

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Simulasi

2.1.1 Pengertian simulasi

Menurut Emshoff dan Simmon (1970) yang mana mereka mendefenisikan simulasi sebagai suatu model sistem dimana komponennya direpresentasikan oleh proses aritmatika dan logika yang dijalankan komputer untuk memperkirakan sifat-sifat dinamis sistem tersebut. Menurut Law dan Kelton (1991) simulasi didefenisikan sebagai kumpulan metode dan aplikasi untuk menirukan atau merepresentasikan perilaku dari suatu sistem nyata yang biasanya dilakukan pada komputer dengan menggunakan perangkat lunak tertentu. Menurut Khosnevis (1994) simulasi merupakan proses aplikasi membangun model dari sistem nyata atau susulan sistem, melakukan eksperimen dengan model tersenut untuk menjelaskan perilaku sistem, mempelajari kinerja sistem atau untuk membangun sistem baru sesuai dengan kinerja yang diinginkan. Dari ketiga pengertian diatas secara umum dapat disimpulkan bahwa simulasi adalah proses perancangan model matematis atau logis dari sistem nyata, melakukan eksperimen

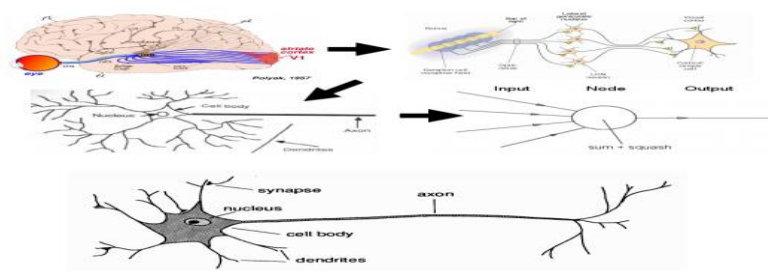
terhadap model dengan menggunakan komputer untuk menggambarkan, menjelaskan dan memprediksi perilaku sistem[Soepono Soeparlan,1995].

2.2 Konsep Dasar Jaringan Saraf Tiruan dan Pemodelannya

2.2.1 Struktur Dasar Jaringan Biologi

Pembuatan struktur dasar jaringan syaraf tiruan diilhami oleh struktur Jaringan Syaraf Tiruan biologi, khususnya jaringan otak manusia. Untuk lebih mengenal asal usul serta bagaimana suatu struktur Jaringan Syaraf Tiruan dibuat dan dapat dipakai sebagai suatu alat penghitung, berikut ini akan diulas sedikit istilah yang secara umum digunakan.

Neuron adalah satuan unit pemroses terkecil pada otak, bentuk sederhana sebuah neuron yang para ahli yang dianggap sebagai satuan unit pemroses tersebut digambarkan sebagai berikut :



Gambar 2.1 Struktur dasar JST dan Struktur Sederhana Sebuah Neuron.

Struktur pada gambar 2.1 tersebut adalah bentuk standar dasar satuan unit jaringan syaraf tiruan otak manusia yang telah disederhanakan. Bentuk standar ini mungkin dikemudian hari akan berubah bila ada ilmuwan yang menciptakan bentuk standar yang lebih baik ataupun memperbaiki bentuk standar yang digunakan saat ini. Jaringan otak manusia tersusun tidak kurang dari 10^{13} buah neuron yang masing-masing terhubung sekitar 10^{15} buah dendrite. Fungsi dendrite adalah sebagai penyampai sinyal dari neuron tersebut ke neuron yang terhubung dengannya. Sebagai keluaran setiap neuron memiliki axon, sedangkan bagian penerima sinyal disebut synapse. Penjelasan lebih jelas tentang hal ini dapat diperoleh dalam disiplin ilmu biologi *molecular*.

Secara umum jaringan syaraf tiruan terbentuk dari jutaan (bahkan lebih) struktur dasar neuron yang terinterkoneksi dan terintegrasi antara satu dengan yang lainnya sehingga dapat melaksanakan aktifitas secara teratur dan terus menerus sesuai dengan kebutuhan.

