

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi uraian dan penjabaran tentang penelitian-penelitian terdahulu yang berhubungan dengan permasalahan yang ada dalam penelitian ini. Selain itu pada bab ini juga diuraikan mengenai objek dan teori-teori pendukung yang berhubungan dengan penelitian ini, yaitu air sungai, 7 parameter pengukuran kualitas air sungai, metode STORET, Jaringan Saraf Tiruan, metode *Extreme Learning Machine* (ELM), Algoritme Evolusi, metode Algoritme Genetika dan pengujian akurasi.

2.1 Kajian Pustaka

Pada penelitiannya, Guo, Cheng dan Wang (2016) mempertimbangkan algoritme *Two Stage Capacitated Facility Location Problem* (TSCFLP) dimana produk yang diproduksi akan disampaikan kepada pelanggan melalui ruang penyimpanan. Permintaan pelanggan adalah subjek dalam menilai tingkat kepuasan untuk produksi tanaman dan kapasitas ruang penyimpanan terbatas. Tujuan dari penelitian ini adalah untuk menentukan lokasi tanaman dan ruang penyimpanan untuk meminimalkan total biaya termasuk biaya perbaikan dan biaya transportasi. Algoritme *hybrid* yang diusulkan oleh penulis dapat memberikan solusi yang lebih baik dalam memecahkan masalah lokasi fasilitas dua-tahap dibandingkan dengan algoritme yang ada, seperti Algoritme Genetika (GA), *Tabu Search* (TS) dan Algoritme Genetika *Hybrid Firefly* (HFGA). Dengan menggunakan algoritme yang diusulkan, fasilitas dapat ditempatkan lebih efisien, yang berarti biaya perbaikan dan biaya transportasi dapat dikurangi secara signifikan.

Pada penelitian kedua yang telah dilakukan oleh Henriquez dan Ruz (2017), PL-ELM dan ELM menunjukkan performa yang serupa. Selain itu, semua hasil analisis statistik pada akurasi adalah dengan $p < 0,05$. Oleh karena itu, kami dapat mengkonfirmasi statistik bahwa algoritme yang diusulkan (PL-ELM) melebihi ELM. Secara khusus, untuk masalah klasifikasi, PL-ELM tidak hanya memperoleh akurasi terbaik pada saat dilakukan pengujian, tetapi juga dengan memiliki dispersi (standar deviasi) terendah membuat prediksi ini lebih handal. Selain itu, algoritme PL-ELM dengan generasi yang telah ditentukan menggunakan LDSs (*Low Discrepancy Points*) / Titik Ketidakcocokkan Terendah, dapat diperluas dengan modifikasi algoritme ELM lainnya

Penelitian ketiga yang dilakukan oleh Han et al. (2017), digunakan metode yang dimodifikasi dari ELM, yaitu EM-ELM. *Error Minimized Extreme Learning Machine* (EM-ELM) adalah pendekatan yang sederhana dan efisien untuk menentukan jumlah *hidden node*. Namun, mirip dengan ELM konstruktif lainnya, EM-ELM meletakkan banyak penekanan pada keakuratan konvergensi, yang dapat memperoleh *Single Hidden Layer Feedforward Neural Network* (SLFN) dengan kinerja konvergensi yang baik tetapi dalam kondisi yang buruk. Dalam penelitian ini, pendekatan yang efektif yaitu EM-ELM dan *Particle Swarm Optimization* (PSO) diusulkan untuk menentukan struktur SLFN secara adaptif dalam menangani

permasalahan regresi. Pada metode baru, untuk membangun SLFN dengan kondisi yang baik, *hidden node* akan dioptimalkan menggunakan PSO, kemudian ditambahkan ke SLFN satu per satu. Selain itu, tidak hanya akurasi regresi tetapi juga nilai kondisi matriks *hidden output* jaringan, dipertimbangkan dalam proses optimasi. Hasil percobaan pada berbagai masalah regresi ini telah membuktikan bahwa algoritme yang diusulkan dapat mencapai kinerja yang lebih baik dengan lebih sedikit *hidden node* dari ELM konstruktif lainnya.

Pada penelitian keempat yang dilakukan oleh Anam dan Al-Jumaily (2016), menurut mereka dengan menggunakan yang *classifier* terbaik ELM, yaitu *Radial Basis Function Extreme Learning Machine* (RBF-ELM), sistem pengenalan pola mencapai akurasi 98,55% pada subjek yang diamputasi 99,5% dan pada subjek yang bertubuh sehat, menggunakan enam saluran EMG. Sementara itu, dengan menggunakan dua saluran EMG, pengenalan pola mencapai akurasi 97,11% dan 92,73% masing-masing pada non-amputasi dan diamputasi. Dalam hal akurasi, keluarga ELM sebanding dengan *classifier* yang paling terkenal, SVM. Namun, dalam hal waktu pemrosesan, ELM lebih unggul dari SVM.

