

**ANALISIS RUNTUN WAKTU DENGAN METODE
QUANTUM ARIMA NEURAL NETWORK
(STUDI KASUS: DATA PERTAMBAHAN JUMLAH
KASUS HARIAN COVID 19 DI JAKARTA)**

BAB I PENDAHULUAN

1.1 Latar Belakang

Penelitian terkait analisis runtun waktu (*time series*) dan peramalan (*forecasting*) adalah bidang penelitian yang aktif, artinya sampai saat ini masih terus dilakukan penelitian mengenai keakuratan dalam proses peramalan runtun waktu terkait dengan proses pengambilan keputusan. Beberapa penelitian yang melakukan riset pada runtun waktu menggunakan metode statistik, jaringan syaraf (*neural network*), *wavelet*, maupun *system fuzzy*. [1, 2]

Metode peramalan runtun waktu lebih menitikberatkan proses pada upaya mencari pola data, dan bukan untuk menjelaskan mengapa ada pola seperti itu. Metode ini akan lebih cocok jika kebutuhan hanyalah ingin mengetahui berapa angka yang akan diprediksi untuk periode mendatang dan tidak perlu campur tangan dari luar untuk mempengaruhi hasil prediksi, karena sandaran satu-satunya adalah data historis yang memang sudah terjadi dan tidak dapat ditarik kembali. Metode peramalan data runtun waktu secara umum terbagi menjadi dua bagian. Pertama, metode peramalan dengan cara matematika statistik seperti *moving average*, *exponential smoothing*, analisis *trend* serta yang paling sering digunakan adalah ARIMA (Box Jenkins). Kedua, metode peramalan yang didasarkan pada kecerdasan buatan seperti *neural network*, algoritma genetika, *simulated annealing*, *genetic programming*, klasifikasi dan model *hybrid*. Tidak semua metode dapat digunakan secara efektif pada semua data. Metode-metode tersebut memiliki kekurangan dan kelebihan yang berbeda. Terlebih lagi, masalah dalam dunia nyata seringkali merupakan masalah yang kompleks dan satu model mungkin tidak mampu mengatasi hal tersebut dengan baik [2].

Pemilihan metode-metode tersebut tergantung pada berbagai aspek yang mempengaruhi yaitu aspek waktu, pola data, tipe model system yang diamati, tingkat keakuratan ramalan yang diinginkan dan sebagainya. Karena itulah muncul suatu masalah jika pengamatan atau pengujian dilakukan pada suatu system dinamis yang memiliki system pola data dengan formulasi yang selalu berubah-ubah atau dengan kata lain system yang memiliki tingkat kesulitan yang tinggi untuk dibuatkan formulasi modelnya pada suatu kurun waktu tertentu. Di samping itu, untuk menerapkan metode statistik tersebut data harus memenuhi beberapa asumsi-asumsi tertentu sesuai dengan pola data

Dengan teknologi di bidang kecerdasan buatan (*Artificial Intelligence*) yaitu teknologi jaringan syaraf tiruan (*Neural Network*) maka identifikasi pola data dari system dapat dilakukan dengan metode pendekatan pembelajaran atau pelatihan yaitu untuk menentukan bobot penghubung antar simpul yang optimum. Keunggulan utama jaringan syaraf tiruan adalah kemampuan komputasi yang paralel dengan cara belajar dari pola-pola yang diajarkan. Berdasarkan kemampuan belajar yang dimilikinya, maka jaringan syaraf tiruan dapat dilatih untuk mempelajari dan menganalisa pola data masa lalu dan berusaha mencari suatu formula atau fungsi yang akan menghubungkan pola data masa lalu dengan keluaran yang diinginkan pada saat ini maupun pada masa mendatang. Dalam sebuah studi komparatif (studi perbandingan) telah dilakukan penelitian terhadap model peramalan runtun waktu yang memiliki trend dan pola musiman untuk mengetahui apakah model yang lebih kompleks selalu menghasilkan forecast (ramalan) yang lebih baik daripada model statistika. Dalam studi perbandingan tersebut, metode yang dibandingkan adalah model *Winter's Decomposition*, *Time Series Regression*, *ARIMA* dan *Neural Network*. Hasilnya diperoleh kesimpulan bahwa model yang kompleks tidak selalu menghasilkan forecast (ramalan) yang lebih baik dibandingkan model statistika sederhana. Data yang digunakan dalam penelitian tersebut adalah data penumpang pesawat internasional dari bulan Januari 1949 sampai bulan Desember 1960. [3]

