

BAB II TINJAUAN PUSTAKA

1.1 Saham

Menurut Darmadji dan Fakhruddin (2012), tanda kepemilikan seseorang atau badan terhadap suatu perusahaan disebut saham (*stock*). Saham memiliki wujud berupa selembar kertas yang berisi informasi bahwa pemilik kertas tersebut merupakan pemilik perusahaan yang menerbitkannya. Menurut Fahmi (2012), saham merupakan instrument pasar modal yang memberikan tingkat keuntungan paling menarik sehingga banyak diminati investor.

Menurut Darmadji dan Fakhruddin (2012), ada beberapa jenis saham yaitu:

1. Ditinjau dari segi kemampuan dalam hak tagih atau klaim, maka saham terbagi atas:

- a) Saham yang menempatkan pemiliknya paling akhir terhadap pembagian dividen, dan hak atas harta kekayaan perusahaan apabila perusahaan tersebut dilikuidasi disebut saham biasa (*common stock*).
- b) Saham yang memiliki karakteristik gabungan antara obligasi dan saham biasa, karena bisa menghasilkan pendapatan tetap (seperti bunga obligasi), tetapi juga bisa

tidak menghasilkan bunga tetap disebut saham preferen (*preferred stock*).

2. Dilihat dari cara pemeliharaannya, saham dibedakan menjadi:

a) Saham yang tidak tertulis nama pemiliknya, agar mudah dipindahtangankan dari satu investor ke investor lain disebut saham atas unjuk (*bearer stock*).

b) Saham yang ditulis dengan jelas siapa pemiliknya, dan apabila ingin dipindahtangankan maka harus melalui prosedur tertentu disebut saham atas nama (*registered stock*).

2. Ditinjau dari kinerja perdagangannya, maka saham dapat dikategorikan menjadi:

a) Saham biasa dari suatu perusahaan yang memiliki reputasi tinggi serta sehat secara finansial. Saham jenis ini memiliki pendapatan yang stabil dan konsisten dalam membayar dividen, bahkan ketika bisnis berjalan lebih buruk dari biasanya. Jenis saham ini disebut saham unggulan (*blue-chip stock*)

b) Saham biasa dari suatu emiten yang memiliki kemampuan membayar dividen secara teratur bahkan lebih tinggi dari rata-rata dividen yang dibayarkan pada tahun sebelumnya disebut saham pendapatan (*income stock*)

c) Saham pertumbuhan (*growth stock*), terbagi menjadi dua yaitu *well known* dan *lesser known*. *Well known* adalah

saham dari emiten yang merupakan pemimpin dalam industrinya dan memiliki reputasi tinggi serta pertumbuhan pendapatan yang tinggi. Sedangkan *lesser known* adalah saham dari emiten yang umumnya bukan sebagai pemimpin dalam industrinya, namun saham ini memiliki ciri *growth stock*.

- d) Saham suatu perusahaan yang tidak bisa secara konsisten memperoleh penghasilan yang baik di masa mendatang disebut saham spekulatif (*spekulative stock*).
- e) Saham yang paling stabil karena tidak terpengaruh oleh kondisi ekonomi makro maupun situasi bisnis secara umum disebut saham siklikal (*counter cyclical stock*).

2.2 Saham Syariah

Islam telah mengatur seluruh kehidupan umat manusia dan menawarkan berbagai macam cara dan kiat untuk menjalaninya sesuai dengan norma – norma dan aturan Allah. Dunia investasi Islam juga memberikan petunjuk dan rambu – rambu. Menurut Huda dan Nasution (2008), rambu – rambu tersebut adalah sebagai berikut:

a) Terbebas dari Unsur Riba

Riba berarti tumbuh dan bertambah. Riba juga berarti mengambil atau memperoleh harta dengan cara tidak benar. Ayat berikut melarang riba dengan lebih jelas dan tegas:

“Hai orang-orang yang beriman, janganlah kamu memakan riba dengan berlipat ganda dan bertakwalah kamu kepada Allah supaya kamu mendapat keberuntungan.” (Q.S Ali Imran: 130).

b) Terhindar dari Unsur Gharar

Gharar secara etimologi bermakna kekhawatiran atau resiko. Gharar juga berarti semua jual beli yang mengandung ketidakjelasan, atau keraguan yang bisa mengakibatkan suatu kecelakaan, kerugian, dan kebinasaan.

c) Terhindar dari Unsur Judi (Maysir)

Maysir secara etimologi bermakna mudah. Maysir merupakan cara untuk mendapatkan uang dengan mudah tanpa kerja keras dan bertentangan dengan nilai serta aturan syariah. Allah SWT telah melarang segala jenis perjudian, hal tersebut tertuang dalam Al-Qur'an surat al-Maidah ayat 90-91:

