebih kecil dari *Evolving Fuzzy Neural Network* (EFuNN) yang setara. *Simple Evolving*

Connectionist System (SECoS) terdiri dari tiga lapisan neuron, pertama adalah lapisan input, kedua lapisan evolusi dan ketiga setara dengan lapisan aturan dalam *Evolving Fuzzy Neural Network* (EFuNN). Aktivasi neuron di lapisan ini adalah fungsi jarak antara vektor input saat ini dan vektor bobot masuk ke setiap neuron lapisan evolusi, di mana jarak dapat diukur dengan fungsi apa pun yang mengembalikan jarak dalam rentang nol hingga satu. Di sini aktivasi dilakukan melalui operasi perkalian dan penjumlahan sederhana antara aktivasi lapisan evolusi dan bobot koneksi dari lapisan evolusi ke lapisan output. Fungsi aktivasi linear jenuh diterapkan pada lapisan tersembunyi dan lapisan output (Watts, 2004).

(Al-Khowarizmi, 2021) Menjelaskan *Simple Evolving Connectionist System* (SECoS) merupakan versi sederhana dari *Evolving Fuzzy Neural Network* (EFuNN). Ada beberapa keuntungan menggunakan *Simple Evolving Connectionist System* (SECoS) dari pada *Evolving Fuzzy Neural Network* (EFuNN). Pertama, arsitektur *Simple Evolving Connectionist System* (SECoS) jauh lebih sederhana dimana *Simple Evolving Connectionist System* (SECoS) lebih mudah untuk memahami dan menganalisis. Kedua, ruang input lebih rendah sehingga memungkinkan *Simple Evolving Connectionist System* (SECoS) bekerja dengan data pelatihan dengan lebih sedikit. Adapun architecture *Simple Evolving Connectionist System* (SECoS) dapat dilihat pada Gambar II.4 sebagai berikut.



Gambar II.4 Arsitektur

Simple Evolving Connectionist System (SECoS) (Al-Khowarizmi, 2021)

Pada Arsitektur menjelaskan Jika fungsi aktivasi linear digunakan, maka aktivasi A pada lapisan *evolvingnode* n ditentukan dengan:

$$A_n = 1 - D_n \dots \dots \dots (II.1)$$

Dimana A_n merupakan nilai aktivasi pada node n dan D_n merupakan nilai *normalized distance* antara *input vector* dengan *incoming weight vector* pada node tersebut. Nilai *distance* D_n dapat dihitung dengan menggunakan *normalized Hamming distance*:

$$D_n = \frac{\sum_i^K |I_i - W_i|}{\sum_i^K |I_i - W_i|} \dots \dots \dots (II.2)$$

Pada penjelasan rumus diatas K adalah jumlah input nodes pada *Simple Evolving Connectionist System* (SECoS), I merupakan *input vector* dan W merupakan *input weight matrix* pada *evolving layer*.

Adapun Algoritma *Simple Evolving Connectionist System* (SECoS) adalah sebagai berikut (Rahmat, ea al. 2016):

1. Lakukan propagasi input vector I ke dalam jaringan.
2. Jika aktivasi maksimum (A_{max}) dari node lebih kecil dari koefisien *sensitivity threshold* (S_{thr}), maka:

- a. Tambah node baru

Jika tidak lakukan:

- 1) Hitung nilai *error* antara hasil prediksi (*output vector* O_c) dan nilai aktual (*output vector* O_d).
- 2) Jika nilai *error* lebih besar dari koefisien *error threshold* (E_{thr}) atau *output node* yang diinginkan tidak aktif, maka:
 - Tambahkan *node* baru *else*:
 - Lakukan perubahan bobot koneksi pada *winning hidden node*.
- 3) Ulangi langkah tersebut untuk setiap *input vector*. Ketika sebuah *node* ditambahkan, bobot *input* diberi inisialisasi sesuai dengan *input vector* I dan bobot *output* diinisialisasi sesuai dengan *output vector* O_d . Proses propagasi dari *hidden layer* ke *output layer* dapat dilakukan dengan dua cara sebagai berikut:
 - Dengan metode propagasi *One-of-N* yaitu propagasi hanya dilakukan oleh *hidden node* dengan nilai aktivasi tertinggi.
 - Dilakukan dengan metode propagasi *Many-of-N* yaitu propagasi hanya dilakukan oleh *hidden node* yang memiliki nilai aktivasi diatas *activation threshold*.