Untuk metode peramalan yang kompleks dengan data yang sangat banyak, dibutuhkan suatu metode yang tidak hanya berbasis kecerdasan buatan lagi. Saat *machine learning* sudah dikembangkan untuk mengenali dan mempelajari pola suatu data sehingga dapat memberikan hasil prediksi yang benar berdasarkan pola data tersebut. Karena peningkatan jumlah data yang akan diprediksi, maka dalam beberapa tahun terakhir, para peneliti telah menyelidiki apakah komputer kuantum dapat digunakan untuk meningkatkan kinerja algoritma *machine learning*. Dengan adanya efek mekanika kuantum seperti superposisi dan keterlibatan (entanglement) maka komputer kuantum diharapkan dapat memberikan keuntungan untuk memecahkan masalah tertentu lebih cepat daripada komputer klasik. Meskipun komputer kuantum masih dalam tahap percobaan, namun algoritma kuantum telah dikembangkan selama dua dekade terakhir. Algoritma kuantum melibatkan masalah seperti faktorisasi bilangan-bilangan besar dan optimalisasi dengan yang terakhir efektif digunakan dalam percepatan algoritma *machine learning*. Salah satu algoritma *machine learning* yang sudah dikembangkan saat ini untuk metode

Berdasarkan latar belakang tersebut maka dalam penelitian ini penulis tertarik untuk membuat suatu metode peramalan berbasis komputasi kuantum dari metode peramalan *Hybrid Arima Neural Network (Quantum ARIMA Neural Network)* dengan studi kasus data pertambahan jumlah kasus harian covid 19 di Jakarta. Studi kasus ini dilakukan untuk mengendalikan penyebaran Covid 19 melalui analisis prediksi penambahan kasus harian baik jumlah yang positif, meninggal dan sembuh dari Covid 19 di periode yang akan datang. Beberapa penelitian telah dilakukan untuk membuat model prediksi Covid 19 ini, di antaranya menggunakan model pertumbuhan logistik dan model dinamik [6] model ARIMA [7, 8], model hibrid ARIMA dan wavelet [9], model ARIMA *machine learning* [10] dan model-model lainnya [11-15]. Dalam penelitian ini akan dilakukan analisis runtun waktu pertambahan jumlah kasus harian Covid 19 di Jakarta dengan metode Hybrid Quantum ARIMA NN berdasarkan dataset dari situs <https://corona.jakarta.go.id> tanggal 6 Maret 2020 sampai 30 Juni 2021 sebagai data training dan nanti akan diprediksi untuk tanggal 1 Juli 2021 sampai dengan 31 Juli 2021 sebagai data uji.

Tujuan umum dari penelitian ini adalah untuk mengembangkan model prediksi Quantum ARIMA NN yang diharapkan dapat menghasilkan tingkat akurasi yang tinggi dengan komputasi yang tidak terlalu lama dalam pengolahan data yang cukup besar. Tujuan khususnya adalah menganalisis data pertambahan jumlah kasus harian Covid 19 di Jakarta menggunakan metode ARIMA, Neural Network, Hybrid Arima NN dan Quantum ARIMA NN dan membandingkan akurasinya

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah tersebut maka yang menjadi rumusan masalah pada penelitian ini adalah :

1. Bagaimana pola data runtun waktu pertambahan jumlah kasus harian Covid 19 di Jakarta dari tanggal 6 Maret 2020 sampai tanggal 30 Juni 2021 ?
2. Bagaimana membuat model peramalan yang sesuai dan cocok untuk data runtun waktu tersebut dengan metode ARIMA ?
3. Bagaimana membuat model peramalan data runtun waktu tersebut dengan metode Neural Network (NN) ?

4. Bagaimana cara membuat model Hybrid ARIMA Neural Network (ARIMA-NN) yang sesuai dan cocok dengan data runtun waktu tersebut?
5. Bagaimana mengembangkan model Quantum Arima Neural Network untuk data runtun waktu?
6. Bagaimana tingkat akurasi dan hasil peramalan untuk jumlah kasus harian Covid 19 di Jakarta pada periode tanggal 1 Juli sampai 31 Juli 2020 berdasarkan model-model tersebut?