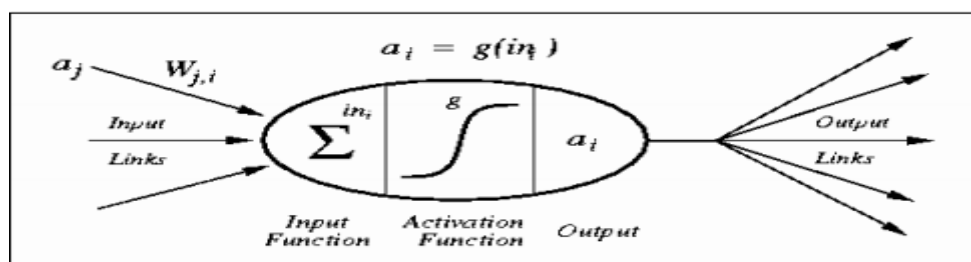
Adapun karakteristik jaringan syaraf tiruan antara lain :

1. Memiliki kemampuan menghasilkan output terhadap pola yang belum pernah dipelajari.
2. Memiliki kemampuan untuk memproses input yg terdapat kesalahan didalamnya dengan tingkat toleransi tertentu.

3. Mampu beradaptasi dengan perubahan yang terjadi terhadap nilai input dan output. Bentuk adaptasi ini diwujudkan dalam perubahan nilai bobot.
4. Akurasi prediksi pada umumnya cukup tinggi.
5. Memerlukan waktu yang relatif lama dan pembelajaran.

2.2.2 Pemodelan Jaringan syaraf tiruan

Tiruan neuron dalam sebuah jaringan syaraf tiruan adalah sebagai elemen pemroses seperti pada gambar 2.2 yang dapat berfungsi seperti halnya sebuah neuron. Sejumlah sinyal masukan a dikalikan dengan masing-masing penimbang yang bersesuaian w . Kemudian dilakukan penjumlahan dari seluruh hasil perkalian tersebut dan keluaran yang dihasilkan dilakukan kedalam fungsi pengaktif untuk mendapatkan tingkatan derajat sinyal keluarannya $F(a,w)$. Walaupun masih jauh dari sempurna, namun kinerja dari tiruan neuron ini identik dengan kinerja dari sel biologi yang kita kenal saat ini.



Gambar 2.2 Model tiruan sebuah neuron

Keterangan gambar 2.2 :

- a_j : nilai aktivasi dari unit j
- $w_{j,i}$: bobot dari unit j ke i
- in_i : penjumlahan bobot dan masukkan ke unit i
- g : fungsi aktivasi
- a_i : nilai aktivasi dari unit i

Misalkan ada n buah sinyal dan n buah penimbang, fungsi keluaran dari neuron adalah seperti persamaan 2.1 berikut :

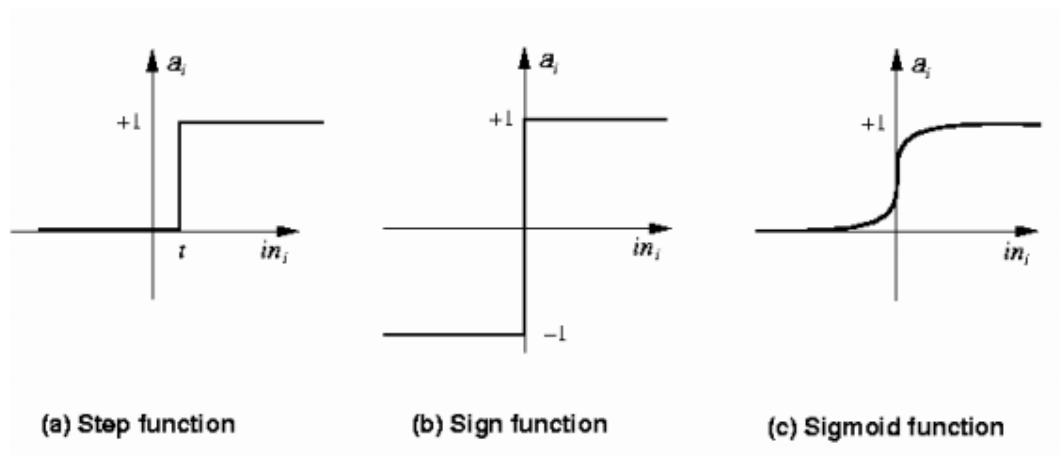
$$in_i = \sum_j W_{ji} \cdot a_j \quad (2.1)$$

Kumpulan dari neuron dibuat menjadi sebuah jaringan yang akan berfungsi sebagai alat komputasi. Jumlah neuron dan jaringan untuk setiap problema yang akan diselesaikan adalah berbeda.

