

## **BAB 3**

### **METODE PENELITIAN**

#### **3.1 BAHAN DAN ALAT PENELITIAN**

Bahan penelitian ini terdiri dari objek penelitian dan data penelitian yang terdiri dari :

1. Objek penelitian

Objek penelitian yang digunakan dalam penelitian kali ini adalah toko Vetza Jeans.

2. Data penelitian

Data penelitian yang digunakan dalam penelitian ini adalah data penjualan jeans pada toko Vetza Jeans selama tiga sampai lima tahun yang berisi tanggal, harga, total jeans, serta data lainnya yang mendukung proses penelitian ini.

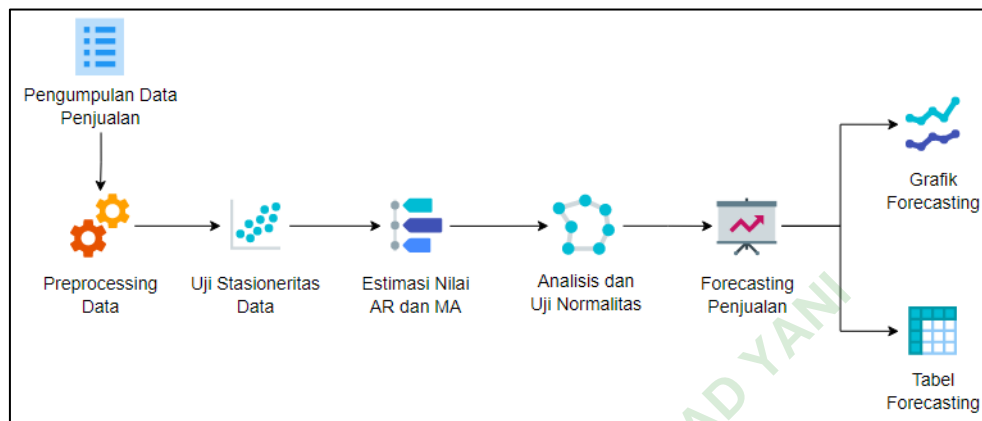
Alat yang digunakan dalam penelitian ini adalah komputer dengan spesifikasi yang cukup untuk menjalankan sistem operasi dan perangkat lunak, serta koneksi internet yang lancar. Sistem operasi dan perangkat lunak yang digunakan untuk mengembangkan aplikasi ini adalah :

1. Sistem Operasi: Windows 8
2. Bahasa pemrograman : Python 3.8
3. *Framework* : Django 3.0.6
4. Aplikasi web Jupyter Notebook
5. *Code editor* : Sublime
6. Google Chrome sebagai web browser

#### **3.2 JALAN PENELITIAN**

Jalan penelitian yang dilakukan pertama kali adalah melakukan pengumpulan data penjualan produk jeans di toko Vetza Jeans. Setelah selesai mengumpulkan data – data yang dibutuhkan, data akan dilakukan diproses terlebih dahulu untuk memudahkan dalam melakukan *forecasting*. Kemudian, dalam melakukan *forecasting*, hal yang pertama dilakukan adalah uji stasioneritas data, setelah selesai

melakukan uji stasioneritas data yaitu melakukan estimasi nilai AR dan MA. Setelah mendapatkan nilai AR dan MA tahap terakhir yang dilakukan adalah melakukan *forecasting* penjualan. Tahap – tahap tersebut terlihat pada Gambar 3.1.



**Gambar 3.1** Jalan Penelitian

### 3.2.1 Pengumpulan Data Penjualan Produk

Tahap pertama adalah pengumpulan data, dalam pengumpulan data sendiri ada riset lapangan dan studi kepustakaan.

#### 1. Riset lapangan

Pertama adalah observasi lapangan, kegiatan ini adalah melakukan pengamatan penjualan pada toko Vetza Jeans secara langsung. Kedua adalah melakukan wawancara terhadap pemilik dan pegawai toko mengenai proses penjualan jeans yang terjadi.

#### 2. Studi kepustakaan

Kegiatan ini adalah melakukan penelitian dengan menggunakan sumber lain yang berasal dari penelitian – penelitian terdahulu yang sudah pernah dilakukan oleh instansi atau peneliti lain.

Data yang digunakan dalam *forecasting* ini adalah data penjualan produk jeans yang berasal dari toko Vetza Jeans. Data yang didapatkan ini merupakan data *time series* yang dimulai dari tahun 2018 hingga 2022 yang berisi tanggal dan total penjualan.

