

BAB II

LANDASAN TEORI

2.1 Konsep Dasar

2.1.1 Pendapatan

Menurut Harnanto (2019, h. 102) pendapatan adalah kenaikan atau bertambahnya aset dan penurunan atau berkurangnya liabilitas perusahaan yang merupakan akibat dari aktivitas operasi atau pengadaan barang dan jasa kepada masyarakat atau konsumen pada khususnya.

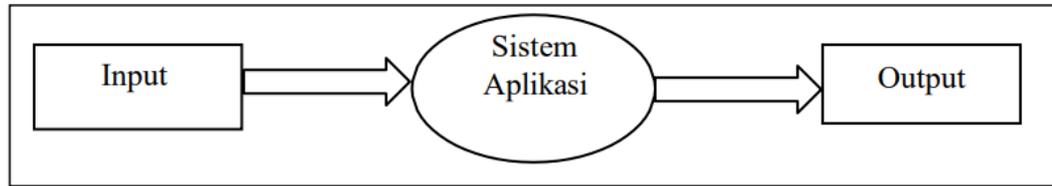
Menurut Sohib (2018, h. 47) pendapatan merupakan aliran masuk yang timbul dari penyerahan barang atau jasa yang dilakukan oleh suatu unit usaha selama periode tertentu. Pendapatan adalah arus masuk bruto dari manfaat ekonomi yang timbul akibat aktivitas normal perusahaan selama satu periode arus masuk itu mengakibatkan kenaikan modal (ekuitas) dan tidak berasal dari kontribusi penanaman modal.

2.1.2 Sistem

Menurut Nugraha dan Sofyan dalam jurnal (Yuliana, Zahrudin, dan Utari, 2016, h. 47) Suatu sistem dapat di definisikan sebagai satu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan.

Sedangkan menurut Tata Sutabri dalam jurnal (Abdurahman, Safi, dan Abdullah, 2018) pada dasarnya sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Dalam sebuah Sistem terdapat beberapa komponen dasar serta karakteristik yang mendukung suatu sistem tersebut.



Gambar 2.1 Andri Kristanto, 2008

Hubungan antar elemen-elemen yang terdapat dalam sistem menurut Andri Kristanto (2008, hal. 12), meliputi :

a) *Input* (Masukan)

Masukan (input) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud (tampak secara fisik) maupun yang tidak tampak.

b) Tujuan Sistem

Sistem yang dibuat harus memiliki tujuan (Goal). Sistem bisa memiliki hanya satu tujuan namun juga bisa memiliki lebih dari satu tujuan. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali.

c) *Output* (Keluaran)

Merupakan hasil dari input yang telah diproses oleh bagian pengolahan dan merupakan tujuan akhir sistem. Output dapat berupa informasi berguna yang dapat ditangkap oleh indera manusia, semisal berupa cetakan laporan dan informasi.

2.1.3 Algoritma

Dalam suatu program aplikasi tentunya pembuat harus menentukan terlebih dahulu mengenai input dan output yang diterima oleh aplikasi tersebut nantinya. Dari input tersebut nantinya akan diolah menjadi output yang diinginkan dan dapat mempermudah pengguna aplikasi dalam berinteraksi dengan aplikasi tersebut. Alur proses yang sistematis tentang jalannya suatu aplikasi dari awal hingga akhir disebut dengan algoritma. (Winda Swastika, 2018).

2.1.4 Time Series

Time Series adalah salah satu pemodelan dalam penelitian dinamis yang bertujuan untuk mengumpulkan dan mempelajari pengamatan deret waktu sebelumnya. Hal ini bertujuan untuk mengembangkan model yang menggambarkan struktur yang melekat dari deret tersebut. Deret waktu dibedakan menjadi analisis deret waktu dan peramalan deret waktu. Terdapat dua hal yang harus diperhatikan untuk kesempurnaan model deret waktu yang dihasilkan yaitu adanya autokorelasi dan data harus stasioner.

Menurut Hanke dan Wichren (2005) Metode *time series* adalah metode peramalan dengan menggunakan analisa pola hubungan antara variabel yang akan dipekirakan dengan variabel waktu. Peramalan suatu data *time series* perlu memperhatikan tipe atau pola data. Secara umum terdapat empat macam pola data *time series*, yaitu horizontal, trend, musiman, dan siklis.