1.3 Tujuan dan Batasan Penelitian

1.3.1 Tujuan Penelitian :

Berdasarkan rumusan masalah di atas, maka tujuan penelitian ini adalah sebagai berikut :

1. Mempelajari pola data runtun waktu pertambahan jumlah kasus harian Covid 19 di Jakarta dari tanggal 6 Maret 2020 sampai tanggal 30 Juni 2021 apakah merupakan runtun waktu *seasonal* atau *non seasonal*
2. Menentukan model peramalan ARIMA yang sesuai dengan data runtun waktu tersebut
3. Menentukan model peramalan untuk data runtun waktu tersebut menggunakan metode *Neural Network (NN)*
4. Menentukan model peramalan *Hybrid Arima Neural Network* untuk data runtun waktu tersebut
5. Membuat dan mengembangkan model Hybrid Quantum ARIMA-NN yang dapat memberikan hasil peramalan yang lebih akurat
6. Membuat analisis hasil peramalan untuk jumlah kasus harian Covid 19 di Jakarta pada periode tanggal 1 Juli sampai 31 Juli 2020 berdasarkan model-model tersebut

Berdasarkan perbandingan telaah tersebut, pada penelitian ini akan diusulkan sebuah metode untuk menghasilkan model peramalan yang sesuai untuk data runtun waktu yang ada, yaitu data jumlah kasus harian Covid 19 di Jakarta dengan metode Hybrid ARIMA-QNN berdasarkan dataset dari situs <https://corona.jakarta.go.id> tanggal 6 Maret 2020 sampai 30 Juni 2021 sebagai data training dan nanti akan diprediksi untuk tanggal 1 Juli 2021 sampai dengan 31 Juli 2021 sebagai data uji. Metode Hybrid ARIMA-QNN diusulkan dengan harapan akan memberikan nilai peramalan yang lebih akurat dengan presisi yang tinggi

BAB III

METODE PENELITIAN

3.1 Gambaran Umum Penelitian

Motivasi dari Metodologi yang diusulkan adalah membuat suatu metode peramalan yang sesuai dengan data runtun waktu yang ada serta meningkatkan akurasi dengan tetap memperhatikan efisiensi waktu komputasinya.

Langkah-langkah yang dilakukan dalam penelitian ini adalah menganalisis data jumlah kasus harian Covid 19 di Jakarta berdasarkan dataset dari situs <https://corona.jakarta.go.id> tanggal 6 Maret 2020 sampai 30 Juni 2021 sebagai data training dan nanti akan diprediksi untuk tanggal 1 Juli 2021 sampai dengan 31 Juli 2021 sebagai data uji dengan tahapan sebagai berikut :

1. Mempersiapkan data runtun waktu yang akan dianalisis
2. Menganalisis data runtun waktu yang ada menggunakan metode statistika ARIMA
3. Menganalisis data runtun waktu yang ada menggunakan metode Quantum Neural Network
4. Mengembangkan model Hybrid ARIMA-Quantum Neural Network
5. Menentukan model yang cocok untuk setiap variabel
6. Menguji kecocokan masing-masing model
7. Melakukan peramalan dengan menggunakan model yang cocok
8. Melakukan perbandingan tingkat akurasi hasil peramalan dengan tiap model

Untuk mendapatkan model peramalan yang diharapkan sesuai dengan data runtun waktu yang ada, maka perlu dilakukan pendekatan ilmiah yaitu dengan melihat pola data runtun waktu yang ada terlebih dahulu. Dengan melihat pola data awal yang dimiliki maka akan memudahkan dalam memilih model yang sesuai untuk data tersebut. Pendekatan lainnya adalah menggunakan tools untuk menentukan secara otomatis Bentuk model statistik ARIMA yang sesuai dengan runtun waktu yang ada, lalu model tersebut dilatih menggunakan quantum neural network agar diketahui pola-pola data yang sudah ada dan dapat diuji akurasi.

3.2 Model ARIMA

Bentuk umum model ARIMA dapat dinyatakan dalam persamaan berikut:

$$\phi_p(B)(1-B)^d Z_t = \theta_0 + \theta_q(B) a_t \dots\dots\dots 1$$

Operator AR adalah

$$\phi_p(B) = (1 - \phi_1 B - \dots - \phi_p B^p) \dots\dots\dots 2$$

Operator MA adalah

$$\theta_q(B) = (1 - \theta_1 B - \dots - \theta_q B^q) \dots\dots\dots 3$$

1. Autoregressive Integrated Moving Average (ARIMA) Notasi

model : ARIMA (p, d, q)

p : orde untuk proses *autoregressive* (AR),

d : orde yang menyatakan banyaknya proses diferensi dilakukan pada data *time series* yang tidak stasioner,

q : orde yang menyatakan proses *moving average* (MA).

