

BAB II

TINJAUAN PUSTAKA

2.1 Definisi Peramalan

Peramalan (*forecasting*) yaitu prediksi nilai -nilai sebuah peubah berdasarkan kepada nilai yang diketahui dari peubah tersebut atau peubah yang berhubungan. Meramal juga dapat didasarkan pada keahlian keputusan (*judgement*), yang pada gilirannya didasarkan pada data historis dan pengalaman [15].

Perpaduan antara seni dan ilmu dalam memperkirakan keadaan di masa yang akan datang, dengan cara memproyeksikan data - data masa lampau ke masa yang akan datang dengan menggunakan model matematika maupun perkiraan yang subjektif disebut sebagai peramalan [16].

2.1.1 Jenis - Jenis Metode Peramalan

Terdapat 2 pendekatan umum untuk jenis metode peramalan yaitu kualitatif dan kuantitatif. Metode peramalan kualitatif sangat penting dimana pada saat data historis tidak ada, tetapi metode ini bersifat sangat subjektif dan membutuhkan penilaian dari para ahli. Di lain pihak peramalan kuantitatif menggunakan data historis yang ada. Tujuan metode ini adalah mempelajari apa yang telah terjadi dimasa lalu untuk dapat meramalkan nilai- nilai dimasa yang akan datang [17].

a. Metode Peramalan Deret Berkala

Metode peramalan deret berkala, atau yang biasa disebut sebagai deret waktu (*time series*), merupakan salah satu metode yang termasuk dalam metode peramalan kuantitatif selain metode regresi atau kausal.

Metode peramalan deret berkala melibatkan proyeksi nilai yang akan datang dari sebuah variabel dengan berdasarkan seluruhnya pada pengamatan masa lalu dan sekarang dari variabel tersebut [17].

b. Metode Pemulusan

Metode pemulusan atau biasa disebut metode *smoothing*, termasuk dalam metode peramalan deret berkala. Metode pemulusan memiliki dasar metode yaitu pembobotan sederhana atau pemulusan pengamatan masa lalu dalam suatu sumber deret berkala untuk memperoleh ramalan masa mendatang [18]. Dalam pemulusan nilai-nilai historis ini, galat acak dirata-ratakan untuk menghasilkan ramalan “halus”. Diantara keuntungannya yaitu biaya yang rendah, mudah digunakan dalam penerapannya, dan cepat dalam penyampaiannya. Karakteristik ini dapat membuatnya menarik terutama bilamana horison waktunya relatif pendek (kurang dari 1 tahun).

Metode pemulusan terdiri atas metode pemulusan perataan, dimana pada saat melakukan pembobotan yang sama terhadap nilai-nilai pengamayan sesuai dengan pengertian konvensional tentang nilai tengah, dan metode pemulusan eksponensial menggunakan bobot berbeda untuk data masa lalu, karena bobotnya berciri menurun seperti eksponensial dari titik data yang terakhir sampai terawal.

2.1.2 Jenis-jenis Pola Data

Langkah penting dalam memilih suatu metode deret berkala (*time series*) yang tepat adalah dengan mempertimbangkan jenis pola data, sehingga metode yang paling tepat dengan pola tersebut dapat diuji.

Pola data dapat dibedakan menjadi empat jenis [15], yaitu:

1. Pola Horizontal atau Horizontal Data Pattern

Pola data ini terjadi bilamana data berfluktuasi di sekitar nilai rata-rata. Suatu produk yang penjualannya tidak meningkat atau menurun selama waktu tertentu termasuk jenis ini.

2. Pola Trend atau Trend Data Pattern

Pola data ini terjadi bilamana terdapat kenaikan atau penurunan sekuler jangka panjang dalam data. Contohnya penjualan perusahaan, GNP (*Gross National Product*) dan berbagai indikator bisnis atau ekonomi lainnya, selama perubahan sepanjang waktu.

3. Pola Musiman atau Seasonal Data Pattern

Pola data ini terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulan atau hari-hari pada minggu tertentu). Penjualan dari produk seperti minuman ringan, es krim dan bahan bakar pemanas ruang semuanya menunjukkan jenis pola ini.

4. Pola Siklis atau Cycled Data Pattern

Pola data ini terjadi bilamana datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Contohnya penjualan produk seperti mobil, baja.

2.1.3 Jangka Waktu Peramalan

Jangka waktu peramalan dapat dikelompokkan menjadi tiga kategori [16], yaitu :

1. Peramalan jangka pendek, peramalan untuk jangka waktu kurang dari tiga bulan.
2. Peramalan jangka menengah, peramalan untuk jangka waktu antara tiga bulan sampai tiga tahun.
3. Peramalan jangka panjang, peramalan untuk jangka waktu lebih dari tiga tahun.

2.1.4 Proses Peramalan

Didalam melakukan proses peramalan, apapun bentuk dan jenis peramalan yang akan dilakukan, terdapat lima langkah proses peramalan yang bisa dilakukan [20], yaitu :

1. Formulasi masalah dan pengumpulan data.

Jika metode peramalan kuantitatif yang dipakai maka data yang relevan harus tersedia dan benar. Jika data yang sesuai tidak tersedia maka mungkin perumusan masalah perlu dikaji ulang atau memeriksa kembali metode peramalan kuantitatif yang dipakai.

2. Manipulasi dan pembersihan data

Ada kemungkinan kita memiliki terlalu banyak atau terlalu sedikit data yang dibutuhkan. Sebagian data mungkin tidak relevan pada masalah. Sebagian data mungkin memiliki nilai yang hilang yang harus diestimasi. Sebagian data mungkin harus dihitung dalam unit selain unit aslinya. Sebagian data mungkin harus diproses terlebih dahulu (misal, dijumlahkan dari berbagai sumber). Data yang lain kemungkinan sesuai tetapi hanya pada periode historis tertentu. Biasanya perlu usaha untuk mengambil data dalam suatu bentuk yang di butuhkan untuk menggunakan prosedur peramalan tertentu.

3. Pembentukan dan evaluasi model

Pembentukan dan evaluasi model menyangkut pengepasan data yang terkumpul pada suatu model peramalan yang sesuai dengan meminimalkan galat peramalan.

4. Implementasi model (peramalan sebenarnya)

Implementasi model terdiri dari model peramalan aktual yang dibuat ketika data yang sesuai telah terkumpul dan terpilihnya model peramalan yang sesuai. Peramalan untuk periode sekarang dengan nilai historis aktual diketahui sering kali digunakan untuk mengecek keakuratan dari proses.

5. Evaluasi peramalan

Implementasi model terdiri dari model peramalan aktual yang dibuat ketika data yang sesuai telah terkumpul dan terpilihnya model peramalan yang sesuai. Peramalan untuk periode sekarang dengan nilai historis aktual diketahui sering kali digunakan untuk mengecek keakuratan dari proses.