Modifikasi bobot masuk pada *winning node* j dilakukan sesuai persamaan (II.3) sebagai berikut:

$$WW_{ii,jj}(tt+1) = WW_{ii,jj}(tt) + \eta\eta_1 I_{ii} - WW_{ii,jj}(tt) \dots (\text{II.3})$$

Penjelasan:

- $WW_{ii,jj}(tt)$ merupakan bobot masuk ii, jj pada saat (tt) .
- $WW_{ii,jj}(tt+1)$ merupakan bobot masuk ii, jj pada saat $(tt+1)$.
- $\eta\eta_1$ merupakan *learning rate* pertama.
- I_{ii} merupakan komponen ke- i pada input vector I .

Sedangkan modifikasi bobot keluar dari *node j* dilakukan sesuai persamaan (II.4):

$$WW_{jj,pp}(tt+1) = WW_{jj,pp}(tt) + \eta\eta_2 AA_{jj} \times EE_{pp} \dots \dots \dots (\text{II.4})$$

Penjelasan:

- $WW_{jj,pp}(tt)$ merupakan bobot keluar jj, pp pada saat (tt)
- $WW_{jj,pp}(tt+1)$ merupakan bobot masuk jj, pp pada saat $(tt+1)$
- $\eta\eta_2$ merupakan *learning rate* kedua
- AA_{jj} merupakan nilai aktivasi dari *node*

II.2 Artificial Intelligence (AI)

Kecerdasan buatan atau *artificial intelligence* (AI) merupakan salah satu bagian ilmu computer 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 jaman, maka peran computer semakin mendominasi

kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh 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 oleh 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. Lebih detailnya, 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.

Domain yang sering dibahas oleh para peneliti meliputi:

- a. *Mundane task*

- Persepsi (*vision & speech*).
- Bahasa alami (*understanding, generation & translation*).
- Pemikiran yang bersifat *commonsense*.
- *Robot control*.

b. *Formal task*

- Permainan / games.
- Matematika (geometri, logika, kalkulus, integral, pembuktian)

c. *Expert task*

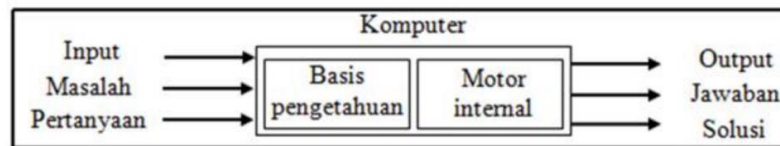
- Analisis finansial.
- Analisis medikal.
- Analisis ilmu pengetahuan.
- Rekayasa (desain, pencarian kegagalan, perencanaan manufaktur)

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 melakukan aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan, yaitu (Gambar 2.1):

- Basis pengetahuan (*knowledge base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.

- Motor inferensi (*inference engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman (Hendra Jaya, et.all, 2018).



Gambar II.5 Penerapan Konsep Kecerdasan Buatan

Menurut Luger dan William (1993), kecerdasan buatan adalah cabang ilmu computer yang berhubungan dengan otomasi perilaku yang cerdas. Kemudian menurut Haag dan Peter (1996) kecerdasan buatan adalah bidang studi yang berhubungan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia ke dalam sebuah sistem teknologi informasi sehingga sistem tersebut dapat digunakan sebagai proses pengambilan keputusan. Dapat disimpulkan bahwa kecerdasan buatan adalah cabang ilmu komputer yang membahas tentang penangkapan, pemodelan, dan penyimpanan kecerdasan manusia ke dalam sebuah teknologi informasi pengambilan keputusan (Athanasia Octaviani Puspita Dewi, 2020).