2.1.5 Analisa *Time Series*

Analisis data time series adalah Pola seperti keacakan, tren, pergeseran level, periode atau siklus, pengamatan yang tidak biasa, atau campuran pola dapat ditemukan. Setelah itu, pola umum dalam data deret waktu dibahas, bersama dengan contoh situasi yang mendorong pola tersebut. Metode time series adalah metode peramalan dengan menggunakan analisa pola hubungan antara variabel yang akan diperkirakan dengan variabel waktu. Peramalan suatu data time series perlu memperhatikan pola data. Secara umum terdapat empat macam komponen data time series yaitu :

a) *Horizontal*

Pola horizontal merupakan kejadian yang tidak terduga dan bersifat acak, tetapi kemunculannya dapat memengaruhi keadaan yang tidak tetap pada data time series.

b) *Trend*

Pola trend merupakan kecenderungan arah data dalam jangka panjang, dapat berupa kenaikan maupun penurunan.

c) *Musiman*

Pola musiman merupakan keadaan yang tidak tetap dari data yang terjadi secara periodik dalam kurun waktu satu tahun, seperti bulanan, mingguan, atau harian.

d) Siklus

Pola siklus merupakan keadaan yang tidak tetap dari data untuk waktu yang lebih dari satu tahun.

2.1.6 Forecasting

Forecasting adalah suatu seni dan ilmu pengetahuan dalam memprediksi peristiwa pada masa mendatang. Peramalan akan melibatkan pengambilan data historis (penjualan tahun lalu) dan memproyeksi mereka ke masa yang akan datang dengan model matematika. (Heizer dan Render, 2015).

2.1.7 Perbandingan

Menurut Nazir (2005) Perbandingan adalah sejenis penelitian deskriptif yang ingin mencari jawaban secara mendasar tentang sebab-akibat, dengan menganalisis faktor-faktor penyebab terjadinya ataupun munculnya suatu fenomena tertentu. Bersifat membandingkan antara dua kelompok atau lebih dari suatu variabel tertentu.

2.2 Metode Penelitian

Menurut Subagyo yang dikutip dalam Syamsul Bahry dan Fakhry Zamzam (2015) Metode Penelitian adalah suatu cara atau jalan untuk mendapatkan kembali pemecahan terhadap segala permasalahan yang diajukan. Sedangkan menurut Priyono (2016) Metode Penelitian adalah cara melakukan sesuatu dengan menggunakan pikiran secara seksama untuk mencapai suatu tujuan.

Pengertian metode penelitian menurut Sugiyono (2017) adalah sebagai berikut: Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Berdasarkan hal tersebut terdapat empat kata kunci yang perlu diperhatikan yaitu cara ilmiah, data, tujuan dan kegunaan.

Berdasarkan pengertian di atas dapat disimpulkan bahwa metode penelitian adalah suatu cara ilmiah atau teknik yang digunakan demi memperoleh data mengenai suatu objek dari penelitian yang memiliki tujuan untuk memecahkan suatu permasalahan.

2.2.1 Metode Observasi

Menurut Esterberg dalam Sugiyono (2013, h. 231) observasi merupakan suatu proses yang tersusun dari berbagai faktor seperti biologis ataupun psikologis dari pengamatan dan ingatan seseorang.

2.2.2 Metode Wawancara

Menurut Esterberg dalam Sugiyono (2013, h. 231) wawancara merupakan pertemuan dua orang untuk bertukar informasi dan ide melalui tanya jawab, sehingga dapat dikonstruksikan makna dalam suatu topik tertentu. Tanya jawab ‘sepihak’ berarti bahwa pengumpul data yang aktif bertanya, sementara pihak yang ditanya aktif memberikan jawaban atau tanggapan. Dari definisi itu, kita juga dapat mengetahui bahwa Tanya jawab dilakukan secara sistematis, telah terencana, dan mengacu pada tujuan penelitian yang dilakukan.