Pada Tabel 3.1 menunjukkan pengelompokan total dihitung perbulan selama kurang lebih 4 (empat) tahun.

**Tabel 3.1** Data Penjualan

No	Tahun	Tanggal	Total
1	2018	2018-04-30	295
2		2018-05-31	323
3		2018-06-30	460
4		2018-07-31	366
5		2018-08-31	250
6		2018-09-30	263
7		2018-10-31	256
8		2018-11-30	248
9		2018-12-31	370
10	2019	2019-01-31	258
11		2019-02-28	302
12		2019-03-31	335
13		2019-04-30	454
14		2019-05-31	372
15		2019-06-30	331
16		2019-07-31	492
17		2019-08-31	315
18		2019-09-30	330
19		2019-10-31	386
20		2019-11-30	234
21		2019-12-31	330
22	2020	2020-01-31	261
23		2020-02-29	210
24		2020-03-31	332
25		2020-04-30	354
26		2020-05-31	343
27		2020-06-30	331
28		2020-07-31	275
29		2020-08-31	245
30		2020-09-30	354
31		2020-10-31	340

32		2020-11-30	260
33		2020-12-31	241
34	2021	2021-01-31	370
35		2021-02-28	308
36		2021-03-31	353
37		2021-04-30	466
38		2021-05-31	434
39		2021-06-30	500
40		2021-07-31	477
41		2021-08-31	383
42		2021-09-30	299
43		2021-10-31	380
44		2021-11-30	388
45		2021-12-31	472
46	2022	2022-01-31	490
47		2022-02-28	350
48		2022-03-31	293
49		2022-04-30	275
50		2022-05-31	430

### 3.2.2 Preprocessing Data

Setelah itu akan dilakukan pengambilan data yang dibutuhkan dalam penelitian ini, yaitu data penjualan jeans selama tiga sampai empat tahun. Setelah data diambil, data akan diolah dulu atau dilakukan *preprocessing* data untuk menghilangkan data yang sekiranya tidak diperlukan. Hasil data tersebut bisa digunakan sebagai bahan *forecasting* penjualan jeans.

Mulai dari proses *preprocessing* ini akan dilakukan di *software* Jupyter Notebook, pada tahap ini dilakukan *preprocessing* untuk mengubah format pada kolom “Bulan” menjadi datetime dan melakukan plot pada data seperti pada kode di bawah.

```
data['Bulan'] = pd.to_datetime(data['Bulan'], format='%Y-%m-%d')
data.set_index(['Bulan'], inplace=True)
data.plot
plt.ylabel('Total Penjualan')
```

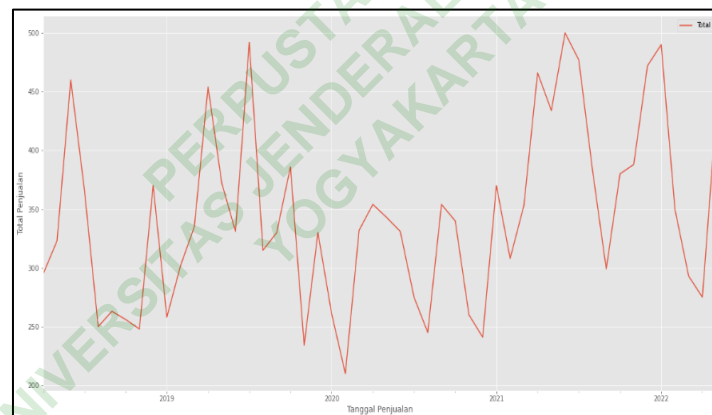
```
plt.xlabel('Tanggal Penjualan')
plt.show()
```

Setelah mengubahnya menjadi datetime, tampilan grafik data yang diperoleh seperti pada Gambar 3.2. Pada grafik tersebut sumbu y menunjukkan total penjualan dan sumbu x menunjukkan tanggal penjualan.

