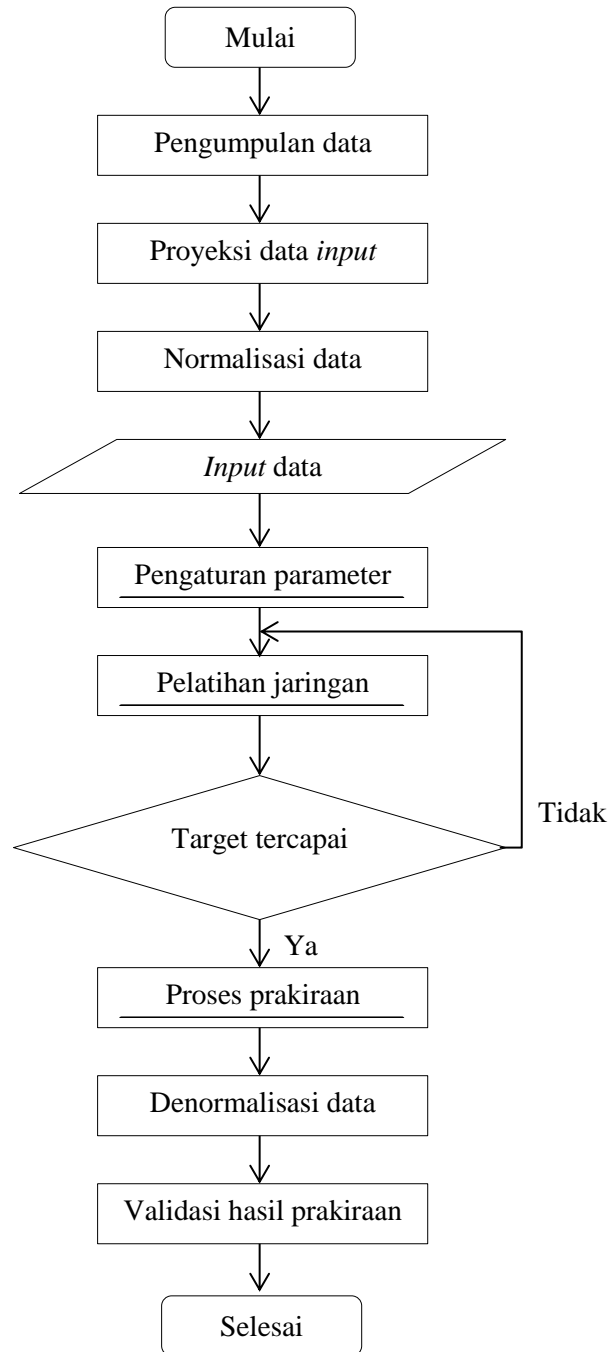


# BAB III

## METODELOGI PENELITIAN

### 3.1 Alur Penelitian

Diagram alir penelitian berguna dalam memberikan informasi dari awal hingga akhir penelitian. Langkah-langkah yang dilakukan dalam proses penelitian dijelaskan secara singkat pada gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

Berdasarkan diagram alir penelitian pada Gambar 3.1 langkah-langkah penelitian adalah sebagai berikut:

1. Penentuan data masukan dan data target

Data masukan penelitian berupa data masukan dan data target, data masukan dijadikan variabel pada lapisan masukan sedangkan data target dijadikan variabel pada lapisan keluaran. Ini adalah data masukan yang ditunjukkan pada tabel 3.1.

Tabel 3.1 Data Masukan Pelatihan Jaringan Syaraf Tiruan

Tahun	PDRB	Penduduk	Pelanggan Listrik (ribu)			
			RT	Bisnis	Publik	Industri
2011	290545838,9	11005518	1659	74	39	5.2
2012	310385592,4	11248947	1795	84	42	5.3
2013	331099105,5	11452491	1992	93	45	5.6
2014	349584667,8	11704877	2178	101	47	5.7
2015	369209288,1	11955243	2399	111	52	6
2016	389543932,3	12203148	2548	132	56	6.1
2017	412639618,3	12448160	2712	152	62	6.3
2018	436581428,1	12689736	2894	174	62	6.3
2019	459828497,4	12927316	3078	175	76	6.7
2020	445672956,7	11904562	3224	191	76	6.8

Berdasarkan Tabel 3.1 data masukan yang akan diolah pada sistem JST *backpropagation* memuat enam variabel yaitu data masukan pelatihan berupa demografi, Pendapatan Domestik Regional Bruto (PDRB), jumlah pengunjung barang pengguna listrik pada rumah tangga, bisnis, publik dan industri. Data target akan disajikan pada tabel 3.2.