2.1.5 Karakteristik Peramalan

Karakteristik dari peramalan yang baik harus memenuhi beberapa kriteria yaitu dari hal-hal sebagai berikut:

1. Ketelitian/ Keakuratan

Tujuan utama peramalan adalah menghasilkan prediksi yang akurat. Peramalan yang terlalu rendah mengakibatkan kekurangan persediaan (inventory). Peramalan yang terlalu tinggi akan menyebabkan inventory yang berlebihan dan biaya operasi tambahan.

2. Biaya

Biaya untuk mengembangkan model peramalan dan melakukan peramalan akan menjadi signifikan jika jumlah produk dan data lainnya semakin besar. Mengusahakan melakukan peramalan jangan sampai menimbulkan ongkos yang terlalu besar ataupun terlalu kecil. Keakuratan peramalan dapat ditingkatkan dengan mengembangkan model lebih kompleks dengan konsekuensi biaya menjadi lebih mahal. Jadi ada nilai tukar antara biaya dan keakuratan.

3. Responsif

Ramalan harus stabil dan tidak terpengaruhi oleh fluktuasi demand.

4. Sederhana

Keuntungan utama menggunakan peramalan yang sederhana yaitu kemudahan untuk melakukan peramalan. Jika kesulitan terjadi pada metode sederhana, diagnosa dilakukan lebih mudah. Secara umum, lebih baik menggunakan metode paling sederhana yang sesuai dengan kebutuhan peramalan.

2.1.6 Perbandingan Algoritma

Beberapa penelitian yang melakukan riset pada time series adalah statistik, jaringan syaraf, wavelet, dan sistem fuzzy. Komponen utama pembentuk *soft computing* seperti jaringan syaraf memiliki keunggulan dalam penyelesaian masalah yang mengandung ketidakpastian, ketidaktepatan, dan kebenaran parsial. Komponen tersebut memiliki kelebihan dan kekurangan yang berbeda. Namun hal tersebut dapat menjadi celah bagi perkembangan ilmu pengetahuan. Karena kelebihan suatu komponen dapat melengkapi kekurangan komponen lainnya. Terlebih lagi, masalah dalam dunia nyata seringkali merupakan masalah yang

kompleks dan satu model mungkin tidak mampu mengatasi masalah tersebut dengan baik.

Tabel 2.1 Arsitektur Perbandingan *Performance Algoritma* [21]

No	Algoritma	Accuracy	AUC
1	Neural Network	89.71%	0.872
2	Naive Bayes	84.70%	0.854
3	Decision Tree	89.10%	0.959
4	Logistic Regression	89.32%	0.993
5	K-Nearest Neighbor	87.79%	0.962

Dalam pembuatan model menggunakan klasifikasi yaitu *Neural Network*, *Naive Bayes*, *Decesion Tree*, *K-NN*, dan *Logistic Regreesion* menggunakan data pemasararan pada Bank. Algoritma Neural Network memiliki Akurasi yang lebih tinggi dengan nilai 89.71% dibandingkan dengan 4 algoritma lainnya [21].

2.2 Jaringan Saraf Tiruan (*Neural Network*)

Neural Network merupakan suatu metode *Artificial Intelligence* yang konsepnya meniru sistem jaringan syaraf yang ada pada tubuh manusia, dimana dibangun *node - node* yang saling berhubungan satu dengan yang lainnya. *Node node* tersebut terhubung melalui suatu *link* yang biasa disebut dengan istilah *weight*. Ide dasarnya adalah mengadopsi cara kerja otak manusia yang memiliki ciri-ciri *pararel processing*, *processing element* dalam jumlah besar dan *fault tolerance*.

Neural Network (Jaringan Saraf Tiruan) adalah prosesor tersebar paralele yang sangat besar dan memiliki kecenderungan untuk menyimpan pengetahuan yang bersifat pengalaman dan membuatnya siap untuk digunakan. NN ini merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Secara sederhana NN adalah sebuah alat pemodelan data statistik non-linear. NN dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data. Neuron juga terdiri dari satu output. Outputnya adalah terbentuk dari pengolahan dari berbagai input oleh neuron-neuron [22].

Pada umumnya *neural network* dibagi berdasarkan *layer-layer* yaitu *input layer*, *hidden layer* dan *output layer*. Setiap *node* pada masing-masing *layer* memiliki suatu *error rate*, yang akan digunakan untuk proses *training*.

2.2.1 Neural System

1. Node

Node adalah sebuah sel neuron yang di setiap nodenya memiliki output, error, dan weight. Jadi di setiap node, dimanapun itu pasti memiliki ketiga unsur tersebut. Hubungan antar node diasosiasikan dengan suatu nilai yang disebut dengan bobot atau weight. Setiap node pasti memiliki *output*, *error* dan *weight*nya masing - masing. Output merupakan keluaran dari suatu node. *Error* merupakan tingkat kesalahan yang terdapat dalam suatu node dari proses yang dilakukan. Weight merupakan bobot dari node tersebut ke node yang lain pada layer yang berbeda. Nilai weight berkisar antara -1 dan 1. Bobot-bobot atau weight yang tersimpan di dalam jaringan syaraf tiruan ini disebut sebagai bobot interkoneksi. Nilai bobot yang baik akan memberikan keluaran yang sesuai, dalam arti mendekati keluaran yang diharapkan (target output) untuk suatu input yang diberikan. Bobot awal dalam suatu jaringan syaraf tiruan biasanya diperoleh secara random dan sebaiknya di inialisasi dengan nilai yang relatif kecil, yaitu berkisar antara -0,1 sampai 0,1 [23]. Memasuki tahap pelatihan, bobot tersebut akan mengalami penyesuaian melalui suatu proses perhitungan matematik agar tercapai nilai bobot yang sesuai.

2. Input, Hidden dan Output Layer

Input layer merupakan *layer* tempat sebuah input dimasukkan (*inialisasi input*), dan dari layer ini dilakukan proses-proses selanjutnya.

hidden layer berfungsi untuk membantu proses. Semakin banyak hidden layer yang digunakan, maka semakin bagus dan semakin cepat pula output yang diinginkan didapat. Akan tetapi waktu training akan berlangsung semakin lama.

Output layer adalah layer yang menampung hasil proses dari suatu neural network. *Forward propagation* dilakukan untuk mencari *error* di *output layer*. *Forward propagation* bertujuan untuk menentukan *output* dari suatu node. Output yang dimaksud di sini adalah output dari *output layer*. Karena masing-masing node tersebut memiliki output.

