

## BAB II LANDASAN TEORI

### 2.1. Studi Literatur

Berikut ini adalah beberapa penelitian yang dilakukan sebelumnya terkait dengan topik yang dibahas yaitu:

Pada penelitian Khairunnisa Samosir yang diterbitkan pada Jurnal Teknologi dan Sistem Informasi (JURTEKSI) dengan judul “Prediksi Minat Konsumen Terhadap Produk Perusahaan *Direct Selling* Tianshi menggunakan Artificial Neural Network (ANN)”, Penelitian ini bertujuan untuk memprediksi minat konsumen terhadap produk Tianshi dengan tepat. Metode yang digunakan salah satu teknik *Artificial Neural Network* (ANN), yaitu Backpropagation dengan Momentum. Data penjualan yang diuji bersumber dari Stokist 319 Padang. Hasil dari penelitian ini yang dapat dengan tepat menentukan minat konsumen adalah arsitektur 5-2-1 dan 5-3-1. Sehingga penelitian ini sangat membantu sekali dalam pengadaan barang untuk meningkatkan nilai penjualan.

Pada penelitian Castaka Agus Sugianto yang diterbitkan pada Jurnal Informatika: Jurnal Pengembangan IT (JPIT) dengan judul “Prediksi Pergerakan Harga Valas menggunakan Algoritma *Neural Network*”, Hasil penelitian berupa nilai Root Mean Squared Error (RMSE) serta label tambahan angka hasil prediksi yang didapatkan setelah dilakukan validasi menggunakan sliding windows validation dengan hasil paling baik yaitu pada pengujian yang menggunakan algoritma neural network yang menggunakan windowing yaitu sebesar 0,006 dan 0,003 pada pengujian yang tidak menggunakan windowing dan untuk pengujian pada algoritma linear regression yang menggunakan windowing yaitu sebesar 0,007

dan pada pengujian yang tidak menggunakan windowing yaitu sebesar 0,004 dan setelah dilakukan pengujian T-test menunjukkan bahwa pada penelitian ini didapat hasil bahwa pengujian menggunakan neural network yang dibandingkan dengan linear regression memiliki hasil yang tidak signifikan dengan nilai T-test untuk pengujian dengan windowing sebesar 1,000 dan pengujian tanpa windowing sebesar 0,077.

Pada penelitian Syukri yang diterbitkan pada Jurnal Nasional Komputasi dan Teknologi Informasi dengan judul “Pengujian Algoritma *Artificial Neural Network* (ANN) untuk Prediksi Kecepatan Angin”. Hasil pengujian menunjukkan bahwa metode ANN Backpropagation cukup baik diterapkan untuk proses prediksi, kemampuan ANN dalam melakukan prediksi memiliki tingkat akurasi rata – rata yang lebih baik yaitu 96 %. Sedangkan nilai rata – rata kerapatan daya angin jam per harian yaitu 45.030 W/m<sup>2</sup>.

## **2.2. Prediksi**

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2013).

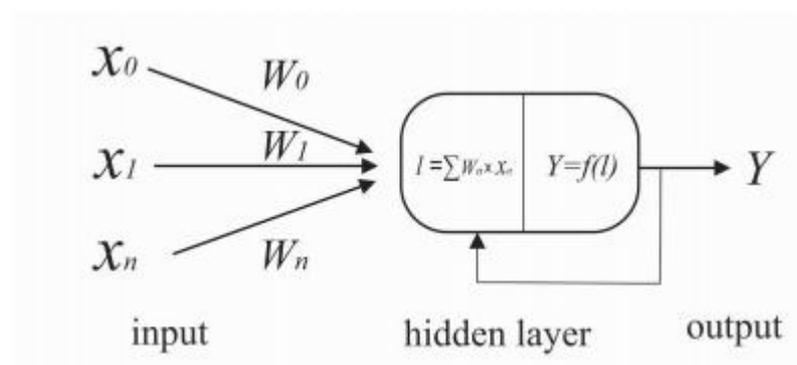
Prediksi yaitu suatu proses untuk memperkirakan suatu hal yang akan terjadi dimasa depan secara sistematis berdasarkan informasi yang didapat dari masa lalu. Prediksi dilakukan untuk mencari informasi sedekat dan semirip mungkin terhadap

hal yang akan terjadi. Dalam memprediksi ada dua hal yang harus dilakukan yaitu (Pandji, 2019):

1. Pembangunan model sebagai prototipe untuk disimpan dalam memori
2. Pembangunan model tersebut untuk pengenalan/prediksi pada suatu objek data lain agar diketahui kelas data tersebut berdasarkan model yang sudah disimpan.