2.2.3 Mengaktifkan Jaringan syaraf tiruan

Mengaktifkan jaringan syaraf tiruan berarti mengaktifkan setiap neuron yang dipakai pada jaringan tersebut. Banyak fungsi yang dipakai sebagai pengaktif, seperti fungsi-fungsi goniometri dan

hiperboliknya, fungsi unit step, impulse, sigmoid, dan lain sebagainya seperti pada gambar 2.3, tetapi yang lazim digunakan adalah fungsi sigmoid karena dianggap lebih mendekati kinerja sinyal pada otak.



Gambar 2.3 Fungsi pengaktif

- $\text{Stept}(x) = 1$ if $x \geq t$ else 0
- $\text{Sign}(x) = +1$ if $x \geq 0$ else -1
- $\text{Sigmoid}(x) = 1/(1+e^{-x})$
- Fungsi Identitas

2.3 Metode Propagasi Balik (*Backpropagation*)

Algoritma pelatihan Backpropagasi (*Backpropagation*) atau ada yang menerjemahkan sebagai Propagasi balik, pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada jaringan syaraf tiruan. Algoritma ini termasuk pelatihan metode

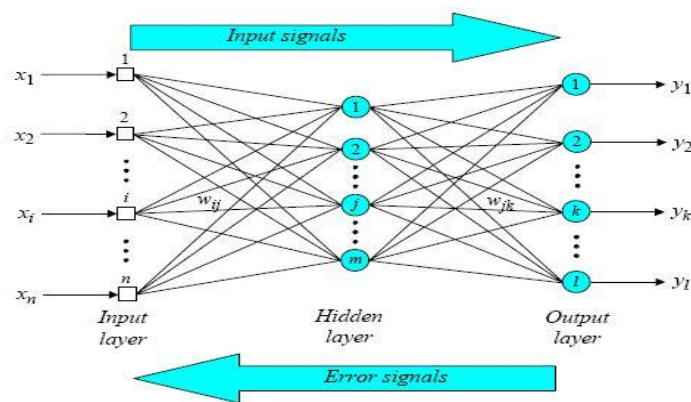
supervised dan didesain untuk operasi pada jaringan *feed forward* multi lapis.

Metode ini banyak diaplikasikan secara luas. Sekitar 90% bahkan lebih backpropagasi telah berhasil diaplikasikan diberbagai bidang diantaranya diterapkan dibidang finansial, pengenalan pola, prediksi dll.

Algoritma ini juga banyak dipakai pada aplikasi pengaturan karena proses pelatihannya didasarkan pada hubungan yang sederhana, yaitu : jika keluaran memberikan hasil yang salah, maka penimbang (*weight*) dikoreksi supaya *error* atau galatnya dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar. Backpropagasi juga memiliki kemampuan untuk memperbaiki penimbang pada lapisan tersembunyi (*hidden layer*) .

Secara garis besar disebut propagasi balik dapat dideskripsikan sebagai berikut, ketika jaringan diberi pola masukkan sebagai pola pelatihan maka pola tersebut menuju unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit lapisan keluaran. Kemudian unit-unit lapisan keluaran memberikan tanggapan yang disebut sebagai keluaran jaringan. Saat keluaran jaringan tidak sama dengan dengan keluaran yang diharapkan maka keluaran akan menyebar mundur (*backward*) pada lapisan tersembunyi diteruskan ke unit pada lapisan masukkan. Oleh karenanya maka mekanisme pelatihan tersebut dinamakan *backpropagation*/ propagasi balik.

