

BAB III. LANDASAN TEORI

3.1. Sistem Pendukung Keputusan

Pengambilan keputusan adalah proses memilih tindakan dari berbagai alternatif yang ada, untuk mencapai tujuan tertentu. Setiap harinya seseorang harus mengambil berbagai keputusan. Mulai dari memutuskan untuk bangun tidur pada pukul tertentu, hingga tidur kembali pada pukul tertentu. Hal ini juga terjadi pada pihak manajemen (dalam penelitian ini adalah penjual). Penjual harus mengambil berbagai keputusan salah satunya adalah keputusan dalam pengadaan stok.

Sistem Pendukung Keputusan adalah sistem berbasis komputer yang membantu pembuat keputusan untuk membuat keputusan dari alternatif yang ada dengan memanfaatkan informasi dari berbagai sumber serta pemodelan (Sauter, 2010). Sistem pendukung keputusan memungkinkan pengambil keputusan untuk mengakses data dan melakukan simulasi terhadap alternatif solusi yang ada. Sistem pendukung keputusan terdiri dari empat komponen utama yaitu komponen manajemen data, komponen manajemen model, komponen *user interface* dan komponen manajemen pengetahuan.

Komponen manajemen data meliputi sekumpulan data yang relevan, yang dapat digunakan dalam berbagai situasi pengambilan keputusan, serta dikelola dengan perangkat lunak DBMS (*Database Management System*). Komponen manajemen model adalah perangkat lunak yang mencakup ilmu keuangan, statistik, manajemen, atau model kuantitatif lainnya yang memberikan sistem kemampuan analitis. Perangkat lunak ini sering disebut sebagai MBMS (*model base management system*). Komponen *user interface* adalah komponen yang menjembatani komunikasi antara pengambil keputusan atau pengguna dengan sistem yang akan dibangun. Pengguna akan menginputkan masukan-masukan yang diperlukan untuk melakukan simulasi

melalui komponen *user interface*. Setelah sistem mengolah inputan, sistem akan menampilkan keluaran keputusan melalui komponen *user interface*. Komponen manajemen pengetahuan adalah komponen yang dapat mendukung semua komponen lain atau bertindak sebagai komponen yang independen. Sebuah sistem pendukung keputusan harus memiliki tiga komponen utama yaitu DBMS, MBMS, dan *user interface* (Turban, Aronson, & Liang, 2005).

3.2. Peramalan

Istilah peramalan berkaitan dengan klasifikasi dan pengelompokan (*clustering*). Dalam KBBI, kata meramalkan diartikan sebagai melihat atau menduga keadaan atau hal yang akan terjadi. Peramalan penjualan adalah penggunaan berbagai anggapan untuk memproyeksikan permintaan potensial di masa mendatang (Baktiar, Wibowo, & Adipranata, 2013). Klasifikasi adalah pemetaan data ke dalam kelompok atau kelas yang telah ditentukan sebelumnya. *Clustering* memiliki pengertian yang sama dengan klasifikasi. Perbedaannya adalah dalam *clustering* kelompok dan kelas belum ditentukan sebelumnya, kelompok dan kelas tersebut ditentukan oleh data itu sendiri (Dunham, 2003).

Peramalan dan perencanaan dilakukan karena terdapatnya selang waktu (*time lag*) antara kebutuhan dimasa mendatang dengan peristiwa itu sendiri. Dalam berbagai kasus, perencanaan akan memegang peranan yang penting jika terdapat waktu tenggang yang panjang dan peristiwa yang diramalkan bergantung pada faktor-faktor yang dapat diketahui. Kisaran waktu tenggang tersebut dapat berupa tahun, kuartal, hingga jam. Proses pengadaan barang, penjadwalan produksi, hingga penanaman modal adalah contoh kasus yang memerlukan perencanaan. Kebutuhan akan peramalan meningkat seiring dengan usaha manajemen dalam mengurangi kebergantungan terhadap faktor-faktor yang belum pasti. Peramalan bertujuan untuk memperoleh suatu perkiraan untuk kejadian yang akan datang dengan meminimalkan *forecast error*. Pengukuran *forecast error* dapat dilakukan dengan metode seperti *Mean Absolute Error (MAE)* atau *Mean Squared Error* (Ervinasari, 2008).

Teknik peramalan dibagi ke dalam dua kategori utama yaitu metode kualitatif dan metode kuantitatif.

1. Metode Kualitatif

Metode peramalan kualitatif atau teknologis tidak bergantung pada data-data historis atau data-data seperti pada metode kuantitatif. Metode ini memerlukan input yang bergantung pada metode tertentu dan biasanya merupakan hasil dari perkiraan, pengetahuan, atau pemikiran yang intuitif. Inputan untuk metode ini diperoleh dari sejumlah orang yang terlatih secara khusus pada bidang yang bersangkutan. Metode kualitatif dibagi menjadi metode eksploratis dan normatif. Metode eksploratis seperti Dalphi, kurva-S, analogi, dan penelitian morfologis. Sedangkan Metode normatif seperti matriks keputusan, dan pohon relevansi (Makridakis, 1997).