Pola teoretis ACF dan PACF dari proses yang stasioner

Tipe Model	Pola Tipikal ACF	Pola Tipikal PACF
AR (p)	Menurun secara eksponensial/ sinusoidal	Terputus setelah lag p
MA (q)	Terputus setelah lag q	Menurun secara eksponensial/ sinusoidal
ARMA (p, q)	Menurun secara eksponensial/ sinusoidal	Menurun secara eksponensial/ sinusoidal

Sumber : Aswi dan Sukarna 2006

2. Tahapan Analisis Time Series (ARIMA)

a. Membuat Plot Time Series

- Identifikasi asumsi stasioneritas data runtun waktu.
- Suatu deret pengamatan dikatakan stasioner apabila proses tidak berubah seiring dengan perubahan waktu
- Tidak stasioner dalam mean : jika trend tidak datar (tidak sejajar smbu waktu)
- Tidak stasioner dalam varian : jika trend datar atau hampir datar, tetapi data tersebar membangun pola melebar atau menyempit (pola terompet)

- Tidak stasioner dalam mean & varians : jika trend tidak datar dan data membentuk pola terompak.

Augmented Dickey- Fuller (Uji Formal untuk Stasioneritas)

Hipotesis :

H_0 : Terdapat akar unit dan data tidak stasioner ($=0$)

H_1 : Tidak terdapat akar unit dan data stasioner (<0 span=" ">)

Taraf Signifikansi : $\alpha = \dots \%$

Statistik Uji :

$$\Delta Y_t = \alpha_0 + \gamma Y_{t-1} + \sum_{i=2}^p Y_{t-i+1} + \varepsilon_t \dots 4$$

$$\varepsilon_t \sim IID(0, \sigma^2)$$

$$\gamma = -(1 - \sum_{i=1}^p \alpha_i) \dots 5$$

$$\beta_i = -\sum_{j=1}^p \alpha_j \dots 6$$

Kriteria Uji :

H_0 ditolak jika nilai mutlak dari *Augmented Dickey- Fuller* > nilai kritis *MacKinnon* , atau nilai prob.* < α .

b. Menghitung/ Membuat Plot ACF dan PACF

- Mengidentifikasi model runtun waktu yang mungkin
- Mengestimasi parameter model

c. Uji Signifikansi Parameter

Hipotesis :

H_0 : $\phi_i = 0$ dan/atau $\theta_i = 0$ (parameter tidak signifikan terhadap model)

H_1 : $\phi_i \neq 0$ dan/atau $\theta_i \neq 0$ (parameter signifikan terhadap model)

Taraf Signifikansi : $\alpha = \dots \%$

Statistik Uji :

$$t_{hitung} = \frac{\hat{\phi}}{se(\hat{\phi})} \quad \text{dan/atau} \quad t_{hitung} = \frac{\hat{\theta}}{se(\hat{\theta})}$$

Kriteria Uji :

Tolak H_0 jika $|t_{hitung}| > t_{(\alpha/2; n-1)}$ atau p-value < alpha

d. Verifikasi Model

- Independensi Residual

Hipotesis :

$H_0 : \rho_{12} = \rho_{24} = \rho_{36} = \rho_{48} = 0$ (tidak ada korelasi antar-lag)

H_1 : Paling sedikit ada satu $\rho_k \neq 0$

dengan $k=12, 24, 36, 48$ (ada korelasi antar-lag)

Statistik Uji :

$$Q = n(n+2) \sum_{k=1}^m (n-k)^{-1} \hat{\rho}_k^2$$

Kriteria Uji :

Tolak H_0 jika $Q > \chi^2_{(\alpha, m-s)}$ atau p-value < alpha

dengan m = lag maksimum, s = jumlah parameter yang diestimasi,
dan α = taraf signifikans

- Normalitas Residual

Hipotesis :

$H_0 : F(a_t) = F_0(a_t)$ (residual berdistribusi normal)

$H_1 : F(a_t) \neq F_0(a_t)$ (residual tidak berdistribusi normal)

Statistik Uji :

$$D_0 = \max |F_0(a_t) - S_N(a_t)|$$

$F_0(a_t)$: fungsi peluang kumulatif residual

$S_N(a_t)$: distribusi kumulatif yang diobservasi dari suatu sampel acak
sebanyak N observasi

Kriteria Uji :