Tahap pelatihan ini merupakan langkah bagaimana suatu Jaringan Syaraf Tiruan itu berlatih (pembelajaran), yaitu dengan cara melakukan perubahan penimbang (sambungan antar lapisan yang membentuk jaringan melalui masing-masing unitnya). Sedangkan pemecahan masalah baru akan dilakukan jika proses pelatihan tersebut selesai, fase tersebut adalah fase mapping atau disebut juga proses pengujian.



Gambar 2.4 Arsitektur Jaringan syaraf tiruan backpropagasi

2.3.1 Konsep Dasar Metode Propagasi Balik (Backpropagation)

Pada Jaringan syaraf tiruan propagasi balik terdapat 2 fase, yaitu:

2.3.1.1 Fase pembelajaran

Pada umumnya tujuan pembelajaran/*training* pada jaringan syaraf tiruan adalah untuk mendapatkan

keseimbangan antar sistem jaringan saat diberi masukan pola. Keseimbangan antar sistem jaringan tersebut berupa nilai numerik yang dikenal dengan istilah bobot (weight). Dengan nilai bobot (weight) yang sudah termemori dalam sistem jaringan, maka jaringan syaraf tiruan bisa menghafal jenis pola yang sudah diajarkan kepadanya. Sehingga pada saat pengujian atau penggunaan, jaringan syaraf tiruan bisa mengenali pola input yang diujikan kepadanya. Hal ini sama dengan kemampuan manusia, pada saat pertama kali melihat benda yang belum dikenalnya, maka manusia belum tahu nama benda tersebut. Setelah ada yang mengajarnya, maka secara otomatis otak manusia bisa mengingatnya.

Pada proses pembelajaran *backpropagation error* terdapat 3 tahapan, yaitu:

1. Pemberian pola masukan saat proses pembelajaran.
2. Perhitungan dan proses *backpropagation* dari *error*
3. Pengaturan nilai penimbang.

2.3.1.2 Fase Pengujian

Pada fase pengujian ini, pola yang akan dihitung dimasukkan pada masukan jaringan (*node* pada *layer* masukan). Pola ini akan dikomputasi dengan bobot-bobot interkoneksi hasil fase pelatihan, kemudian hasil aktivasi sel-sel pada lapisan akhir adalah merupakan keputusan dari jaringan saraf tiruan.

Kemudian hasil aktivasi *layer output* tersebut dibandingkan dengan target dan dicari jarak yang terdekat dengan target, dalam hal ini adalah *error* yang terkecil.

Jaringan syaraf tiruan mengenali pola yang diujikan dengan cara membandingkan bentuk pola yang diujikan dengan target pola yang sudah diajarkan sebelumnya. Pengenalan pola diproses dengan cara menggunakan nilai bobot-bobot interkoneksi untuk mengambil keputusan tentang pola yang di ajarkan. Jika bentuk pola yang di ujikan berbeda dengan pola yang di ajarkan, maka jaringan syaraf tiruan memberikan informasi bahwa pola yang diujikan tidak dikenali, atau terjadi *error* yang tidak sesuai dengan standar *error* yang sudah di tetapkan.

2.3.2 Algoritma Metode Propagasi balik (*Backpropagation*)

Langkah 1 : Inisialisasi bobot dengan bilangan nilai acak kecil

Langkah 2 : Selama kondisi berhenti salah, kerjakan langkah 3 s.d. 9

Umpan Maju (Feedforward)

Langkah 3 : Tiap unit masukan (x_i , $i=1, \dots, n$) menerima isyarat masukan x_i dan diteruskan ke unit-unit tersembunyi (hidden layer)

Langkah 4 : Tiap unit tersembunyi (z_j , $j=1, \dots, p$) menjumlahkan bobot sinyal input.

$$Z_in_{jk} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

dengan menerapkan fungsi aktivasi hitung :

$$Z_j = f(Z_in_j)$$

misal, fungsi aktivasi yang digunakan adalah sigmoid :