2.2.3 Metode Studi Pustaka

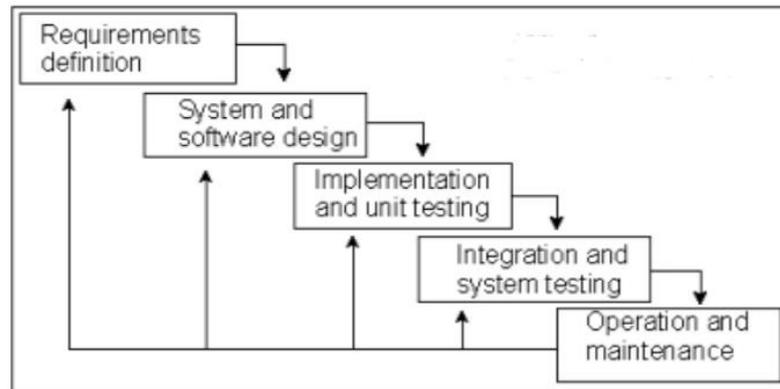
Menurut Esterberg dalam Sugiyono (2012, h. 231) metode studi pustaka adalah kajian teoritis, referensi serta literatur ilmiah lainnya yang berkaitan dengan budaya, nilai dan norma yang berkembang pada situasi sosial yang diteliti. Secara garis besar, Studi Kepustakaan yaitu mengadakan penelitian dengan cara mempelajari dan membaca literatur-literatur yang ada hubungannya dengan permasalahan yang menjadi obyek penelitian.

2.2.4 Waterfall

Dalam melakukan pengembangan sistem aplikasi sistem pengiriman barang ini, penulis menggunakan metode *waterfall*. Menurut Pressman (2015, h. 42), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah *Linear Sequential Model*.

Model ini sering disebut juga dengan *classic life cycle* atau metode *waterfall*. Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam Software Engineering (SE).

Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Dibawah ini merupakan gambar dari tahapan-tahapan metode waterfall.



Gambar 2.2 Metode Waterfall Pressman, Roger S

Berikut ini merupakan rincian penjelasan dari tahapan-tahapan metode waterfall :

1. *Requirement Definition* (Definisi Kebutuhan)

Merupakan tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and Software Design* (Desain Sistem dan Perangkat Lunak)

Dalam tahapan ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar sistem perangkat lunak dan hubungan- hubungannya.

3. *Implementation and Unit Testing* (Implementasi dan Testing Unit)

Dalam tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

4. *Integration and System Testing* (Integrasi dan Testing Sistem)

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

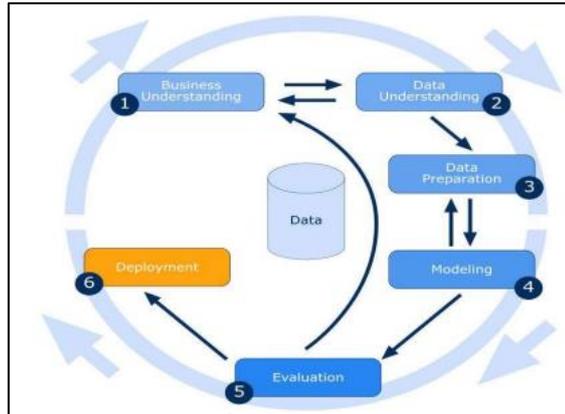
5. *Operation and Maintenance* (Operasional dan Pemeliharaan)

Dalam tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

2.2.5 CRISP-DM

Cross-Industry Standard Process for Data Mining atau CRISP-DM adalah salah satu model proses *datamining* (*datamining framework*) yang awalnya (1996) dibangun oleh 5 perusahaan yaitu Integral Solutions Ltd (ISL), Teradata, Daimler AG, NCR Corporation dan OHRA. Framework ini kemudian dikembangkan oleh ratusan organisasi dan perusahaan di Eropa untuk dijadikan *methodology standard non-proprietary* bagi *datamining*. Versi pertama dari metodologi ini dipresentasikan pada 4th CRISP-DM SIG Workshop di Brussels pada bulan Maret 1999 (Pete Chapman, 1999) dan langkah langkah proses *datamining* berdasarkan model ini di publikasikan pada tahun berikutnya (Pete Chapman, 2000).

Pada penelitian ini, digunakannya metodologi data mining CRISP-DM sebagai pemecah masalah yang umum untuk bisnis dan penelitian. Metodologi ini terdiri dari enam tahapan yaitu *Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, dan Deployment*. Proses metodologi ini terdiri dari 6 tahapan yang dapat dijelaskan sebagai berikut.



Gambar 2.3 Skema CRISP-DM

1. Business Understanding

Beberapa hal yang dilakukan pada tahap ini seperti memahami kebutuhan serta tujuan dari sudut pandang bisnis selanjutnya mengartikan pengetahuan ke dalam bentuk pendefinisian masalah pada data mining dan kemudian menentukan rencana serta strategi untuk mencapai tujuan data mining.

2. Data Understanding

Tahapan ini diawali dengan mengumpulkan data, mendeskripsikan data, serta mengevaluasi kualitas data.