Di penelitian kelima membahas mengenai sensor inersia. Distribusi data sensor sangat dipengaruhi oleh penempatan sensor. Performanya akan menurun ketika model dilatih pada satu penempatan yang digunakan pada orang lain. Dalam rangka untuk mengatasi masalah ini, model pengenalan aktivitas manusia yang cepat dan kuat yang disebut TransM-RKELM diusulkan dalam makalah ini (*Transfer learning Mixed and Reduced Kernel Extreme Learning Machine*); Menggunakan metode *kernel* gabungan untuk mengurangi pengaruh dengan pilihan fungsi *kernel* dan *kernel* yang direduksi dimanfaatkan untuk mengurangi biaya komputasi. Setelah menyadari model pengenalan aktivitas awal dengan M-RKELM, dalam fase *online*, M-RKELM digunakan untuk mengklasifikasikan aktivitas dan mengadaptasi model untuk lokasi baru berdasarkan hasil pengenalan yang tinggi secara *real time* (Wang et al., 2017).

Pada penelitian ini, akan digunakan metode ELM dan Algoritme Genetika untuk memprediksi kualitas air sungai. Perbandingan objek dan metode penelitian yang telah dilakukan terhadap tinjauan pustaka dari penelitian sebelumnya ditunjukkan oleh Tabel 2.1 berikut.

Tabel 2.1 Kajian pustaka

No.	Judul	Objek	Metode	Hasil
1.	<i>Hybrid Evolutionary Algorithm with Extreme Machine Learning Fitness Function Evaluation for Two-Stage Capacitated Facility Location Problems</i>	Lokasi logistik tanaman.	- <i>Extreme Learning Machine</i> (ELM). - Algoritme Evolusi <i>Hybrid / Fitness Approximation</i> (FA)	Algoritme <i>hybrid</i> yang diusulkan oleh penulis dapat memberikan solusi yang lebih baik dalam memecahkan masalah lokasi fasilitas dua-tahap dibandingkan dengan

				<p>algoritme yang ada, seperti Algoritme Genetika (GA), <i>Tabu Search</i> (TS) dan Algoritme Genetika <i>Hybrid Firefly</i> (HFGA).</p>
2.	<p><i>Extreme learning machine with a deterministic assignment of hidden weights in two parallel layers</i></p>	<p>Datasets pada UCI Repositoy</p>	<p><i>Parallel Layer Extreme Learning Machine</i> (PL-ELM)</p>	<p>Secara khusus, untuk masalah klasifikasi, PL-ELM tidak hanya memperoleh akurasi terbaik pada saat dilakukan pengujian, tetapi juga dengan memiliki dispersi terendah (standar deviasi) membuat prediksi ini lebih handal.</p>
3.	<p><i>An improved incremental constructive single-hidden-layer feedforward networks for extreme learning machine based on particle swarm optimization</i></p>	<p>Bank</p>	<p><i>Incremental Particle Swarm Optimization Error Minimized Extreme Learning Machine</i> (IPSO-EM-ELM)</p>	<p>Karena menggunakan PSO untuk memilih parameter bobot, IPSO-EM-ELM menghabiskan lebih banyak waktu daripada ELM lainnya. Dengan ini algoritme yang diusulkan oleh penulis dapat mencapai kinerja yang lebih baik dengan lebih sedikit <i>hidden node</i> dari ELM konstruktif lainnya.</p>
4.	<p><i>Evaluation of extreme learning machine for classification of individual and combined finger movements using electromyography</i></p>	<p>Jari tangan</p>	<p><i>Radial Basis Function Extreme Learning Machine</i> (RBF-ELM)</p>	<p>Perbedaan keakuratan penerapan metode M-PR pada orang yang diamputasi dan yang tidak diamputasi tidak terlalu besar, dengan akurasi 98,55% pada</p>

	<i>on amputees and non-amputees</i>			orang diamputasi dan 99,5% pada yang tidak diamputasi menggunakan enam saluran <i>electromyography</i> (EMG).
5.	<i>Kernel fusion based extreme learning machine for cross-location activity recognition</i>	Sensor Inersia	<i>Transfer learning Mixed and Reduced Kernel Extreme Learning Machine</i> (TransM-RKELM)	Hasil penelitian menunjukkan bahwa metode yang diusulkan dapat mengklasifikasi lokasi sensor baru dan mudah beradaptasi dengan cepat serta menghasilkan kinerja yang baik.

Sumber: (Guo et al., 2016), (Henriquez dan Ruz, 2017), (Han et al., 2017), (Anam dan Al-Jumaily, 2016), (Wang et al., 2017).

2.2 Air Sungai

Sungai merupakan tempat/wadah serta jaringan pengaliran air, mulai dari mata air sampai muara yang dibatasi di sisi kanan dan kirinya serta sepanjang pengalirannya oleh garis luar pengamatan (Peraturan Pemerintah No.35 tahun 1991). Kualitas air sungai dapat diketahui melalui faktor fisik, kimia maupun biologi (Nadhiroh, 2014). Sungai memiliki fungsi dan nilai untuk kesejahteraan manusia serta kehidupan yang ada di dalam sungai itu sendiri. Karena itu sungai merupakan salah satu ekosistem yang penting bagi manusia (Siahaan et al., 2011 disitasi dalam Budiarta et al., 2016, p2). Bahkan peradaban manusia zaman dahulu dimulai dari daerah yang berada dekat dengan aliran sungai. Sejak zaman dahulu air sungai banyak dimanfaatkan oleh manusia untuk berbagai macam keperluan, seperti untuk mencuci, mandi atau pun sebagai sanitasi.

