Prediksi kebutuhan Energi Listrik Dengan Jaringan Saraf Tiruan Metode Backpropagation*

Demak Damianus

Kebutuhan energi listrik merupakan salah satu hal utama yang diprioritaskan oleh penyedia listrik. Perlunya dilalukan perencanaan terhadap pemenuhan kebutuhan energi listrik oleh penyedia listrik setiap tahunnya. Penelitian dilakukan untuk memprediksi kebutuhan energi listrik setiap provinsi di Indonesia pada tarif industry dengan mengembangkan suatu model Jaringan Saraf Tiruan metode Backpropagation menggunakan software Matlab. Beberapa variabel yang digunakan yaitu jumlah daya listrik. Penelitian menghasilkan score sebesar 99% .

I. PENDAHULUAN

Energi listrik merupakan salah satu sumber energi yang terpenting dalam kehidupan sehari-hari, khususnya pada era modernisasi saat ini. Adanya gangguan pasokan energi listrik dapat mengakibatkan terganggunya rutinitas perekonomian masyarakat . Jumlah konsumsi listrik yang tidak tentu dan tidak diperkirakan terlebih dahulu dapat berpengaruh pada kesiapan dari unit pembangkit untuk menyediakan pasokan listrik kepada konsumen. Jika daya yang dikirim dari pembangkit tenaga listrik jauh lebih besar daripada permintaan daya pada beban listrik, maka akan timbul permasalahan pemborosan energi pada perusahaan listrik, terutama pada pembangkit termal. Sedangkan jika daya yang dibangkitkan dan dikirimkan lebih rendah atau tidak memenuhi kebutuhan beban konsumen, maka akan terjadi pemadaman lokal secara bergilir yang akibatnya merugikan pihak konsumen.(Nazmi Fadilah et al., 2020)

Penelitian dilakukan dalam meramal kebutuhan energi listrik di setiap provinsi yang ada di Indonesia yang berfokus pada sektor industri. Prakiraan dilakukan dengan menggunakan Algoritma Jaringan Syaraf Tiruan Backpropagation. Jaringan saraf tiruan (JST) adalah suatu model yang mencoba meniru struktur dan cara kerja jaringan saraf pada otak manusia.(Reddy & Jung, 2016) Jaringan saraf tiruan merupakan metode peramalan yang memiliki tingkat error data yang rendah dan cukup baik untuk melakukan proses generalisasi karena didukung oleh data training dan pada proses pembelajaran yang menyesuaikan bobot sehingga model ini mampu dalam meramalkan data time series untuk periode waktu ke depan.(Nugraha & SN, 2014)

II. TINJAUAN PUSTAKA

Mengacu dari judul yang diangkat terdapat beberapa penelitian terdahulu yang relevan untuk dijadikan referensi dalam penelitian ini. Adapun penelitian tersebut dari (Hammaines et al., 2021) membahas tentang metode Feedforward Neural Network (FFNN) dengan algoritma Backpropagation. Penelitian tersebut memprediksi penggunaan energi listrik dengan model neural network. Feed forward nural network (FFNN), yang telah dikenal akan keunggulannya, memiliki nilai prediksi yang sangat tinggi atau mendekati nilai aktualnya sehingga menghasilkan error yang kecil berdasarkan MSE (Mean Square Error), dan juga mampu mendeteksi atau melakukan analisis untuk permasalahan yang bersifat sangat kompleks. Hasil penelitian ini di lakukan pada Gedung P Fakultas Teknik Elektro, dengan menggunakan parameter terbaik yaitu menggunakan partisi data 70% dan training 30% dan testing, learning rate 0.001, dan epoch sebesar 80 sehingga menghasilkan nilai MSE sebesar 0.35037.

.A. Listrik

Listrik adalah salah satu sumber energi utama yang digunakan pada seluruh aspek kehidupan. Kebutuhan energi listrik semakin meningkat seiring dengan kemajuan pembangunan di bidang teknologi, industri dan informasi. Energi listrik merupakan energi yang berkaitan dengan akumulasi arus electron, dinyatakan dalam watt-jam atau kilowatt-jam.

B. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) ialah sebuah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi atau jaringan syaraf manusia. Beberapa istilah dalam jaringan syaraf tiruan adalah sebagai berikut (Sari, 2016) :

a. Input adalah nilai input akan diproses menjadi nilai Output.
b. Output merupakan solusi dari nilai input.
c. Bobot merupakan nilai matematis dalam koneksi antara neuron.
d. Fungsi aktivasi merupakan fungsi yang digunakan untuk menentukan nilai keluaran.
e. Fungsi aktivasi sederhana digunakan untuk mengalikan input dengan bobotnya kemudian menjumlahkannya (disebut penjumlahan sigma) berbentuk linier atau tidak linier dan sigmoid.

f. Neuron atau node adalah unit dari sel syaraf tiruan yang merupakan elemen pengolahan jaringan syaraf tiruan. Setiap neuron menerima input, memproses input, kemudian mengirimkan hasilnya berupa output.
g. Lapisan tersembunyi (hidden layer) adalah lapisan tidak secara langsung berinteraksi dengan dunia luar. Lapisan ini memperluas kemampuan jaringan syaraf tiruan dalam menghadapi masalah-masalah yang kompleks

C. Pelatihan Standar Backpropagation

Pelatihan backpropagation meliputi tiga fase yaitu sebagai berikut :

a. Fase I : Propagasi Maju

Dalam propagasi maju, sinyal masukan (= xi) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang di insialisasi. Keluaran dari setiap unit layar tersembunyi (= zj) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan (= yk).

Berikutnya, keluaran jaringan (= yk) dibandingkan dengan target yang harus dicapai (= tk). Selisih tk – yk adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

b. Fase II : Propagasi mundur

Berdasarkan kesalahan tk – yk, dihitung faktor δk (k = 1,2 , ... , m) yang digunakan untuk mendistribusikan kesalahan di unit yk ke semua unit tersembunyi yang langsung terhubung dengan yk. δk juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor δj di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

c. Fase III : Perubahan bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δk yang ada di unit keluaran.

Ketiga fase terebut diulang-ulang terus-menerus sampai kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Adapun algoritma pelatihan untuk jaringan pada satu layar tersembumyi (dengan fungsi aktivasi sigmoid biner) sebagai berikut :

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan Langkah 2-9

Langkah 2 : Untuk setiap pasang data pelatihan, lakukan Langkah 3-8

Fase I : Propagasi maju

Langkah 3 : Tiap unit masukan menerima sinyal dan melanjutkannya ke unit yang tersembunyi di atasnya

Langkah 4 : Hitung semua keluaran di unit tersembunyi zj¬ (j=1,2,..,p)

z_net j = Vjo + ∑_(i=1)^n▒ 〖Xi Vji〗

zj = f(z-net j) =1/(1+ e^(-z_net) j)

Langkah 5 : Hitung semua keluaran jaringan di unit yk (k=1, 2,...,m)

y_net k = wko + ∑_(j=1)^p▒ 〖Zi Wkj〗

yk = f(y-net k) = 1/(1+ e^(-y_net) k)

Fase II : Propagasi mundur

Langkah 6 : Hitung faktor δ unit kerluaran berdasarkan kesalah di setiap unit keluaran yk (k=1,2,..,m)

δ k = (t¬k – yk) f' (y_netk) = (tk – yk) yk (1- yk)

δ k merupakan unti kesalahan yang akan dipakai dalam perubahan bobot layer di bawahnya (Langkah 7)

Hitung suku perubahan bobot wkj (yang akan dipakai nanti untuk merubah bobot wkj) dengan laju percepatan α

Δ wkj = α δ k zj ; k = 1,2,…,m ; j = 0,1,..,p

Langkah 7 : Hitung faktor δ unit tersembunyi berdasarkan kesalah di setipa unit tersembunyi zj (j = 1,2,..,p)

δ _ netj = + ∑_(k=1)^m▒ 〖δ k Wkj〗

faktro δ unit tersembunyi :