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

dan mengirimkan isyarat ini ke semua unit pada unit keluaran

Langkah 5 : Tiap unit keluaran (y_k , $k=1, \dots, m$) menjumlahkan isyarat masukan berbobot

$$Y_in_k = w_{0j} + \sum_{k=1}^p z_j v_{jk}$$

dengan menerapkan fungsi aktivasi hitung :

$$Y_j = f(Y_in_k)$$

Perambatan Galat Mundur (Backpropagation)

Langkah 6 : Tiap unit keluaran (y_k , $k=1, \dots, m$) menerima pola pelatihan masukannya. Hitung galat (error) informasinya :

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Hitung koreksi bobot dan biasnya :

$$\begin{aligned}\Delta w_{jk} &= \alpha \delta_k x_j \\ \Delta w_{0k} &= \alpha \delta_k\end{aligned}$$

Langkah 7 : Tiap unit tersembunyi (z_j , $z=1, \dots, p$) menjumlahkan delta masukannya (dari unit-unit yang berada pada lapisan atasnya).

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk}$$

Hitung galat (error) informasinya :

$$\delta_j = \delta_in_j f'(x_in_j)$$

Hitung koreksi bobot dan biasnya :

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Perbaiki bobot dan bias

Langkah 8 : Tiap unit keluaran (y_k , $k=1,\dots,m$) memperbaharui bobot dan bias ($j=0,1,\dots,p$)

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

Tiap unit tersembunyi (z_j , $j=1,\dots,p$) memperbaharui bobot dan biasnya ($i=0,1,\dots,n$)

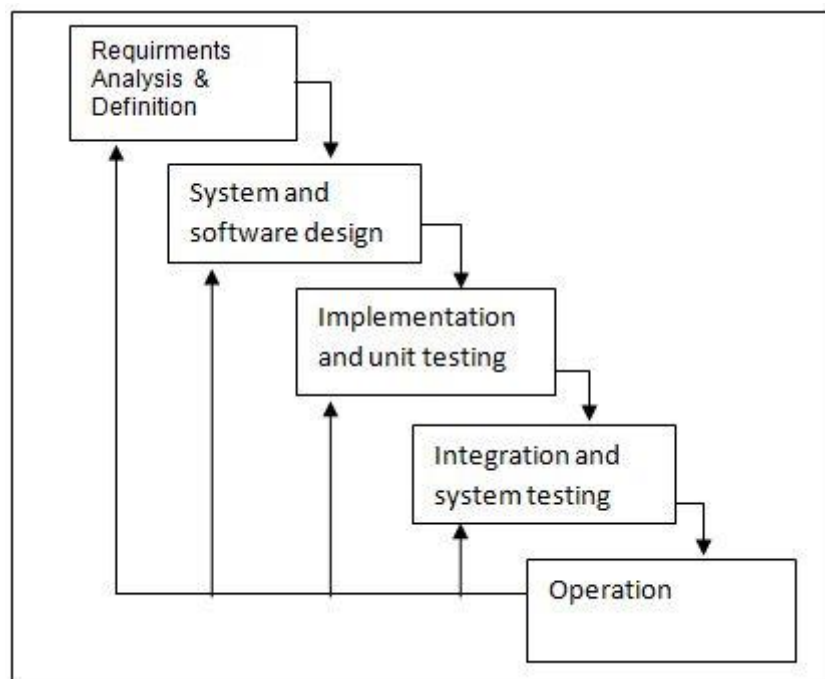
$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

Langkah 9 : Uji syarat berhenti.

2.4 Metode Perancangan Sistem

Dalam membuat perancangan sistem simulasi ini, penulis menggunakan metode waterfall sebagai acuan dalam pengerjaan agar dapat memudahkan tahap-tahap yang akan dikerjakan sehingga simulasi dapat berjalan sesuai dengan yang diinginkan.

Model ini adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Fase-fase dalam Waterfall model menurut referensi Sommerville adalah seperti gambar 2.5 berikut :



Gambar 2.5 Fase-fase Metode Waterfall

Keterangan gambar 2.5 :

1. Requirements analysis dan definition

Mengumpulkan kebutuhan data secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

2. System and software design

Desain dikerjakan setelah kebutuhan data selesai dikumpulkan secara lengkap.