Tolak H_0 jika $D_0 > D_{tabel}$ atau p-value < alpha

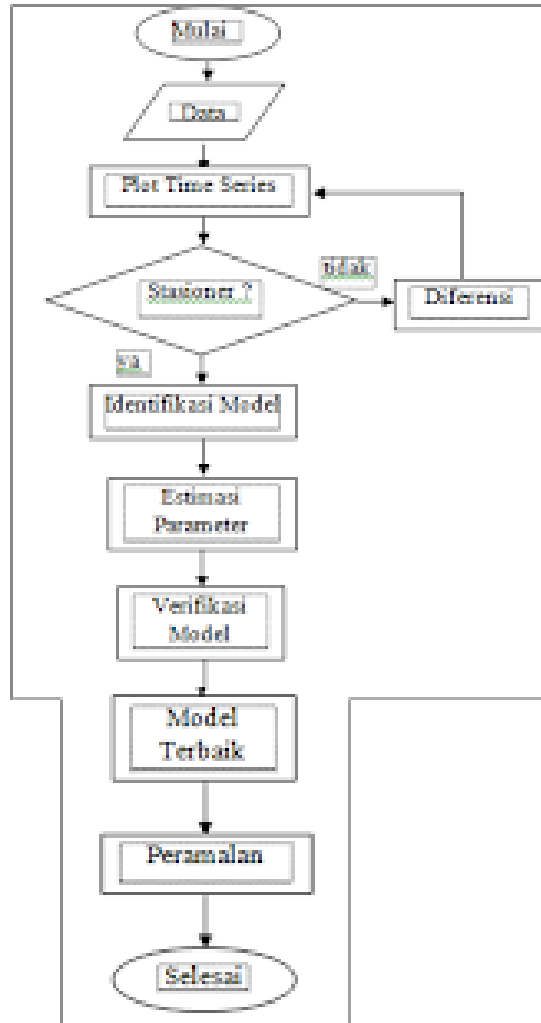
- Ukuran Ketepatan Ramalan

Model dengan ukuran ketepatan peramalan yang baik adalah model yang menghasilkan *error* yang kecil.

Nilai tengah kesalahan kuadrat (*Mean Square Error*)

$$MSE = \frac{\sum_{i=1}^n e_i^2}{n}$$

Berikut flowchart langkah-langkah membuat model
ARIMA :



Gambar 1. Flowchart Analisis Runtun Waktu ARIMA

3.3 Model Neural Network

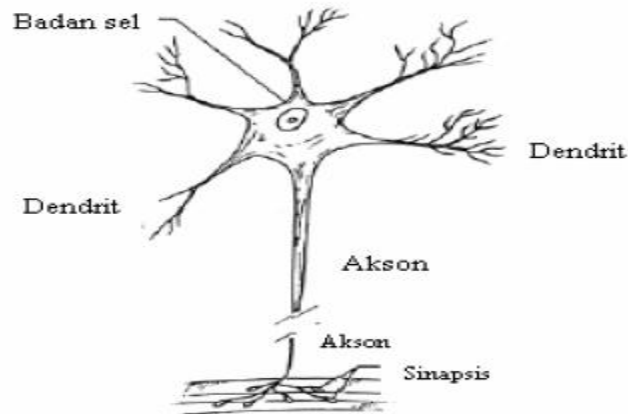
Dalam buku Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB, Drs. Jong Jek Siang, M.Sc menyebutkan bahwa Jaringan Syaraf Tiruan adalah system pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. Jaringan Syaraf Tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron)

- Sinyal dikirimkan di antara neuron-neuron melalui penghubung-penghubung
- Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal
- Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya

Jaringan Syaraf Tiruan ditentukan oleh tiga hal :

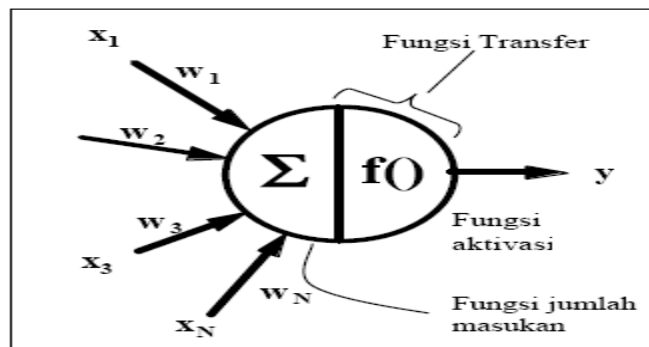
- Pola hubungan antar neuron (disebut arsitektur jaringan)
- Metode untuk menentukan bobot penghubung (disebut metode training/learning algoritma)
- Fungsi Aktivasi