δ j = δ _ netj f'(z_netj) = δ _ net j zj (1-zj)

Hitung suku perubahan bobot vji (yang akan dipakai nanti untuk merubah bobot v¬ji)

Δ vji = α δ j xi ; j = 1,2,…,p ; i = 0,1,..,n

Fase III : Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran

wkj (baru) = wkj (lama) + Δ wkj (k = 1,2,..m ; j = 0,1,..,p)

Perubahan bobot garis yang menuju ke unit tersembunyi :

vji (baru) = vji (lama) + Δ vji (j = 1,2,..p ; i = 0,1,..,n)

Sesudah melakukan pelatihan, jaringan bisa dipakai untuk pengenalan pola. Untuk hal ini, hanya propagasi maju (Langkah 4 dan 5) yang dipakai untuk menentukan keluaran jaringan.

Apabila fungsi aktivasi yang dipakai bukan sigmoid biner, maka Langkah 4 dan 5 harus disesuaikan. Demikian juga turunannya pada Langkah 6 dan 7.(Siang, 2005)

III. METODE ANALISIS

Setelah mengumpulkan data distribusi energi listrik tarif industri di Indonesia, dan juga data tersebut sudah di normalisasi. Langkah selanjutnya pada perancangan ini algoritma yang digunakan penelitian penerapan metode jaringan syaraf tiruan algoritma backpropation dengan data numerik dari data jumlah energi listrik, sebagai berikut proses penerapan metode algoritma backpropagation..

A. Memasukkan Data

Untuk memaca data csv pada python menggukan perintah sebagai berikut

```
df = pd.read_csv('datalis.csv')
df.
```