3. Data Preparation

Dalam tahapan ini yaitu membangun dataset akhir dari berupa data mentah. Ada beberapa hal yang akan dilakukan mencakup melakukan pembersihan data (Data Cleaning), melakukan pemilihan data (Data Selection), record dan atribut-atribut, dan juga melakukan transformasi terhadap data (Data Transformation) untuk dijadikan masukan dalam tahap pemodelan.

4. *Modelling*

Pada tahapan ini secara langsung melibatkan Machine Learning untuk penentuan teknik data mining, alat bantu data mining serta algoritma data mining.

5. *Evaluation*

Tahap ini dilakukan dengan melihat tingkat performa dari pola yang dihasilkan oleh algoritma.

6. *Deployment*

Tahapan ini dilakukan dengan pembuatan laporan dan artikel jurnal menggunakan model yang dihasilkan.

2.2.6 *Object Oriented Programming (OOP)*

Object Oriented Programming (OOP) adalah sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan tehnik berorientasi atau berbasis pada sebuah obyek dalam pembangunan program aplikasi, maksudnya bahwa orientasi pembuatan program tidak lagi menggunakan orientasi linear melainkan berorientasi pada objek-objek yang terpisah-pisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah Obyek yang didalamnya berisi beberapa perintah-perintah standar sederhana. Obyek ini dikumpulkan dalam Modul form atau Report atau modul lain dan disusun didalam sebuah project.

2.2.7 *Metode ARIMA*

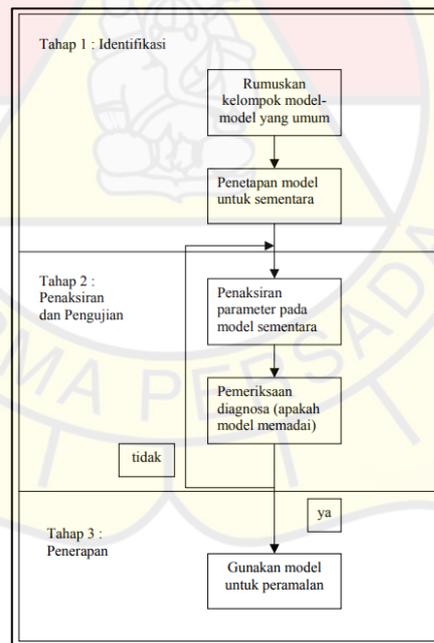
ARIMA sering juga disebut metode runtun waktu Box-Jenkins. *ARIMA* sangat baik ketepatannya untuk peramalan jangka pendek, sedangkan untuk peramalan jangka panjang ketepatan peramalannya kurang baik. Biasanya akan cenderung flat(mendatar/konstan) untuk periode yang cukup panjang.

Model *Autoregresif Integrated Moving Average (ARIMA)* adalah model yang secara penuh mengabaikan independen variabel 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).

- **Model Matematis dan Algoritma Pokok Analisis**

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 *ARIMA* :



Gambar 2.3 Skema *ARIMA*

- **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.

- **Klasifikasi Model *ARIMA***

Menurut Athanasopoulos, G. & Hyndman, R.J. (2018) Model Box-Jenkins (*ARIMA*) dibagi kedalam 3 kelompok, yaitu: model *autoregressive* (AR), *moving average* (MA), dan model campuran *ARIMA* (*autoregressive moving average*), *ARMA* (*Autoregressive and Moving Average*) yang mempunyai karakteristik dari dua model pertama.

1. Model *Autoregressive* (AR)

Model AR (p) atau *ARIMA* (p,0,0) yang ditulis dalam persamaan berikut:

$$X_t = \mu' + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + e_t$$

Keterangan :

X_t = nilai series yang stasioner

X_{t-1}, X_{t-p} = nilai lampau series

μ' = nilai konstan

ϕ_p = parameter autoregresif

e_t = White Noise (galat)

2. Model *Moving Average* (MA)

Model MA (q) atau *ARIMA* (0,0,q) yang ditulis dalam persamaan berikut:

$$X_t = \mu + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_p e_{t-q}$$

Keterangan :

X_t = nilai series yang stasioner

μ = suatu nilai konstan

θ_p = parameter moving average

e_t = White Noise (galat)

3. Model *Autoregressive and Moving Average* (ARMA)

Pada metode ARMA merupakan gabungan antara *Autoregressive* (AR) dan *Moving Average* (MA). Model ARMA (p,q) yang dinotasikan dalam *ARIMA* (p,0,q). Bentuk umum model ARMA adalah

$$X_t = \mu + \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + u_t + \theta_1 e_{t-1} + \dots + \theta_q e_{t-q}$$