3. Training

Proses belajar suatu *neural network* terdiri dari proses *Forward*, *Backward*, dan *Update Weight*. Sekali melewati 3 tahap itu disebut dengan 1 kali *training* (1 cycle). Semakin banyak *training* yang dilakukan maka akan semakin kecil pula tingkat *error* yang dihasilkan di *output layer*-nya. Dengan demikian semakin kecil juga *error* suatu sistem. Ada dua metode *learning* dalam *neural network* [24], yaitu :

a. *Supervised Learning*

Supervised Learning adalah suatu metode dimana *neural network* belajar dari pasangan data *input* dan *target*, pasangan ini disebut *training pair*. Biasanya jaringan dilatih dengan sejumlah *training pair*, dimana suatu input vektor diaplikasikan, menghasilkan nilai di output, lalu hasil pada output tersebut akan dibandingkan dengan target output. Selisihnya akan dikembalikan ke jaringan, kemudian dihitung *error*-nya, melalui *error* ini akan didapatkan selisih yang terdapat di dalam *weight*. Oleh karena itu terdapat *weight* baru yang cenderung memiliki *error* yang lebih kecil, sehingga akan didapat *error* yang lebih minimum dari *error* yang pertama. Vektor-vektor dalam *training set* diaplikasikan seluruhnya secara berurutan. Pertama-tama *error* dihitung, kemudian *weight* disesuaikan sampai seluruh *training set* menghasilkan *error* yang sekecil-kecilnya. Pada dasarnya konsep ini berawal dari konsep *human brain*.

Model *Neural Network* yang menggunakan metode *supervised learning* diantaranya adalah sebagai berikut :

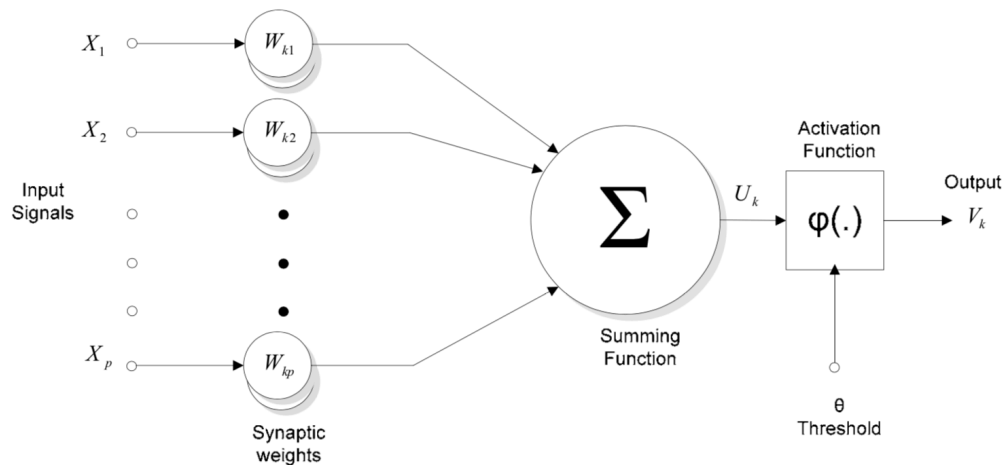
- Model *Back Propagation*
- Model *Bidirectional Associative Memory*
- *Hopfield Network*

b. *Non-Supervised (Unsupervised) Learning*

Unsupervised Learning dianggap sebagai model dalam konsep sistem biologis [25]. Dalam *unsupervised learning* tidak diperlukan target output. *Training* hanya terdiri dari vector-vektor input, tanpa memiliki pasangan target. Algoritma *training* merubah *weight* jaringan untuk menghasilkan output yang konsisten. Aplikasi dari vector-vektor yang cukup serupa akan menghasilkan pola output yang sama. Dengan demikian proses *training* akan menghasilkan sifat-sifat statistik dalam bentuk pengelompokan vector-vektor dalam beberapa kelas. Dengan mengaplikasikan suatu vektor dari suatu kelas sebagai input, maka akan menghasilkan vektor *output* yang spesifik.

2.2.2 Model Neuron

Neuron adalah unit pemrosesan informasi yang merupakan dasar dari operasi jaringan syaraf tiruan. Sel-sel syaraf tiruan ini dirancang berdasarkan sifat-sifat dari neuron biologis. Sel syaraf tiruan ini biasa disebut sebagai *processing elements*, unit atau *node*. Satu sel syaraf terdiri dari tiga bagian, yaitu: fungsi penjumlahan (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*).



Gambar 2.1 Model Neuron ANN

Jika kita lihat, neuron buatan diatas mirip dengan sel neuron biologis. Informasi (*input*) akan dikirim ke neuron dengan bobot tertentu. Input ini akan diproses oleh suatu fungsi yang akan menjumlahkan nilai-nilai bobot yang ada. Hasil penjumlahan kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, jika tidak, maka neuron tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya.

Sehingga dapat disimpulkan bahwa neuron terdiri dari 3 elemen pembentuk, yaitu:

1. Himpunan unit-unit yang dihubungkan dengan jalur koneksi. Jalur-jalur tersebut memiliki bobot yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan yang bernilai negatif akan memperlemah sinyal yang dibawa. Jumlah, struktur, dan pola hubungan antar unit-unit tersebut akan menentukan arsitektur jaringan.
2. Suatu unit penjumlah yang akan menjumlahkan input-input sinyal yang sudah dikalikan dengan bobotnya.
3. Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan ke neuron lain atau tidak.

2.2.3 Kelebihan dan Kekurangan ANN

1. Kelebihan Artificial Neural Network

- a. Mampu mengakuisisi pengetahuan walau tidak ada kepastian
- b. Mampu melakukan generalisasi dan ekstraksi dari suatu pola data tertentu
- c. ANN dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar (*self organizing*)
- d. Memiliki fault tolerance, gangguan dapat dianggap sebagai noise saja
- e. Kemampuan perhitungan secara paralel sehingga proses lebih singkat
- f. ANN mampu :
 - *Klasifikasi*: memilih suatu input data ke dalam kategori tertentu yang sudah ditetapkan
 - *Asosiasi*: menggambarkan suatu obyek secara keseluruhan hanya dengan bagian dari obyek lain
 - *Self organizing*: kemampuan mengolah data-data input tanpa harus mempunyai target
 - *Optimasi*: menemukan suatu jawaban terbaik sehingga mampu meminimalisasi fungsi biaya.

2. Kekurangan Artificial Neural Network

- a. Black Box
- b. Kurang mampu untuk melakukan operasi operasi numerik dengan presisi tinggi
- c. Kurang mampu melakukan operasi algoritma aritmatik, operasi logika dan simbolis
- d. Lamanya proses training yang mungkin terjadi dalam waktu yang sangat lama untuk jumlah data yang besar[26].