	Provinsi	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
0	Aceh	53.87	58.50	69.67	80.67	97.49	106.56	122.45	144.78	159.59	172.01	190.56
1	Sumatera Utara	2016.23	2134.96	2134.05	2094.13	2076.06	2127.51	2419.85	2819.74	291.53	2946.75	3082.97
2	Sumatera Barat	722.48	772.62	801.43	844.41	840.89	826.31	1001.06	989.05	925.12	857.01	945.24
3	Riau	122.26	121.04	156.97	155.92	199.68	224.33	296.30	403.77	447.31	649.89	1,593.61
4	Jambi	71.84	74.19	96.06	101.72	100.38	93.53	109.65	117.97	151.43	162.34	175.26
5	Sumatera Selatan	536.08	648.03	646.84	680.07	751.18	794.57	911.69	963.34	969.99	930.16	997.88
6	Bengkulu	22.83	25.87	26.39	27.17	31.13	48.41	52.32	66.07	73.95	74.18	79.73
7	Lampung	394.98	490.55	671.10	709.50	725.60	798.03	850.70	918.53	974.28	1005.18	1,100.99
8	Bangka Belitung	29.11	39.44	42.92	45.02	50.72	69.62	107.76	148.84	185.66	224.70	334.89
9	Kepulauan Riau	486.21	538.69	605.93	610.73	547.25	30.26	555.87	636.03	775.80	39.42	39.99
10	DKI Jakarta	9975.67	10958.99	11409.25	11400.94	10303.59	4170.66	4321.57	4508.73	4349.23	3831.81	4,184.30
11	Jawa Barat	17050.46	18535.99	20088.38	21119.03	21226.23	22187.93	26316.17	27478.81	27646.05	21427.96	24,077.60
12	Jawa Tengah	5235.82	5738.43	6475.75	6898.15	6901.46	7227.80	7223.07	8142.04	8269.00	7592.91	8,332.48
13	D.I Yogyakarta	193.86	209.89	225.17	222.38	237.84	239.62	240.07	253.15	253.15	244.51	265.47
14	Jawa Timur	10609.40	12295.75	12737.55	13227.12	13080.88	13838.62	14695.72	15494.65	15695.49	15081.36	16,469.27
15	Banten	6470.10	6661.35	7454.88	6331.97	6412.78	12810.74	14464.39	15363.05	15644.68	13027.20	14,233.05
16	Bali	116.28	125.97	147.52	160.44	167.67	178.98	173.85	187.52	202.88	189.25	183.47
17	Nusa Tenggara Barat	22.40	34.19	48.87	60.52	68.93	77.17	85.52	105.82	132.18	154.03	178.56
18	Nusa Tenggara Timur	4.81	5.61	7.09	25.43	41.92	41.83	35.03	39.97	48.35	47.97	41.19
19	Kalimantan Barat	77.84	85.78	98.53	92.77	97.13	107.48	135.75	158.83	183.33	197.74	1,100.99
20	Kalimantan Tengah	19.24	20.82	21.79	26.22	29.43	28.31	33.31	58.70	89.82	129.50	167.24
21	Kalimantan Selatan	137.87	155.98	169.11	187.44	194.45	210.79	248.64	319.77	358.77	391.85	433
22	Kalimantan Timur	144.77	146.87	161.31	157.70	171.92	176.43	236.50	274.17	312.89	351.70	408
23	Kalimantan Utara	29.99	26.99	27.20	31.61	27.98	28.56	25.40	28.13	45.53	42.80	50.86
24	Sulawesi Utara	73.24	85.02	103.50	107.81	118.52	156.08	221.50	274.04	320.89	369.90	384.49
25	Sulawesi tenggah	16.78	19.94	21.68	24.75	27.88	25.68	30.77	34.48	36.25	46.21	64.61
26	Sulawesi Selatan	686.78	754.87	741.28	782.49	824.86	904.35	958.73	1057.30	1262.67	1173.91	1,635.68
27	Sulawesi Tenggara	24.56	24.74	24.99	26.82	27.93	32.92	36.59	45.76	47.08	53.80	59.27
28	Gorontalo	15.50	18.10	17.60	17.34	17.65	18.95	23.36	31.40	30.13	36.11	39.8
29	Sulawesi Barat	2.13	2.81	3.89	5.50	6.65	7.68	9.56	13.16	17.59	36.00	38.79
30	Maluku	5.72	7.42	7.89	8.35	9.63	9.91	10.15	10.45	9.99	10.13	10.51
31	Maluku Utara	1.74	1.97	1.86	1.83	1.52	2.40	3.27	5.33	5.97	5.83	6.15
32	Papua Barat	4.56	5.16	5.80	6.25	6.79	8.21	8.36	9.87	8.72	8.20	9.47
33	Papua	2.08	2.08	3.36	5.66	5.75	4.72	5.22	6.26	7.64	9.11	11.84

Gambar 1 Data Asli yang di input pada Python

Sehingga data tersebut dapat dibaca di python.

B. Memisahkan antara Data Input dan Data Target

Sebelum melakukan memisahkan data uji dan data latih, sebaiknya menentukan data input dan data target. Dimana data target berguna sebagai acuan untuk menentukan tingkat akurasi pada data tersebut, sehingga nantinya akan membentuk model yang baik, jika akurasi nya tinggi

```
x=df[['2011','2012','2013','2014','2015','2016','2017','2018','2019','2020']]
y=df[['2021']]
```

.

C. Menentukan Data Latih dan Data Uji

Kemudian memisahkan data latih dan data uji. Dimana data latih itu sebesar 70% dan data uji tersebut 30%.

```
X_train, X_test, y_train, y_test = train_test_split(x, y, random_state=1, test_size=0.3)
```