Tabel 3.2 Data Target Pelatihan Jaringan Syaraf Tiruan

Tahun	Konsumsi listrik (GWh)			
	RT	Bisnis	Publik	Industri
2011	3681	1931	404,02	11471
2012	4050	1755	410,52	12353
2013	3640	2299	419,08	12920
2014	3892	2243	438,00	12569
2015	4370	2147	440,65	11645
2016	4543	2344	469,10	12811
2017	4599	2521	500,63	13623
2018	4825	2736	543,67	14803
2019	5231	2930	587,46	14601
2020	5794	2890	556,62	13027

Berdasarkan Tabel 3.2 pelatihan model JST *backpropagation* menggunakan empat variabel sebagai sasaran pelatihan berupa data aktual penjualan tenaga listrik pada sektor rumah tangga, bisnis, industri, dan umum. Data target diatas merupakan nilai output yang diinginkan dari proses pelatihan JST *backpropagation*. Nilai keluaran pelatihan dikatakan baik jika mendekati nilai target atau memiliki nilai *error* yang kecil dibandingkan nilai target.

## 2. Proyeksi data *input*

Data proyeksi masukan tersebut dijadikan sebagai contoh data yang akan digunakan dalam proses peramalan kebutuhan energi listrik. Contoh nilai *input* perkiraan sektoral ditunjukkan pada tabel 3.3.

Tabel 3.3 Data *Input* Sampel Prakiraan Sektoral

Tahun	PDRB	Penduduk	Pelanggan Listrik (ribu)			
			RT	Bisnis	Publik	Industri
2021	497393409,9	12951609,05	3631	231	84	7,1
2022	524011559,3	13144004,51	3930	262	92	7,3
2023	552129116,8	13339435,09	4257	298	100	7,6
2024	581834581,2	13537951,33	4614	341	110	7,9
2025	613221904,2	13739604,62	5005	390	120	8,1
2026	646390839,7	13944447,25	5434	448	132	8,4
2027	681447317,1	14152532,45	5904	517	145	8,8
2028	718503838,9	14363914,36	6420	598	159	9,1
2029	757679905,1	14578648,07	6987	694	176	9,4
2030	799102466	14796789,65	7609	808	194	9,8

Berdasarkan Tabel 3.3 sampel data masukan prakiraan industri diperoleh dari masukan data prakiraan menggunakan *trend* eksponensial dengan menggunakan persamaan (2.7). Hasil peramalan menggunakan *trend* eksponensial setiap variabel sebagai sampel nilai *input* untuk *input* data pada proses peramalan industri dari tahun 2021 hingga tahun 2030.

## 3. Normalisasi data

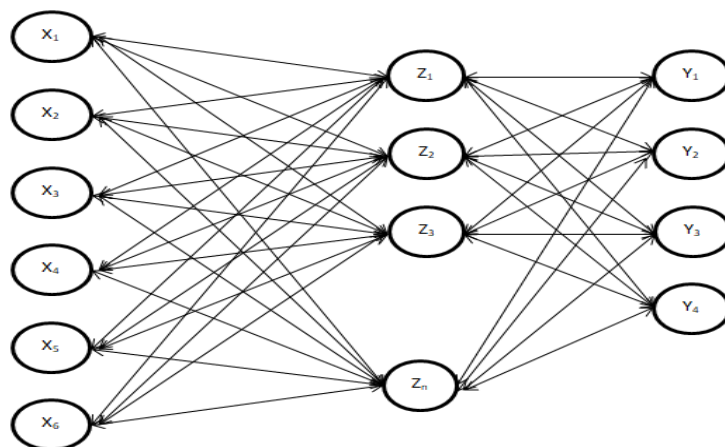
Normalisasi adalah teknik desain database logis yang mengelompokkan atribut hubungan untuk membentuk struktur hubungan yang baik tanpa redundansi yang dapat menghilangkan sebagian besar ambiguitas. Semua data *input*, target, dan sampel dikonversi ke rentang (-1,1). Normalisasi data dilakukan menggunakan Persamaan (3.1).

$$X_n = 2 * (X - X_{min}) / (X_{max} - X_{min}) - 1 \quad (3.1)$$

Nilai  $X_n$  merupakan nilai data normalisasi yang diperoleh menggunakan Persamaan (3.1). Nilai  $X$  merupakan nilai data *real*, nilai  $X_{min}$  merupakan nilai minimum data, nilai  $X_{max}$  merupakan nilai maksimum data.