“Hai orang-orang yang beriman, Sesungguhnya (meminum) khamar, berjudi, (berkorban untuk) berhala, mengundi nasib dengan panah, adalah termasuk perbuatan syaitan. Maka jauhilah perbuatan-perbuatan itu agar kamu mendapat keberuntungan.” (Al-Maidah: 90)

“*Sesungguhnya syaitan itu bermaksud hendak menimbulkan permusuhan dan kebencian di antara kamu lantaran (meminum) khamar dan berjudi itu, dan menghalangi kamu dari mengingat Allah dan sembahyang; Maka berhentilah kamu (dari mengerjakan pekerjaan itu).*” (Al-Maidah: 91)

d) Terhindar dari unsur haram

Investasi yang dilakukan oleh investor muslim diharuskan terhindar dari unsur haram. Sesuatu yang haram merupakan segala sesuatu yang dilarang Allah SWT.

e) Terhindar dari unsur syubhat

Kata syubhat berarti mirip, serupa, semisal, dan bercampur. Dalam terminologi syariah syubhat berarti sebagai sesuatu perkara yang tercampur (antara halal dan haram), akan tetapi tidak diketahui secara pasti apakah ia sesuatu yang dihalalkan atau diharamkan.

Saham syariah merupakan saham biasa yang memiliki karakteristik khusus yang penerapan prinsip syariah berupa kontrol ketat dalam hal kehalalan ruang lingkup kegiatan usaha. Menurut Yuliana (2010) prinsip dasar saham syariah meliputi:

- 1) Jika ditawarkan secara terbatas maka bersifat *musyarakah*
- 2) Jika ditawarkan kepada publik maka bersifat *mudharabah*
- 3) Tidak boleh ada perbedaan jenis saham, karena risiko harus ditanggung oleh semua pihak
- 4) Prinsip bagi hasil laba rugi

5) Tidak dapat dicairkan kecuali dilikuidasi

Saham syariah adalah saham - saham perusahaan yang sesuai dengan prinsip syariah. Secara keseluruhan daftar saham syariah terdapat dalam Daftar Efek Syariah (DES). Sedangkan dalam prinsip syariah, penyertaan modal hanya dilakukan pada perusahaan-perusahaan yang tidak melanggar prinsip-prinsip syariah, seperti bidang riba, perjudian dan memproduksi barang yang diharamkan.

Sejak November 2007, keberadaan DES kemudian ditindaklanjuti oleh Bursa Efek Indonesia (BEI) dengan meluncurkan Indeks Saham Syariah Indonesia (ISSI) pada tanggal 12 Mei 2011. ISSI merupakan indeks saham yang mencakup keseluruhan saham syariah yang tercatat di BEI dan DES. Saham-saham yang masuk dalam penghitungan ISSI sifatnya tidak tetap, tapi berubah tiap 6 bulan sekali (Mei dan November) dan dipublikasikan pada awal bulan berikutnya. Hal ini dilakukan karena penyesuaian apabila ada saham syariah yang baru tercatat atau dihapuskan dari DES. Metode perhitungan indeks ISSI adalah rata-rata tertimbang 16 dari kapitalisasi pasar. Tahun dasar yang digunakan dalam perhitungan ISSI adalah awal penerbitan DES yaitu Desember 2007 (Bursa Efek Indonesia, 2010).

2.3 Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang mungkin terjadi di masa yang akan datang berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahan dapat diperkecil. Prediksi harus memberikan jawaban secara pasti kejadian yang akan datang, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi.

Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan.

Baik tidaknya suatu prediksi yang disusun selain ditentukan oleh metode yang digunakan, juga ditentukan oleh baik tidaknya informasi yang digunakan. Selama informasi yang digunakan tidak dapat menyakinkan untuk mendapat hasil yang bagus, hasil peramalan yang disusun juga akan sukar dipercaya ketepatannya. Keberhasilan dari suatu prediksi sangat ditentukan oleh:

1. Pengetahuan teknik tentang pengumpulan informasi (data) masa lalu, dapat ataupun informasi tersebut bersifat kuantitatif.
2. Teknik dan metode yang tepat dan sesuai dengan pola data yang dikumpulkan.

2.4 Time Series

Time series adalah pengamatan pada suatu variabel dari waktu lampau dan dicatat secara berurutan menurut urutan waktu dengan periode yang tetap (Hanke & Wichern, 2004: 58). Pada umumnya pencatatan ini dilakukan dalam periode tertentu misalnya harian, mingguan, bulanan, dan sebagainya, sedangkan analisis *time series* adalah suatu metode kuantitatif untuk menentukan pola data tersebut, maka dapat digunakan untuk peramalan di masa mendatang. Beberapa konsep dasar dalam analisis *time series* adalah autokorelasi, konsep *white noise* dan plot data. Terdapat empat jenis plot data pada peramalan *time series*, antara lain Trend (T), Musiman/seasonal (S), siklik (C), dan horizontal (H).