	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
22	144.77	146.87	161.31	157.70	171.92	176.43	236.50	274.17	312.89	351.70
4	71.84	74.19	96.06	101.72	100.38	93.53	109.65	117.97	151.43	162.34
2	722.48	772.62	801.43	844.41	840.89	826.31	1001.06	989.05	925.12	857.01
21	137.87	155.98	169.11	187.44	194.45	210.79	248.64	319.77	358.77	391.85
23	29.99	26.99	27.20	31.61	27.98	28.56	25.40	28.13	45.53	42.80
10	9975.67	10958.99	11409.25	11400.94	10303.59	4170.66	4321.57	4508.73	4349.23	3831.81
29	2.13	2.81	3.89	5.50	6.65	7.68	9.56	13.16	17.59	36.00
28	15.50	18.10	17.60	17.34	17.65	18.95	23.36	31.40	30.13	36.11
18	4.81	5.61	7.09	25.43	41.92	41.83	35.03	39.97	48.35	47.97
6	22.83	25.87	26.39	27.17	31.13	48.41	52.32	66.07	73.95	74.18
13	193.86	209.89	225.17	222.38	237.84	239.62	240.07	253.15	253.15	244.51
7	394.98	490.55	671.10	709.50	725.60	798.03	850.70	918.53	974.28	1005.18
33	2.08	2.08	3.36	5.66	5.75	4.72	5.22	6.26	7.64	9.11
1	2016.23	2134.96	2134.05	2094.13	2076.06	2127.51	2419.85	2819.74	291.53	2946.75
16	116.28	125.97	147.52	160.44	167.67	178.98	173.85	187.52	202.88	189.25
0	53.87	58.50	69.67	80.67	97.49	106.56	122.45	144.78	159.59	172.01
15	6470.10	6661.35	7454.88	6331.97	6412.78	12810.74	14464.39	15363.05	15644.68	13027.20
5	536.08	648.03	646.84	680.07	751.18	794.57	911.69	963.34	969.99	930.16
11	17050.46	18535.99	20088.38	21119.03	21226.23	22187.93	26316.17	27478.81	27646.05	21427.96
9	486.21	538.69	605.93	610.73	547.25	30.26	555.87	636.03	775.80	39.42
31	1.74	1.97	1.86	1.83	1.52	2.40	3.27	5.33	5.97	5.83
8	29.11	39.44	42.92	45.02	50.72	69.62	107.76	148.84	185.66	224.70
12	5235.82	5738.43	6475.75	6898.15	6901.46	7227.80	7223.07	8142.04	8269.00	7592.91

Gambar 2 Data latih

	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
14	10609.40	12295.75	12737.55	13227.12	13080.88	13838.62	14695.72	15494.65	15695.49	15081.36
19	77.84	85.78	98.53	92.77	97.13	107.48	135.75	158.83	183.33	197.74
3	122.26	121.04	156.97	155.92	199.68	224.33	296.30	403.77	447.31	649.89
27	24.56	24.74	24.99	26.82	27.93	32.92	36.59	45.76	47.08	53.80
32	4.56	5.16	5.80	6.25	6.79	8.21	8.36	9.87	8.72	8.20
26	686.78	754.87	741.28	782.49	824.86	904.35	958.73	1057.30	1262.67	1173.91
20	19.24	20.82	21.79	26.22	29.43	28.31	33.31	58.70	89.82	129.50
25	16.78	19.94	21.68	24.75	27.88	25.68	30.77	34.48	36.25	46.21
24	73.24	85.02	103.50	107.81	118.52	156.08	221.50	274.04	320.89	369.90
30	5.72	7.42	7.89	8.35	9.63	9.91	10.15	10.45	9.99	10.13
17	22.40	34.19	48.87	60.52	68.93	77.17	85.52	105.82	132.18	154.03

Gambar 3 Data Test

Kemudian melakukan transformasi serta normalisasi data latih dan data uji, dengan membuat perintah sbb:

```
sc_X = StandardScaler()
X_trainscaled = sc_X.fit_transform(X_train)
X_testscaled = sc_X.fit_transform(X_test)
```