#### 4. Perancangan arsitektur *backpropagation*

Arsitektur jaringan adalah representasi visual dari koneksi antara berbagai lapisan yang digunakan dalam proses pembelajaran. Di bawah ini adalah desain dari arsitektur model yang telah dikembangkan dalam gambar 3.2.



Gambar 3.2 Arsitektur Jaringan Syaraf Tiraun

Gambar 3.2 menggambarkan jaringan propagasi balik. Setiap unit di lapisan masukan terhubung dengan setiap unit di lapisan tersembunyi. Unit di lapisan tersembunyi juga terhubung ke setiap unit di lapisan keluaran. Arsitektur yang akan digunakan adalah jaringan *multilayer* yang terdiri dari:

- a. Terdiri dari beberapa *input neuron* yang disesuaikan dengan pola masukan di dalam satu lapisan masukan.
- b. Lapisan tersembunyi akan melibatkan proses trial and error dalam penelitian ini, dengan membandingkan tiga model arsitektur berbeda. Model-model ini adalah:
  - Arsitektur dengan 1 lapisan tersembunyi yang memiliki sejumlah neuron yang merupakan kelipatan dari jumlah neuron pada lapisan masukan.
  - Arsitektur dengan 2 lapisan tersembunyi yang juga memiliki sejumlah neuron yang merupakan kelipatan dari jumlah neuron pada lapisan masukan.

- Arsitektur dengan 3 lapisan tersembunyi yang juga memiliki sejumlah neuron yang merupakan kelipatan dari jumlah neuron pada lapisan masukan.

Proses trial and error akan digunakan untuk menentukan model arsitektur yang paling optimal, termasuk mengevaluasi jumlah neuron dalam setiap lapisan tersembunyi tersebut..

- c. Terdapat satu lapisan keluaran yang dapat memiliki beberapa neuron, tergantung pada jumlah keluaran yang diinginkan atau yang diperlukan dalam model ini.

#### 5. Pengaturan parameter jaringan

Proses pembentukan jaringan menggunakan parameter-parameter berikut:

- a. Jumlah *hidden layer* dan *node* pada *hidden layer*

Lapisan tersembunyi berfungsi sebagai tempat untuk memberikan bobot pada masukan yang diterima dari lapisan masukan di setiap *node*. Penentuan jumlah lapisan tersembunyi dan jumlah *node* yang tepat dalam suatu jaringan seringkali tidak dapat dijelaskan dengan rumus matematis tertentu. Dalam penelitian ini, jumlah lapisan tersembunyi ditentukan melalui metode *trial and error*, dengan membandingkan 1 lapisan tersembunyi, 2 lapisan tersembunyi, dan 3 lapisan tersembunyi untuk mencari hasil terbaik yang memiliki tingkat akurasi yang memadai dan sekaligus proses yang lebih cepat. Pemilihan jumlah *node* dalam penelitian ini juga dilakukan melalui pendekatan *trial and error*. Jumlah *node* memiliki dampak signifikan dalam membentuk jaringan yang tepat dan efektif.

- a. Fungsi pelatihan

Fungsi pelatihan memiliki peran penting dalam melatih jaringan untuk memahami pola-pola dalam data dan melakukan peramalan dengan tingkat akurasi yang tinggi. Dalam perangkat lunak MATLAB, terdapat total 12 fungsi pelatihan yang dapat digunakan, antara lain Lavenberg-Marquardt (LM), BFGS Quasi-Newton (BFG), Gradient Descent with Momentum (GDM), Gradient Descent with Momentum and Adaptive Learning (GDX), serta Bayesian Regularization (BR).