1. Trend (T)

Pola data trend terjadi bilamana data pengamatan mengalami kenaikan atau penurunan selama periode jangka panjang. Suatu data pengamatan yang mempunyai trend disebut data nonstasioner.

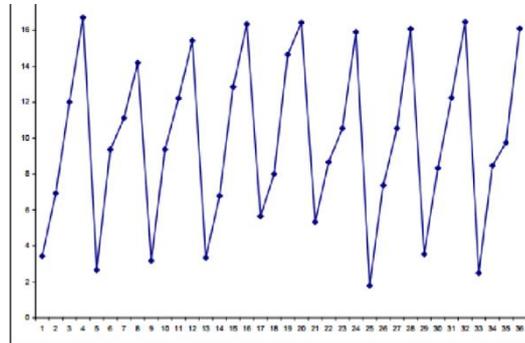


Gambar 2. 1 Pola Data Trend

2. Musiman Atau Seasonal (S)

Pola data musiman terjadi bilamana suatu deret dipengaruhi oleh faktor musiman. Pola data musiman dapat mempunyai pola musim yang berulang dari periode ke periode berikutnya. Misalnya pola yang

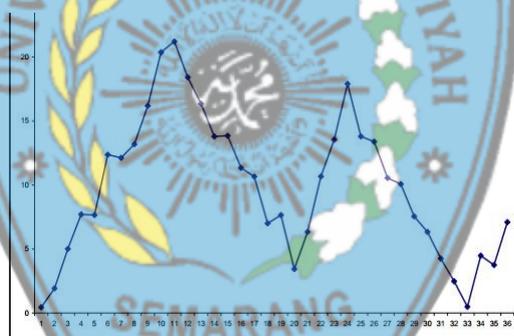
berulang setiap bulan tertentu, tahun tertentu atau pada minggu tertentu.



Gambar 2. 2 Pola Data Musiman

3. Siklis (C)

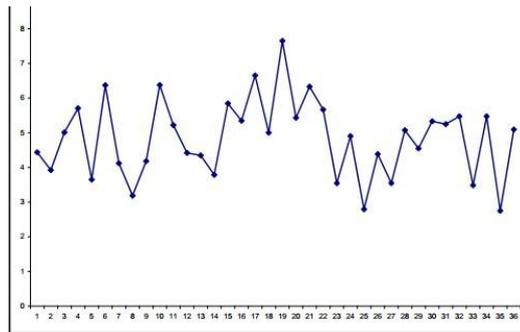
Pola data siklis terjadi bilamana deret data dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis.



Gambar 2. 3 Pola Data Siklis

4. Horosontal (H)

Pola data ini terjadi pada saat observasi berfluktuasi disekitaran suatu nilai konstan atau membentuk garis horizontal. Data ini disebut juga dengan data stasioner.



Gambar 2. 4 Barpola Data Horisontal

2.5 Neural Network

Jaringan saraf tiruan (JST) atau sering dikenal dengan istilah *neutral network* (NN) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi manusia, ini adalah dasar dari kecerdasan buatan (AI) dan memecahkan masalah yang terbukti tidak mungkin atau sulit menurut standar manusia atau statistik. JST memiliki kemampuan belajar mandiri yang memungkinkannya menghasilkan hasil yang lebih baik saat lebih banyak data tersedia. *Neural network* telah dimanfaatkan di berbagai bidang salah satunya adalah peramalan. Pada dasarnya *Neural network* merupakan kumpulan elemen – elemen proses yang saling berhubungan yang disebut *unit-unit* atau syaraf-syaraf. *Neural network* berkerja berdasarkan pola pada inputnya. Setiap *neuron* dihubungkan dengan *neuron* lainnya dengan satu koneksi link, yang direpresentasikan dengan bobot atau *weight*. Metode yang menentukan *weight* disebut *training*, *learning* atau *algoritma*. Setiap *neuron* menggunakan aktivasi pada input yang digunakan untuk memprediksi *output*. NN termasuk dalam kategori *time series* dan memiliki bentuk fungsional yang fleksibel

sehingga tidak memiliki asumsi – asumsi yang harus terpenuhi. *Neuron – neuron* disusun dalam grup yang disebut layer (lapis). NN memiliki dua tahapan dalam system pemrosesan informasi yaitu:

a) Tahap pelatihan (training)