Pemanfaatan air harus dilakukan secara bijak dengan mempertimbangkan kepentingan generasi sekarang dan yang akan datang. Akan tetapi, seiring meningkatnya proses pembangunan di berbagai bidang dan laju pertumbuhan penduduk yang semakin pesat menyebabkan pemanfaatan air tidak lagi dilakukan dengan baik. Hal ini memberikan dampak buruk yang mempengaruhi sifat fisik dan sifat kimia air sehingga menurunkan kualitas air. Pengelolaan sumber daya air bertujuan menyediakan air dalam jumlah yang cukup dengan kualitas yang sesuai dengan tujuannya (Azizah, 2016).

2.3 Parameter Pengukuran Kualitas Air Sungai

Kualitas air merupakan istilah yang menggambarkan kesesuaian atau tingkat kecocokan air untuk penggunaan tertentu, misalnya: air minum, rumah tangga, perikanan, pengairan/irigasi, peternakan, industri, rekreasi dan sebagainya. Kualitas air dapat diketahui dengan melakukan pengujian tertentu terhadap air tersebut. Pengujian yang umumnya dilakukan adalah uji kimia, fisik, biologi, atau uji wujud air (bau dan warna). Pada penelitian ini, akan digunakan parameter fisika yaitu TSS, parameter kimia organik yang meliputi BOD, COD, DO, pH dan parameter kimia anorganik yang meliputi fenol, minyak dan lemak.

2.3.1 TSS

Padatan tersuspensi (*Total Suspended Solid/TSS*) terdiri dari partikel-partikel yang ukuran maupun beratnya lebih kecil dari pada sedimen, misalnya tanah liat, bahan-bahan organik tertentu, sel-sel mikroorganisme, dan sebagainya. (Fardiaz, 1992 disitasi dalam Nadhiroh, 2014, p6).

Menurut Annisah (2011), disitasi dalam Azizah (2016), p10, kadar TSS yang tinggi akan menghalangi masuknya sinar matahari ke dalam air, sehingga akan mengganggu proses fotosintesis yang menyebabkan turunnya kadar oksigen terlarut yang dilepas kedalam air oleh tanaman. Apabila sinar matahari terhalangi dari dasar tanaman, maka tanaman akan berhenti memproduksi oksigen dan akan mati. TSS juga menyebabkan penurunan kejernihan dalam air.

2.3.2 BOD

BOD adalah banyaknya oksigen yang diperlukan untuk menjernihkan limbah kembali dengan menguraikan benda organik oleh bakteri yang diukur dalam ppm atau milligram per liter (mg/l) (Sugiharto, 1987 disitasi dalam Ghazali, 2015 p.47). Apabila dalam air banyak mengandung bahan-bahan organik, akan mengakibatkan semakin banyaknya oksigen yang diperlukan oleh bakteri untuk menguraikan bahan-bahan organik tersebut, sehingga kandungan oksigen dalam air akan semakin menurun. Semakin besar angka BODnya, semakin kotor pula air tersebut. Pengukuran BOD diperlukan, karena BOD merupakan parameter yang digunakan untuk menentukan tingkat pencemaran air limbah.

2.3.3 COD

COD adalah banyaknya oksigen yang diperlukan untuk menguraikan benda organik secara kimiawi. COD diukur dalam satuan ppm atau milligram per liter (mg/l) (Sugiharto, 1987 disitasi dalam Ghazali, 2015 p.47).

2.3.4 DO

DO adalah banyaknya oksigen yang terkandung di dalam air dan diukur dalam satuan milligram per liter (mg/l). Oksigen yang terlarut ini dipergunakan sebagai penanda tingkat pengotoran limbah yang ada. Semakin besar oksigen yang larut, maka semakin kecil tingkat pengotorannya (Sugiharto, 1987 disitasi dalam Ghazali, 2015 p.47).

2.3.5 pH

Konsentrasi ion hidrogen yaitu ukuran kualitas dari air maupun air limbah. Kadar yang di mana masih memungkinkan kehidupan di dalam air berjalan dengan baik adalah termasuk kadar yang baik. Air limbah dengan konsentrasi tidak netral akan menghambat proses biologis, sehingga akan mengganggu proses penjernihannya. pH yang baik untuk air minum dan air limbah adalah netral (7). Semakin kecil nilai pH-nya, maka akan menyebabkan air tersebut asam (Sugiharto, 1987 disitasi dalam Ghazali, 2015 p.47).

2.3.6 Fenol

Fenol merupakan penyebab timbulnya rasa yang ada di dalam air minum. Fenol ini dihasilkan dari industri dan apabila konsentrasi mencapai 500 mg/l masih dapat dioksidasi melalui proses biologis (Sugiharto, 1987 disitasi dalam Ghazali, 2015 p.47).

2.3.7 Minyak dan lemak

Lemak dan minyak merupakan salah satu komponen utama dari bahan makanan yang juga banyak didapatkan di dalam air limbah. Apabila lemak tidak dihilangkan sebelum dibuang ke saluran air limbah dapat menyebabkan munculnya lapisan tipis di permukaan air yang membentuk selaput sehingga dapat memengaruhi kehidupan yang ada di permukaan air (Sugiharto, 1987 disitasi dalam Ghazali, 2015 p.47).