### 3.2.3 Uji Stasioneritas Data

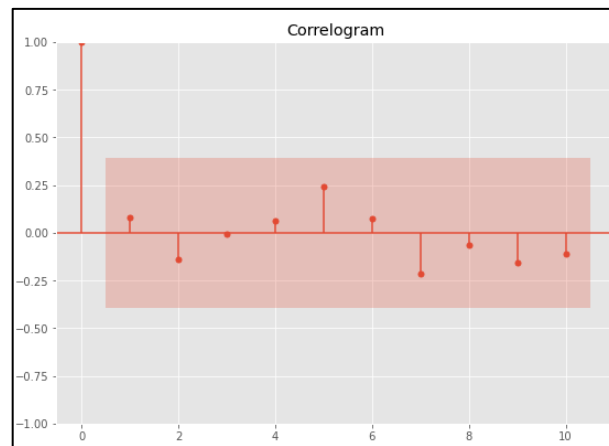
Setelah melakukan plot data, akan terlihat apakah data tersebut stasioner atau tidak. Cara menguji stasioneritas data dapat dilihat melalui grafik data maupun correlogram.

Seperti pada Gambar 3.2 bagian *preprocessing* data, grafik data tersebut menunjukkan data stasioner yang artinya data tersebut konstan dan tidak mengalami kenaikan atau penurunan sepanjang waktu.



**Gambar 3.2** Grafik Data Penjualan

Selain menggunakan grafik data, uji stasioneritas bisa menggunakan correlogram seperti pada Gambar 3.3. Pada gambar tersebut menunjukkan data stasioner karena perubahan titik data terlihat jelas yang artinya data ini bisa digunakan untuk melakukan *forecasting* secara time series.



**Gambar 3.3** Correlogram

### 3.2.4 Estimasi Nilai AR Dan MA

Pertama adalah mengatur model awal menggunakan model baru *Seasonal* ARIMA seperti pada kode di bawah. Dalam estimasi nilai AR dan MA pada ARIMA terdapat integer yang sering digunakan yaitu p, d, q. Integer p digunakan untuk nomor AR, integer d digunakan untuk nilai *differencing*, integer q digunakan untuk nomor MA.

```
q = d = range(0, 2)
p = range(0, 4)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in
    list(itertools.product(p, d, q))]
print('Examples parameters combination for ARIMA : ')
print('ARIMA: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('ARIMA: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('ARIMA: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('ARIMA: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```

Selanjutnya adalah pembagian data menjadi *data train* dan *data test* dengan kode seperti di bawah

```
train_data = data['2018-04-30':'2021-04-30']
test_data = data['2021-05-31':'2022-05-31']
```

Pada *forecasting* dengan metode ARIMA, proses ini juga termasuk mengidentifikasi pola ACF (pada nilai p atau AR) dan PACF (pada nilai q atau

MA). Pada kode di bawah menunjukkan kode yang harus digunakan untuk mendapatkan nilai p, d, q.

```
warnings.filterwarnings("ignore")
AIC = []
SARIMAX_model = []
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(train_data, order=param,
                seasonal_order=param_seasonal, enforce_stationarity=False,
                enforce_invertibility=False)
            results = mod.fit()
            print('ARIMA{ }x{ } - AIC:{ }'.format(param, param_seasonal,
                results.aic), end='\r')
            AIC.append(results.aic)
            SARIMAX_model.append([param, param_seasonal])
        except:
            continue
```

Hasil yang diperoleh setelah menjalankan kode tersebut terlihat pada Gambar 3.4 dengan nilai p yaitu 3, nilai d yaitu 1 dan terakhir nilai q yaitu 1.

ARIMA(3, 1, 1)x(3, 1, 1, 12) - AIC:18.046267380207642

**Gambar 3.4** Hasil AR dan MA

Setelah mendapat sebuah model sebagai patokan, selanjutnya adalah mencari model dengan nilai AIC paling kecil dengan kode seperti pada kode di bawah. Pada proses pembuatan model ARIMA selanjutnya, model yang akan digunakan adalah model yang memiliki hasil *forecasting* terbaik dengan akurasi yang paling tinggi.

```
print('The smallest AIC is { } for model ARIMA{ }x{ }'.
    format(min(AIC), SARIMAX_model[AIC.index(min(AIC))][0],
    SARIMAX_model[AIC.index(min(AIC))][1]))
```

### 3.2.5 Analisis Uji Normalitas

Tahap selanjutnya adalah melakukan analisis dan uji normalitas. Setelah menentukan nilai AR dan MA (p, d, q), selanjutnya adalah menjadikannya sebuah model ARIMA dengan kode seperti pada kode di bawah.