Keterangan :

X_t = nilai series yang stasioner

X_{t-1}, X_{t-p} = nilai lampau series

μ = suatu nilai konstan

ϕ_p = parameter autoregresif

e_t = White Noise (galat)

θ_q = parameter moving average

4. Model Autoregressive Integrated Moving Average (ARIMA)

Model ARIMA (p,d,q) merupakan model umum dari deret waktu sebab ARIMA (p,0,0) sama dengan AR (p), ARIMA (0,0,q) sama dengan MA (p) dan ARIMA (p,0,q) sama dengan ARMA (p,q). Bentuk umum model ARIMA :

$$X_t^* = \mu + \phi_1 X_{t-1}^* + \dots + \phi_p X_{t-p}^* + u_t + \theta_1 e_{t-1} + \dots + \theta_q e_{t-q}$$

Keterangan :

X_t^* = nilai series yang stasioner

X_{t-1}^*, X_{t-p}^* = nilai lampau series

μ = suatu nilai konstan

ϕ_p = parameter autoregresif

e_t = White Noise (galat)

θ_q = parameter moving average.

2.2.8 Metode *Double Moving Average*

Menurut Rizky Yudaruddin (2019, h. 29) *Double Moving Average* adalah teknik rata-rata bergerak dua kali, sekali ke data asli dan kemudian ke data rata-rata bergerak tunggal yang dihasilkan. Metode ini kemudian menggunakan kedua set data yang dihaluskan untuk memproyeksikan ke depan.

Dalam upaya untuk memperhitungkan tren, musiman, atau pola siklus dalam data, anda harus mencari metodologi perkiraan yang akan mempertimbangkan faktor-faktor ini. Metode rata-rata bergerak ganda (*Double Moving Average*) digunakan ketika data deret waktu memiliki trend linier.

Dalam metodologi ini, anda menghitung satu set *moving average* (MA), dan kemudian set kedua dihitung sebagai rata-rata bergerak dari set pertama (MA). Artinya rata-rata bergerak kedua merupakan hasil dari rata-rata bergerak sebelumnya dan seterusnya. Adapun langkah-langkah dalam peramalan dengan menggunakan metode *Double Moving Average* adalah sebagai berikut:

1. Menghitung moving average/rata-rata bergerak pertama, diberisimbol St^I . ini dihitung dari data historis yang ada. Hasilnya diletakkan pada periode terakhir moving average pertama.
2. Menghitung moving average/rata-rata bergerak kedua, diberi simbol St^{II} . Ini dihitung rata-rata bergerak pertama. Hasilnya diletakkan pada periode terakhir moving average kedua.

3. Menentukan besarnya nilai a_t (konstanta)

$$a_t = S_t^I + (S_t^I - S_t^{II})$$

4. Menentukan besarnya nilai b_t (slope)

$$b_t = \frac{2}{v-1}(S_t^I - S_t^{II})$$

V adalah jangka waktu moving average.

5. Menentukan besarnya forecast

$$F_t + m = a + b(m)$$

Di mana, M adalah jangka waktu Forecast ke depan

2.3. *Differensiasi*

Pemodelan ARMA memiliki teori dasar korelasi dan stasioneritas. Maksudnya ARMA dapat digunakan ketika deret waktu telah membentuk grafik yang stasioner, atau tidak membentuk *trend* naik maupun turun. Namun bila data deret waktu tidak stasioner, maka perlu dilakukan proses *differensiasi* untuk mengubah data hingga menjadi stasioner dahulu sebelum dapat diproses melalui ARMA. Data yang telah di *diferensiasi* lalu diolah dengan ARMA ini disebut dengan ARIMA dengan parameter ARIMA (p,d,q) dengan d menunjukkan jumlah proses *differensiasi* yang dilakukan.

2.4. Aplikasi Berbasis Web

Aplikasi berbasis *web* yang digunakan para *programmers* atau pengembang aplikasi berbasis *web*. Fitur-fitur aplikasi berbasis *web* biasanya berupa data persistence, mendukung transaksi dan komposisi halaman *web* dinamis yang dapat dipertimbangkan sebagai hibridisasi antara hypermedia dan sistem informasi.

Aplikasi berbasis *web* dapat dijalankan apabila tersedia minimal dua *software* utama, yaitu *web server* dan *web browser* (Sunyoto, 2007). Aplikasi berbasis *web* merupakan aplikasi yang terletak pada server, yang merupakan perpanjangan dari *web server* untuk mengirim dan memproses informasi yang dikirim ke browser menjadi dinamis dengan adanya aplikasi tersebut (Supriyanto dan Muhsin, 2008).