3. Implementation and unit testing

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

4. Integration and system testing

Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).

5. Operation

Mengoperasikan program dilingkungannya.

2.5 Konsep Dasar Sistem

2.5.1 Pengertian sistem

Sistem adalah sekumpulan elemen – elemen yang saling berhubungan dan berinteraksi dengan berbagai cara atau tanggapan untuk melaksanakan tujuan khusus.

2.5.2 Pendekatan sistem

Pendekatan sistem sendiri dibagi menjadi tiga, yakni sebagai berikut :

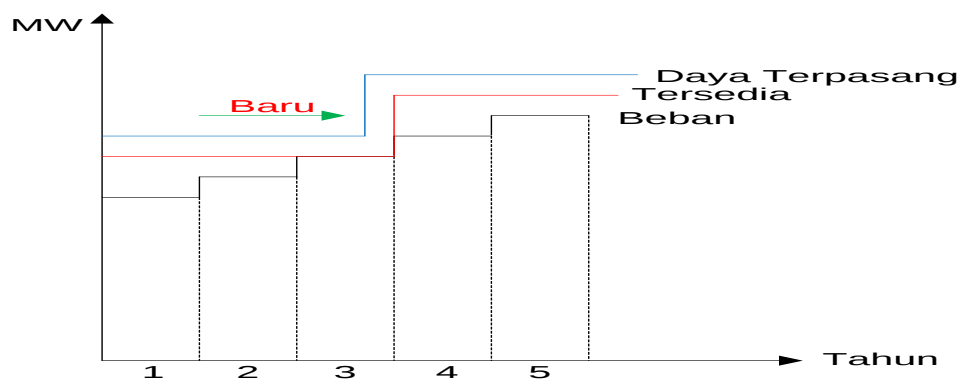
1. Cut-Over, adalah sistem yang lama diganti dengan sistem yang baru
2. Parallel, adalah sistem yang lama dan sistem yang baru sama – sama dijalankan kemudian sistem yang lama diganti dengan sistem yang baru
3. Phase-In, adalah sistem yang baru dicoba dibagian tertentu pada sistem yang lama hingga melewati suatu fase maka sistem yang lama diganti dengan sistem yang baru.

2.5.3 Pemodelan sistem

Secara garis besar pemodelan sistem dapat dibagi menjadi dua, yaitu penerapan struktur model dan pemasokan data. Penerapan struktur merupakan kendala atau batas sistem dan data memberikan nilai yang dapat dimiliki oleh atribut serta menentukan hubungan yang terkandung dalam aktivitas. Kedua cara ini tidak dapat dipisahkan karena yang satu tidak dapat dikerjakan tanpa yang lainnya.

2.6 Tinjauan Umum Perkiraan Beban

Untuk dapat membuat rencana pengembangan sistem kelistrikan perlu dibuat perkiraan beban terlebih dahulu untuk menganalisa kebutuhan energi. Perkiraan beban ini dibuat untuk jangka panjang, sampai 5 atau 10 tahun ke depan, karena untuk membangun pusat listrik baru diperlukan waktu antara 2 sampai 10 tahun. Gambaran perkiraan beban puncak 5 tahun yang akan datang [Maria Amzhelmia Fernandes, 2010].



Gambar 2.6 Perkiraan beban puncak 5 tahun yang akan datang

Berdasarkan gambar 2.6, dapat dilihat bahwa pada tahun ke-3 antara beban puncak dan daya tersedia telah sama besar sehingga perlu dibangun pusat listrik baru atau ditambahkan unit pembangkit baru untuk memenuhi kebutuhan beban.

Faktor-faktor penting yang berpengaruh dalam perkiraan beban melalui suatu investigasi yang cermat dan teliti pada pemilihan parameter yang terkait dengan faktor-faktor ekonomi untuk ramalan beban jangka

panjang, faktor-faktor di bawah ini diperkirakan sangat mempengaruhi ramalan beban tenaga listrik dimasa datang :

1. Jumlah Rumah Tangga (*Household*) adalah jumlah penduduk berdasarkan kepala keluarga.
2. Produk Domestik Bruto (PDB) adalah nilai pasar semua barang dan jasa yang diproduksi oleh suatu negara pada periode tertentu.
3. Rasio Elektrifikasi adalah jumlah rumah tangga yang berlistrik.
4. Elastisitas merupakan perbandingan pertumbuhan ekonomi terhadap kebutuhan listrik.