Tahapan ini dimulai dengan memasukan data latih ke dalam jaringan. Menggunakan jaringan, jaringan akan mengubah bobot yang menjadi penghubung antar *node*. Tahapan ini berlangsung pada beberapa iterasi dan akan berhenti setelah menemukan bobot yang sesuai dengan nilai *error* yang diinginkan.

b) Tahapan pengujian (testing)

Pengujian dilakukan dengan memasukan suatu pola yang belum pernah dilatih sebelumnya (data uji) dengan menggunakan bobot hasil tahap pelatihan. Diharapkan bobot-bobot hasil pelatihan yang telah menghasilkan error minimal juga akan menghasilkan error minimal pada tahap pengujian. Data training digunakan dalam membuat arsitektur terbaik dalam proses pelatihan, sedangkan data testing digunakan untuk pengujian keakuratan dari arsitektur yang telah terbentuk.

2.6 Komponen Neural Network

Neural Network memiliki karakteristik dengan karakteristik syaraf manusia, berdasarkan karakteristiknya NN dibangun dengan komponen yang sama dengan manusia diantaranya:

1. Neuron/node

Bertugas memproses semua informasi yang diterima, sama dengan fungsi *neuron* pada otak manusia, dimana semua proses perhitungan dilakukan disini.

2. Input

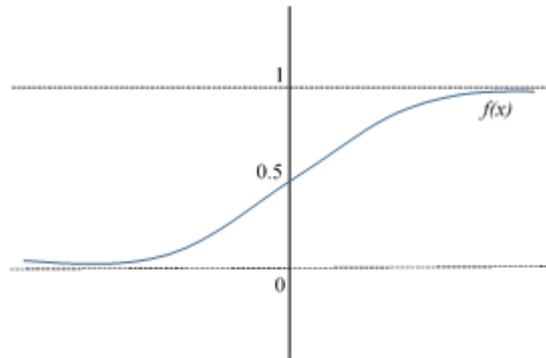
Informasi yang diproses dalam *neuron* input berasal dari lingkungan ataupun dari *node* lain. NN dapat memeproses data masukan berupa numerik sehingga apabila ada masalah yang meliubatkan data kualitatif seperti grafik, gambar, sinyal harus di transformasikan terlebih dahulu kedalam data numerik.

3. Fungsi aktivasi

Fungsi aktivasi yang akan menentukan apakah sinyal dari input neuron akan diteruskan atau tida. Fungsi aktivasi jaringan syaraf tiruan yang dilatih menggunakan optimasi *levenberg marquardt* adalah fungsi sigmoid biner. Fungsi *sigmoid biner* mempunyai interval range 0 sampai 1, sehingga fungsi ini digunakan untuk jaringan syaraf tiruan yang memiliki nilai output yang terletak pada interval 0 sampai 1. Fungsi *sigmoid biner* dirumuskan sebagai:

$$y = f x = \frac{1}{1+e^{-x}} \quad (2.2)$$

dengan, $y' x = (x) [1 - f(x)]$



Gambar 2. 5 Fungsi Aktivasi Singmoid

Fungsi *sigmoid* memiliki nilai maksimum = 1. Maka untuk pola yang targetnya > 1 , pola *input* dan *output* harus terlebih dahulu ditransformasi sehingga semua polanya memiliki range yang sama seperti fungsi *sigmoid* yang dipakai.

4. Bias dan threshold

Bias dapat dipandang sebagai input yang bernilai 1. Bias berfungsi untuk mengubah nilai threshold menjadi 0. Sedangkan threshold berperan sebagai penimbang dalam suatu hubungan dari sebuah unit tertentu.

5. Bobot

Bobot merupakan suatu nilai yang menunjukkan kekuatan antar code. Jika bobot diatur beda maka hasilnya akan beda. Bobot – bobot yang optimal memungkinkan system menerjemahkan data secara benar dan menghasilkan keluaran yang diinginkan.

6. Output

Suatu nilai yang dihasilkan dari fungsi aktivasi, yang bisa berupa output dari jaringan atau menjadi input bagi *node* lain disebut sebagai *output*.

2.7 Artificial Neural Network

Arsitektur jaringan NN merupakan susunan dari *neuran – neurana* dalam lapisan *input*, *hidden* dan *output* yang berhubungan dengan bobot, fungsi aktivasi dan fungsi pembelajaran. Arsitektur ini merupakan salah satu karakteristik yang sangat penting yang membedakan NN. Secara umum ada tiga lapis untuk menentukan NN.

1. Lapis Input

Unit dilapisan *input* disebut unit – unit *input*. Unit – unit *input* tersebut menerima pola inputan dari luar yang menggambarkan suatu masalah. Banyak *node* dalam lapisan input tergantung banyaknya input dalam model dan setiap input menentukan satu *neuron*.