### 2.3. *Artificial Neural Network* (ANN)

*Artificial Neural Network* (ANN) atau Jaringan Syaraf Tiruan (JST) merupakan suatu sistem pemrosesan informasi dengan suatu karakteristik menyerupai sistem saraf pada manusia yang dapat memecahkan masalah SVM dan KNN dengan melakukan training data yang besar dan ANN memiliki kemampuan untuk mentoleransi kesalahan sehingga dapat menghasilkan prediksi yang baik (Kamairudin, 2015). Selain itu metode ini juga dapat digunakan untuk memodelkan hubungan yang kompleks antara masukan (input) dan keluaran (output) dalam menemukan pola-pola pada data. Namun masalahnya adalah ANN memiliki kelemahan yaitu sulit untuk mengetahui berapa banyak neuron dan lapisan yang diperlukan, dan mengalami perlambatan saat learning.



Gambar 2. 1 Model Neuron Tiruan

Keterangan Gambar 2.1 di atas adalah sebagai berikut:

- a. Input merupakan masukan yang digunakan baik saat pembelajaran maupun dalam mengenali suatu objek.
- b. *Weight*, beban yang selalu berubah setiap kali diberikan input sebagai proses pembelajaran.
- c. Processing unit, tempat berlangsungnya proses pengenalan suatu objek berdasarkan pembebanan yang diberikan.

Arsitektur jaringan dan algoritma pelatihan sangat menentukan model-model ANN. Arsitektur tersebut gunanya untuk menjelaskan arah perjalanan sinyal atau data di dalam jaringan. Sedangkan algoritma belajar menjelaskan bagaimana bobot koneksi harus diubah agar pasangan masukan-keluaran yang diinginkan dapat tercapai (Syukri, 2018). Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut neuron layers. Layer-layer penyusun ANN dapat dibagi menjadi 3, yaitu:

- a. *Input layer*

*Input layer* adalah lapisan yang terdiri dari unit-unit (*neuron*) input yang langsung menerima sinyal input dari luar dan mengirim informasi input yang diterima ke setiap neuron yang ada pada *hidden layer* melalui bobot yang menghubungkan lapisan input dan lapisan tersembunyi.

- b. *Hidden layer*

*Hidden layer* adalah lapisan yang terdiri dari unit-unit (*neuron*) tersembunyi yang terletak antara *input layer* dan *output layer* dimana outputnya tidak secara langsung diamati. Penambahan *hidden layer* ini dapat meningkatkan kemampuan jaringan dalam pengenalan pola.

c. *Output layer*

*Output layer* adalah lapisan yang terdiri dari unit-unit output dimana keluaran dari lapisan *output* merupakan solusi ANN pada suatu masalah. Terdapat beberapa fungsi aktivasi, antara lain sigmoid biner, sigmoid bipolar, hiperbolik dan lain sebagainya. Penelitian ini akan menggunakan fungsi aktivasi sigmoid biner yang mana dapat dilihat persamaan 1 berikut:

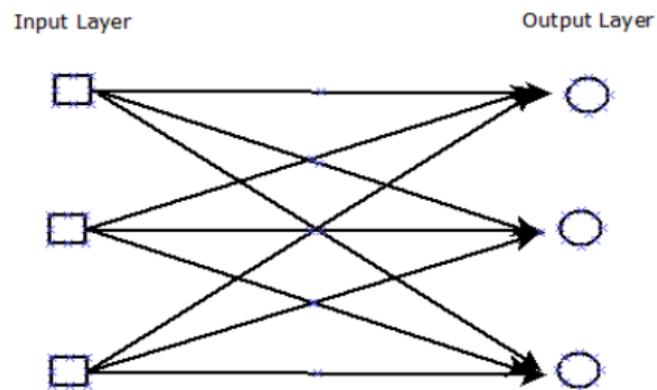
$$f_1(x) = \frac{1}{1+e^{-x}} \quad (1)$$

Dalam proses pelatihan backpropagation terdapat tiga tahap yaitu (Syukri, 2018):

- a. Tahap maju Pada tahap ini seluruh proses awal inialisasi bobot-bobot input dilakukan.
- b. Tahap mundur Pada tahap ini, nilai error ( $\delta_k$ ) yang diperoleh pada lapisan output digunakan untuk mengoreksi bobot-bobot yang ada pada lapisan tersembunyi yang berhubungan langsung dengan lapisan output.
- c. Tahap pengoreksian bobot Setelah seluruh bobot pada lapisan input dan lapisan tersembunyi dimodifikasi sesuai dengan besar faktor errornya, maka ketiga fase ini diulang secara terus menerus sampai kondisi berhenti dipenuhi.