2.4 Metode STORET

Metode STORET merupakan metode yang paling umum digunakan dalam perhitungan penentuan mutu air. Dengan metode STORET dapat diketahui parameter-parameter yang telah memenuhi maupun yang tidak memenuhi baku mutu air. Metode ini memiliki prinsip untuk melakukan perbandingan antara baku mutu air dengan data kualitas air guna menentukan status mutu air. Menurut Keputusan Menteri Lingkungan Hidup (2003), dalam menentukan status mutu air dapat dilakukan dengan menggunakan sistem nilai dari "US-EPA(*Environmental Protection Agency*)" yaitu mengklasifikasikan mutu air ke dalam empat kelas yang ada pada Tabel 2.2.

Tabel 2.2 Klasifikasi status mutu air berdasarkan "US-EPA"

No	Kelas	Kategori	Skor	Keterangan
1	A	Baik Sekali	0	Memenuhi baku mutu
2	B	Baik	-1 s/d -10	Tercemar ringan
3	C	Sedang	-11 s/d -30	Tercemar sedang
4	D	Buruk	≥ -31	Tercemar berat

Sumber: (Keputusan Menteri Negara Lingkungan Hidup, 2003 disitasi dalam Ghazali, 2015, p.15)

Langkah-langkah dalam penentuan status mutu air dengan menggunakan STORET adalah sebagai berikut (Keputusan Menteri Negara Lingkungan Hidup, 2003 disitasi dalam Ghazali, 2015, p.15):

1. Mengumpulkan data hasil pengukuran kualitas air sungai dan debit air secara periodik.
2. Membandingkan data hasil pengukuran dan nilai baku mutu dari masing-masing parameter air sesuai dengan kelasnya.
3. Jika hasil pengukuran memenuhi nilai baku mutu air (hasil pengukuran < baku mutu) maka diberi skor 0,
4. Jika hasil pengukuran tidak memenuhi baku mutu air (hasil pengukuran > baku mutu), maka diberi skor sesuai pada Tabel 2.3.

Tabel 2.3 Penentuan sistem nilai untuk menentukan status mutu air

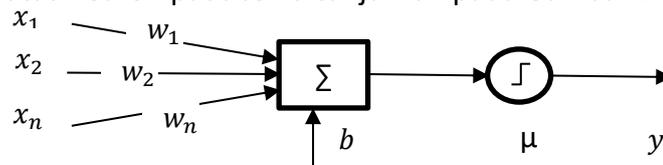
Jumlah Parameter	Nilai	Parameter		
		Fisika	Kimia	Biologi
< 10	Maksimum	-1	-2	-3
	Minimum	-1	-2	-3
	Rata-rata	-3	-6	-9
≥ 10	Maksimum	-2	-4	-6
	Minimum	-2	-4	-6
	Rata-rata	-6	-12	-18

Sumber: (Keputusan Menteri Negara Lingkungan Hidup, 2003 disitasi dalam Ghazali, 2015, p.15-16)

5. Menjumlahkan bilangan negatif pada data hasil perbandingan dari tiap-tiap parameter dari jumlah skor yang diperoleh dengan menggunakan sistem nilai pada Tabel 2.3.
6. Menghitung total jumlah bilangan negatif dari seluruh parameter untuk menentukan status mutunya dengan menggunakan sistem nilai US-EPA pada Tabel 2.2.

2.5 Jaringan Saraf Tiruan

Jaringan saraf tiruan terdiri atas beberapa *node* atau *neuron* dan terdapat *relasi* (hubungan) antara *neuron-neuron* tersebut yang selanjutnya akan mentransformasikan informasi yang akan diterima melalui *neuron-neuron* lainnya. Struktur *node* atau *neuron* pada JST ditunjukkan pada Gambar 2.1 berikut ini.



Gambar 2.1 Struktur *neuron* jaringan saraf tiruan

Struktur *neuron* mempunyai n sinyal masukan, yaitu x_1, x_2, \dots, x_n dengan $x \in \{0; 1\}$. Masing-masing sinyal tersebut kemudian akan dimodifikasi oleh bobot w_1, w_2, \dots, w_n sehingga sinyal yang diterima oleh *neuron* adalah $x_i^1 = x_i w_i, i = 1, 2, \dots, n$. Selanjutnya *neuron* menghitung hasil penjumlahan semua sinyal *input*. Fungsi aktivasi yang terdapat dalam model JST umumnya berupa fungsi *non-linier*. Tingkat aktivasi diwujudkan dalam nilai ambang (*threshold*). Apabila *input* tersebut melebihi suatu nilai ambang tertentu, maka *neuron* tersebut diaktifkan dan mengirimkan *output* (Sinuhaji, 2009).