2. Lapis Tersembunyi (*hidden layer*)

Unit-unit dalam lapisan tersembunyi disebut unit-unit tersembunyi, di mana *outputnya* tidak dapat diamati secara langsung. Lapis tersembunyi terletak di antara lapis *input* dan lapis *output*, yang dapat terdiri atas beberapa lapis tersembunyi.

3. Lapis Output

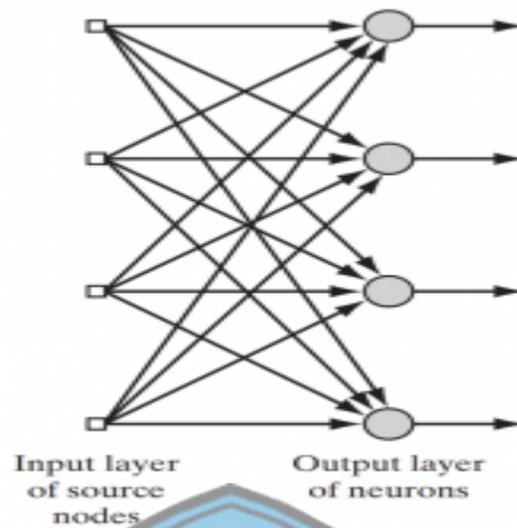
Unit-unit dalam lapisan *output* disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi *Neural Network* terhadap suatu permasalahan. Setelah melalui proses training, *network* merespon *input* baru untuk menghasilkan *output* yang merupakan hasil prediksi.

Pemodelan ANN ada 3 macam yaitu:

1. Single Layer



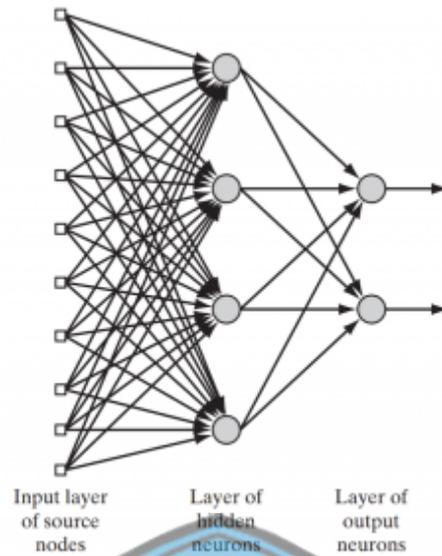
Dalam ANN, *neuron* disusun dalam bentuk lapisan (*layer*). Pembentukan ANN yang paling sederhana yaitu *single layer*. Cara kerja dari *single layer*, *input layer* yang berasal dari sumber *node* di proyeksikan langsung ke *output layer* dari *neuron* (node komputasi), tetapi tidak berlaku sebaliknya. Permodelan ini merupakan jenis jaringan *feedforward* yang dapat dilihat pada gambar 2.6. Pada gambar tersebut input dan output memiliki 4 *node*, namun yang dimaksud dengan *single layer* yaitu *output* dari jaringan, sedangkan inputnya tidak memiliki pengaruh karena pada saat melakukan *input* tidak terjadi proses komputasi.



Gambar 2. 6 Single Layer

2. Multi Layer

Pada *single layer* apabila terdapat tambahan satu atau dua *hidden layer* maka jaringan akan terganggu karena *input* dan *output* dari jaringan tidak dapat melihat *hidden layer* yang di masukkan. Sehingga memerlukan jaringan yang bisa menampungnya yaitu bernama *multi layer*. Cara kerja *multi layer* adalah *input layer* menyuplai *input vektor* pada jaringan, kemudian *input* yang dimasukkan melakukan komputasi pada layer yang kedua, lalu *output* dari layer yang kedua digunakan sebagai *input* dari layer yang ketiga dan seterusnya. Ilustrasi jaringan *multi layer* dapat di lihat pada gambar 2.8.