2.2.4 Kegunaan Artificial Neural Network

1. Pengenalan pola (pattern recognition)
 - Huruf, tanda tangan, suara, gambar yang sudah sedikit berubah (mengandung noise)
 - Identifikasi pola saham

- Pendeteksian uang palsu, kanker
2. Signal Processing
 - Menekan noise pada saluran telepon
 3. Peramalan
 - Peramalan saham
 4. Autopilot dan simulasi
 5. Kendali otomatis otomotif

2.2.5 Konsep Dasar ANN

Setiap pola-pola informasi input dan output yang diberikan kedalam ANN diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut neuron layers. Lapisan-lapisan penyusun ANN tersebut dapat dibagi menjadi 3, yaitu :

1. Lapisan input Unit-unit di dalam lapisan input disebut unit-unit input. Unit-unit input tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan.
2. Lapisan tersembunyi Unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Dimana outputnya tidak dapat secara langsung diamati.
3. Lapisan output Unit-unit di dalam lapisan output disebut unit-unit output. Output dari lapisan ini merupakan solusi ANN terhadap suatu permasalahan.

2.2.6 Arsitektur Jaringan

Arsitektur jaringan adalah jaringan yang menggambarkan pola keterhubungan antara neuron, baik didalam lapisan yang sama maupun antara lapisan yang berbeda. Arsitektur ANN tersebut, antara lain:

1. Jaringan layar tunggal (*single layer network*)

Jaringan dengan lapisan tunggal terdiri dari 1 layer input dan 1 layer output. Setiap neuron/unit yang terdapat di dalam lapisan/layer input selalu terhubung dengan setiap neuron yang terdapat pada layer output. Jaringan ini hanya menerima input kemudian secara langsung akan

mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi. Contoh algoritma ANN yang menggunakan metode ini yaitu : ADALINE, Hopfield, Perceptron.

2. Jaringan layar jamak (*multi layer network*)

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis layer yakni layer input, layer output, dan juga layer tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma ANN yang menggunakan metode ini yaitu : *Adaline, Backpropagation, Neocognitron*.

3. Jaringan dengan lapisan kompetitif (*competitive layer network*) Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan metode ini adalah LVQ (*Learning Vector Quantization*).

2.2.7 Algoritma Umum ANN

Algoritma pembelajaran/pelatihan ANN:

Dimasukkan n contoh pelatihan kedalam ANN, lakukan :

1. Inisialisasi bobot-bobot jaringan. Set $i = 1$.
2. Masukkan contoh ke-i (dari sekumpulan contoh pembelajaran yang terdapat dalam set pelatihan) kedalam jaringan pada lapisan input.
3. Cari tingkat aktivasi unit-unit input menggunakan algoritma aplikasi
If e kinerja jaringan memenuhi standar yang ditentukan sebelumnya
(memenuhi syarat untuk berhenti)
Then e exit
4. Update bobot-bobot dengan menggunakan aturan pembelajaran jaringan.
5. If $i = n$ then reset $i = 1$
Else $i = i + 1$
Ke langkah 2.

Algoritma aplikasi/inferensi jaringan saraf tiruan :

Dimasukkan sebuah contoh pelatihan kedalam jaringan saraf tiruan, lakukan :

1. Masukkan kasus kedalam jaringan pada lapisan input.
2. Hitung tingkat aktivasi node-node jaringan.
3. Untuk jaringan koneksi umpan maju, jika tingkat aktivasi dari semua unit outputnya telah dikalkulasi, maka exit. Untuk jaringan dengan kondisi balik, jika tingkat aktivasi dari semua unit output menjadi konstan atau mendekati konstan, maka exit. Jika tidak, kembali ke langkah 2. Jika jaringannya tidak stabil, maka *exit* dan *fail*.

2.3 *Autoregressive Integrated Moving Average (Arima)*

2.3.1 Prinsip Dasar dan Tujuan Analisis

1. Prinsip Dasar

ARIMA sering juga disebut metode runtun waktu *Box-Jenkins*. ARIMA sangat baik ketepatannya untuk peramalan jangka pendek, sedangkan untuk peramalan jangka panjang ketepatan peramalannya kurang baik. Biasanya akan cenderung flat (mendatar/konstan) untuk periode yang cukup panjang [27].

Model Autoregresif Integrated Moving Average (ARIMA) adalah model yang secara penuh mengabaikan variable independen dalam membuat peramalan. ARIMA menggunakan nilai masa lalu dan sekarang dari variabel dependen untuk menghasilkan peramalan jangka pendek yang akurat. ARIMA cocok jika observasi dari deret waktu (*time series*) secara statistik berhubungan satu sama lain (*dependent*).

2. Tujuan Analisis

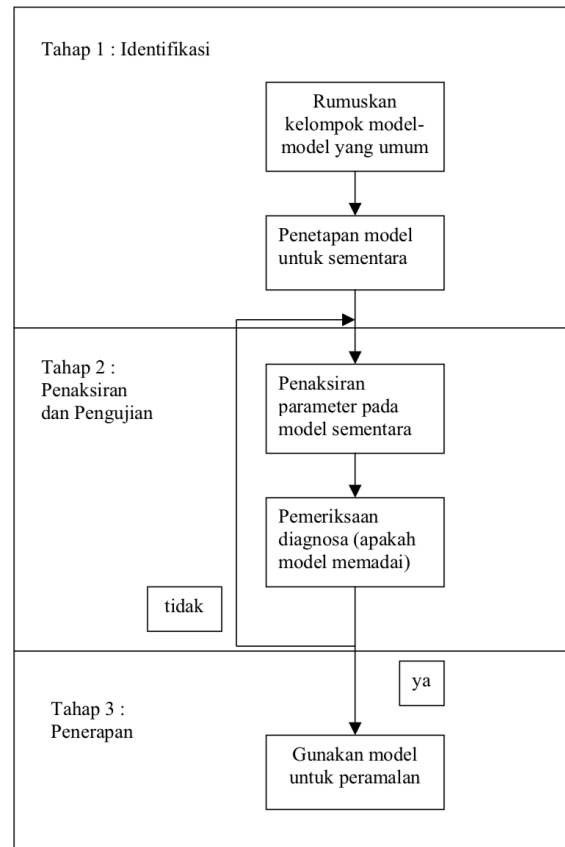
Tujuan model ini adalah untuk menentukan hubungan statistik yang baik antar variabel yang diramal dengan nilai historis variabel tersebut sehingga peramalan dapat dilakukan dengan model tersebut.