Gambar skematik tipikal neuron dapat dilihat pada gambar 2



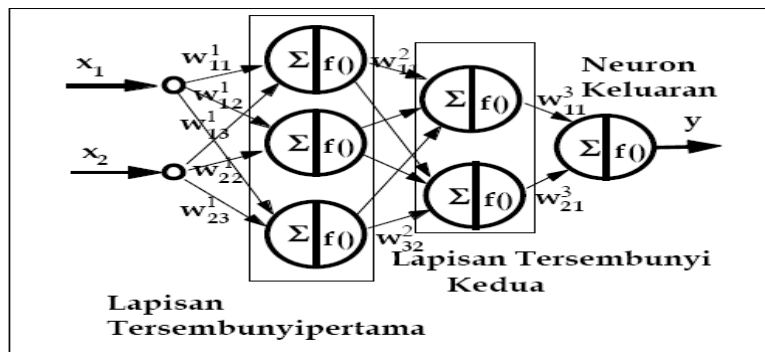
Gambar 2 Syaraf Biologis

Pemrosesan informasi dalam Jaringan Syaraf Tiruan dapat disingkat sebagai berikut : Sinyal (baik berupa aksi ataupun potensial) muncul sebagai masukan unit (sinapsis); efek dari tiap sinyal ini dinyatakan sebagai bentuk perkalian dengan sebuah nilai bobot untuk mengindikasikan kekuatan dari sinapsis. Semua sinyal yang diberi pengali bobot ini kemudian dijumlahkan satu sama lain untuk menghasilkan unit aktivasi. Jika aktivasi ini melampaui sebuah batas ambang tertentu maka unit tersebut akan memberikan keluaran dalam bentuk respon terhadap masukan. Unit aktivasi ini kemudian dibandingkan dengan sebuah nilai ambang, dan hasilnya dimasukkan kedalam fungsi transfer (fungsi non-linier) yang akan menghasilkan sebuah keluaran. Secara ringkas proses tersebut dapat digambarkan dalam gambar 3



Gambar 3 Neuron buatan McCulloch-Pitts sebagai operator matematis

Aktivasi dari unit masukan diatur dan diteruskan melalui jaringan hingga nilai dari keluaran dapat ditentukan. Jaringan berperan sebagai fungsi vektor yang mengambil satu vektor pada masukan dan mengeluarkan satu vektor lain pada keluaran. Model Jaringan Syaraf Tiruan dapat memiliki sebuah lapisan bobot, dimana masukan dihubungkan langsung dengan keluaran, atau beberapa lapisan yang didalamnya terdapat beberapa lapisan tersembunyi, karena berada tersembunyi diantara neuron masukan dan keluaran. Jaringan syaraf menggunakan unit tersembunyi untuk menghasilkan representasi pola masukan secara internal didalam jaringan syaraf. Fungsi transfer (*non-linier*) yang digunakan dalam tiap neuron (baik dilapisan masukan, keluaran, atau lapisan tersembunyi) dapat berupa fungsi nilai ambang, fungsi linier, fungsi sigmoid, ataupun fungsi gaussian, tergantung dari karakter neuron sesuai keinginan kita. Hal ini dapat dilihat pada gambar 4



Gambar 4 Tipikal Sebuah Jaringan Syaraf Tiruan

3.3.1 Komponen Jaringan Syaraf

Terdapat beberapa tipe jaringan syaraf, hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaraf juga terdiri atas beberapa neuron dan ada hubungan antar neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Neuron ini sebenarnya mirip dengan sel neuron biologis. Neuron-neuron buatan tersebut bekerja dengan cara yang sama pula dengan neuron biologis. Informasi (disebut dengan: input) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi

maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya.

Pada Jaringan syaraf, neuron-neuron akan dikumpulkan dalam lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layer*). Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan input dan lapisan output). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan. Mulai dari lapisan input sampai ke lapisan output melalui lapisan lainnya, yang sering disebut sebagai lapisan tersembunyi (*hidden layer*).

3.3.2 Arsitektur Jaringan Syaraf

3.3.2.1 Jaringan dengan lapisan tunggal (*single layer net*)

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.

3.3.3.3 Jaringan dengan banyak lapisan (*multilayer net*)

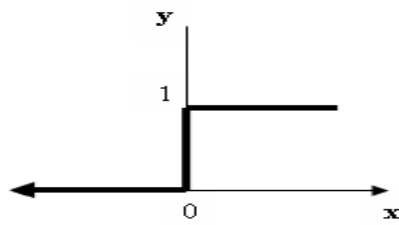
Jaringan dengan banyak lapisan memiliki 1 atau lebih lapisan yang terletak diantara lapisan input dan lapisan output (memiliki 1 atau lebih lapisan tersembunyi). Umumnya, ada lapisan bobot-bobot yang terletak antara 2 lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada jaringan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Namun demikian, pada banyak kasus, pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah.

