

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini dijelaskan mengenai beberapa kajian pustaka dan teori terkait implementasi algoritma *Extreme Machine Learning* (ELM) untuk prediksi beban pemanasan dan pendinginan bangunan.

2.1 Kajian Pustaka

Kajian pustaka dalam bagian ini menjelaskan tentang penelitian sebelumnya yang diperlukan untuk dianalisa sebagai acuan pustaka. Tiap pustaka dianalisa untuk mengetahui bagian terpenting yaitu *input* yang diberikan, proses dari penyelesaian masalah, dan *output* yang dihasilkan. Pada Tabel 2.1 terdapat penjelasan singkat 4 referensi penelitian sebelumnya yang digunakan, tiga dari 4 penelitian merupakan referensi internasional. Sedangkan 1 penelitian merupakan referensi Indonesia. Berikut merupakan rangkuman penelitian yang dimaksud:

Tabel 2.1 Perbandingan Metode Penelitian

Judul	Obyek	Metode	Output
	Input dan Parameter	Proses	Hasil penelitian
<i>Accurate Quantitative Estimation of Energy Performance of Residential Buildings using Statistical</i>	Data desain bangunan	<i>Classical linear regression, Iteratively Reweighted Least Squares (IRLS) dan random forest</i>	Heating Load dan Cooling Load serta nilai pengaruh masing-masing variabel masukan terhadap variabel keluaran (HL dan CL)
<i>Machine Learning Tools</i> (Tsanas & Xifara, 2012)	<ul style="list-style-type: none"> - Relative compactness - Surface area - Wall area - Roof area - Overall height - Orientation - Glazing area - Glazing area distribution 	<ul style="list-style-type: none"> - Eksplorasi data dan analisis statistik - <i>Statistical mapping</i> menggunakan <i>Iteratively Reweighted Least Squares (IRLS)</i> dan <i>Random Forest</i> - <i>Cross validasi</i> dan generalisasi model 	glazing area mempunyai nilai pengaruh 93.12 terhadap HL dan 86.92 terhadap CL tinggi bangunan dengan nilai pengaruh 8.97 terhadap HL dan 9.41 terhadap CL
<i>Modeling Heating and Cooling Loads by Artificial Intelligence for</i>	Data desain bangunan	<i>Support Vector Regression (SVR), Artificial Neural Network (ANN), SVR+ANN,</i>	HL dan CL

Judul	Obyek	Metode	Output
	Input dan Parameter	Proses	Hasil penelitian
<i>Energy-Efficient Building Design</i> (Chou & Bui, 2014)	<ul style="list-style-type: none"> - Relative compactness - Surface area - Wall area - Roof area - Overall height - Orientation - Glazing area - Glazing area distribution - Hidden layer 		<i>ensemble</i> nilai CL menghasilkan MAPE di bawah 4% dan rerata 25,2% lebih baik daripada metode <i>random forest</i> (RF). Prediksi nilai HL metode SVR nilai RMSE 65,9%, MAE 53.8%, dan MAPE 48,1%
<i>Prediction of Parkinson's Disease using Speech Signal with Extreme Learning Machine</i> (Agarwal et al, 2016)	Data pasien	<i>Extreme Machine Learning, SVM dan Neural Network</i>	Akurasi masing-masing metode
	<ul style="list-style-type: none"> - Amplitude - Pulse - Frekuensi - Suara - Pitch - Harmonicity - Bobot masukan - Bias - Bobot keluaran 	<ul style="list-style-type: none"> - Menentukan fungsi aktivasi dan jumlah <i>hidden node</i> - Menentukan bobot masukan dan nilai bias - Menghitung matrik H - Menentukan bobot keluaran 	Test ke-1: akurasi NN 65.18%, SVM 68.66% dan ELM 89.95%. Test ke-2: akurasi NN 67.50%, SVM 72.50 % dan ELM 87.50%. Test ke-3: akurasi <i>Neural Network</i> 72,61%, SVM 82,14% dan ELM 81,55%.
Sistem Pendukung Keputusan Peramalan Jumlah Kunjungan Pasien Menggunakan Metode <i>Extreme Learning Machine</i> (Fardani et al., 2015)	Data Kunjungan Pasien	<i>Extreme Machine Learning</i>	Jumlah Kunjungan Pasien
	<ul style="list-style-type: none"> - Bobot - Nilai bias - Fungsi aktivasi 	<ul style="list-style-type: none"> - Menentukan fungsi aktivasi dan jumlah <i>hidden node</i> - Menentukan bobot masukan dan nilai bias - Menghitung matrik H - Menentukan bobot keluaran 	Dari penelitian tersebut didapatkan nilai MSE yang kecil yaitu 0.027 dengan menggunakan hidden layer sejumlah 7.