2.3.2 Model Matematis

Model ARIMA terdiri dari tiga langkah dasar, yaitu tahap identifikasi, tahap penaksiran dan pengujian, dan pemeriksaan diagnostik. Selanjutnya

model ARIMA dapat digunakan untuk melakukan peramalan jika model yang diperoleh memadai.

Skema Pendekatan Box Jenkins



Gambar 2.2 Skema Pendekatan Box Jenkins [27]

2.3.3 Kelebihan dan Kekurangan ARIMA

1. Kelebihan ARIMA

- Merupakan model tanpa teori karena variabel yang digunakan adalah nilai-nilai lampau dan kesalahan yang mengikutinya.
- Memiliki tingkat akurasi peramalan yang cukup tinggi karena setelah mengalami pengukuran kesalahan peramalan MAE (*mean absolute error*), nilainya mendekati nol.
- Cocok digunakan untuk meramal sejumlah variabel dengan cepat, sederhana, akurat dan murah karena hanya membutuhkan data variabel yang akan diramal [27].

2. Kekurangan ARIMA

- a. Untuk data peramalan dalam periode yang cukup panjang ketepatannya kurang baik karena biasanya akan cenderung flat (datar/konstan).
- b. ARIMA akan mengalami penurunan keakuratan apabila terdapat komponen nonlinier time series pada data pengamatan [27].

2.3.4 Stasioneritas dan Nonstasioneritas

Hal yang perlu diperhatikan adalah bahwa kebanyakan deret berkala bersifat nonstasioner dan bahwa aspek-aspek AR dan MA dari model ARIMA hanya berkenaan dengan deret berkala yang stasioner.

Stasioneritas berarti tidak terdapat pertumbuhan atau penurunan pada data. Data secara kasarnya harus horizontal sepanjang sumbu waktu. Dengan kata lain, fluktuasi data berada di sekitar suatu nilai rata-rata yang konstan, tidak tergantung pada waktu dan varians dari fluktuasi tersebut pada pokoknya tetap konstan setiap waktu.

Suatu deret waktu yang tidak stasioner harus diubah menjadi data stasioner dengan melakukan differencing. Yang dimaksud dengan differencing adalah menghitung perubahan atau selisih nilai observasi. Nilai selisih yang diperoleh dicek lagi apakah stasioner atau tidak. Jika belum stasioner maka dilakukan differencing lagi. Jika varians tidak stasioner, maka dilakukan transformasi logaritma.

2.3.5 Klasifikasi Model ARIMA

Model Box-Jenkins (ARIMA) dibagi kedalam 3 kelompok, yaitu: model AR (*autoregressive*), MA (*moving average*), dan model campuran ARMA (*Autoregressive Moving Average*) yang mempunyai karakteristik dari dua model pertama.

1) *Autoregressive Model (AR)*

Bentuk umum model *autoregressive* dengan ordo p (AR(p)) atau model ARIMA ($p,0,0$) dinyatakan sebagai berikut:

$$X_t = \mu' + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \dots + \varphi_p X_{t-p} + e_t [0]$$

dimana: μ' = suatu konstanta

φ_p = parameter autoregresif ke- p

e_t = nilai kesalahan pada saat t

2) *Moving Average Model (MA)*

Bentuk umum model moving average ordo q (MA(q)) atau ARIMA ($0,0,q$) dinyatakan sebagai berikut:

$$X_t = \mu' + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-k}$$

dimana: μ' = suatu konstanta

θ_1 sampai θ_q adalah parameter-parameter *moving average*

e_{t-k} = nilai kesalahan pada saat $t - k$

3) *Model campuran*

a. Proses ARMA

Model umum untuk campuran proses AR(1) murni dan MA(1) murni, misal ARIMA (1,0,1) dinyatakan sebagai berikut:

$$X_t = \mu' + \varphi_1 X_{t-1} + e_t - \theta_1 e_{t-1}$$

atau

$$(1 - \varphi_1 B) X_t = \mu' + (1 - \theta_1 B) e_t$$

AR(1) MA(1)

b. Proses ARIMA

Apabila nonstasioneritas ditambahkan pada campuran proses ARMA, maka model umum ARIMA (p,d,q) terpenuhi. Persamaan untuk kasus sederhana ARIMA $(1,1,1)$ adalah sebagai berikut:

$$(1-B)(1-\phi_1 B)X_t = \mu' + (1-\theta_1 B)e_t$$

Perbedaan pertama AR(1) MA(1)

2.3.6 Musiman dan Model ARIMA

Musiman didefinisikan sebagai suatu pola yang berulang-ulang dalam selang waktu yang tetap. Untuk data yang stasioner, faktor musiman dapat ditentukan dengan mengidentifikasi koefisien autokorelasi pada dua atau tiga time-lag yang berbeda nyata dari nol. Autokorelasi yang secara signifikan berbeda dari nol menyatakan adanya suatu pola dalam data. Untuk mengenali adanya faktor musiman, seseorang harus melihat pada autokorelasi yang tinggi.

Untuk menangani musiman, notasi umum yang singkat adalah:

$$\text{ARIMA } (p,d,q) (P,D,Q)^S$$

Dimana (p,d,q) = bagian yang tidak musiman dari model

(P,D,Q) = bagian musiman dari model

S = jumlah periode per musim

2.3.7 Identifikasi Model

Proses identifikasi dari model musiman tergantung pada alat-alat statistik berupa autokorelasi dan parsial autokorelasi, serta pengetahuan terhadap sistem atau proses yang dipelajari.

2.3.8 Penaksiran Parameter

Ada dua cara yang mendasar untuk mendapatkan parameter-parameter tersebut:

- a) Dengan cara mencoba-coba (*trial and error*), menguji beberapa nilai yang berbeda dan memilih satu nilai tersebut (atau

sekumpulan nilai, apabila terdapat lebih dari satu parameter yang akan ditaksir) yang meminimumkan jumlah kuadrat nilai sisa SSR (*Sum of Squared Residual*).

- b) Perbaiki secara iteratif, memilih taksiran awal dan kemudian membiarkan program komputer memperhalus penaksiran tersebut secara iteratif.

2.3.9 Pengujian Parameter Model

1. Pengujian masing-masing parameter model secara parsial (t-test)
2. Pengujian model secara keseluruhan (Overall F test)

Model dikatakan baik jika nilai error bersifat random, artinya sudah tidak mempunyai pola tertentu lagi. Dengan kata lain model yang diperoleh dapat menangkap dengan baik pola data yang ada. Untuk melihat kerandoman nilai error dilakukan pengujian terhadap nilai koefisien autokorelasi dari *error*, dengan menggunakan salah satu dari dua statistik berikut:

1) Uji *Q* Box dan Pierce:

$$Q = n' \sum_{k=1}^m r_k^2$$

2) Uji Ljung-Box:

$$Q = n'(n' + 2) \sum_{k=1}^m \frac{r_k^2}{(n' - k)}$$

Menyebar secara Khi Kuadrat (χ^2) dengan derajat bebas (db) = (k-p-q-P-Q) dimana:

$$n' = n - (d + SD)$$

d = ordo pembedaan bukan faktor musiman

D = ordo pembedaan faktor musiman

S = jumlah periode per musim

m = lag waktu maksimum

r_k = autokorelasi untuk time lag 1, 2, 3, 4, ..., k

Kriteria pengujian:

Jika $Q \leq \chi^2_{(\alpha, db)}$ berarti: nilai error bersifat random (model dapat diterima).