2. Metode Kuantitatif

Metode kuantitatif adalah teknik peramalan menggunakan data historis. Terdapat kondisi khusus untuk dapat menerapkan peramalan kuantitatif (Makridakis, 1997) yaitu :

- a. Tersedianya informasi tentang masa lalu.
- b. Informasi dapat dikuantitatifkan dalam bentuk data numerik.
- c. Diasumsikan bahwa beberapa aspek pola masa lalu akan terus berlanjut di masa mendatang atau berkesinambungan.

Dalam metode kuantitatif terdapat dua jenis model peramalan yang utama yaitu model deret berkala (*time series*) dan model regresi atau kausal. Pada model deret berkala, peramalan dilakukan berdasarkan nilai-nilai masa lalu dari suatu variabel dan/ atau kesalahan masa lalu. Tujuannya adalah untuk menemukan pola data dan mengekstrapolasikan pola tersebut ke masa depan. Sedangkan model regresi menggunakan asumsi bahwa faktor yang diramalkan memiliki hubungan sebab-akibat terhadap satu atau beberapa variabel bebas.

Model regresi bertujuan untuk menemukan bentuk hubungan tersebut dan menggunakannya untuk meramalkan nilai masa depan dari variabel tak bebas.

3.3. Model ARIMA

3.3.1. Deret Berkala

Deret berkala atau *time series* adalah rangkaian observasi atau pengamatan yang diambil pada waktu yang berurutan. Waktu pengambilan data dapat berupa periode dalam detik, hari, tahun, dan satuan waktu lainnya. Contoh data deret berkala adalah jumlah barang yang dikirim dari pabrik setiap bulan, jumlah kecelakaan di jalan setiap minggunya, jumlah curah hujan harian dan pengamatan per jam pada proses kimia (Box, Jenkins, Reinsel, & Ljung, 2016). Deret berkala dituliskan dengan $Z_1, Z_2, \dots, Z_n$ dengan Z adalah variabel yang diperoleh pada urutan waktu tertentu.

3.3.2. Stasioneritas Data

Proses peramalan dilakukan apabila data deret berkala telah stasioner. Data bersifat stasioner apabila varian, rata-rata, dan auto-correlation bersifat konstan dari waktu ke waktu. Pengujian stasioneritas data dapat dilakukan dengan menggunakan *Augmented Dickey Fuller Test* atau ADF test (Smith, 2018).

H_0 : data tidak stasioner

H_1 : data stasioner

H_0 ditolak jika nilai ADF test < critical value, dengan taraf signifikansi $\alpha = 5\%$.

Data yang tidak stasioner dapat ditangani dengan melakukan *differensiasi*. Banyaknya *differensiasi* yang dilakukan terhadap data dinotasikan dengan d . Bentuk *differensiasi* pertama ($d = 1$) adalah:

$$\nabla Z_t = Z_t - Z_{t-1}$$

Bentuk *differensiasi* kedua ($d = 2$) adalah:

$$\nabla^2 Z_t = \nabla Z_t - \nabla Z_{t-1}$$

dengan:

Z_t : pengamatan waktu ke- t

Z_{t-1} : pengamatan waktu ke- $(t-1)$

∇Z_t : *differensiasi* pertama pada waktu ke- t

$\nabla^2 Z_t$: *differensiasi* kedua pada waktu ke- t

∇Z_{t-1} : *differensiasi* pertama pada waktu ke- $(t-1)$

3.3.3. Model ARIMA

Model *Auto-Regressive Integrated Moving Average* (ARIMA) adalah kelas model paling umum untuk peramalan data deret berkala yang dapat dibuat menjadi stasioner dengan melakukan diferensiasi dan mungkin bersamaan dengan transformasi nonlinier seperti *logging* atau *deflating* jika diperlukan. *Lags* dari deret stasioner dalam persamaan peramalan disebut dengan istilah "*autoregressive*", *lags* dari kesalahan peramalan disebut dengan istilah "*moving average*", dan sebuah deret berkala yang membutuhkan proses pembedaan untuk membuat deret tersebut menjadi stasioner disebut sebagai versi "*integrated*" dari deret stasioner. Model *random-walk*, *random trend*, *autoregressive*, dan *exponential smoothing* adalah kasus-kasus khusus model ARIMA (Nau, 2018). Model ARIMA nonseasonal memiliki kerangka notasi ARIMA (p, d, q).

dengan:

p : jumlah *autoregressive*.

d : jumlah pembedaan nonseasonal yang dibutuhkan untuk membuat deret menjadi stasioner.

q : jumlah *lag* kesalahan peramalan dalam persamaan peramalan.