Sumber : (Tsanas & Xifara, 2012), (Chou & Bui, 2014), (Agarwal et al, 2016), dan (Fardani et al., 2015)

Pustaka pertama berjudul *Accurate Quantitative Estimation of Energy Performance of Residential Buildings using Statistical Machine Learning Tools* yang dilakukan oleh Tsanas dan Xifara, dalam penelitiannya mereka mengembangkan *statistical learning machine framework* untuk mengetahui pengaruh delapan variabel masukan untuk memprediksi HL dan CL suatu desain bangunan. Variabel tersebut antara lain *relative compactness*, luas permukaan, luas dinding, area atap, tinggi, orientasi, *glazing area* dan *glazing area distribution*. Mereka menghitung pengaruh hubungan masing-masing variabel masukan dengan variabel keluaran (HL dan CL) dengan tujuan mengetahui variabel masukan mana yang paling berpengaruh. Dari penelitian tersebut didapatkan bahwa variabel masukan *glazing area* mempunyai pengaruh hubungan tertinggi dalam perhitungan HL dan CL dengan nilai pengaruh 93.12 terhadap HL dan 86.92 terhadap CL sedangkan variabel masukan dengan nilai pengaruh terendah ialah tinggi bangunan dengan nilai pengaruh 8.97 terhadap HL dan 9.41 terhadap CL (Tsanas & Xifara, 2012).

Selanjutnya pustaka kedua yakni penelitian untuk memprediksi HL dan CL yang pernah dilakukan oleh Chou dan Bui dengan penelitian yang berjudul "*Modeling Heating and Cooling Loads by Artificial Intelligence for Energy-Efficient Building Design*". Menurut penelitian tersebut metode *ensemble* (SVR + ANN) dan *Support Vector Regression* dapat digunakan dalam memprediksi HL dan CL secara substansial. Penggunaan metode *ensemble* untuk prediksi nilai CL menghasilkan *mean absolute percentage error* (MAPE) di bawah 4% dan rerata 25,2% lebih baik daripada metode *random forest* (RF). Sedangkan untuk prediksi nilai HL metode SVR meningkatkan nilai RMSE sebanyak 65,9%, MAE 53.8%, dan MAPE sebanyak 48,1% bila dibandingkan metode RF (Chou & Bui, 2014).

Penelitian selanjutnya tentang algoritma ELM dilakukan oleh Agarwal et al dengan judul *Prediction of Parkinson's Disease using Speech Signal with Extreme Learning Machine*. Tujuan penelitian tersebut untuk memprediksi penyakit *Parkinson* dengan sinyal suara menggunakan ELM. Dalam penelitian ini dilakukan perbandingan antara algoritma ELM, *Neural Network* (NN) dan SVM. Pada tahap pengujian dilakukan tiga kali perbandingan akurasi terhadap data uji yang berbeda. Pengujian pertama dilakukan terhadap data latih dan menghasilkan rata-rata akurasi NN 65.18%, SVM 68.66% dan ELM 89.95%. Kemudian pada pengujian kedua dilakukan terhadap data 40 sampel acak dan menghasilkan akurasi NN 67.50%, SVM 72.50 % dan ELM 87.50%. Sedangkan untuk pengujian ketiga dilakukan terhadap 168 data independen, hasilnya adalah bahwa akurasi *Neural Network* 72,61%, SVM 82,14% dan ELM 81,55%. Dari hasil pengujian tersebut disimpulkan bahwa ELM masih lebih baik daripada algoritma *Neural Network* dan SVM (Agarwal et al, 2016).