II.3 Machine Learning

Machine Learning merupakan praktik pemrograman komputer untuk belajar dari data dan program dengan mudah dapat menentukan apakah diberikan penting atau "spam" (Russell, 2018). *Machine Learning* diciptakan oleh Samuel pada tahun 1959, istilah *Machine Learning* diberikan untuk bidang studi komputer yang mampu belajar tanpa diprogram secara eksplisit, dengan kata lain, *machine Learning* dapat memperoleh pengetahuan langsung dari data dan belajar memecahkan masalah. *Machine Learning* tidak lama lagi akan mempengaruhi

komunitas statistik (Bruce Ratner, 2017).

Big data memerlukan metode otomatis analisis data, itulah yang disediakan oleh *Machine learning*. Secara khusus, mendefinisikan *Machine learning* sebagai kumpulan metode yang dapat secara otomatis mendeteksi pola dalam data dan kemudian menggunakan pola yang terungkap untuk memprediksi data di masa depan, atau melakukan jenis pengambilan keputusan lain dalam situasi ketidakpastian (Murphy, n.d.).

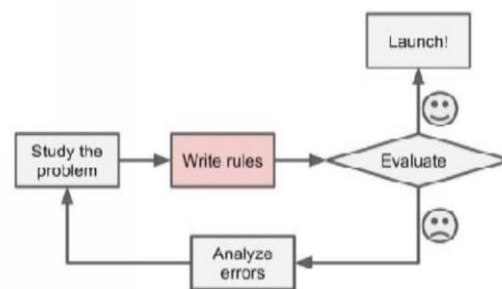
Machine learning atau pembelajaran mesin yang luas, masing-masing diilustrasikan dengan tugas pembelajaran sederhana. Dalam bagian ini, kami mempertimbangkan dua aspek dari pembelajaran mesin praktis. Yang pertama melibatkan menemukan algoritma yang mampu belajar untuk mengenali angka tulisan tangan dan memperoleh kinerja prediktif yang optimal darinya. Yang kedua melibatkan hal yang berbeda—menunjukkan bahwa mendapatkan, membersihkan, dan merepresentasikan data dapat setidaknya sama pentingnya dengan rekayasa algoritma (Joshi, 2017).

Pembelajaran, seperti kecerdasan, mencakup rentang proses yang sangat luas sehingga sulit untuk didefinisikan secara tepat. Definisi dalam kamus mencakup frasa-frasa seperti mendapatkan pengetahuan, pemahaman, atau keterampilan melalui studi, instruksi, atau pengalaman dan modifikasi kecenderungan perilaku melalui pengalaman. Zoolog dan psikolog mempelajari pembelajaran pada hewan dan manusia. Dalam buku ini, kami fokus pada pembelajaran pada mesin. Ada beberapa paralel antara pembelajaran hewan dan pembelajaran mesin. Tentu saja, banyak teknik dalam pembelajaran mesin berasal dari upaya psikolog untuk

membuat teori pembelajaran hewan dan manusia mereka menjadi lebih presisi melalui model komputasi. Terlihat juga kemungkinan bahwa konsep dan teknik yang sedang diteliti oleh para peneliti dalam pembelajaran mesin dapat menerangi aspek-aspek tertentu dari pembelajaran biologis (Nilsson, 1998).

Mari kita asumsikan bahwa Anda ingin menulis program filter tanpa menggunakan metode *machine learning*. Dalam hal ini, Anda harus melakukan langkah-langkah berikut:

1. Anda akan melihat seperti apa e-mail spam itu. Anda mungkin memilih kata-kata atau frasa yang digunakan, seperti "kartu debit," "gratis," dan sebagainya, serta pola yang digunakan dalam nama pengirim atau dalam isi email.
2. Anda akan menulis algoritma untuk mendeteksi pola-pola yang telah Anda lihat, dan kemudian perangkat lunak akan menandai email sebagai spam jika sejumlah pola tersebut terdeteksi.