Dalam penelitian ini, pemilihan fungsi pelatihan Lavenberg-Marquardt (LM) didasarkan pada hasil percobaan yang dilakukan oleh MATLAB terhadap berbagai fungsi pembelajaran lainnya, serta berdasarkan temuan dalam penelitian sebelumnya.

b. Fungsi aktivasi

Nilai-nilai yang sebelumnya terdapat di dalam *node* lapisan tersembunyi telah diproses dan dihitung. Proses komputasi selama pemrosesan data dilakukan melalui penggunaan fungsi aktivasi. Setiap lapisan tersembunyi menggunakan fungsi aktivasi tertentu untuk melakukan perhitungan dan mengirimkan hasilnya ke lapisan keluaran. Dalam penelitian ini, fungsi aktivasi yang digunakan pada lapisan tersembunyi adalah tansigmoid (Tansig). Pemilihan fungsi ini didasarkan pada fakta bahwa data penelitian telah dinormalisasi ke dalam rentang skala (-1,1) sebelum dimasukkan ke dalam jaringan. Sementara itu, lapisan keluaran hanya menggunakan fungsi *Pureline*.

c. Fungsi performa

Fungsi performa pada jaringan saraf tiruan menggunakan metode *Mean Squared Error* (MSE). Fungsi performa ini berperan dalam menghitung tingkat kesalahan dengan membandingkan selisih antara nilai data asli dan nilai yang dihasilkan oleh proses pelatihan. Dengan menggunakan fungsi performa ini, kita dapat mengukur kinerja proses peramalan dan menentukan metode yang memberikan hasil peramalan terbaik berdasarkan tingkat kesalahan yang ditemukan.

d. Fungsi pembelajaran

Penentuan seberapa cepat jaringan saraf tiruan dapat belajar dan bagaimana gradien yang sesuai untuk mengubah bobot adalah faktor penting dalam mencapai prediksi yang akurat. Jaringan saraf tiruan memiliki fungsi pembelajaran yang memainkan peran kunci dalam proses ini. Dalam konteks ini, terdapat dua fungsi pembelajaran yang digunakan, yaitu *learn*gd dan *learn*gdm.

Dibawah ini adalah tabel yang berisi parameter-parameter jaringan yang digunakan selama proses pelatihan untuk mengatur bagaimana jaringan belajar dan mengoptimalkan bobot ditunjukkan pada Tabel 3.4.

Tabel 3.4 Parameter Jaringan

Parameter	Spesifikasi
Jumlah <i>Neuron</i> Lapisan <i>Input</i>	6
<i>Neuron</i> Pada <i>Hidden Layer</i> 1	6, 12, 18, 24 dan 30
<i>Neuron</i> Pada <i>Hidden Layer</i> 2	6, 12, 18, 24 dan 30
<i>Neuron</i> Pada <i>Hidden Layer</i> 3	6, 12, 18, 24 dan 30
Jumlah <i>Neuron</i> Lapisan <i>Output</i>	4
Fungsi Aktifasi <i>Hidden Layer</i> 1	<i>Tansig</i>
Fungsi Aktifasi <i>Hidden Layer</i> 2	<i>Tansig</i>
Fungsi Aktifasi <i>Hidden Layer</i> 3	<i>Tansig</i>
Fungsi Aktifasi <i>Output Layer</i>	<i>Pureline</i>
Jumlah Iterasi	1000
<i>Goal</i>	0,01
<i>Time</i>	<i>Infinite</i>
<i>Learning rate</i>	0,001

Berdasarkan Tabel 3.4 dapat dilihat parameter jaringan yang digunakan pada proses pelatihan JST *backpropagation*. Parameter pelatihan yang digunakan adalah sebagai berikut.

- a. Batas maksimum *epoch* yang digunakan adalah 1000. Ini berarti proses pelatihan akan berhenti setelah mencapai 1000 iterasi melalui seluruh set data pelatihan, kecuali jika kriteria lain yang telah ditentukan terpenuhi sebelum mencapai batas tersebut. *Epoch* merupakan salah satu parameter penting dalam pelatihan jaringan dan dapat memengaruhi tingkat akurasi dan kecepatan konvergensi jaringan.
- b. Nilai target yang ditetapkan adalah 0,01. Artinya, proses pelatihan akan terus berlanjut hingga nilai error yang dihasilkan oleh jaringan mencapai atau mendekati 0,01, atau pelatihan akan berhenti jika parameter lain yang telah ditentukan tercapai sebelum mencapai nilai target tersebut. Penetapan nilai goal ini penting untuk mengontrol tingkat akurasi yang ingin dicapai oleh jaringan selama proses pelatihan.