```
mod = sm.tsa.statespace.SARIMAX(train_data,
    order = SARIMAX_model[AIC.index(min(AIC))][0],
```

```
seasonal_order = SARIMAX_model[AIC.index(min(AIC))][1],
enforce_invertibility = False)
results = mod.fit()
```

Setelah selesai, selanjutnya menggunakan plot untuk menampilkan grafik yang akan digunakan untuk uji normalitas dengan kode seperti pada kode di bawah.

```
results.plot_diagnostics(figsize=(20, 14))
plt.show()
```

### 3.2.6 Forecasting Penjualan Produk

Hal pertama yang dilakukan dalam *forecasting* adalah melakukan *forecasting* dengan menggunakan data test seperti pada kode di bawah, gunanya adalah untuk memperkirakan titik atau patokan *forecasting* selanjutnya.

```
pred0 = results.get_prediction(Date='2019-05-31', dynamic=False)
pred0_ci = pred0.conf_int()
```

Selanjutnya melakukan *forecasting* dengan data yang sama namun dengan menggunakan *dynamic = True*, yang artinya *forecasting* dilakukan setelah tanggal sebelumnya dilakukan sampai seterusnya dengan menggunakan kode seperti pada kode di bawah.

```
pred1 = results.get_prediction(Date='2021-05-31', dynamic=True)
pred1_ci = pred1.conf_int()
```

Terakhir adalah melakukan *forecasting* untuk data yang baru yaitu data mulai dari bulan Juni 2021 hingga bulan Mei 2023 dengan kode seperti pada kode di bawah.

```
pred2 = results.get_forecast('2023-05-31')
pred2_ci = pred2.conf_int()
print(pred2.predicted_mean['2022-04-30':'2023-05-31'])
```

Setelah selesai melakukan *forecasting*, hasilnya akan ditampilkan dalam bentuk grafik dengan kode seperti pada kode di bawah agar lebih mudah jika akan dilakukan analisis lebih lanjut.



```

ax = data.plot(figsize=(20, 16))
pred0.predicted_mean.plot(ax=ax, label='Prediksi Data Training
(dynamic=False)')
pred1.predicted_mean.plot(ax=ax, label='Prediksi Data Testing
(dynamic=True)')
pred2.predicted_mean.plot(ax=ax, label='Prediksi - 2023
(get_forecast)')
ax.fill_between(pred2_ci.index, pred2_ci.iloc[:, 0],
pred2_ci.iloc[:, 1], color='k', alpha=.1)
plt.ylabel('Jumlah Penjualan')
plt.xlabel('Tanggal')
plt.legend()
plt.show()

```

Tahap paling akhir dalam *forecasting* adalah melakukan perhitungan *Mean Absolute Percentage Error* (MAPE). Bisa diartikan sebagai perhitungan tingkat *error* dalam melakukan *forecasting*, kode yang digunakan seperti pada kode di bawah.

```

prediction = pred2.predicted_mean['2023-05-01':'2023-5-31'].values
truth=list(itertools.chain.from_iterable(test_data.values))
MAPE = np.mean(np.abs((truth - prediction) / truth)) * 100
print('The Mean Absolute Percentage Error for the forecast of year
2023 is {:.2f}%'.format(MAPE))