Jika $Q > \chi^2_{(\alpha, db)}$ berarti: nilai error tidak bersifat random (model tidak dapat diterima).

2.3.10 Pemilihan Model Terbaik

Untuk menentukan model yang terbaik dapat digunakan MAD (*Mean Absolute Deviation*) berikut :

$$MAD = \frac{\sum_{t=1}^n |Y_t - F_t|}{n}$$

Dimana :

Y_t = Data aktual pada periode t

F_t = Nilai peramalan pada periode t

n = Jumlah data

MAD merupakan ukuran pertama kesalahan peramalan keseluruhan untuk sebuah model.

MSE (*Mean Square Error*) merupakan cara kedua untuk mengukur kesalahan peramalan keseluruhan. MSE merupakan rata-rata selisih kuadrat antara nilai yang diramalkan dan yang diamati.

$$MSE = \frac{\sum_{t=1}^n |Y_t - F_t|^2}{n}$$

Dimana :

Y_t = Data aktual pada periode t

F_t = Nilai peramalan pada periode t

n = Jumlah data

RMSE (*Root Mean Square Error*) merupakan nilai rata-rata dari jumlah kuadrat kesalahan juga dapat menyatakan ukuran besarnya kesalahan yang dihasilkan oleh suatu model prakiraan. Nilai RMSE rendah menunjukkan bahwa variasi nilai yang dihasilkan oleh suatu model prakiraan mendekati variasi nilai obeservasinya.

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (Y_t - F_t)^2}{n}}$$

Dimana :

Y_t = Data aktual pada periode t

F_t = Nilai peramalan pada periode t

n = Jumlah data

Selain nilai RMSE (*Root Mean Square Error*), nilai rata-rata persentase kesalahan peramalan MAPE (*Mean Absolute Percentage Error*) dapat juga digunakan sebagai bahan pertimbangan dalam menentukan model yang terbaik yaitu:

$$MAPE = \frac{\sum_{t=1}^T \frac{|Y_t - F_t|}{Y_t}}{t} \times 100\%$$

Dimana :

t = banyaknya periode peramalan/dugaan.

2.3.11 Peramalan Dengan Model ARIMA

Notasi yang digunakan dalam ARIMA adalah notasi yang mudah dan umum. Misalkan model ARIMA (0,1,1)(0,1,1)¹² dijabarkan sebagai berikut:

$$(1 - B)(1 - B^{12}) X_t = (1 - \theta_1 B)(1 - \theta_{12} B^{12}) e_t$$

Tetapi untuk menggunakannya dalam peramalan mengharuskan dilakukan suatu penjabaran dari persamaan tersebut dan menjadikannya sebuah persamaan regresi yang lebih umum. untuk model diatas bentuknya adalah:

$$X_t = X_{t-1} + X_{t-12} - X_{t-13} + e_t - \theta_1 e_{t-1} - \theta_{12} e_{t-12} + \theta_{13} e_{t-13}$$

Untuk meramalkan satu periode ke depan, yaitu X_{t+1} maka seperti pada persamaan berikut:

$$X_{t+1} = X_t + X_{t-11} - X_{t-12} + e_{t+1} - \theta_1 e_t - \theta_{11} e_{t-11} + \theta_1 \theta_{12} e_{t-12}$$

Nilai e_{t+1} tidak akan diketahui, karena nilai yang diharapkan untuk kesalahan random pada masa yang akan datang harus ditetapkan sama dengan nol. Akan tetapi dari model yang disesuaikan (*fitted model*) kita boleh mengganti nilai e_t , e_{t-11} dan e_{t-12} dengan nilai nilai mereka yang ditetapkan secara empiris (seperti yang diperoleh setelah iterasi terakhir algoritma Marquardt). Tentu saja bila kita meramalkan jauh ke depan, tidak akan kita peroleh nilai empiris untuk “ e ” sesudah beberapa waktu, dan oleh sebab itu nilai harapan mereka akan seluruhnya nol.

Untuk nilai X , pada awal proses peramalan, kita akan mengetahui nilai X_t , X_{t-11} , X_{t-12} . Akan tetapi sesudah beberapa saat, nilai X akan berupa nilai ramalan (forecasted value), bukan nilai-nilai masa lalu yang telah diketahui.

2.4 *Vector Autoregression (VAR)*

Metode *Vector Autoregression* atau VAR adalah pendekatan non-struktural (lawan dari pendekatan struktural, seperti pada persamaan simultan) yang menggambarkan hubungan yang “saling menyebabkan” (*kausalistis*) antar variabel dalam sistem. Metode ini mulai dikembangkan oleh Sims pada tahun 1980 yang mengasumsikan bahwa semua variabel dalam model bersifat endogen (ditentukan di dalam model) sehingga metode ini disebut sebagai model yang a-teoritis (tidak berlandaskan teori).

Hal ini dilakukan karena sering dijumpai keadaan dimana teori ekonomi saja ternyata tidak dapat menangkap (tidak cukup kaya menyediakan spesifikasi) secara tepat dan lengkap hubungan dinamis antar variabel. Apabila data tidak stasioner pada level - nya, maka data harus ditransformasi (*first difference*) untuk mendapatkan data yang stasioner. Hubungan jangka panjang hilang dalam transformasi. Untuk tetap mendapatkan hubungan jangka panjang, model VAR

akan dimodifikasi menjadi model koreksi kesalahan VECM (*Vector Error Correction Model*), jikaterdapat kointegrasi dalam model.