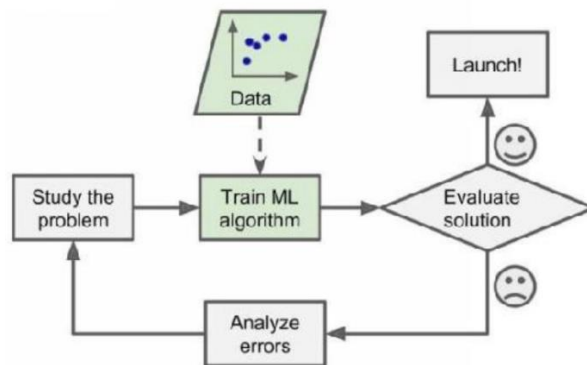


Gambar II.6 Flow Mendeteksi Pola

3. Anda akan menguji program tersebut, dan kemudian mengulangi dua langkah pertama lagi sampai hasilnya cukup baik.

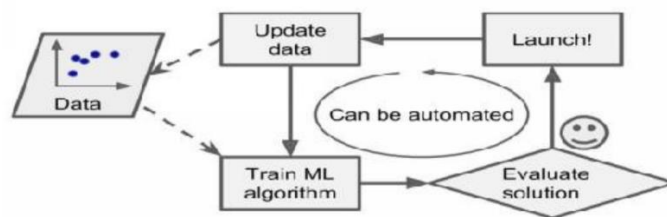
Karena program tersebut bukanlah perangkat lunak, ia mengandung daftar aturan yang sangat panjang yang sulit untuk dipelihara. Namun, jika Anda

mengembangkan perangkat lunak yang sama menggunakan machine learning, Anda akan dapat memeliharanya dengan baik.



Gambar II.7 Flow Menguji Program

Selain itu, pengirim email dapat mengubah *template email* mereka sehingga kata seperti "4U" sekarang menjadi "for you," karena *email* mereka telah diketahui sebagai spam. Program yang menggunakan teknik tradisional akan membutuhkan pembaruan, yang berarti bahwa jika ada perubahan lain, Anda perlu memperbarui kode Anda berulang kali.



Gambar II.8 Flow Proses *Machine Learning*

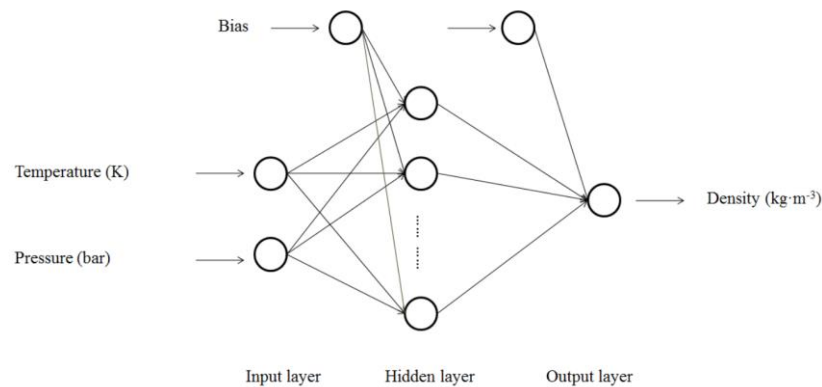
Di sisi lain, program yang menggunakan teknik *machine learning* akan secara otomatis mendeteksi perubahan ini oleh pengguna, dan mulai menandainya tanpa Anda harus memberitahunya secara manual. Selain itu, kita dapat menggunakan machine learning untuk menyelesaikan masalah yang sangat kompleks bagi perangkat lunak non-machine learning. Misalnya, pengenalan suara:

ketika Anda mengatakan "satu" atau "dua", program harus dapat membedakan perbedaannya. Oleh karena itu, dengan menggunakan machine learning, program dapat belajar dan beradaptasi dengan perubahan data dan masalah yang kompleks secara lebih efisien. Mengembangkan sebuah algoritma yang mengukur suara. Pada akhirnya, machine learning akan membantu kita untuk belajar, dan algoritma machine learning dapat membantu kita melihat apa yang telah kita pelajari (Russell, 2018).