- c. *Time* pada pelatihan jaringan saraf merupakan parameter waktu selama proses pelatihan. Pengaturan ini bertindak sebagai pengatur waktu selama pelatihan. Waktu percobaan diatur tidak terbatas sehingga parameter penghentian bergantung pada parameter lainnya.
  - d. Nilai skala learning rate yang digunakan adalah 0,001. Pengaturan nilai learning rate ini memiliki dampak yang signifikan dalam proses pelatihan. Nilai learning rate yang terlalu tinggi dapat menyebabkan konvergensi yang cepat tetapi dapat melewati minimum lokal, sedangkan nilai learning rate yang terlalu rendah dapat menyebabkan pelatihan berjalan sangat lambat atau bahkan tidak konvergen sama sekali. Penggunaan nilai learning rate 0,001 bertujuan untuk meminimalkan perubahan nilai bobot agar dapat mencapai hasil pelatihan yang maksimal dengan menghindari masalah yang disebutkan di atas. Dengan kata lain, nilai ini dipilih agar proses pelatihan berjalan stabil dan konvergen ke minimum global dengan baik..
6. Pelatihan model
- Proses pelatihan model dilakukan dengan tujuan untuk meminimalkan nilai error sehingga model jaringan dapat cocok dan digunakan untuk keperluan prediksi. Dalam pelatihan ini, data dari tahun 2010 hingga 2021 digunakan sebagai data latih untuk mengembangkan model Jaringan Syaraf Tiruan (JST) dengan algoritma backpropagation. Data pelatihan yang digunakan mencakup berbagai informasi seperti data kependudukan, Produk Domestik Regional Bruto (PDRB), dan jumlah pelanggan listrik pada sektor domestik, komersial, publik, dan industri. Sementara itu, target pelatihan adalah data penjualan listrik aktual pada sektor rumah tangga, bisnis, publik, dan industri. Proses pelatihan ini dilakukan secara iteratif, di mana setiap data target diolah satu per satu dengan data masukan yang sama. Tujuannya adalah untuk mengoptimalkan bobot dan parameter jaringan sehingga model dapat melakukan prediksi dengan akurasi yang tinggi berdasarkan data yang diberikan. Dengan demikian, pelatihan model JST backpropagation dapat membantu dalam menghasilkan prediksi yang lebih baik dalam konteks energi listrik.



7. Proses prakiraan

Pelatihan terhadap masing-masing kelompok pelanggan dilakukan untuk mendapatkan nilai bobot dan bias yang kemudian digunakan untuk meramalkan konsumsi listrik. Proses selanjutnya adalah data jumlah pelanggan listrik yang diharapkan dan PDRB atau sampelnya menjadi masukan untuk simulasi pada jaringan yang dilatih. Nilai bobot dan bias yang digunakan untuk pelatihan dan simulasi yang dihasilkan diasumsikan sama untuk mencari nilai konsumsi energi listrik tahun 2021 hingga tahun 2030 semua sektor wilayah pelanggan.

8. Denormalisasi data

Denormalisasi dilakukan untuk memulihkan data yang dinormalisasi lebih lanjut dalam skala (-1,1). Proses ini berdampak pada terciptanya nilai yang telah dinormalisasi ke nilai aslinya. Proses denormalisasi dilakukan dengan menggunakan rumusan Persamaan (3.2).

$$X = 0.5 * (X_n + 1) * (X_{max} - X_{min}) + X_{min} \quad (3.2)$$

Nilai  $X$  merupakan nilai data denormalisasi yang diperoleh menggunakan Persamaan (3.2) nilai  $X_n$  merupakan nilai hasil normalisasi data, nilai  $X_{min}$  merupakan nilai minimum data dan nilai  $X_{max}$  merupakan nilai maksimum data.

9. Validasi dengan data aktual

Eksperimen terhadap data nyata dilakukan untuk mengetahui keakuratan ramalan dengan menurunkan nilai MAPE menggunakan persamaan (2.12). Data aktual yang digunakan dalam proses validasi adalah data aktual yang diketahui pada tahun 2021. Data aktual tahun 2021 dijadikan sebagai data peramalan tahun 2021 dengan menggunakan metode *neural network backpropagation* dan RUPTL PLN tahun 2021 hingga tahun 2030 telah diketahui.