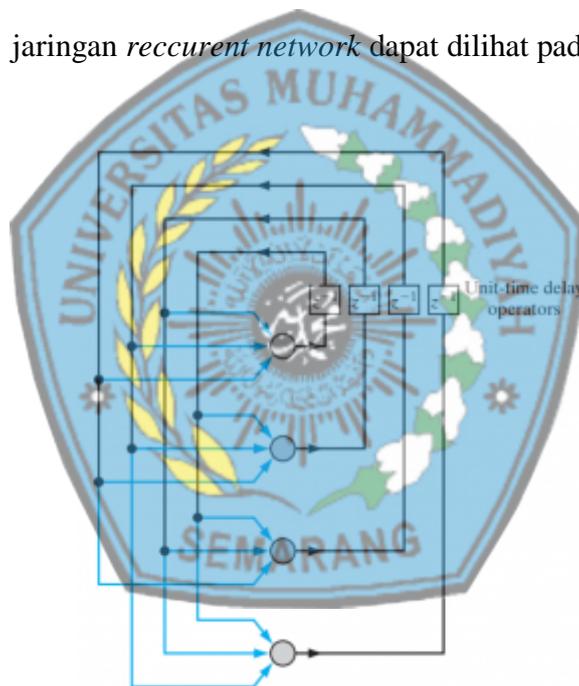
Gambar 2. 7 Multi Layer

3. Recurrent Network

Recurrent Neural Network merupakan jaringan saraf berulang. Dikatakan jaringan saraf berulang karena nilai *neuron* pada *hidden layer* sebelumnya akan digunakan kembali sebagai data input. Penggunaan *neuron* pada *hidden layer* akan disimpan ke dalam sebuah *layer* yang dinamakan *context layer*. *Recurrent neural network* adalah model pada *neural network* yang mengakomodasi output jaringan untuk menjadi input jaringan kembali bisa disebut jaringan umpan balik. *Algoritma* ini dikembangkan dari *feedforward neural network*. *Recurrent neural network* merupakan metode yang kompleks dan dinamis

dikarenakan hasil yang dihasilkan tidak hanya terpengaruh oleh *input* saja, melainkan juga dari hasil *output* sebelumnya.

Recurrent network terbentuk karena pada jaringan *single layer* dan *multi layer* harus memiliki *feedback* untuk dirinya sendiri pada setiap loop jaringannya, pada *recurrent network* jaringan tidak memerlukan *feedback* untuk dirinya sendiri melainkan *feedback* dari input yang digunakan. Ilustrasi jaringan *recurrent network* dapat dilihat pada gambar 2.9.



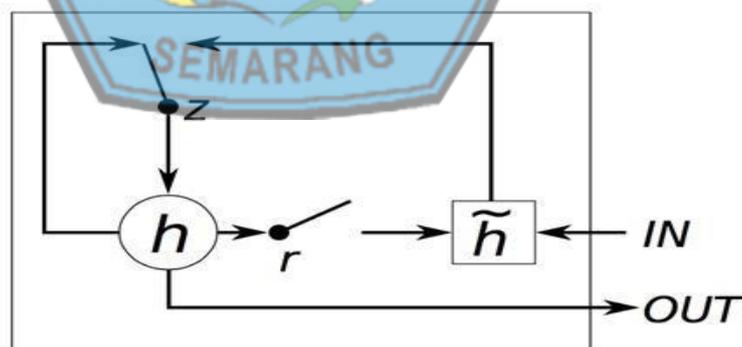
Gambar 2. 8 Recurrent Network

2.8 Gated Recurrent Unit

GRU merupakan variasi dari LSTM yang lebih sederhana. Sementara GRU adalah struktur berulang yang dirancang dengan hati-hati yang membuat *trade-off* yang baik antara kinerja dan kecepatan. Oleh

karena itu, GRU telah lazim digunakan dalam bidang akademik dan industri. GRU pertamakali diperkenalkan oleh Chung et al pada tahun 2014. Tujuan utama dari pemuatan GRU adalah untuk membuat setiap *recurrent unit* untuk dapat menangkap *dependencies* dalam skala waktu yang berbeda-beda secara adaptif. Sebagai analogi, manusia tidak perlu menggunakan semua informasi pada masa lalu untuk dapat membuat keputusan sekarang.

Di dalam GRU, komponen pengatur alur informasi tersebut disebut sebagai *gate* dan GRU mempunyai 2 *gate*, yaitu *reset gate* dan *update gate*. Bila kita ingin membuat keputusan untuk makan seperti analogi diatas, *reset gate* pada GRU akan menentukan bagaimana untuk menggabungkan input baru dengan informasi masa lalu, dan *update gate*, akan menentukan berapa banyak informasi masa lalu yang harus tetap disimpan.



Gambar 2. 9 Struktur GRU

Pada gambar diatas, r melambangkan *reset gates*, dan z melambangkan *update gates*. Sedangkan h dan \tilde{h} adalah *activation* dan *candidate activation*. *Activation* dan *candidate activation* adalah fungsi aktivasi. Sekarang,

GRU masih belum terlalu di eksplorasi cukup mendalam, karena umurnya yang masih cukup muda. Masih banyak kemungkinan yang dapat dikembangkan dari GRU untuk dapat menjadi RNN yang efisien dan mempunyai akurasi tinggi.