II.4 Jaringan Syaraf Tiruan

Istilah *Artificial Neural Networks* atau Jaringan Syaraf Tiruan digunakan untuk menghasilkan mimpi atau khayalan dan ketakutan yang tidak berguna. Jaringan Syaraf Tiruan (JST) adalah teknik pemrosesan data yang pada dasarnya dipahami saat ini; Jaringan Syaraf Tiruan (JST) harus menjadi bagian dari kotak alat semua ilmuwan yang ingin memanfaatkan data yang tersedia bagi mereka, termasuk melakukan ramalan, merancang model prediktif, mengenali pola atau sinyal, dan lain sebagainya (Dreyfus, 2004). pada jaringan syaraf tiruan, yang umumnya disebut sebagai "jaringan saraf," telah dipicu sejak awal dengan pengakuan bahwa otak manusia melakukan komputasi secara sepenuhnya berbeda dari komputer digital konvensional. Pemikiran komputer yang sangat kompleks, nonlinear, dan parallel (sistem pemrosesan informasi). Komputer memiliki kemampuan untuk mengorganisasikan komponen strukturalnya, yang dikenal sebagai neuron, sehingga dapat melakukan beberapa komputasi (misalnya, pengenalan pola, persepsi, dan pengendalian motor) dengan kecepatan yang jauh

lebih cepat daripada komputer digital tercepat yang ada saat ini. Fungsi sistem visual adalah memberikan representasi lingkungan sekitar kita dan yang lebih penting, menyediakan informasi yang kita butuhkan untuk berinteraksi dengan lingkungan. Secara khusus, otak secara rutin menyelesaikan tugas pengenalan perseptual (misalnya, mengenali wajah yang familiar di dalam suasana yang tidak dikenal) dalam waktu sekitar 100-200 ms, sedangkan tugas yang jauh lebih sederhana bisa memakan waktu sehari-hari pada komputer konvensional (Simon Haykin, 1994). *Artificial Neural Networks* adalah algoritma pembelajaran mesin dengan fungsi estimasi dan aproksimasi berdasarkan masukan, yang terinspirasi dari jaringan saraf biologis dalam otak manusia. Berbeda dari jaringan dengan hanya satu atau dua lapisan logika arah tunggal, mereka menggunakan algoritma dalam pengendalian penentuan dan pengorganisasian fungsi. Jaringan yang saling terhubung biasanya terdiri dari neuron yang dapat menghitung nilai dari masukan dan beradaptasi dengan berbagai situasi. Dengan demikian, *Artificial Neural Networks* memiliki kapasitas yang kuat dalam prediksi numerik dan pengenalan pola, yang telah mendapatkan popularitas luas dalam menginferensikan fungsi dari pengamatan, terutama ketika objeknya terlalu rumit untuk ditangani oleh otak manusia. Gambar II.5 menunjukkan struktur skematik dari *Artificial Neural Networks* untuk prediksi densitas cairan terkompresi R227ea, yang terdiri dari lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Lapisan masukan terdiri dari dua node yang mewakili suhu dan tekanan yang dimasukkan. Lapisan keluaran terdiri dari neuron yang mewakili densitas R227ea (Christian Dawson (Ed.), n.d.).



Gambar II.9 Struktur Skematik

Jaringan saraf (*neural network*), kadang disebut *artificial neural network* (ANN) atau komputasi saraf (*neural computing*), adalah suatu bidang kecerdasan buatan yang meniru pola pemrosesan dalam otak manusia yang berbasis pada pengenalan pola. (Athanasia Octaviani Puspita Dewi, 2020).

II.5 Surat Perintah Perjalanan Dinas (SPPD)

Surat Perintah Perjalanan Dinas (SPPD) merupakan Surat Dinas digunakan untuk kepentingan pekerjaan formal seperti instansi, dinas dan tugas kantor, dalam kedinasan ketika seorang Pegawai Negeri Sipil (PNS) ataupun pegawai swasta diperintahkan untuk melakukan tugas tertentu atau diperintahkan untuk melakukan tugas ke Daerah, luar kota maupun keluar negara (Hendry Fonda, 2019). Menurut (Nindya dan Putri, 2021) Surat perjalanan dinas (SPPD) adalah jenis surat tugas yang diberikan atasan untuk seorang pegawai dalam melaksanakan tugas kerjanya dan harus melakukan perintah perjalanan pergi ke daerah lain dalam rangka perjalanan dinas seperti tugas kerja berupa studi banding, audit ke instansi kota lain, rapat kerja nasional maupun pekerjaan-pekerjaan lain yang dilakukan di luar kota.