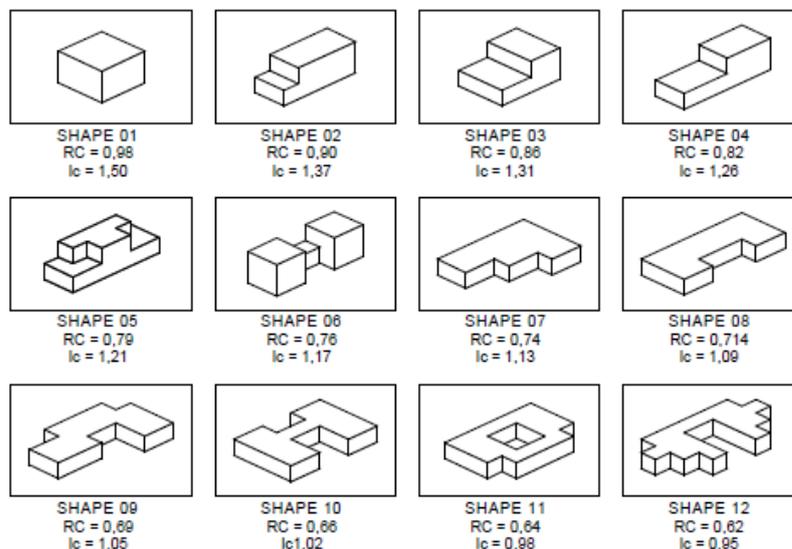
Penelitian keempat berjudul *Sistem Pendukung Keputusan Peramalan Jumlah Kunjungan Pasien Menggunakan Metode Extreme Learning Machine* (Studi Kasus: Poli Gigi RSUD. Wahidin Sudiro Husodo Mojokerto) dilakukan oleh Fardani et al. Dari penelitian tersebut didapatkan nilai MSE yang kecil yaitu 0.027 dengan menggunakan hidden layer sejumlah 7.

2.2 Beban Pemanasan dan Beban Pendinginan Bangunan

Prinsip dasar dari bangunan yang hemat energi adalah mengurangi penggunaan energi yang biasa digunakan untuk pemanas, pendingin dan pencahayaan tanpa mempengaruhi kesehatan dan kenyamanan dari pengguna bangunan tersebut (Duarte & Grasielle, 2017). Bangunan yang hemat energi dapat dicapai dengan desain bangunan yang memperhatikan beban pemanasan (HL) dan beban pendinginan (CL). HL dan CL merupakan ukuran energi yang harus ditambahkan atau ditiadakan dari suatu ruangan dengan cara sistem *heating ventilation and air conditioning* (HVAC) untuk memberikan tingkat suhu yang nyaman dalam suatu ruangan (Chou & Bui, 2017). CL adalah laju aliran kalor yang harus diambil dari suatu ruangan untuk mempertahankan temperatur dan kelembaban udara relatif ruangan pada kondisi yang diinginkan begitu pula sebaliknya dengan HL.

Beban pemanasan dan pendinginan memiliki satuan energi antara lain kCal/jam atau kJ/jam (Harahap, 2014). Beban tersebut akan digunakan dalam perhitungan beban daya penggunaan alat pemanas atau pendingin ruangan. Beban ini dapat dikonversi ke satuan daya lain antara lain Watt, PK (*Pardeen Kraft*), HP (*Horse Power*) dan PS (*PferderStaerke*). Pada penelitian ini akan difokuskan pada faktor desain bangunan yakni *Relative Compactness*, *Surface Area* dan *Wall Area*.

Relative Compactness dari bentuk bangunan diperoleh dari perbandingan volume suatu bangunan dengan rasio permukaan area pada bentuk bangunan yang memiliki kepadatan tertinggi. Berikut contoh relative compactness dari sebuah bentuk bangunan yang bias dilihat pada Gambar 2.1.