2.4.1 Kelebihan dan Kekurangan VAR

Beberapa keunggulan metode VAR dibandingkan dengan metode ekonometrika lainnya, antara lain:

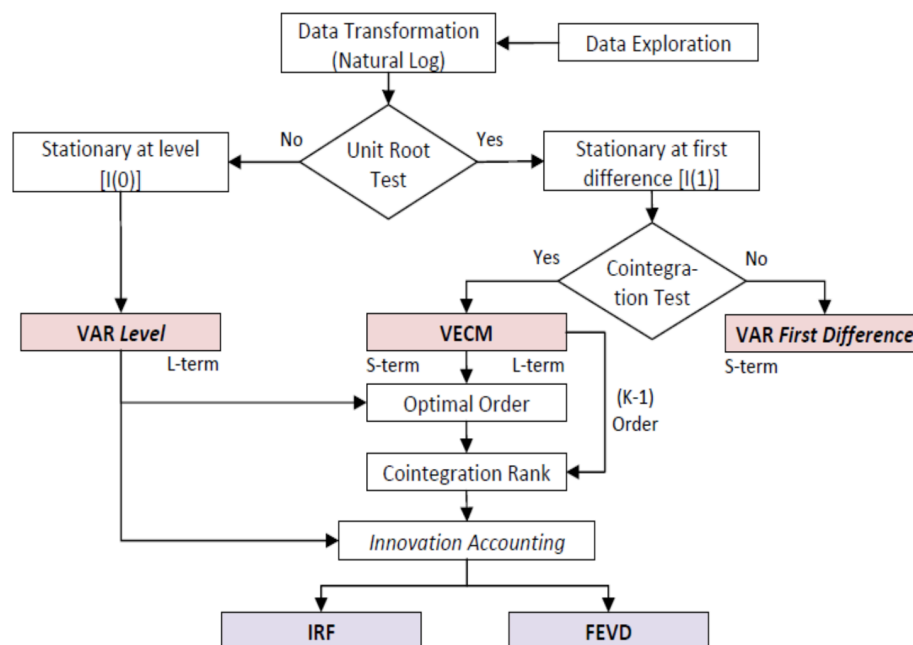
1. Metode VAR terbebas dari berbagai batasan teori ekonomi yang sering muncul, seperti gejala *spurious variable* endogeneity and exogeneity, karena bekerja berdasarkan data.
2. VAR membangun model secara bersamaan di dalam suatu sistem yang kompleks (*multivariate*), sehingga dapat menangkap hubungan keseluruhan variabel di dalam persamaan itu.
3. Uji VAR yang multivariat dapat menghindari parameter yang bias akibat tidak dimasukkannya variabel yang relevan.
4. Uji VAR dapat mendeteksi hubungan antar variabel di dalam suatu sistem persamaan, dengan menjadikan seluruh variabel sebagai endogenous.
5. Metode VAR sederhana, ketika seseorang tidak perlu khawatir untuk menentukan variabel mana yang endogen dan variabel mana yang eksogen.
6. Metode VAR sederhana, karena metode OLS (*Ordinary Least Square*) biasa dapat diterapkan pada masing-masing persamaan secara terpisah;
7. Hasil estimasi prediksi (*forecast*) yang diperoleh melalui metode VAR dalam banyak kasus lebih baik dari pada hasil estimasi dari model-model persamaan simultan yang lebih kompleks.

Sementara itu, beberapa kelemahan metode VAR, antara lain:

3. Model VAR dianggap a-teoritis, karena menggunakan lebih sedikit informasi dari teori-teori terdahulu, tidak seperti model persamaan

- simultan, dimana pemasukan dan pengeluaran variabel tertentu memainkan peran penting dalam identifikasi model;
4. Model VAR kurang sesuai untuk analisis kebijakan, disebabkan terlalu menekankan pada prediksi (*forecast*);
 5. Pemilihan panjang lag menjadi tantangan terbesar, khususnya ketika variabel terlalu banyak dengan lag panjang, sehingga ada terlalu panjang parameter yang akan mengurangi degree of freedom dan memerlukan ukuran sampel yang besar;
 6. Semua variabel harus stasioner. Jika tidak, data harus ditransformasi dengan benar (misalnya, diambil first difference-nya). Hubungan jangka panjang yang diperlukan dalam analisis akan hilang dalam transformasi;
 7. IRF (*Impulse Response Function*), yang merupakan inti dari analisis menggunakan metode VAR, masih diperdebatkan oleh para peneliti.

2.4.2 Proses Analisis VAR



Gambar 2.3 Proses Analisis VAR

Berdasarkan alur bagan di atas maka analisis VAR mensyaratkan beberapa pengujian antara lain : Uji Stasioneritas (*Unit Root Test*), Uji Stabilitas Model

VAR, Uji Optimum Lag, Uji Kointegrasi, Model VECM (Jangka Panjang), Analisis IRF (*Impulse Response Function*), dan Analisis FEVD (*Forecast Error Variance Decomposition*).

1. Uji Stasioneritas

Pengujian stasioneritas dapat dilakukan untuk melihat perilaku data. Uji stasioneritas dapat dilakukan dengan menggunakan metode ADF (*Argumented Dickey Fuller*) sesuai dengan bentuk tren determinasi yang dikandung oleh setiap variabel. Hasil stasioner akan berujung pada penggunaan VAR dengan model sederhana. Sedangkan variabel non stasioner meningkatkan kemungkinan keberadaan hubungan kointegrasi antar variabel.

2. Uji Stabilitas Model VAR

Stabilitas model VAR dapat dilihat pada nilai modulus yang dimiliki oleh setiap variabel. Model VAR dikatakan stabil apabila nilai modulus berada pada radius < 1 , dan tidak stabil jika nilai modulus > 1 . Jika nilai Modulus yang paling besar kurang dari satu dan berada pada titik optimal, maka komposisi tadi sudah berada pada posisi optimal dan model VAR sudah stabil.

3. Uji Optimum Lag

Penentuan optimum lag berguna untuk menghilangkan masalah dalam autokorelasi dalam sebuah sistem VAR. Untuk menetapkan besarnya lag yang optimal dapat dilakukan dengan menggunakan beberapa kriteria antara lain : AIC (*Akaike Information Criteria*), SIC (*Schwarz Information Criterion*), HQ (*Hanna Quinn Information Criterion*). Namun, dalam memberikan kestabilan dan konsisten nilai panjang lag optimum pada umumnya menggunakan SIC.

4. Uji Kointegrasi

Uji kointegrasi dilakukan untuk mengetahui apakah akan terjadi keseimbangan dalam jangka panjang, yaitu terdapat kesamaan pergerakan dan stabilitas hubungan diantara variabel-variabel di dalam penelitian ini

atau tidak. Dalam penelitian ini uji kointegrasi dilakukan dengan menggunakan metode Johansen's Cointegration Test.

5. Estimasi Model VAR

Estimasi model VAR mensyaratkan data dalam kondisi stasioner. Estimasi model VAR dimulai dengan menentukan berapa panjang lag optimal (tahap VAR ke-3).

6. Uji Kausalitas

Uji kausalitas dilakukan untuk mengetahui apakah suatu variabel endogen diperlakukan sebagai variabel eksogen. Uji kausalitas dapat dilakukan dengan berbagai metode diantaranya metode *Granger's Causality* dan ECM (*Error Correction Model*) Causality.