Berikut ini merupakan fungsi aktivasi sigmoid yang sering digunakan dalam JST menggunakan metode ELM (Singh & Balasundaram, 2007) yaitu:

1. Fungsi aktivasi *sigmoid*

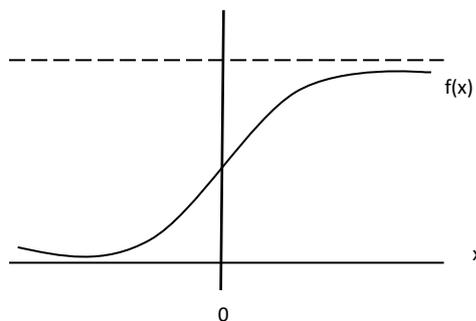
Fungsi aktivasi *sigmoid* memiliki interval *output* 0 sampai 1. Fungsi aktivasi *sigmoid* dapat dirumuskan seperti Persamaan 2.1. Ilustrasi fungsi aktivasi *sigmoid* ditunjukkan pada Gambar 2.2.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

Keterangan:

$f(x)$ = Fungsi aktivasi *sigmoid*

e^{-x} = Eksponensial pangkat minus data ke- x



Gambar 2.2 Ilustrasi fungsi aktivasi *sigmoid*

2.6 Extreme Learning Machine (ELM)

Extreme Learning Machine (ELM) merupakan metode pembelajaran baru dari JST. Metode ini diperkenalkan oleh Huang (2004). ELM adalah jaringan saraf tiruan *feedforward* sederhana dengan satu *hidden layer* yang lebih dikenal dengan istilah *Single Hidden Layer Feedforward Neural Networks* (SLFNs).

Metode ELM ini dibuat untuk mengurangi kelemahan-kelemahan JST *feedforward* terutama mengenai *learning speed*. Huang et al mengemukakan alasan mengapa JST *feedforward* memiliki *learning speed* rendah, yaitu:

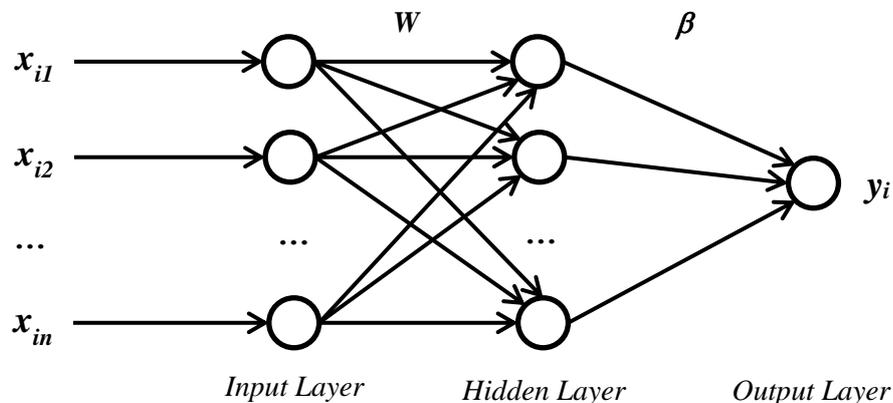
1. Pada proses *training* JST *feedforward* menggunakan algoritme *slow gradient based learning*.

2. Seluruh parameternya ditentukan secara iteratif.

Secara umum, semua parameter dari jaringan *feedforward* ditentukan secara manual. Parameter yang dimaksud yaitu *input weight* dan bias. Parameter antar *layer* saling berhubungan, menyebabkan *learning speed* yang lama (Huang et al., 2005).

Sedangkan pada ELM, parameter-parameter seperti *input weight* dan bias dipilih secara acak dari nilai tertentu, sehingga ELM memiliki *learning speed* yang cepat dan mampu menghasilkan *good generalization performance*. Dengan *random* pada *range* tertentu, bisa menghindari hasil prediksi yang tidak stabil.

Secara umum struktur jaringan saraf tiruan yang menggunakan ELM sebagai metode pembelajarannya menurut Huang, et al. (2005) dapat dilihat pada Gambar 2.3.



Gambar 2.3 Struktur ELM

Metode ELM mempunyai model komputasi yang berbeda dari jaringan saraf tiruan *feedforward*. Model komputasi dari ELM lebih sederhana dan efektif. Struktur dari metode ELM bisa dilihat pada Gambar 2.4.

Langkah-langkah perhitungan dengan metode ELM ini dibedakan menjadi 2 proses, yaitu proses *training* dan proses *testing*.

- **Proses Training**

Sebelum digunakan untuk proses prediksi, ELM harus melalui proses *training* terlebih dahulu. Tujuan dari proses ini adalah mendapatkan *output weight* dengan tingkat kesalahan yang rendah. Langkah-langkah proses *training* metode ELM yaitu sebagai berikut:

1. Inisialisasi *input weight* dan bias dengan bilangan acak yang kecil.
2. Menghitung semua keluaran di *hidden layer* dengan menggunakan fungsi aktivasi. Perhitungan keluaran *hidden layer* ditunjukkan pada Persamaan 2.2. Setelah keluaran *hidden layer* (H) didapatkan, kemudian (H) dihitung dengan menggunakan fungsi aktivasi yang ditunjukkan pada Persamaan 2.1.

$$H_{ij} = \left(\sum_{k=1}^n x_{ik} \cdot w_{jk}^T \right) + b_j \quad (2.2)$$

Keterangan:

H_{ij} = Matriks keluaran *hidden layer*.

i = [1, 2, ..., N], dimana N adalah jumlah data.

j = [1, 2, ..., Ñ], dimana Ñ adalah jumlah *hidden neuron*.

k = Jumlah *input neuron*.

w = Bobot *input* dengan ukuran ordo matriks, *hidden neuron x input neuron*.

w^T = Bobot *input* yang di-*transpose*.

x = Data *input*.

b = Bias.