2.7 Prediksi

Dalam riset ini, membahas masalah prediksi jumlah konsumsi listrik tahun 2011-2019 untuk wilayah Jawa-Bali berdasarkan data yang diperoleh dari tahun 2003-2010 pada Rancangan Usaha Penyedia Tenaga Listrik PLN.

Pemakaian daya dari tahun ke tahun berbeda-beda sehingga diperlukan sebuah sistem yg dapat melihat kemungkinan konsumsi listrik tahun 2011-2019 yang akan datang.

Prediksi itu sendiri sebenarnya mempertimbangkan nilai yang belum terlihat pada masa yang akan datang dengan melihat sebuah pola sebelumnya. Prediksi merupakan meramalkan sesuatu dengan mengoreksi

aksi sebelumnya, untuk meramalkan suatu kondisi dibutuhkan perhitungan yg tepat guna menjawab permasalahan tersebut. Perhitungan tersebut dapat menggunakan Jaringan syaraf tiruan dengan metode backpropagasi.

2.8 Alat Bantu Perancangan Sistem

2.8.1 Flowchart (Bagan Alir)

Flowchart adalah skema atau bagan yang menggambarkan urutan kegiatan dari akhir suatu program dari awal sampai akhir. Atau suatu diagram yang menggambarkan susunan logika suatu program. Diagram alir program merupakan alat yang berguna bagi programmer untuk mempersiapkan program yang rumit. Bagan alir terdiri dari simbol-simbol yang mewakili fungsi-fungsi langkah program dan garis alir menunjukkan urutan dari simbol-simbol yang akan dikerjakan.

Terdapat dua macam *flowchart* yaitu :

1. Sistem *flowchart*

Yaitu bagan yang menggambarkan urutan prosedur dan proses dari beberapa file di dalam media tertentu.

2. Program *flowchart*

Yaitu bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program.

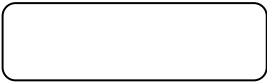
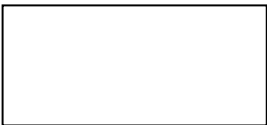
Untuk pengolahan suatu data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu :

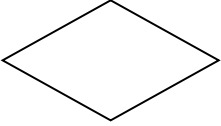
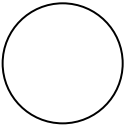
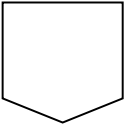
1. *START* : berisi intruksi untuk persiapan peralatan yang diperlukan sebelum menangani pemecahan persoalan.
2. *READ* : berisi intruksi untuk membaca data dari suatu peralatan input.
3. *PROCESS* : berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
4. *WRITE* : berisi intruksi untuk merekam hasil kegiatan ke peralatan output.
5. *END* : mengakhiri kegiatan pengolahan.

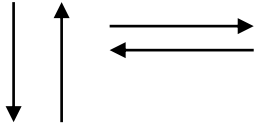
2.8.2 Simbol-simbol *Flowchart*

Berikut rincian mengenai symbol – symbol yang digunakan dalam *flowchart*.

Tabel 2.1 Simbol-simbol Flowchart

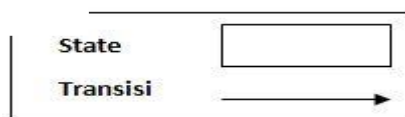
Simbol	Keterangan
	Terminator / Interrupt (mulai/berhenti) : Simbol ini dipergunakan untuk menunjukkan awal kegiatan atau akhir kegiatan atau berhentinya suatu program.
	Input atau output (data atau hasil) : Untuk mewakili data input dan menuliskan outputnya.
	Persiapan atau pemberian harga awal
	Tampilan (layar atau monitor)
	Prose (Pengolahan) : Suatu symbol yang melambangkan diprosesnya suatu data.