2.4.1. Website

Pengertian *website* menurut Sebok, Vermat, dan tim (2018 : 70) adalah kumpulan halaman yang saling terhubung yang di dalamnya terdapat beberapa item seperti dokumen dan gambar yang tersimpan di dalam *web server*. *Web app* adalah sebuah aplikasi yang berada dalam *web server* yang bisa user akses melalui browser, *Web app* biasanya menampilkan data user dan informasi dari server.

2.4.2. PHP

Menurut Supono & Putratama (2018, hal. 1) mengemukakan bahwa *hypertext preprocessor (PHP)* adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang ditambahkan ke *HTML*.

Menurut Solichin (2016) mengemukakan bahwa *PHP* merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk pengembang *web*. Kumpulan kutipan diatas menerangkan bahwa *hypertext preprocessor (PHP)* merupakan bahasa pemrograman untuk membuat/mengembangkan aplikasi berbasis *web* dan bersifat *open source* dan ditanamkan ke dalam *script HTML*.

2.4.3. Javascript

Javascript sering digunakan di dalam aplikasi berbasis *web* untuk memberikan pesan menarik kepada user dan tampilan pendukung yang ada pada *website*. Menurut Wahana Komputer (2010) “*JavaScript* merupakan bahasa yang berbentuk kumpulan skrip yang berfungsi untuk memberikan tampilan yang tampak lebih interaktif pada dokumen *web*”.

Menurut Sunyoto (2007) “*JavaScript* adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar *browser* populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape, dan Opera”. *JavaScript* dikembangkan oleh Netscape dengan nama awal LiveScript yang berfokus pada proses pengolahan data di sisi client dan menyajikan komponen *web* yang lebih interaktif serta berfungsi untuk menambah fungsionalitas dan kenyamanan halaman *web* (Solichin, 2016).

Dari pendapat para ahli di atas, maka javascript merupakan bahasa pemrograman untuk aplikasi berbasis *web* yang berbasis client dan script untuk tampilan pendukung pada *website* sehingga membuat halaman *web* bisa melakukan tugas tambahan yang tidak dilakukan pada *script HTML*.

2.4.4. Cascading Style Sheet (CSS)

Menurut (Wahyudi, 2017), CSS adalah suatu Bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* akan lebih rapi, terstruktur, dan seragam.

Menurut Sulistiyawan, dkk (2008) mengemukakan bahwa “*cascading style sheet* adalah suatu bahasa stylesheet yang digunakan untuk mengatur *style* suatu dokumen. Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa *HTML* dan *XHTML*”. CSS atau *cascading style sheet* bahasa pemrograman yang diusulkan oleh Hakon Wilum Lie pada tahun 1994 dan distandarisasi oleh W3C yang berfungsi untuk mempercantik tampilan *web* (Solichin, 2016).

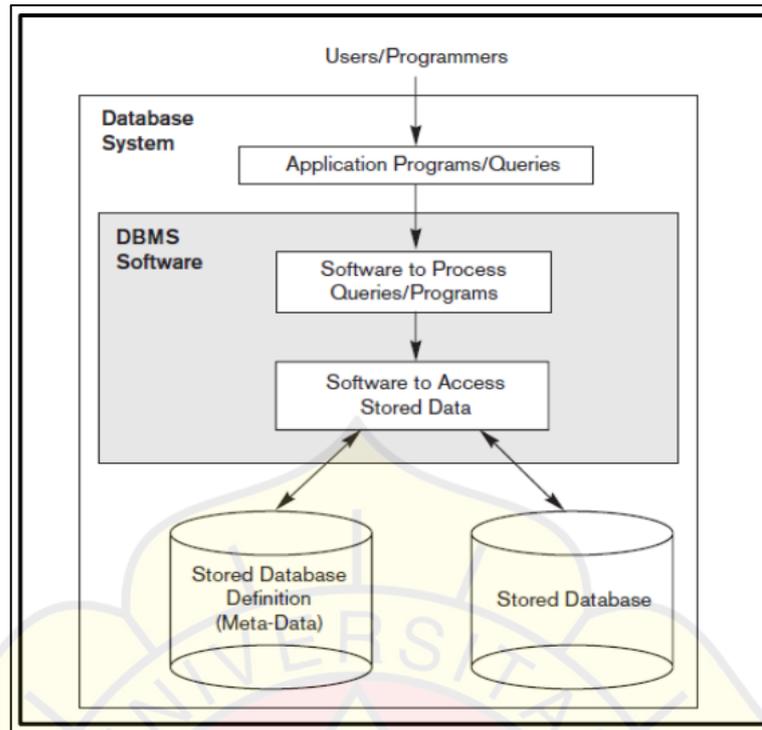
Maka dari itu, *cascading Style Sheet* (CSS) merupakan bahasa pemrograman *web* yang digunakan untuk mengatur konten dalam sebuah halaman *web* yang ditulis dalam bahasa markup agar halaman *web* tersebut lebih menarik dan terstruktur.