Prosedur ini mengambil rata-rata jumlah linear dengan keadaan yang telah ada dan jumlah linear keadaan yang baru. Namun GRU tidak mempunyai mekanisme atau cara untuk mengontrol tingkatan untuk memilih keadaan yang akan ditampilkan, tetapi menampilkan seluruh keadaan setiap waktu.

Aktivasi h_t^i pada GRU dengan waktu t adalah interpolasi linier antara aktivasi sebelumnya h_{t-1}^i . Maka aktivasi \tilde{h}_t^j adalah

$$h_t^i = (1 - z_t^j) h_{t-1}^i + z_t^j \tilde{h}_t^j \quad (2.3)$$

Dimana *update gate* z_t^j menentukan banyak unit baru sebagai aktivasi. Batas baru di berikan sebagai berikut:

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \quad (2.4)$$

Kandidat aktivasi \tilde{h}_t^j di hitung seperti recurrent unit.

$$\tilde{h}_t^j = \tanh(W x_t + U(r_t \odot h_{t-1}))^j \quad (2.4)$$

Dimana r_t adalah *reset gate* dan \odot adalah elemen multiplikasi. Ketika berhenti (r_t^j mendekati 0), *reset gate* secara efektif membuat unit bertindak seperti sedang membaca simbol pertama dari urutan input. *Reset gate* r_t^j di hitung seperti *update gate*:

$$r_t^j = (W_r x_t + U_t h_{t-1}) \quad (2.5)$$

2.9 Mean Absolute Percetage Error

MAPE atau *Mean Absolute Percetage Error* adalah ukuran akurasi dari suatu prediksi atau suatu peramalan, untuk menghitung seberapa baik sistem dalam meramal inflasi maka diperlukan tahapan evaluasi. Pada tahap ini MAPE digunakan untuk mengukur tingkat akurasi model yang sudah dilati. Berikut adalah rumus dari MAPE.

$$MAPE = \left(\frac{100\%}{n}\right) \sum_{t=1}^n \left| \frac{A_t - f_t}{A_t} \right| \quad (2.6)$$

Dimana:

A_t : Jumlah pada periode ke-t,

F_t : Hasil prediksi/peramalan pada periode ke-t,

N : Jumlah observasi



Interpretasi dari perhitungan MAPE menurut (Chen *et al*, 2007) adalah sebagai berikut:

1. nilai MAPE < 10% sangat baik untuk melakukan prediksi.
2. nilai MAPE berada diantara 10% - 20% menghasilkan nilai prediksi yang baik.
3. nilai MAPE berada diantara 20% - 50% menghasilkan prediksi yang proporsional/wajar artinya masih dapat digunakan dalam memprediksi.
4. Nilai MAPE > 50% tidak dapat digunakan dalam prediksi.

2.10 Mean Squere Error

MSE atau *Mean Squere Error* adalah rata – rata umlah kuadrat selisih target dalam kesalahan meramal dan hasil output jaringan. Semakin kecil nilai MSE dari suatu suatu model yang dignakan maka semakin baik yang dihasilkan. Rumus MSE dihitung sebagai berikut.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.7)$$

Keterangan:

Y_i = Nilai actual pada periode i

\hat{Y}_i = Niai hasil peramalan pada periode i

N = Jumlah peramalan

2.11 Inisialisasi Hyperparameter Model GRU

Hyperparameters merupakan salah satu faktor yang mempengaruhi kinerja dari model GRU-RNN untuk mendapatkan performa yang lebih tinggi saat melatih model GRU-RNN tersebut. Ketika nilai hyperparameters diubah, kinerja model juga akan berubah. Inisialisasi Hyperparameter pembelajaran dalam GRU – RNN ini meliputi *algoritma optimasi*, jumlah iterasi/*epoch*, dan *batch size*.

1. Algoritma Optimasi

Algoritma yang digunakan dalam penelitian ini adalah Adaptive Moment Estimation (Adam). Adam pertama kali di presentasikan oleh Diederik Kingma dari OpenAI dan Jimmy Ba dari University of Toronto. Adam adalah algoritma pengoptimalan yang dapat di gunakan untuk memperbaharui *weight network* secara iteratif berdasarkan data training. Metode ini menghitung tingkat pembelajaran adaptif individu untuk parameter yang berbeda dari perkiraan momen pertama dan kedua dari gradient. Adam adalah algoritma yang populer di bidang deep learning karena ia mencapai hasil yang baik dengan cepat. Hasil empiris menunjukkan bahwa Adam bekerja dengan baik dalam praktiknya dan lebih baik dibandingkan dengan stochastic optimization method lainnya. Adam adalah algoritma yang digunakan untuk optimalisasi *gradient* pada *neural networks* berdasarkan data latih. Metode ini lebih mudah diimplementasikan, efisien secara komputasi, memerlukan kebutuhan memori yang kecil, dan sesuai untuk masalah yang memiliki banyak

data maupun parameter. Adam berbeda dengan algoritma *Stochastic Gradient Descent* yang memiliki laju pembelajaran (*learning rate*) yang tidak berubah setiap pembaruan bobot (*weight update*). Algoritma adam menghitung rata-rata bergerak eksponensial (*Exponential Moving Average*) dari gradient dan gradient kuadra. Parameter β_1 dan β_2 mengendalikan laju pengurangan (*decay rate*) dari rata-rata bergerak.