Sedangkan untuk memanggil data latih yang sudah di normalisasi dan tranformasi. Sbb

```
X_trainscaled
```

```
array([[-0.43096637, -0.43263284, -0.43360721, -0.42647648, -0.42188408,
        -0.40504775, -0.39146387, -0.39414285, -0.37172237, -0.39179726],
       [-0.44886655, -0.44907089, -0.44728485, -0.43788487, -0.43665884,
        -0.42108931, -0.41260713, -0.41891974, -0.39704992, -0.4292528 ],
       [-0.28917134, -0.29110683, -0.29942589, -0.28652911, -0.28372543,
        -0.27929281, -0.26402762, -0.28074653, -0.27568444, -0.29184659],
       [-0.43265993, -0.43057243, -0.43197218, -0.42041565, -0.41723109,
        -0.39839893, -0.38944038, -0.38690964, -0.36452537, -0.38385556],
       [-0.45913835, -0.45974612, -0.46171921, -0.45217286, -0.45161121,
        -0.43366132, -0.42664985, -0.43317042, -0.41366201, -0.45289789],
       [ 1.98196188,  2.01274664,  1.92417478,  1.86482819,  1.67055337,
         0.36785519,  0.28943235,  0.27755518,  0.26144123,  0.29657093],
       [-0.4659764 , -0.46521492, -0.46660543, -0.45749393, -0.45601638,
        -0.4377017 , -0.42929005, -0.435545  , -0.41804484, -0.45424294],
       [-0.46269482, -0.46175678, -0.46373156, -0.455081  , -0.45374461,
        -0.4355209 , -0.42698987, -0.43265172, -0.41607774, -0.45422118],
       [-0.46531861, -0.46458164, -0.46593465, -0.45343231, -0.44873226,
        -0.43109351, -0.42504473, -0.43129232, -0.41321964, -0.45187526],
       [-0.46089572, -0.45999943, -0.46188901, -0.45307771, -0.45096066,
        -0.42982025, -0.42216285, -0.42715226, -0.40920388, -0.44669091],
       [-0.41891756, -0.41837959, -0.42022094, -0.41329509, -0.40826999,
        -0.39282018, -0.39086882, -0.3974771 , -0.38109353, -0.41299951],
       [-0.36955401, -0.35490266, -0.32674554, -0.31402296, -0.30753563,
        -0.28476513, -0.2890895 , -0.29193262, -0.26797292, -0.26253846],
       [-0.46598867, -0.46538002, -0.46671653, -0.45746132, -0.45620225,
        -0.43827447, -0.43001344, -0.4366395 , -0.41960565, -0.4595618 ],
       [ 0.02837089,  0.0170139 , -0.02008341, -0.0318437 , -0.02863264,
        -0.02750428, -0.02754483,  0.00964275, -0.37507303,  0.12150544],
       [-0.43795905, -0.4373598 , -0.43649786, -0.42591809, -0.42276181,
        -0.40455432, -0.40190632, -0.40788752, -0.38897918, -0.42392998],
       [-0.45327716, -0.4526195 , -0.4528167 , -0.44217473, -0.4372557 ,
        -0.41856794, -0.41047363, -0.41466706, -0.3957699 , -0.42734007],
       [ 1.12154332,  1.04074849,  1.09526368,  0.8318026 ,  0.86700602,
         2.03975272,  1.98002953,  1.99929848,  2.03331075,  2.11542548],
       [-0.33492196, -0.31928537, -0.33183089, -0.32002062, -0.30225274,
        -0.28543465, -0.27892374, -0.28482473, -0.26864587, -0.27737747],
       [ 3.71842148,  3.72643834,  3.74348537,  3.84531647,  3.92634546,
         3.85428463,  3.95547481,  3.92113501,  3.91591496,  3.77710167],
       [-0.34716222, -0.34401483, -0.34040641, -0.33415169, -0.34436926,
        -0.43333236, -0.33823153, -0.33674358, -0.29910764, -0.45356646],
       [-0.46607212, -0.4654049 , -0.46703096, -0.45824185, -0.45707585,
        -0.4387234 , -0.43033846, -0.43678702, -0.41986762, -0.46021059],
       [-0.45935434, -0.4569303 , -0.458424  , -0.44943999, -0.44691484,
        -0.42571601, -0.41292215, -0.41402305, -0.3916804 , -0.41691795],
       [ 0.81859759,  0.83201148,  0.89001942,  0.94718668,  0.96793039,
         0.95942666,  0.77305207,  0.85388121,  0.87631899,  1.04051913]])
```