2.5. Basis Data

Basis data dan teknologi basis data memiliki dampak signifikan terhadap peningkatan jumlah pengguna komputer. Tidak berlebihan untuk mengatakan bahwa *database* memainkan peran penting di hampir setiap bidang penggunaan komputer, termasuk bisnis, *e-commerce*, teknik, kedokteran, genetika, hukum, pendidikan, dan ilmu perpustakaan. Kata basis data sangat umum digunakan, maka harus dimulai dengan mendefinisikan apa arti dari basis data itu sendiri. Menurut Ramez Elmasri, Shamkant B. Navathe dalam *Fundamentals of Basis data Systems 6th edition* (2011) Basis data adalah kumpulan data yang terkait. Dengan data, fakta yang diketahui, direkam dan memiliki makna yang implisit.

2.5.1. *Database Management System*

Menurut Ramez Elmasri, Shamkant B. Navathe dalam *Fundamentals of Basis data Systems 6th edition* (2011, hal. 5) *Basis data Management System* (DBMS) adalah kumpulan program yang memungkinkan pengguna untuk membuat dan memelihara basis data. DBMS adalah sistem perangkat lunak tujuan umum yang memfasilitasi proses mendefinisikan, membangun, memanipulasi, dan berbagi basis data di antara berbagai pengguna dan aplikasi. Mendefinisikan sebuah basis data melibatkan menentukan tipe data, struktur, dan batasan data yang akan disimpan dalam basis data. Definisi basis data atau informasi deskriptif juga disimpan oleh DBMS dalam bentuk katalog atau kamus basis data itu disebut *meta-data*. Membangun basis data adalah proses penyimpanan data pada beberapa medium penyimpanan yang dikendalikan oleh DBMS.



Gambar 2.4 Skema Sistem Basis Data Elmasri & B.Navathe, 2010

2.5.2. *SQL*

Bahasa *SQL* dapat dianggap sebagai salah satu alasan utama keberhasilan komersial database relasional. Karena menjadi standar untuk basis data relasional, pengguna tidak terlalu khawatir tentang migrasi aplikasi basis data dari jenis sistem basis data lain, seperti jaringan dan sistem hierarkis, ke basis data relasional. Hal ini karena jika pengguna tidak puas dengan produk *DBMS relasional* tertentu yang mereka gunakan, akan memakan biaya dan waktu untuk mengalihkan pengguna ke produk *DBMS relasional* lain karena kedua sistem mematuhi standar bahasa yang sama.

SQL adalah kepanjangan dari *Structured Query Language* yang awalnya disebut *SEQUEL (Structured English Query Language)* yang didesain dan diimplementasi pada IBM Research sebagai tampilan antar muka eksperimental

sistem basis data relasional yaitu SYSTEM R.SQL yang lebih dikenal sekarang sebagai komersial *DBMS relasional*. *SQL* adalah bahasa basis data yang komprehensif dengan memiliki pernyataan untuk mendefinisi, queries, dan pembaruan data.

2.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) Berdasarkan jurnal Dini Agustia Tri Suci, dkk menurut (Ginting, 2013) mengungkapkan *Unified Modeling Language (UML)* bukanlah suatu proses melainkan bahasa pemodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh artifak sistem perangkat lunak. Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain di luarnya.

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan dari sebuah sistem pengembangan *software berbasis object oriented*. (Mamed Rofendy Manalu, 2015) Dari Pengertian diatas penulis menyimpulkan bahwa *Unified Modeling Language (UML)* merupakan bahasa pemodelan yang berbentuk grafis yang digunakan untuk memvisualisasi, menspesifikasikan suatu sistem perangkat lunak.