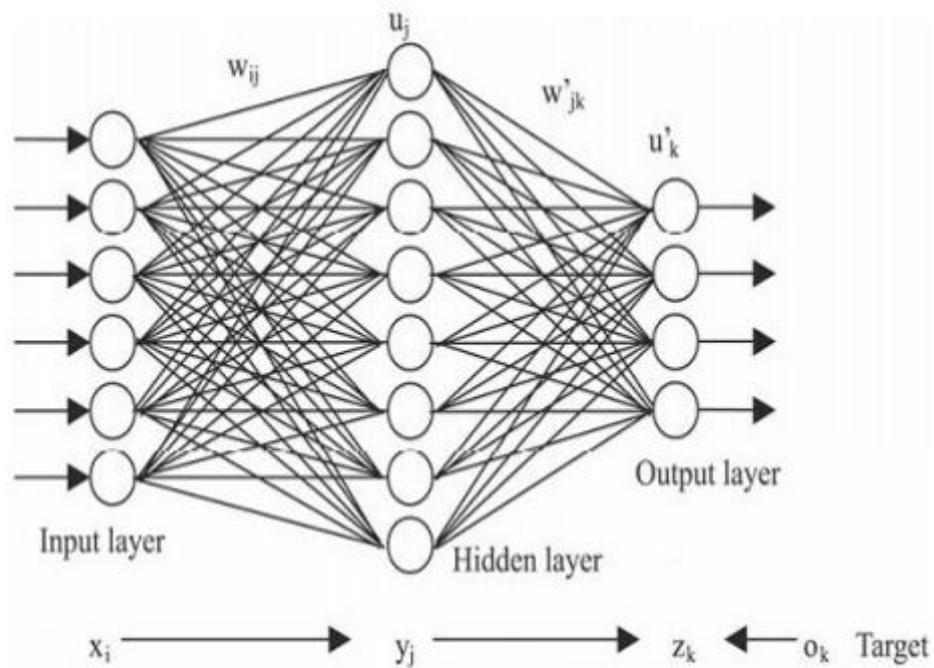
Berdasarkan jumlah Layer arsitektur jaringan Artificial Neural Network (ANN) dapat di klasifikasikan menjadi dua kelas yang berbeda yaitu :

- a. Jaringan layer tunggal (single layer network) merupakan semua unit input dalam jaringan ini dihubungkan dalam semua unit output, meskipun dengan bobot yang berbeda - beda, Berikut contoh jaringan single layer dapat di lihat pada gambar 2.2.



Gambar 2. 2 Jaringan Single Layer Network

- b. Jaringan layar jamak (multi layer network) adalah jaringan layar jamak yang merupakan perluasan dari layar tunggal jaringan layar jamak ini memperkenalkan satu atau layar tersembunyi (hidden layer) yang mempunyai simpun yang disebut neuron tersembunyi (hidden layer), berikut contoh jaringan layar jamak dapat di lihat pada gambar 2.3 (Putra, 2014).



Gambar 2. 3 Jaringan *Multilayer Network*

## 2.4. *Unified Modelling Language (UML)*

Menurut (Nugroho, 2015) menyatakan bahwa dengan adanya *UML*, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. *UML* diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini.

Menurut (A.S, Rosa dan Shalahuddin, 2013) menyatakan bahwa “*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah *system*.”

### a. Tujuan *UML*

- 1) Memberikan model yang siap pakai, bahasa permodelan *visual* yang *ekspresif* untuk mengembangkan model dan dimengerti secara umum.
- 2) Memberikan bahasa permodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- 3) Menyatukan praktek-praktek yang terdapat dalam permodelan.

### b. Diagram-diagram dalam *UML*

Ada beberapa diagram dalam *UML (Unified Modelling Language)* antara lain:

#### 1) *Use Case Diagram*

*Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Adapun simbol dari *use case diagram* antara lain :

Tabel 2. 1 Simbol-simbol *Use Case* Diagram

No	Simbol	Keterangan
1	Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2	Aktor / <i>Actor</i>  Nama Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanyadinyatakan menggunakan kata benda di awal frase nama aktor.
3	Asosiasi / <i>Association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	Ekstensi / <i>Extend</i> <<extend>>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip

		dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5	Generalisasi (  )	Hubungan generalisasi dan spesialisasi (umum- khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan <i>Include / uses</i>  <<include>>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>usecase</i> tambahan dijalankan.