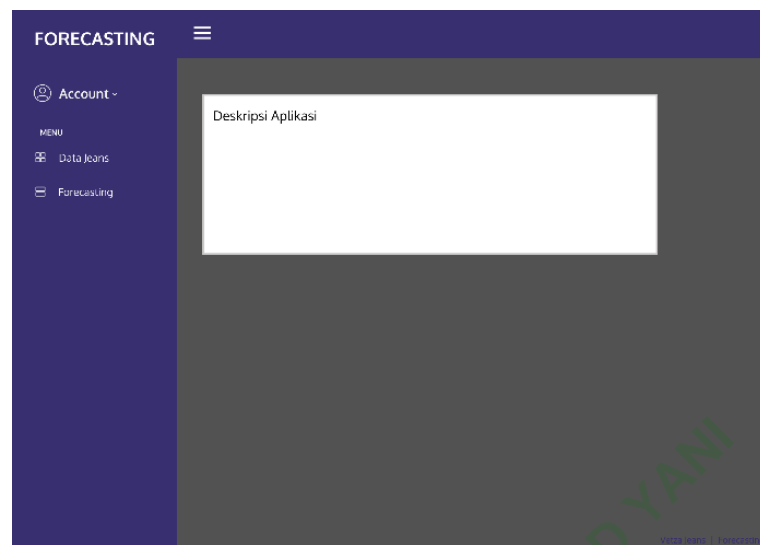
```

### 3.3 DESAIN ANTARMUKA

Desain antarmuka merupakan rancangan tampilan website yang nantinya akan digunakan atau diterapkan pada sistem *forecasting* ini. Berikut adalah rancangan antarmuka yang sudah dibuat yang menunjukkan beberapa menu utama pada sistem ini.

#### 3.3.1 Halaman *Home*

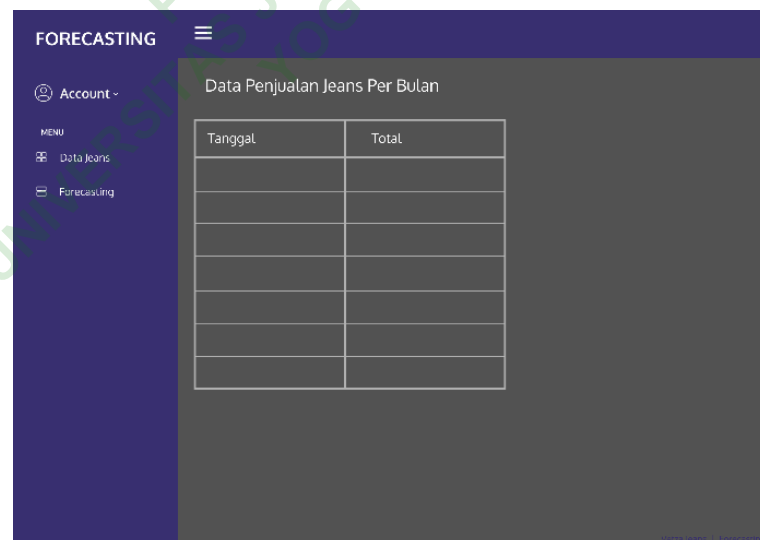
Menu *Home* adalah tampilan *dashboard* pertama. Menu ini digunakan untuk memberikan petunjuk kepada pengguna agar mengerti bagaimana cara menggunakan sistem ini dengan benar. Rancangan antarmuka menu ini ada pada Gambar 3.5.



**Gambar 3.5** Desain Halaman Home

### 3.3.2 Halaman Data Penjualan

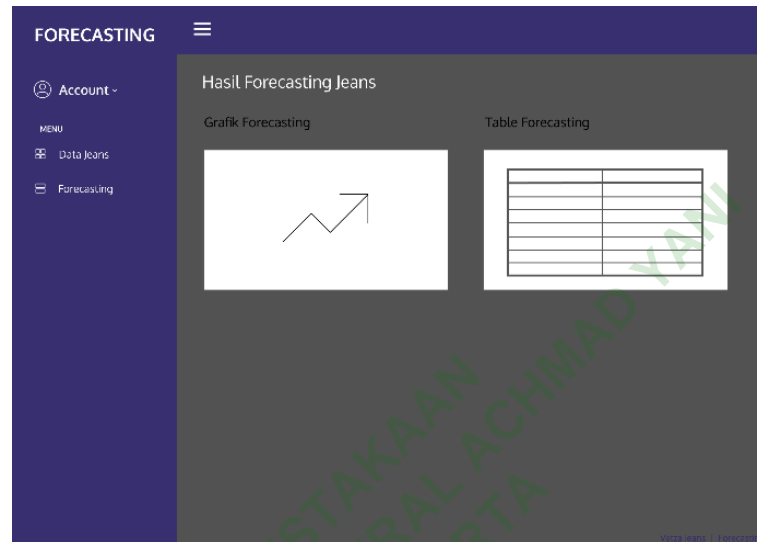
Menu data berisi tabel yang digunakan untuk menampilkan data yang penjualan yang nantinya akan digunakan untuk melakukan *forecasting*. Adapun rancangan antarmukanya pada Gambar 3.6.



**Gambar 3.6** Desain Halaman Data Penjualan

### 3.3.3 Halaman Hasil *Forecasting*

Pada menu ini nantinya akan terdapat grafik dan tabel hasil *forecasting* yang sudah dilakukan pada software aplikasi Jupyter Notebook, tampilan rancangan desain menu *forecasting* terlihat seperti Gambar 3.7.



**Gambar 3.7** Desain Halaman Hasil Forecasting