10. Kesimpulan dan saran

Tujuan dari penelitian ini adalah untuk mengevaluasi kinerja Jaringan Syaraf Tiruan dengan algoritma backpropagation dalam meramalkan kebutuhan energi listrik. Hasil yang diharapkan dari penelitian ini adalah pengembangan model yang mampu memberikan nilai *Mean Absolute*

*Percentage Error* (MAPE) yang rendah. MAPE digunakan sebagai ukuran untuk menilai tingkat ketepatan atau kemampuan model dalam mencapai nilai target atau nilai yang diharapkan dalam konteks peramalan kebutuhan energi listrik. Semakin kecil nilai MAPE, semakin baik kinerja model dalam meramalkan kebutuhan energi listrik dengan akurasi yang tinggi.

### **3.2 Studi Literatur**

Penelitian ini dilakukan dengan cara membaca dan memahami buku, jurnal, serta literatur-literatur terkait yang berkaitan dengan topik peramalan energi listrik. Literatur penelitian ini bertujuan untuk memahami teori-teori khusus dan teori pendukung yang relevan dalam pengolahan data peramalan kebutuhan listrik. Hasil dari penelitian literatur ini digunakan sebagai bahan perbandingan dan referensi dalam melakukan peramalan kebutuhan listrik di Provinsi Banten.

### **3.3 Alat dan Bahan**

Alat dan bahan yang digunakan dalam penelitian skripsi ini terdiri dari:

1. Laptop Lenovo X250 dengan Processor Intel Core i5 2.4 GHz, dan sistem operasi Windows 10, digunakan sebagai media perancangan dan pengujian simulasi.
2. Data-data dari beberapa referensi jurnal yang dijadikan sebagai sumber pendukung dalam penelitian skripsi ini.
3. Perangkat lunak Microsoft Office (Excel 2010 dan Word 2010) digunakan untuk pengolahan dan perhitungan data serta pembuatan laporan berdasarkan hasil perhitungan.
4. Perangkat lunak MATLAB R2020a merupakan perangkat lunak utama yang digunakan dalam perancangan dan perhitungan simulasi penelitian ini, dan sudah terinstal di unit laptop.

### **3.4 Pengumpulan Data**

Pengumpulan data dalam penelitian ini melibatkan beberapa instansi dan perusahaan yang menyediakan informasi yang diperlukan untuk perhitungan dalam rangka menghasilkan prakiraan kebutuhan energi listrik. Data historis

tentang kebutuhan energi listrik di wilayah Provinsi Banten diperoleh dari PT. PLN (Persero) sesuai dengan Rencana Usaha Penyediaan Tenaga Listrik (RUPTL) PLN 2021 hingga tahun 2030. Data demografi dan Produk Domestik Regional Bruto (PDRB) diperoleh dari catatan historis Badan Pusat Statistik (BPS) Provinsi Banten. Selain itu, dalam proses pengumpulan data, dilakukan wawancara dan observasi lapangan untuk mengakses dokumen dan teori yang terkait dengan peramalan kebutuhan energi listrik. Data statistik ketenagalistrikan Provinsi Banten yang digunakan dalam penelitian ini juga diperoleh dari PT. PLN (Persero) dan mencakup rentang tahun dari 2011 hingga 2021. Penelitian ini membutuhkan data sebagai dasar untuk melakukan peramalan. Data yang digunakan untuk menghitung peramalan berasal dari berbagai perusahaan yang memiliki dampak signifikan terhadap permintaan energi listrik. Dalam rangka penelitian ini, data yang diperlukan telah diperoleh melalui observasi terhadap beberapa perusahaan dan instansi, dengan sumber informasi utama berasal dari PT. PLN (Persero) dan Badan Pusat Statistik (BPS) Provinsi Banten. Berikut adalah jenis-jenis data yang dikumpulkan:

1. Data pelanggan PLN jenis rumah tangga.
2. Data pelanggan PLN jenis bisnis.
3. Data pelanggan PLN jenis industri.
4. Data pelanggan PLN jenis publik.
5. Jumlah energi yang terjual di sektor rumah tangga.
6. Jumlah energi yang terjual di sektor bisnis.
7. Jumlah energi yang terjual di sektor publik.
8. Jumlah energi yang terjual di sektor industri.
9. Data jumlah penduduk.
10. Data Produk Domestik Regional Bruto (PDRB).

Data-data ini digunakan sebagai sumber informasi untuk melakukan analisis dan peramalan kebutuhan energi listrik dalam penelitian ini.