Gambar 4 Data latih Tranformasi dan Normalisasi

X_testscaled

```
array([[ 3.15616528,  3.15681161,  3.15740316,  3.15726393,  3.15655904,
         3.15611769,  3.15606096,  3.15534742,  3.15272219,  3.15246601],
       [-0.32470511, -0.32426784, -0.32251346, -0.32538336, -0.32744873,
        -0.32829061, -0.32659658, -0.32849982, -0.33110424, -0.33436115],
       [-0.3100235 , -0.31421517, -0.30642311, -0.30863878, -0.29993088,
        -0.29863881, -0.28819398, -0.27285666, -0.27181782, -0.22843471],
       [-0.34231511, -0.34167043, -0.34276132, -0.34287038, -0.34601759,
        -0.34721093, -0.35031506, -0.354186  , -0.36170419, -0.36808238],
       [-0.34892547, -0.34725272, -0.34804493, -0.34832463, -0.35169021,
        -0.35348133, -0.35706751, -0.36233915, -0.37031934, -0.37876522],
       [-0.12343947, -0.13350935, -0.14554433, -0.1425002 , -0.13217256,
        -0.12607722, -0.1297447 , -0.12439386, -0.08869872, -0.10567108],
       [-0.34407347, -0.34278803, -0.34364238, -0.34302947, -0.34561508,
        -0.34838076, -0.35109962, -0.35124641, -0.35210535, -0.35034793],
       [-0.34488654, -0.34303892, -0.34367267, -0.34341925, -0.346031  ,
        -0.34904815, -0.35170717, -0.35674848, -0.36413647, -0.36986051],
       [-0.3262255 , -0.32448452, -0.32114507, -0.32139542, -0.32170903,
        -0.3159579 , -0.3060857 , -0.3023275 , -0.30021008, -0.29402875],
       [-0.34854207, -0.34660839, -0.34746948, -0.3477678 , -0.35092814,
        -0.35304993, -0.35663935, -0.36220739, -0.37003412, -0.37831307],
       [-0.34302903, -0.33897623, -0.33618641, -0.33393463, -0.33501581,
        -0.33598206, -0.3386113 , -0.34054213, -0.34259186, -0.34460122]])
```

Gambar 5 Data Uji yang sudah di tranformasi dan Normalisasi

D. Klasfikasi Data Daya listrik

Multilayer Perceptron (MLP0 model jaringan saraf tiruan feedforward yang memetakan set data input ke set output yang sesuai. Sebuah MLP terdiri dari beberapa lapisan dan setiap lapisan sepenuhnya terhubung ke yang berikutnya. Node pada layer adalah neuron dengan fungsi aktivasi nonlinier, kecuali node pada input layer. Antara lapisan input dan output mungkin ada satu atau lebih lapisan tersembunyi nonlinier..

Sebagai berikut untuk perintah pada python:

```
clf = MLPClassifier(hidden_layer_sizes=( 128, 64, 32), activation="relu",
random_state=1,max_iter=20000).fit(X_trainscaled, y_train)
y_pred = clf.predict(X_testscaled)
x_pred = clf.predict(X_trainscaled)
print(clf.score(X_trainscaled, y_train))
```

Mendapatkan hasil klasifikasi score prediksi sebesar 95%. Dimana menggunakan hidden layer pertama 128, hidden layer kedua 64 dan hidden layer ketiga 32 neuron.

Sedangkan hasil prediksi pada data latih, menggunakan perintah sbb:

```
print(x_pred)
```

```
['408' '175.26' '945.24' '433' '50.86' '4,184.30' '38.79' '39.8' '41.19'
 '79.73' '265.47' '1,100.99' '6.15' '3082.97' '183.47' '190.56'
 '14,233.05' '997.88' '24,077.60' '39.99' '6.15' '334.89' '8,332.48']
```

Gambar 6 Hasil prediksi data latih

E. MPL-BP Regresion

Multi-layer perceptron (MLP) algoritma pembelajaran terawasi yang mempelajari suatu fungsi f(.) : $R^m \rightarrow R^o$ dengan pelatihan pada dataset, di mana m dalah jumlah dimensi untuk input dan 0 adalah jumlah dimensi untuk output. Diberikan serangkaian fitur X = x1,x2,...,xm dan target, ia dapat mempelajari aproksimator fungsi non-linier untuk klasifikasi atau regresi. Berbeda dengan regresi logistik, di antara lapisan input dan output, bisa ada satu atau lebih lapisan non-linier, yang disebut lapisan tersembunyi.