Gambar 2.1 Relative Compactness

Sumber: (Pessenlehner & Mahdevi, 2003)

2.3 Normalisasi dan Denormalisasi

Proses normalisasi perlu dilakukan agar mendapatkan data dengan ukuran yang lebih kecil antara 0-1 yang mewakili data asli tanpa kehilangan karakteristik data itu sendiri. Persamaan 2.1 merupakan jenis normalisasi Min-Max yang fungsinya adalah menyediakan transformasi linear pada kisaran asli dari data. Persamaan untuk melakukan proses normalisasi dapat dilihat sebagai berikut:

$$y = \frac{x - Min}{Max - Min} * (B_{atas} - B_{bawah}) + B_{bawah} \quad (2.1)$$

Dengan x merupakan nilai data yang akan dinormalisasi, Min ialah nilai data minimum, Max adalah nilai data maksimum, sedangkan B_{atas} dan B_{bawah} adalah batas atas dan batas bawah (*range*) nilai normalisasi. Dalam hal ini batas atas dan batas bawah yang akan digunakan dalam penelitian ini ialah 0.1 sampai 0.99.

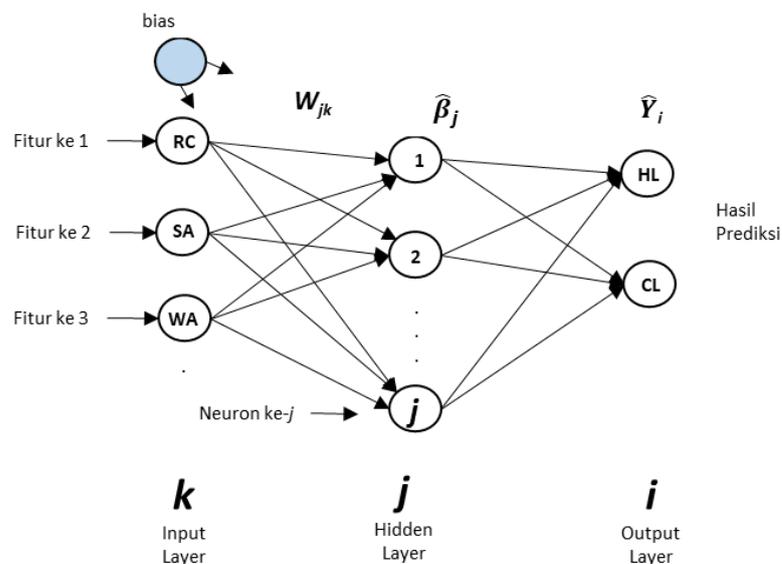
Proses Denormalisasi dilakukan agar mendapatkan data yang asli dengan cara mengembalikan ukuran data yang telah dinormalisasi sebelumnya. Persamaan untuk melakukan proses denormalisasi dapat dilihat pada persamaan 2.2.

$$x = \frac{y - B_{bawah}}{B_{atas} - B_{bawah}} * (Max - Min) + Min \quad (2.2)$$

Dengan y merupakan nilai normalisasi yang akan didenormalisasi. Max dan Min ialah nilai data maksimum dan data minimum. Sedangkan B_{atas} dan B_{bawah} adalah batas atas dan batas bawah (*range*) nilai normalisasi.

2.4 Extreme Learning Machine (ELM)

Extreme learning machine merupakan jaringan syaraf tiruan *feedforward* dengan satu hidden layer atau lebih dikenal dengan istilah *single hidden layer feedforward neural network* (Sun, Choi, Au, & Yu, 2008).