II.6 Penelitian Terkait

Tabel II.1 Penelitian Terkait

No	Penelitian	Metode	Tujuan	Kekurangan	Manfaat
1	<i>Optimization of Applied Detection Rate in the Simple Evolving Connectionist System Method for Classification of Images Containing Protein</i>	<i>MAPE; Detection rate; Simple Evolving Connectionist; Classification;</i>	Penerapan Artificial Neural Network (ANN) sangat cocok untuk dilakukan optimasi klasifikasi menggunakan Metode Simple Evolving Connectionist System (SECoS), dan sebagai hasilnya, klasifikasi gambar yang mengandung protein dengan data pengujiannya adalah telur bekerja dengan bukti optimal mencapai MAPE.	Menentukan nilai parameter untuk menghasilkan tingkat error yang kecil.	Pengolahan citra digital, pada umumnya membuat gambar yang tampak diubah menjadi Fungsi intensitas cahaya yang direpresentasikan dalam bidang dua dimensi. Fungsi adalah suatu nilai yang akan diproses untuk klasifikasi sehingga computer mampu mengenali gambar tersebut. Selain itu, klasifikasi memerlukan pelatihan dan pengujian menghasilkan nilai error yang kecil dan algoritma yang optimal
2	<i>Measuring the Accuracy of Simple Evolving Connectionist System with Varying Distance Formulas</i>	<i>ECoS, SECoS, Normalized Euclidean distance dan Normalized Manhattan distance</i>	Keakuratan pengukuran yang dihasilkan dari ketiga rumus jarak tersebut dihitung menggunakan berarti persentase kesalahan absolut. pelatihan dengan beberapa parameter seperti ambang sensitivitas, ambang batas kesalahan, kecepatan pembelajaran pertama, dan kecepatan pembelajaran kedua,	Menentukan nilai parameter secara manual dan jarak Manhattan untuk menghasilkan tingkat error yang kecil.	Simple Evolving Connectionist System (SECoS) adalah implementasi minimal dari Mengembangkan Sistem Connectionist (ECoS) dalam jaringan saraf tiruan. Jaringan tiga lapis arsitektur SECoS dapat dibangun berdasarkan masukan yang diberikan

			ditemukan bahwa jarak Euclidean yang dinormalisasi lebih akurat daripada jarak Hamming yang dinormalisasi dan menormalisasi jarak Manhattan		
3	<i>Classification of skin cancer images by applying simple evolving connectionist system</i>	<i>Classification Data mining, SECoS ,Skin care</i>	SECoS menggunakan jaringan saraf feedforward tetapi tersembunyi lapisan berevolusi sehingga setiap lapisan input tidak melakukan epoch. SECoS jarak telah dimodifikasi dengan rumus jarak Euclidean yang dinormalisasi untuk mengurangi kesalahan dalam pelatihan	Dataset yang digunakan memiliki masalah seperti labelgambar yang kabur, atau gambar yang tidak representatif, hasil klasifikasi dapat menjadi tidak akurat.	Mengenal kanker kulit dengan cara mengklasifikasikannya gambar tahi lalat kulit jinak ganas menggunakan SECoS berdasarkan parameter kombinasi.
4	<i>The effect of a SECoS in crude palm oil forecasting to improve business intelligence</i>	<i>ANN Business intelligence Crude palm oil e-Commerce SECoS</i>	Kelapa sawit merupakan salah satu tanaman yang masa panennya ± 2 minggu dan kondisinya sangat memprihatinkan perlunya penyebaran informasi menggunakan e-commerce agar bisa memprediksi harga hasil panen perusahaan atau kebun perorangan di dalamnya 2 minggu ke depan untuk meningkatkan studi tentang intelijen bisnis.	Penelitian ini mungkin mengklaim bahwa penggunaan SECoS meningkatkan bisnis intelijen, tetapi penting untuk memberikan bukti konkret dalam bentuk dampak nyata pada pengambilan keputusan bisnis dan kinerja bisnis.	Mengimplementasikan e-commerce dengan memaksimalkan proses B2B dan mengembangkannya Model Intelijen Bisnis yang khusus untuk pendapatan bisnis minyak sawit mentah berbasis aplikasi perkebunan