3.3.3. Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan, antara lain :

a. Fungsi Undak Biner (*Hard Limit*)

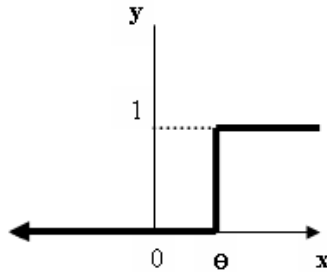
Jaringan dengan lapisan tunggal sering menggunakan fungsi undak (*step function*) untuk mengkonversikan input dari suatu variabel yang bernilai kontinu ke suatu output biner (0 atau 1)



Undak Biner (hard limit)

b. Fungsi undak biner (*Threshold*)

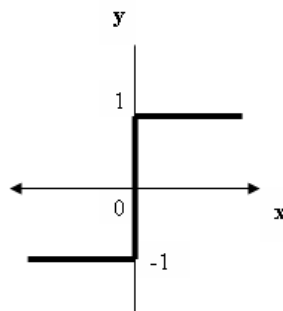
Fungsi undak biner dengan menggunakan nilai ambang sering juga disebut dengan fungsi nilai ambang (*Threshold*) atau fungsi *Heaviside*.



Undak Biner (threshold)

c. Fungsi Bipolar (*Symetric Hard Limit*)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner, hanya saja output yang dihasilkan berupa 1, 0 atau -1



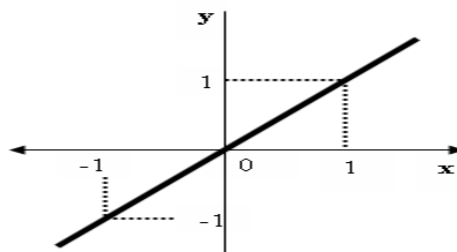
Bipolar (symmetric hard limit)

d. Fungsi Bipolar (dengan threshold)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner dengan threshold. Hanya saja keluaran yang dihasilkan berupa 1, 0, atau -1

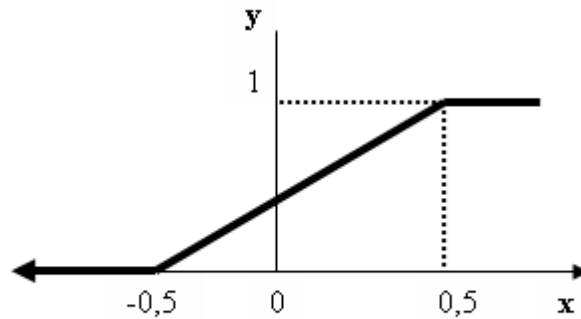
e. Fungsi Linear (*Identitas*)

Fungsi linear memiliki nilai output yang sama dengan nilai inputnya.



f. Fungsi Saturating Linear

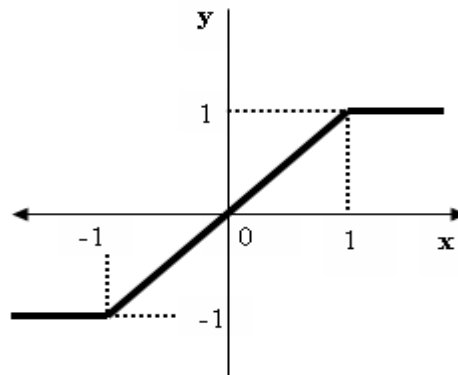
Fungsi ini akan bernilai 0 jika inputnya kurang dari $-\frac{1}{2}$, dan akan bernilai 1 jika inputnya lebih dari $\frac{1}{2}$. Sedangkan jika nilai input terletak antara $-\frac{1}{2}$ dan $\frac{1}{2}$, maka outputnya akan bernilai sama dengan nilai input ditambah $\frac{1}{2}$



Saturating Linear

g. Fungsi Symetric Saturating Linear

Fungsi ini akan bernilai -1 jika inputnya kurang dari -1, dan akan bernilai 1 jika inputnya lebih dari 1. Sedangkan jika nilai input terletak antara -1 dan 1, maka outputnya akan bernilai sama dengan nilai inputnya.



Symmetric Saturating Linear

h. Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode backpropagation. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1.

i. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja output dari fungsi ini memiliki range antara 1 sampai -1

Fungsi ini sangat dekat dengan fungsi hyperbolic tangent. Keduanya memiliki range antara -1 sampai 1. Untuk fungsi hyperbolic tangent,