Gambar 2.2 Arsitektur ELM

Pada Gambar 2.2 dapat dilihat arsitektur algoritme ELM yang terdiri dari tiga layer yaitu *input layer* dengan *input* RC, SA dan WA, *hidden layer* dengan *neuron* sebanyak j dan *output layer* dengan *output target* HL dan CL. Dalam arsitektur ELM tersebut terdapat hanya satu *hidden layer* yang merupakan ciri khas dari algoritme ELM. Pada layer *input layer* terdapat masukan dari fitur ke 1 sampai fitur ke k . Sebelum masuk *hidden layer*, bobot *input* W_{jk} menjadi masukan untuk perhitungan h di *hidden neuron* menggunakan fungsi aktivasi. Setelah perhitungan h , *hidden layer* akan menghasilkan $\hat{\beta}_i$ sebagai masukan *output layer* untuk menghitung keluaran \hat{Y}_i .

Tingkat pembelajaran jaringan syaraf *feed-forward* (*feed-forward neural network* - FFNN) lebih banyak mengkonsumsi waktu. Menurut Huang et al. ada dua alasan kenapa ini bisa terjadi, pertama adalah karena penggunaan algoritma pembelajaran dengan gradien yang lambat untuk pelatihan jaringan syaraf dan yang ke dua adalah karena tuning berulang dari parameter jaringan karena algoritma ini. Untuk mengatasi masalah ini, Huang et al. mengajukan sebuah algoritma pembelajaran yang dinamakan *extreme learning machine* (ELM) untuk jaringan syaraf *single hidden layer feed-forward* (SLFN) yang memilih secara acak bobot input dan secara analitik menentukan bobot output SLNF. Huang et al. menyatakan "Secara teori, algoritma ini cenderung memberikan kinerja generalisasi terbaik pada kecepatan pembelajaran yang sangat cepat." (Huang, Zhu, & Siew, 2004). ELM memiliki fitur yang menarik dan signifikan, berbeda dengan algoritma pembelajaran berbasis gradien yang populer untuk jaringan syaraf *feed-forward*. Fitur yang dimaksud adalah sebagai berikut (Huang, Zhu, & Siew, 2006):

1. Kecepatan belajar ELM sangat cepat. Dalam simulasi yang dilaporkan dalam literatur, fase pembelajaran ELM dapat diselesaikan dalam hitungan detik untuk banyak aplikasi. Sebelumnya, tampaknya ada penghalang kecepatan virtual yang sebageian besar algoritma pembelajaran klasik tidak dapat menembusnya. Bukan hal yang tidak biasa lagi kalau pelatihan jaringan syaraf *feed-forward* yang menggunakan algoritma pembelajaran klasik memerlukan waktu yang cukup lama bahkan untuk aplikasi yang sederhana.
2. ELM memiliki kinerja generalisasi yang lebih baik dibandingkan pembelajaran berbasis *gradien*, seperti *backpropagation* dalam kebanyakan kasus. Algoritma pembelajaran klasik berbasis *gradien* dan beberapa algoritma pembelajaran lainnya menghadapi beberapa masalah seperti minima lokal, tingkat pembelajaran yang tidak tepat, dan lain-lain. Untuk menghindari masalah ini, beberapa metode seperti peluruhan bobot dan metode pemberhentian lebih awal sering digunakan pada algoritma klasik ini.
3. ELM cenderung untuk mencapai solusi sederhana tanpa masalah sepele seperti itu. Algoritma pembelajaran ELM terlihat jauh lebih sederhana dari algoritma pembelajaran jaringan syaraf *feed-forward* kebanyakan. Tidak seperti algoritma pembelajaran berbasis *gradien* yang hanya bekerja untuk fungsi aktivasi terdiferensiasi, algoritma ELM dapat

digunakan untuk melatih SLFN dengan banyak fungsi aktivasi yang tidak terdiferensiasi.