3. Menghitung Matriks *Moore-Penrose Generalized Inverse/ Moore-Penrose Pseudo Inverse* (H^+) dari hasil keluaran *hidden layer* dengan fungsi aktivasi dengan Persamaan 2.3.

$$H^+ = (H^T H)^{-1} H^T \quad (2.3)$$

Keterangan:

H^+ = Matriks *Moore-Penrose Generalized Inverse*.

H^T = Matriks H yang telah di-*transpose*.

H = Matriks H hasil keluaran *hidden layer* yang telah diaktivasi.

$(H^T H)^{-1}$ = *Inverse* dari perkalian matriks H^T dengan H .

4. Menghitung *output weight* dari *hidden layer* ke *output layer* yang ditunjukkan pada Persamaan 2.4.

$$\beta = H^+ T \quad (2.4)$$

Keterangan:

β = Matriks *output weight* dari *hidden layer* ke *output layer*.

H^+ = Matriks *Moore-Penrose Generalized Inverse*.

T = Matriks Target.

- **Proses Testing**

Proses *testing* dilakukan berdasarkan *input weight*, bias, dan *output weight* yang sesuai dari perhitungan *training*. Proses *training* bertujuan untuk mengembangkan model ELM, sedangkan proses *testing* bertujuan untuk mengevaluasi kemampuan ELM sebagai metode untuk memprediksi. Langkah-langkah proses *testing* yaitu sebagai berikut:

1. Inialisasi *input weight* dan bias yang didapatkan pada perhitungan *training*.
2. Menghitung semua keluaran di *hidden layer* dengan menggunakan fungsi aktivasi menggunakan Persamaan 2.2 dan 2.1.
3. Menggunakan hasil *output weight* dari *hidden layer* ke *output layer* dari proses *training*. Menghitung keluaran pada *output layer* yang merupakan hasil prediksi yang ditunjukkan pada Persamaan 2.4.

$$y = H\beta \quad (2.5)$$

Keterangan:

y = *Output* hasil prediksi.

β = *Output weight*.

H = Keluaran *Hidden layer* dengan fungsi aktivasi.

2.7 Algoritme Evolusi

Algoritme evolusi (*evolutionary algorithms*, EAs) merupakan *subset* dari komputasi evolusi (*evolutionary computation*, EC) yang merupakan bentuk generik dari algoritme optimasi *meta-heuristic* berbasis populasi. Algoritme Evolusi merupakan teknik optimasi yang meniru proses evolusi biologi. Menurut teori evolusi terdapat sejumlah individu dalam populasi. Dari generasi ke generasi, individu-individu ini berperan sebagai induk (*parent*) yang melakukan reproduksi menghasilkan keturunan (*offspring*). Individu-individu ini (beserta *offspring*) berevolusi dan individu-individu yang lebih baik (mampu beradaptasi dengan lingkungannya) mempunyai peluang lebih besar untuk melewati seleksi alam (*natural selection*) dan bertahan hidup. Individu yang lebih baik juga cenderung (tidak selalu tapi mempunyai kemungkinan lebih besar) menghasilkan keturunan yang lebih baik sehingga dari generasi ke generasi akan terbentuk populasi yang lebih baik.

2.8 Algoritme Genetika

Algoritme genetika merupakan algoritme yang populer dan telah digunakan untuk menyelesaikan masalah-masalah optimasi yang bersifat kompleks di bidang fisika, biologi, ekonomi, sosiologi dan lain-lain. (Mahmudy,2015). Algoritme genetika merupakan suatu metode yang menggunakan seleksi alam yang merupakan bagian utama dari prinsip evolusi sebagai dasar pemikiran untuk menyelesaikan suatu permasalahan. Prinsip ini dikemukakan oleh Charles Darwin, dimana tanpa menghiraukan prinsip dasar penurunan sifat, Darwin mengemukakan penggabungan kualitas induk pada generasi berikutnya, disamping itu bahwa individu yang mampu beradaptasi dengan lingkungannya akan mempunyai kesempatan hidup yang lebih besar.

Penggunaan prinsip genetika pada komputer dimulai pada tahun 1950 ketika beberapa ahli Biologi menggunakan komputer untuk simulasi sistem biologi. Akhir tahun 1975 John Holland dari Universitas Michigan melalui paper yang berjudul

“Adaption in Natural and Artificial System” menggunakan konsep dasar algoritme genetika. Algoritme genetika bekerja dengan suatu populasi *string* dan melakukan proses pencarian nilai optimal secara *parallel*, dengan menggunakan operator genetika. Algoritme genetika akan melakukan rekombinasi antar individu. Algoritme genetika memiliki elemen dasar berupa *string* yang tersusun dari rangkaian *substring* (gen), yang masing-masing merupakan kode dari parameter dalam ruang solusi dimana suatu *string* (kromosom) menyatakan kandidat solusi. Kumpulan *string* dalam populasi berkembang dari generasi ke generasi melalui operator genetika. Pada setiap iterasi, individu-individu (kromosom) dalam populasi itu akan dievolusi dan diseleksi untuk menentukan populasi pada generasi berikutnya. Populasi ini akan terus berulang sampai menemukan suatu parameter dengan nilai yang paling optimal sesuai dengan yang diinginkan

2.8.1 Representasi Kromosom

Gen adalah suatu sel dari suatu kromosom atau nilai yang terdapat dalam Algoritme Genetika ini dapat dibentuk oleh sebuah *byte* bahkan tidak menutup kemungkinan suatu *string*. Gen ini mewakili sebagian kecil dari solusi permasalahan.