Ada beberapa kelebihan dari metode Adam yaitu ukuran parameter dari parameter tanggal tidak berubah untuk penskalaan gradien, ukurannya kira-kira dibatasi oleh ukuran langkah hyperparameter, itu tidak memerlukan tujuan stasioner, bekerja dengan gradien yang jarang.

2. Epoch

Epoch merupakan banyaknya jumlah iterasi yang digunakan. Siklus algoritma ketika seluruh *dataset* sudah melalui proses training pada Neural Network sampai dikembalikan awal dalam satu kali putaran data training. Satu *epoch* berarti sebuah algoritma dari data training secara keseluruhan. Seiring bertambahnya jumlah *epoch* tidak dapat ditentukan namun bergantung pada dataset yang dimiliki.

3. Batch size

Batch size merupakan jumlah sampel dari dataset yang dimasukkan ke dalam model setiap iterasi. *Batch size* untuk mempermudah dan mempercepat proses training dataset dibagi per *batch size*. *Batch size* efisien secara komputasi ketika berhadapan dengan dataset yang besar. *batch size* yang besar memerlukan memori yang lebih besar dan waktu

pelatihan menjadi lebih cepat. Penentuan nilai dari *batch size* biasanya tergantung peneliti melihat banyaknya sampel.

2.12 Fungsi Aktifasi

Fungsi aktifasi adalah fungsi non linier yang memungkinkan JST untuk dapat mentransformasikan data input menjadi dimensi yang lebih tinggi sehingga dapat dilakukan pemotongan hyperplane sederhana yang memungkinkan dilakukan klasifikasi. Dalam penelitian ini terdapat tiga fungsi aktifasi yaitu:

1. Sigmoid

Fungsi *sigmoid* mentransformasi range nilai dari input x menjadi 0 dan 1 fungsi *sigmoid* memiliki bentuk sebagai berikut.

$$f(x) = \frac{1}{(1+e^{-x})} \quad (2.8)$$

2. Softmax

fungsi *softmax* menghitung probabilitas masing – masing kelas tujuan melalui semua kelas yang ada. Kemudian probabilitas melakukan kalkulasi yang sangat membantu untuk menentukan kelas tujuan dari input yang diberikan. Keuntungan utama dari *softmax* adalah output dengan rentang 0-1 dan jumlah probabilitas akan sama dengan satu. Fungsi *softmax* didefinisikan sebagai berikut.

$$f(x_i) = \frac{\text{Exp}(x_i)}{\sum_{j=0}^k \text{Exp}(x_j)}, \text{ dimana } i = 0,1,2, \dots k \quad (2.9)$$

3 Tanh

Fungsi *tanh* memiliki distribusi yang sangat mirip dengan fungsi *sigmoid* namun dengan range antara -1 – 1 dan distribusi yang lebih sempit. Fungsi *tanh* memiliki bentuk sebagai berikut.

$$f(x) = \tanh(x) \quad (2.10)$$

Karakteristik terpusat pada nol tersebut mengatasi kelemahan utama pada fungsi *sigmoid*, sehingga dalam praktek, fungsi *tanh* selalu bagus dibandingkan fungsi *sigmoid*.

2.13 Training Model

Proses *training* digunakan untuk memperoleh bobot (*wight*) yang nantinya akan dijadikan model pada proses *training*. Model akan mencoba mempelajari pola yang ada pada datam *training* dan melakukan pada setiap *epoch*. Proses *training* akan dihentikan ketika nilai *mean square error* memenuhi target atau iterasi maksimal yang telah ditetapkan. Proses *training*, *epoch* dan *batch size* harus didefinisikan. Semakin banyak *epoch* maka semakin lama juga waktu yang dibutuhkan untuk *training* model.

2.14 Preprocessing

Preprocessing diperlukan sebelum melakukan prediksi pada data yang akan digunakan. Dalam penelitian ini terdapat dua tahapan untuk *preprocessing* data yaitu *splitting data*.

1. Splitting data

Pada *splitting data* akan dibagi menjadi dua menjadi data *training* dan *testing*.