2.4.1 Fungsi Aktivasi

Menurut Srimuang & Intarasothonchun (Srimuang & Intarasothonchun, 2015), terdapat beberapa jenis fungsi aktivasi antara lain:

1. Fungsi Sigmoid Biner

$$H = \frac{1}{1 + \exp(-H_{init})} \quad (2.3)$$

2. Fungsi Linear

$$H = H_{init} \quad (2.4)$$

3. Fungsi Sin

$$H = \sin(H_{init}) \quad (2.5)$$

4. Fungsi Radial Basis

$$H = \exp(-(H_{init})^2) \quad (2.6)$$

5. Fungsi Sigmoid Bipolar

$$H = \frac{1 - \exp(-H_{init})}{1 + \exp(-H_{init})} \quad (2.7)$$

6. Fungsi Hard Limit

$$H = \begin{cases} 1, & \text{jika } H_{init} \geq 0 \\ 0, & \text{jika tidak} \end{cases} \quad (2.8)$$

7. Fungsi Triangular Basis

$$H = \begin{cases} 1 - \text{abs } H_{init}, & \text{jika } -1 \leq H_{init} \leq 1 \\ 0, & \text{jika tidak} \end{cases} \quad (2.9)$$

2.4.2 Training ELM

Langkah-langkah yang dilakukan untuk *training* menggunakan metode ELM adalah sebagai berikut:

1. Tentukan bobot W dan bias secara *random*.
2. Hitung matriks keluaran pada *hidden layer* menggunakan Persamaan 2.10.

$$H = \frac{1}{1 + \exp(-X_{training} \cdot W^T + \text{ones}(N_{train}, 1) * \text{bias})} \quad (2.10)$$

Dimana X adalah matriks data *training* sedangkan W^T adalah *transpose* matriks bobot.

3. Hitung matriks *Moore-Penrose Generalized Inverse* dengan menggunakan Persamaan 2.11

$$H^+ = (H^T . H)^{-1} . H^T \quad (2.11)$$

H^+ merupakan matriks *Moore-Penrose Generalized invers* dari matriks H . Sedangkan matriks H merupakan matriks yang tersusun dari *output* masing-masing *hidden layer*. Sedangkan Y merupakan matriks target.

4. Hitung matriks bobot keluaran dari *hidden layer* menggunakan Persamaan 2.12.

$$\hat{\beta} = H^+ Y \quad (2.12)$$

Dimana $\hat{\beta}$ ialah matriks bobot keluaran, H^+ ialah matriks *Moore-Penrose* dan Y ialah matriks target.

2.4.3 Testing ELM

Langkah-langkah yang dilakukan untuk *testing* menggunakan algoritma ELM adalah sebagai berikut:

1. Diketahui bobot W dan bias serta $\hat{\beta}$ yang telah diperoleh dari proses *training* sebelumnya.
2. Hitung matriks keluaran *hidden layer* menggunakan Persamaan 2.13.

$$H = \frac{1}{1 + \exp(-X_{testing} . W^T + ones(N_{test}, 1) * bias)} \quad (2.13)$$

Dimana X adalah matriks data *testing* sedangkan W^T adalah *transpose* matriks bobot.

3. Menghitung semua keluaran di unit *output* dengan menggunakan Persamaan 2.14.

$$\hat{Y} = H . \hat{\beta} \quad (2.14)$$

Dimana \hat{Y} ialah matriks target hasil prediksi, H adalah matriks keluaran *hidden layer* dan $\hat{\beta}$ adalah matriks bobot keluaran yang diperoleh dari proses *training*.

2.5 Nilai Evaluasi MAPE

Ketepatan ramalan adalah suatu hal yang penting untuk peramalan, yaitu bagaimana mengukur kesesuaian antara data yang sudah ada dengan data peramalan. *Mean Absolute Percentage Error* (MAPE) dihitung dengan menggunakan kesalahan absolut pada tiap periode dibagi dengan nilai observasi yang nyata untuk periode itu. Kemudian merata-rata kesalahan persentase absolut tersebut. MAPE merupakan pengukuran kesalahan yang menghitung

ukuran presentase penyimpangan antara data aktual dengan data peramalan. Persamaan MAPE dapat dilihat dalam persamaan 2.15.

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y - y_i}{y_i} \times 100 \right| \quad (2.15)$$

Dimana N merupakan jumlah data *testing*, y adalah data prediksi sedangkan y_i adalah data target *testing*.