Individu dalam populasi disebut *string*, *genotype* atau kromosom-kromosom terdiri dari unit-unit yang dinamakan Gen, Karakter, *Decoder*. Kromosom ini dapat mewakili suatu solusi, dimana dapat diilustrasikan pada Gambar 2.4.

Kromosom A	123456789
Kromosom B	987654321

Gambar 2.4 Representasi kromosom

2.8.2 Inisialisasi

Tahap inisialisasi ini bertujuan untuk membangkitkan himpunan solusi baru secara acak yang terdiri dari sejumlah *string* kromosom dan ditempatkan pada penampungan yang disebut dengan populasi. Pada tahap ini μ (μ) yang menyatakan ukuran populasi harus ditentukan. Panjang *string* kromosom (*stringLen*) dihitung berdasarkan presisi variabel dan solusi yang dicari (Mahmudy, 2015).

Misalkan populasi inisialisasi dibangkitkan secara *random* dengan $\mu = 4$ seperti pada Tabel 2.4.

Tabel 2.4 Contoh populasi awal

P(t)	x1	x2	σ_1	σ_2
1	1.48980	2.09440	0,14197	0,91090
2	8.49170	2.57540	0,53801	0,86904
3	1.84610	1.70970	0,99835	0,49351
4	5.81140	5.07790	0,40521	0,98911

Sumber: (Mahmudy, 2015)

2.8.3 Reproduksi

Reproduksi merupakan sebuah proses yang dilakukan untuk menghasilkan keturunan (*offspring*) yang berasal dari induk kromosom yang dipilih secara *random*. Terdapat dua operator genetika dalam reproduksi yaitu *crossover* dan mutasi.

- *Crossover*

Persilangan pada algoritme genetika melibatkan dua buah individu yang dipilih secara acak menjadi *parent* yang akan membentuk kromosom baru. Proses ini diawali dengan penentuan tingkat *crossover* (*crossover rate / cr*). Nilai ini menentukan rasio dari keturunan (*offspring*) yang dihasilkan dari proses *crossover*. Hasil *offspring* dari proses *crossover* diperoleh dengan mengalikan *cr* dengan ukuran populasi (*popsiz*). Pada *Extended Intermediate Crossover*, dipilih 2 *parents* untuk melakukan perhitungan *crossover* menggunakan Persamaan 2.6.

$$\begin{aligned} C(1) &= P(1) + \alpha(P(2) - P(1)) \\ C(2) &= P(2) + \alpha(P(1) - P(2)) \end{aligned} \tag{2.6}$$

Keterangan:

C = Hasil/*Offspring crossover*.

P = *Parent*.

α = *alpha*, nilai dibangkitkan secara *random*.

- Mutasi

Pada proses mutasi digunakan 1 kromosom *parent* sebagai masukannya yang diambil secara *random* dan memperhitungkan masukan *mutation rate*-nya. Sistem melakukan perulangan untuk menghasilkan *offspring* dari hasil mutasi sejumlah *mutation rate* dikalikan dengan ukuran populasi (*popsiz*). Proses mutasi menggunakan metode *random mutation* dilakukan dengan memilih satu posisi atau titik dari suatu kromosom pada *parent* terpilih. Kemudian nilai dari posisi kromosom terpilih, akan dimutasi menggunakan Persamaan 2.7. Setelah proses selesai maka akan tampil individu dari hasil mutasi kromosom yang dipilih secara *random*.

$$x'(i) = x(i) + r(max_i - min_j) \tag{2.7}$$

Keterangan:

$x'(i)$ = Hasil/*Offspring* mutasi.

r = nilai yang dibangkitkan secara *random*.

max_i = nilai maksimal dari 1 kolom seluruh individu (kolom kromosom terpilih).

min_j = nilai minimal dari 1 kolom seluruh individu (kolom kromosom terpilih).

2.8.4 Evaluasi

Melakukan evaluasi pada data kromosom terhadap nilai *fitness* yang digunakan. Nilai *fitness* merupakan suatu ukuran baik tidaknya suatu solusi yang dinyatakan sebagai satu individu, atau dengan kata lain nilai *fitness* menyatakan nilai dari fungsi tujuan. Algoritme genetika mempunyai tujuan untuk memaksimalkan nilai *fitness* atau mencari nilai *fitness* maksimal.