	Decission (keputusan) : Menunjukkan suatu perbandingan yang harus dibuat, bila hasilnya 'Ya' maka alir akan masuk ke suatu tempat, bila tidak akan menuju ke tempat lain.
	Connector satu halaman (penghubung) : Bila suatu flowchart sangat panjang dan diputus di tengah sebelum selesai, jika disambung dalam halaman yang sama lagi, maka gunakan simbol ini.
	Connector beda halaman (penghubung) : Bila suatu flowchart sangat panjang dan diputus di tengah sebelum selesai, dan disambung pada halaman yang lain gunakan symbol ini.
	Proses (prosedur) : pengolahan data yang telah didefinisikan.

	Flowlines (Garis alir) : Simbol-simbol dari flowchart dihubungkan dengan garis-garis ini. Garis ini yang menunjukkan arah selanjutnya yang akan dituju.
	Mencetak hasil pada layer.

2.8.3 State Transition Diagram (STD)

State Transition Diagram (STD) merupakan suatu modeling tool yang menggambarkan sifat ketergantungan pada waktu dari suatu sistem. Untuk menyusun suatu STD dibutuhkan beberapa notasi yang dapat menggambarannya. Sebuah kotak persegi panjang merepresentasikan sebuah state, sedangkan setiap transisi atau perubahan state direpresentasikan dengan sebuah garis berarah. Untuk lebih jelasnya perhatikan Gambar 2.7 berikut ini.

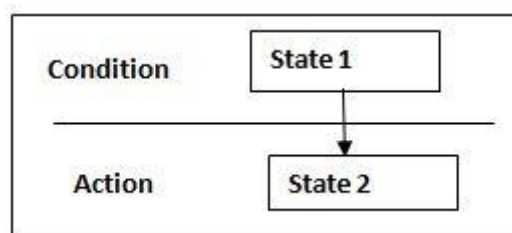


Gambar 2.7 Simbol - Simbol State Transition Diagram

State adalah suatu bentuk yang mewakili keberadaan suatu proses atau komponen sistem yang ditentukan oleh suatu keadaan pada saat tertentu. Sedangkan perubahan state adalah perubahan sistem dari satu state ke state lainnya.

Sebuah STD pun harus dilengkapi dengan *condition* dan *action*. *Condition* adalah suatu event pada external environment yang dapat dideteksi oleh sistem, dapat berupa sinyal, interupsi, atau data yang menyebabkan terjadinya perubahan dari state lainnya. Sedangkan action adalah yang dilakukan sistem bila terjadi perubahan state atau merupakan reaksi terhadap *condition*.

Condition ditulis didekat garis berarah yang menggambarkan arah perubahannya. Sedangkan action yang timbul ditulis dibawah *condition* tadi, dibatasi dengan sebuah garis. Untuk lebih jelasnya perhatikan gambar berikut ini :



Gambar 2.8 Contoh Penggunaan State Transition Diagram

Setiap state selalu dipacu oleh sebuah *condition*. State yang tidak dipacu oleh condition apapun disebut sebagai current state dan state yang dipacu oleh sebuah *condition* namun tidak berubah

menjadi state lain disebut sebagai final state. Dalam sebuah STD, final state tidak harus satu tapi dapat beberapa state, namun untuk initial state tidak boleh lebih dari satu.

Adapun pendekatan yang dapat diikuti untuk pembuatan suatu STD adalah berikut :

1. Identifikasikan setiap kemungkinan state dari sistem dan gambarkan masing state pada sebuah kotak. Lalu buatlah hubungan antara state tersebut.
2. Mulai dengan state pertama dan kemudian lanjutkan dengan state- state berikutnya sesuai dengan *flow* yang diinginkan.
3. Periksa apakah semua state dapat dicapai atau diakses.
4. Setiap state harus memiliki *successor* atau *output* kecuali jika merupakan final state.
5. Perubahan state harus dapat terjadi untuk segala macam kondisi, artinya sistem harus dapat pula mendeteksi bila terdapat *condition* maupun *action* yang tidak diharapkan.