## 2) Activity Diagram

*Activity Diagram* menggambarkan berbagai alur aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, dan *decision* yang mungkin terjadi, dan bagaimana alur berakhir. Adapun simbol dari *Activity Diagram* antara lain :

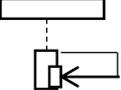
Tabel 2. 2 Simbol *Activity Diagram*

No	Gambar	Keterangan
1.	 <i>Activity</i>	Aktivitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
2.	 <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
3.	 Status Awal	Status awal aktiviatas sistem, sebuah diagram aktivitas memiliki status awal.
4.	 Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memilki sebuah status akhir.
5.	 <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

### 3) *Sequence Diagram*

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna dan *display*) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek–objek yang terkait). Setiap pesan yang dikirimkan bisa memberikan respon (*return*) relatif pada skenario yang dirancang di *Use Case diagram*. Interaksi yang terjadi bisa bersifat instansiasi sebuah objek maupun *static method* dari sebuah *class*. Berikut ini adalah simbol-simbol dari *sequence diagram*.

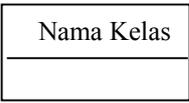
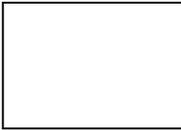
Tabel 2. 3 *Simbol Sequence Diagram*

No	Gambar	Keterangan
1.	 Aktor	Aktor adalah pengguna sistem, pengguna dapat berarti manusia, mesin atau sistem lain atau subsistem dari model apapun yang berinteraksi dengan sistem dari <i>boundary</i> sistem
2.	 <i>Lifeline</i>	Peserta individu dalam interaksi (yaitu jalur hidup yang tidak dapat memiliki multiplisitas)
3.	 <i>Message</i>	Menunjukkan aliran informasi atau kendali transaksi antar elemen
4.	 <i>Sel-Message</i>	Mencerminkan proses baru atau metode pemanggilan operasi <i>lifeline</i> . Ini adalah spesifikasi pesan biasanya dalam <i>sequence diagram</i>

#### 4) *Class Diagram*

*Class Diagram* adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan

Tabel 2. 4 *Simbol-simbol Class Diagram*

No	Gambar	Keterangan
1.		Kelas pada stuktur sistem
2.		Properti dari suatu kelas yang melambangkan batas nilai
3.	 <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	 <i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. Biasanya disertai <i>multiplicity</i>
5.	 <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
6.	 Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

7.	 <p data-bbox="676 376 794 409">Agregasi</p>	<p data-bbox="943 230 1326 409">Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)</p>
----	---	--

## 2.5. Python

Menurut Ibrahim, (2015), Python adalah bahasa pemrograman interpretatif, berorientasi objek dan semantik yang dinamis. Python memiliki high-level struktur data, dynamic typing dan dynamic binding. Python memiliki sintaks sederhana dan mudah dipelajari untuk penekanan pada kemudahan membaca dan mengurangi biaya perbaikan program. Python mendukung modul dan paket untuk mendorong kemodularan program dan code reuse. Interpreter Python dan standard library-nya tersedia secara gratis untuk semua platform dan dapat secara bebas disebar. Bahasa pemrograman ini dibuat oleh Guido van Rossum.

## 2.6. MySQL

Menurut (Syahida. Priyanto, 2015) “MySQL adalah salah satu aplikasi DBMS yang sudah banyak oleh para pemrogram aplikasi *web*. Contoh DBMS lainnya adalah: PostgreSQL (*freeware*), SQL Server, MS Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro, dsb”.

## 2.7. Pengujian Hasil

Tahap pengujian hasil akurasi dengan menggunakan nilai RMSE, dimana RMSE (*Root Mean Squared Error*) sering digunakan untuk mengevaluasi kinerja prediksi yang digunakan (Chai, 2014) dengan mengukur tingkat akurasi dari hasil prediksi. Jika nilai RMSE rendah menunjukkan bahwa bentuk perubahan nilai yang

dihasilkan oleh suatu model prediksi mendekati bentuk nilai aslinya. Dan sebaliknya, jika besar nilai RMSE maka keakuratan yang dihasilkan jauh dari bentuk nilai aslinya:

$$RMSE = \frac{\sum_i^n \sqrt{(y_i - \hat{y}_i)^2}}{n}$$

Dengan:

$y_i$  = data awal

$\hat{y}_i$  = data akhir

$n$  = jumlah data