Pada Algoritme Genetika, suatu individu dievaluasi berdasarkan suatu nilai tertentu berdasarkan suatu fungsi tertentu sebagai ukuran nilai *fitness*-nya. Pada masalah optimasi, jika solusi yang dicari adalah memaksimalkan fungsi h , maka nilai *fitness* yang digunakan adalah nilai dari fungsi h tersebut, yakni $f = h$. Sedangkan jika masalahnya adalah meminimalkan fungsi h , maka fungsi h tidak bisa digunakan secara langsung. Sehingga nilai *fitness* untuk masalah minimasi adalah $f = 1/h$, yang artinya semakin kecil nilai h , maka semakin besar nilai f . Namun fungsi ini akan bermasalah jika h bernilai 0, yang akan mengakibatkan f bernilai tak hingga. Maka, h perlu ditambah dengan sebuah bilangan. Sehingga rumus *fitness* bisa menghasilkan solusi yang optimal. *Fitness* yang digunakan pada penelitian ini adalah maksimasi fungsi yang ditunjukkan pada Persamaan 2.8.

$$f(x) = \text{AkurasiPrediksiELM} \quad (2.8)$$

Keterangan:

- x adalah individu yang akan dihitung nilai *fitness*-nya

Sehingga dari setiap individu baik individu *parent* yang terbentuk dari populasi awal maupun individu *offspring* yang terbentuk dari hasil rekombinasi dan mutasi dapat diperoleh nilai *fitness* yang dapat ditunjukkan pada Tabel 2.5 berikut.

Tabel 2.5 Contoh evaluasi

Individu	x1	x2	σ_1	σ_2	f(x1,x2)
P1	1.48980	2.09440	0,14197	0,91090	19.82128
P2	8.49170	2.57540	0,53801	0,86904	34.70609
P3	1.84610	1.70970	0,99835	0,49351	11.58639
P4	5.81140	5.07790	0,40521	0,98911	14.56208
C1	0,17815	1.90205	0,57016	0,70221	16.64183
C2	3.3228	2.14255	0,76818	0,68128	19.5813
C3	3.6506	3.58615	0,27359	0,95001	9.57005
...					
C8	1.94415	2.04997	0,84499	0,74940	19.90344
C9	3.49731	1.81372	0,30095	1.04501	10.98098
C10	7.44343	3.39807	0,42445	0,83617	5.40751
C11	0,25632	1.60705	0,51314	0,63199	11.26206
C12	2.76338	3.23420	0,77195	0,81544	16.32937

Sumber: (Mahmudy, 2015)

2.8.5 Seleksi

Seleksi merupakan tahapan terakhir pada sekali proses evolusi yang dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang akan dipertahankan hidup pada generasi berikutnya. Semakin besar nilai *fitness* kromosom, maka semakin besar peluang kromosom tersebut terpilih. Hal ini dilakukan agar terbentuk generasi berikutnya yang lebih baik dari generasi sekarang (Mahmudy, 2015). Metode seleksi yang digunakan dalam penelitian ini adalah metode seleksi *elitism*.

Metode seleksi *elitism* bekerja dengan cara mengumpulkan semua individu dalam populasi (*parent*) dan *offspring* dalam satu penampungan. Kemudian melakukan *sorting* secara *descending* (dari terbesar ke terkecil) pada nilai *fitness* nya. Individu terbaik/tertinggi akan di ambil sesuai dengan jumlah populasi awal/*miu* (μ). Individu terbaik dalam penampungan akan lolos untuk masuk dalam generasi berikutnya. Metode seleksi *elitism* menjamin individu yang terbaik akan selalu lolos (Mahmudy, 2015). Contoh dari tahapan seleksi dapat dilihat pada Tabel 2.6.

Tabel 2.6 Contoh seleksi

P(t+1)	Asal	x1	x2	σ_1	σ_2	f(x1,x2)
P1	P2	8.49170	2.57540	0,53801	0,86904	34.70609
P2	C7	0,07068	2.09495	0,62717	0,77243	21.33870
P3	C8	1.94415	2.04996	0,84499	0,74940	19.90344
P4	P1	1.48980	2.09440	0,14197	0,91090	19.82128

Sumber: (Mahmudy, 2015)

2.9 Pengujian Akurasi

Pengujian akurasi bermanfaat untuk melihat seberapa optimal metode yang diterapkan pada sebuah penelitian serta menjadi parameter untuk dilakukan evaluasi pada penelitian selanjutnya. Pada penelitian ini pengujian akurasi yang digunakan adalah *K-fold cross validation*. *K-fold cross validation* adalah salah satu metode untuk mengevaluasi kinerja *classifier*, metode ini dapat digunakan apabila memiliki jumlah data yang terbatas (jumlah *instance* tidak banyak), *K-fold cross validation* merupakan salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut *input* yang acak. *K-fold cross validation* diawali dengan membagi data sejumlah *n-fold* yang diinginkan. Dalam proses *cross validation* data akan dibagi dalam *n* buah partisi dengan ukuran yang sama $D_1, D_2, D_3 \dots D_n$ selanjutnya proses uji dan latih dilakukan sebanyak *n* kali. Dalam iterasi ke-*i* partisi D_i akan menjadi data uji dan sisanya akan menjadi data latih.

Cara kerja *K-fold cross validation* adalah sebagai berikut:

1. Total *instance* dibagi menjadi beberapa bagian.

2. *Fold* ke-1 adalah ketika bagian ke-1 menjadi data uji (data *testing*) dan sisanya menjadi data latih (data *training*). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut. Perhitungan akurasi tersebut dengan menggunakan persamaan 2.9.

$$Akurasi = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \times 100\% \quad (2.9)$$

3. *Fold* ke-2 adalah ketika bagian ke-2 menjadi data uji (data *testing*) dan sisanya menjadi data latih (data *training*). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.

Demikian seterusnya hingga mencapai *fold* ke-k. Hitung rata-rata akurasi dari K buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi *final*.