Sebagai berikut perintah pada python

```
reg =
MLPRegressor(hidden_layer_sizes=(5),activation="relu"
,random_state=1, max_iter=200000).fit(X_trainscaled,
y_train)
```

pada printah tersebut dapat dilihat bahwa jumlah hidden layer nya adalah 5 dan maksimal iterasi sebanyak 200000 iterasi.

Kemudian mencari score regression prediksi dengan menggunakan perintah sbb:

```
y_pred=reg.predict(X_testscaled)
score = r2_score(y_pred, y_test)
print("The Score with ", score)
```

The Score with 0.992150369879066

dan mencari RMSE dengan perintah pada python sebagai berikut

```
from sklearn.metrics import mean_squared_error
rmse = mean_squared_error(y_test, y_pred, squared=False)
```

print("RMSE : ", rmse)
RMSE : 503.53093422485296

IV. KESIMPULAN

Energi listrik merupakan salah satu sumber energi yang terpenting dalam kehidupan sehari-hari, khususnya pada era modernisasi saat ini. Adanya gangguan pasokan energi listrik dapat mengakibatkan terganggunya rutinitas perekonomian masyarakat . Jumlah konsumsi listrik yang tidak tentu dan tidak diperkirakan terlebih dahulu dapat berpengaruh pada kesiapan dari unit pembangkit untuk menyediakan pasokan listrik kepada konsumen. Dengan menggunakan Jaringan syaraf tiruan (JST) ialah sebuah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi atau jaringan syaraf manusia. Untuk data yang digunakan sebesar 70% data latih dan 30% data uji, Mendapatkan hasil MLP klasifikasi score prediksi sebesar 95%. Dimana menggunakan hidden layer pertama 128, hidden layer kedua 64 dan hidden layer ketiga 32 neuron. Sedangkan MLP Regression mendapatkan The Score with 0.992150369879066 dan RMSE : 503.53093422485296..

REFERENCES

[1] Atia Sari, D. (2006). Peramalan Kebutuhan Beban Jangka Pendek Menggunakan Jaringan Syaraf Tiruan Backpropagation. Makalah Seminar Tugas Akhir, 1–12.

[2] Nazmi Fadilah, M., Yusuf, A., & Huda, N. (2020). Muhammad Nazmi Fadilah , Akhmad Yusuf , Nurul Huda digunakan untuk menyelesaikan permasalahan yang berhubungan dengan prediksi , kecerdasan buatan yang digunakan mengidentifikasi pola data history dengan. Matematika Murni Dan Terapan, 14(2), 81–92.

[3] Pembangunan, A. (2007). Prakiraan Kebutuhan Energi Listrik Tahun 2006 – 2015 Pada Pt. Pln (Persero) Unit Pelayanan Jaringan (Upj) Di Wilayah Kota Semarang Dengan Metode Gabungan. Transmisi, 9(1), 52–56.

[4] Setyowati, D. (2019). PRAKIRAAN KEBUTUHAN ENERGI LISTRIK DENGAN JARINGAN SARAF TIRUAN (ARTIFICIAL NEURAL NETWORK) METODE BACKPROPAGATION TAHUN 2020-2025 (Studi Kasus: PT PLN (Persero) UP3 Semarang). 2025, 2019.

[5] Siang, J. J. (2005). Jaringan Syaraf Tiruan dan Pemograman Menggunakan Matlab. In Pemograman Backpropagation Dengan Matlab (pp. 247–275).

[6] Setyowati, D., & Sunardiyo, S. (2020). Prakiraan Kebutuhan Energi Listrik dengan Jaringan Saraf Tiruan (Artificial Neural Network) Metode Backpropagation Tahun 2020-2025. Jurnal EECCIS, 14(1), 6–9. https://jurnaleeccis.ub.ac.id/.