7. Analisis IRF (*Impulse Response Functions*)

IRF dalam VAR digunakan untuk melihat dampak dari perubahan dari satu variabel terhadap perubahan variabel lainnya secara dinamis. IRF merupakan aplikasi vector moving average yang bertujuan untuk melihat jejak respon saat ini dan kedepan suatu variabel terhadap guncangan dari variabel tertentu. Bentuk dari analisis IRF pada umumnya direpresentasikan dalam bentuk grafik. Analisis FEVD (*Forecast Error Variance Decomposition*) Metode yang digunakan untuk melihat bagaimana perubahan dalam suatu variabel yang ditunjukkan oleh perubahan error variance dipengaruhi oleh variabel-variabel lainnya. Analisis ini digunakan untuk menghitung seberapa besar pengaruh acak guncangan dari variabel tertentu terhadap variabel endogen. Dengan metode ini dapat terlihat kekuatan dan kelebihan masing-masing variabel dalam mempengaruhi variabel yang lainnya dalam kurun waktu yang panjang.

2.5 Software R

R adalah suatu kesatuan software yang terintegrasi dengan beberapa fasilitas untuk manipulasi, perhitungan dan penampilan grafik yang handal. R berbasis pada bahasa pemrograman S, yang dikembangkan oleh AT&T Bell Laboratories sekarang Lucent Technologies pada akhir tahun '70 an. R merupakan versi gratis dari bahasa S dari software (berbayar) yang sejenis yakni S-PLUS

yang banyak digunakan para peneliti dan akademisi dalam melakukan kegiatan ilmiahnya.

Pada awalnya, versi pertama R dibuat oleh Ross Ihaka and Robert Gentleman dari Universitas Auckland, namun selanjutnya R dikembangkan oleh tim yang disebut tim inti. Tim inti (*core team*) terdiri dari ahli statistik, ahli komputer & pemrograman, geografi, ekonomi dari institusi yang berbeda dari seluruh dunia yang mencoba membangun sebuah sistem (*software*) yang handal namun dengan biaya yang sangat murah. R dapat secara gratis didownload dan digunakan dengan berlisensi pada GNU (*General Public License*).

Menurut kutipan dari penghargaan *Association for Computing Machinery Software* bagi John Chamber 1998, menyatakan bahwa bahasa pemrograman dapat “memanipulasi, visualisasi dan menganalisis data”. R dibuat searah dengan ide yang ada pada bahasa pemrograman S.

2.5.1 R dan Program Statistik Lainnya

Seperti dijelaskan sebelumnya, R merupakan “kerabat” dekat dari S-PLUS dimana secara fungsi dan sintaks/tata bahasa sama-sama menggunakan bahasa S, namun tidak identik. R dapat berinteraksi dengan program statisik, manipulasi, perhitungan dan penampilan grafik lainnnya, seperti SPSS, yang cukup populer, Microsoft Excel dengan menyediakan fasilitas import dan ekspor data.

Selain software di atas, R dapat melakukan import file dari software lainnya seperti, Minitab, SAS, Stat, Systat dan EpInfo. R adalah bahasa fungsional1, dimana terdapat inti bahasa yang menggunakan bentuk standar notasi aljabar, yang memungkinkan perhitungan numerik seperti $2+3$, atau 3^{11} . Selain itu tersedia pula fasilitas perhitungan dengan menggunakan fungsi.

Dengan beberapa fitur tersebut, R menjadi alat yang tangguh bagi para statistikawan, ahli ekonomi, peneliti dalam membantu risetnya, dikarenakan R dibangun dan didukung dengan model dan teori statistik terdepan dan menggunakan standar tertinggi bagi analisis data. R hampir dapat digunakan

untuk berbagai bidang, mulai dari kalkulasi biasa seperti kalkulator, statistik, ekonometri, geografi, hingga pemrograman komputer.

2.5.2 Kelebihan dan Fitur-fitur R

R mempunyai karakteristik tersendiri, dimana selalu dimulai dengan prompt “ > “ pada console-nya. R mempunyai beberapa kelebihan dan fitur-fitur yang canggih dan berguna, diantaranya:

- a. Efektif dalam pengelolaan data dan fasilitas penyimpanan. Ukuran file yang disimpan jauh lebih kecil dibanding software lainnya.
- b. Lengkap dalam operator perhitungan array
- c. Lengkap dan terdiri dari koleksi tools statistik yang terintegrasi untuk analisis data, diantaranya, mulai statistik deskriptif, fungsi probabilitas, berbagai macam uji statistik, hingga *time series*.
- d. Tampilan grafik yang menarik dan fleksibel ataupun *costumized*.

Dapat dikembangkan sesuai keperluan dan kebutuhan dan sifatnya yang terbuka, setiap orang dapat menambahkan fitur-fitur tambahan dalam bentuk paket ke dalam software R.

Selain kelebihan dan kelengkapan fitur-fiturnya, hal yang terpenting lainnya yakni, R bersifat multiplatform, yakni dapat diinstall dan digunakan baik pada system operasi Windows , UNIX/LINUX maupun pada Macintosh. Untuk dua system operasi disebutkan terakhir diperlukan sedikit penyesuaian. Selain kelebihan disebutkan di atas, R didukung oleh komunitas yang secara aktif saling berinteraksi satu sama lain melalui Internet dan didukung oleh manual atau R-help yang menyatu pada software R. Sebagai catatan, buku ini mengambil contoh pada penggunaan R pada system berbasis Windows.

2.5.3 R, Riset dan Akademis

Software R sangat cocok untuk riset, baik statistik, ekonomi, komputasi numerik dan pemrograman komputer. Karena didukung oleh banyak tenaga ahli dibidangnya, R layak dijadikan suatu perangkat lunak acuan bagi berbagai

kalangan, terlebih di kalangan akademik (dosen, mahasiswa). Selain itu R memiliki fitur yang lengkap dan handal serta faktor tanggung jawab moral dan legal/hukum bukan lagi menjadi kekhawatiran dalam penggunaannya, karena dapat diperoleh secara gratis. Berikut adalah beberapa contoh yang didapat dari R sebagai acuan implementasi pada:

1. Pemodelan matematis seperti software MATLAB dalam membentuk perspektif, cocok jurusan teknik arsitek, sipil, mesin, dan ilmu komputer (pencitraan).
2. Pencitraan dan analisis kontur, cocok untuk jurusan geografi dan sejenis.
3. Proses analisis data statistik, dengan tampilan grafik plot yang costumized dan grafik fungsi densitas yang dapat diparalelkan dnegan histogram. Cocok untuk bidang statistika, ekonomi, dan lain lain.