3.4 MODEL HYBRID ARIMA NEURAL NETWORK

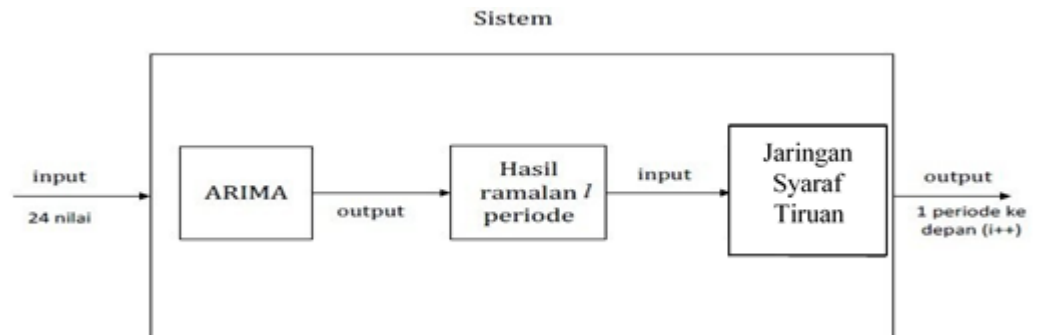
Berdasarkan hasil peramalan model ARIMA, akan dilakukan proses analisis runtun waktu menggunakan metode jaringan syaraf tiruan. Dengan kata lain, output dari peramalan model ARIMA akan menjadi input pada proses pengolahan data menggunakan metode jaringan syaraf tiruan. Kemudian akan ditentukan model jaringan syaraf tiruan yang sesuai dan cocok untuk data runtun waktu tersebut.

Secara matematis, hasil ramalan secara keseluruhan yang diperoleh adalah sebagai berikut :

$$\hat{Z}_t = \hat{L}_t + \hat{N}_t$$

Z_t merupakan hasil peramalan yang merupakan gabungan nilai ramalan dari model ARIMA atau Exponential Smoothing dan nilai ramalan dari model JST.

Berikut ini adalah arsitektur model peramalan hybrid ARIMA-JST dan ES-JST



Gambar 5 Model Hybrid ARIMA-JST dan Hybrid ES-JST

3.5 MODEL QUANTUM HYBRID ARIMA NEURAL NETWORK

Ada banyak pendekatan untuk pengembangan model Quantum Arima NN. Model-model ini fokus pada yang berbeda aspek komputasi kuantum dan pemrosesan saraf. Dalam komputasi kuantum, Sebagai unit informasi terkecil, bit kuantum atau qubit adalah sistem kuantum yang menyatakan terletak di ruang Hilbert dua dimensi. Seperti bit dalam klasik komputer, qubit berlabel dan mengekspresikan satu bit informasi: sesuai dengan bit 0 komputer klasik, dan bit 1. Keadaan qubit menyatakan superposisi keadaan yang koheren :

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \left(|\alpha|^2 + |\beta|^2 = 1\right) \dots\dots\dots 7$$

Di mana dan menentukan probabilitas yang sesuai. Gerbang kuantum yang mencakup karakteristik komputasi kuantum merupakan dasar untuk implementasi fisik dari komputasi kuantum. Himpunan logika universal termasuk dalam logika kuantum. Mirip dengan bit klasik, gerbang dasar dapat membentuk gerbang kuantum bermacam-macam dan menyelesaikan keadaan kuantum dari beberapa logika transformasi. berbasis elemen pada gerbang pergeseran fasa 1 bit dan gerbang kontrol-Tidak 2 bit dalam dinamika kuantum diambil sebagai fungsi aktivasi dalam Jaringan saraf. Untuk memudahkan aplikasi, formulir berikut:
Fungsi kompleks diberikan untuk menyatakan keadaan kuantum:

$$f(\theta) = e^{i\theta} = \cos\theta + i\sin\theta \dots\dots\dots 8$$

$i = \sqrt{-1}$ adalah bilangan imajiner adalah kuantum fase

3.6 Pengukuran Kinerja

3.6.1 Mean Squared Error

Dalam statistik, *Mean Squared Error* (MSE) sebuah estimator adalah nilai yang diharapkan dari kuadrat *error*. *Error* yang ada menunjukkan seberapa besar perbedaan hasil estimasi dengan nilai yang akan diestimasi. Perbedaan itu terjadi karena adanya keacakan pada data atau karena estimator tidak mengandung informasi yang dapat menghasilkan estimasi yang lebih akurat

3.6.2 Komparasi Hasil Peramalan

Setelah nilai Mean Squared Error dari kedua metode didapatkan, maka akan dilakukan komparasi terhadap nilai MSE yang didapatkan pada periode testing (*out-sample*)

- Jika nilai $MSE_{STATISTIKA} < MSE_{ANN}$ maka metode Statistika memiliki performa lebih baik dibandingkan metode ANN karena memiliki tingkat kesalahan relatif lebih kecil.
- Sebaliknya, jika $MSE_{STATISTIKA} > MSE_{ANN}$ maka metode Statistika memiliki performa lebih buruk dibandingkan metode ANN karena tingkat kesalahan yang dihasilkan relatif lebih besar.