2.6.1. Use case Diagram

Use case Diagram adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan. (Mamed Rofendy Manalu, 2015)

Menurut Shalahuddin dalam jurnal Umar Al Faruq (2015) mengungkapkan Diagram *use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.” Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1 *Use case* Diagram

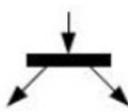
Simbol	Nama	Keterangan
	Aktor	Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan.
	<i>Use case</i>	Perangkat tertinggi dari fungsional yang dimiliki sistem.
	Relasi Asosiasi	Relasi yang terjadi antara actor dengan <i>use case</i> biasanya berupa asosiasi.

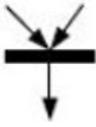
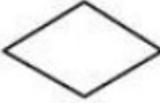
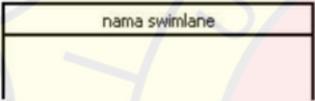
	<p>Include Relationship</p>	<p>Relasi cakupan memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.</p>
	<p>Extends Relationship</p>	<p>Memungkinkan suatu <i>use case</i> memiliki kemungkinan untuk memperluas fungsional yang disediakan <i>use case</i> yang lainnya.</p>

2.6.2. Activity Diagram

Menurut Rosa dalam jurnal Sari dan David mengungkapkan *Activity Diagram* menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem. Simbol-simbol yang digunakan dalam *activity diagram* sebagai berikut :

Tabel 2.2 Activity Diagram

Simbol	Nama	Keterangan
	Start Point	Awal Aktivitas.
	End Point	Akhir Aktivitas.
	Activities	Activities menjelaskan suatu proses/kegiatan.
	Fork (Percabangan)	Fork (percabangan) digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

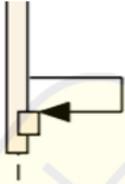
	<p>Join (Penggabungan)</p>	<p>Join (penggabungan) digunakan untuk menunjukkan kegiatan yang digabungkan.</p>
	<p>Decision Point</p>	<p>Decision Points menggambarkan pilihan untuk pengambilan keputusan true atau false.</p>
	<p>Swimlane</p>	<p>Swimlane pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

2.6.3. Sequence Diagram

Sequence Diagram adalah tool yang sangat populer dalam pengembangan sistem informasi secara object-oriented untuk menampilkan interaksi antar objek. (Nofriyadi Jurdam, 2014) Berdasarkan definisi tersebut, dapat disimpulkan bahwa *Sequence* Diagram adalah tool yang digunakan dalam pengembangan sistem.

Tabel 2.3 *Sequence Diagram*

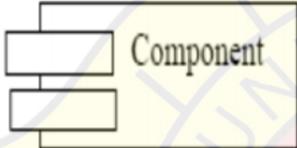
Simbol	Nama	Keterangan
	Entity Class	Entity Class merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas- entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk basis data
	Boundary Class	Boundary Class berisi kumpulan kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.
	Control Class	Control class suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas contohnya adalah kalkulasi dan

		<p>aturan bisnis yang melibatkan berbagai objek.</p>
	<p>Message</p>	<p>Message simbol mengirim pesan antar class.</p>
	<p>Recursive</p>	<p>Recursive menjelaskan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p>Activation</p>	<p>Activation mewakili sebuah eksekusi operasi dari objek panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.</p>
	<p>Lifeline</p>	<p>Lifeline garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.</p>

2.6.4. Deployment Diagram

Deployment diagram menggambarkan detail bagaimana komponen di deploy dalam infrastruktur system, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server dan hal-hal lain yang bersifat fisik.

Tabel 2.4 *Deployment Diagram*

Simbol	Nama	Keterangan
	Component	Pada deployment diagram komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.
	Node	Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.

	Association	Sebuah association digambarkan sebagai sebuah garis yang menghubungkan jalur komunikasi antara elemen - elemen hardware.
---	-------------	--

2.7. *Entity Relationship Diagram (ERD)*

Meskipun banyak implementasi *database* telah dikembangkan, model relasional adalah yang paling umum dan pilihan pertama untuk sebagian besar sistem perangkat lunak . Rekaman berhubungan satu sama lain menggunakan bidang kunci yang merupakan sub kumpulan bidang tabel. Untuk merancang skema relasional yang baik, (yaitu, tabel, fields, dan keys), Seseorang perlu memahami persyaratan system. Skenario interaksi sistem pengguna yang menentukan input dan output data. Model data kemudian diekstraksi dari persyaratan ini, untuk mendukung fungsionalitas sistem yang ditentukan.

Menurut (Sukamto & Shalahuddin, 2018:50) menyatakan bahwa “ERD digunakan untuk permodelan basis data relasional”.