Persamaan tipe-tipe model ARIMA yaitu:

a. ARIMA ($p,0,0$)

$$x_t = \mu' + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + e_t,$$

Merupakan proses autoregresif dengan μ' adalah nilai konstan, ϕ_p adalah parameter autoregresif ke- p , dan e_t adalah nilai kesalahan pada saat t .

b. ARIMA (0,0, q)

$$x_t = \mu' + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q}$$

Merupakan proses *moving average* dengan θ_1 sampai θ_q adalah parameter-parameter *moving average*, e_{t-k} adalah nilai kesalahan saat $t - k$ dan μ' adalah konstanta.

c. ARIMA (0,1,0)

$$x_t = \mu' + X_{t-1}$$

ARIMA model ini disebut *random walk*. Ketika data tidak stasioner maka model paling sederhana yang dapat digunakan adalah *random walk* yang merupakan kondisi tertentu pada model AR (1) dengan nilai koefisien *autoregressive* adalah 1.

d. ARIMA (1,0,1)

$$x_t = \mu' + \phi_1 X_{t-1} + e_t - \theta_1 e_{t-1}$$

Merupakan model umum untuk campuran AR (1) murni dan MA (1) murni yang tidak mengalami *diferensiasi*. Model ini juga disebut proses atau model ARMA.

e. ARIMA (1,1,1)

$$(1 - B)(1 - \phi_1 B)X_t = \mu' + (1 - \theta_1 B)e_t$$

Pembedaan
pertama

AR(1)

MA (1)

3.3.4. Kriteria Pemilihan Model Terbaik

Untuk menilai kualitas model yang baik dapat menggunakan *Akaike Information Criterion (AIC)*. AIC adalah metode yang dapat digunakan untuk memilih model yang ditemukan oleh Akaike yang berdasar pada metode *maximum likelihood estimation (MLE)*. Dengan menggunakan metode AIC,

model terbaik adalah model yang memiliki nilai AIC terkecil (Fathurrahman, 2009). Nilai AIC dapat diperoleh dengan menggunakan rumus berikut:

$$AIC = -2(\log - likelihood) + 2K$$

dengan: - K adalah jumlah parameter yang diestimasi dalam model

- $\log-likelihood$ adalah ukuran kecocokan model (diperoleh dari hasil statistik)

3.3.5. Pengukuran Ketepatan Nilai Peramalan

Sebuah peramalan selalu mengandung ketidakpastian. Penyimpangan hasil peramalan selain disebabkan adanya error juga dapat disebabkan karena tidak ada metode peramalan yang mampu menghasilkan peramalan dengan akurat. Hal ini juga dapat disebabkan karena metode atau model yang digunakan tidak dapat memprediksi dengan baik komponen *trend*, musiman ataupun siklus yang terdapat pada deret data (Sungkawa & Megasari, 2011). *Mean Squared Error (MSE)* adalah salah satu metode yang dapat digunakan untuk mengevaluasi model dengan rumus:

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

3.4. Pengadaan Onderdil Sepeda Motor

Onderdil atau suku cadang sepeda motor adalah komponen-komponen penyusun sepeda motor. Sebuah sepeda motor terdiri dari ribuan komponen penyusun. Apabila komponen-komponen tersebut mengalami kerusakan, salah satu solusinya adalah dengan mengganti komponen atau suku cadang tersebut. Masing – masing onderdil memiliki fungsi tersendiri. Busi atau *spark plug* berfungsi untuk membakar campuran bahan bakar dan udara kemudian mengantarkan hasil energi panas keluar ruang pembakaran. Disarankan untuk melakukan pemeriksaan busi secara rutin setiap 4.000 Km dan penggantian setiap 8.000 Km, untuk pemakaian secara normal (PT. Astra Honda Motor, 2016).



Gambar 3.1 Busi

Selain busi, onderdil atau suku cadang sepeda motor yang sering diganti adalah ban, lampu, kampas dan tali rem, dan rantai. Setiap produsen motor dalam buku manualnya juga menyarankan perawatan serta penggantian onderdil-onderdil tertentu secara rutin contohnya oli, ban, dan rantai. Penggantian onderdil secara rutin bertujuan untuk mempertahankan performa sepeda motor agar tetap berfungsi dengan baik.

Pengadaan onderdil dapat diartikan sebagai kegiatan yang dilakukan oleh pihak penjual onderdil yang bertujuan untuk menambah stok onderdil yang akan dijual. Pengadaan onderdil dapat dilakukan pada periode tertentu seperti setiap bulan sekali atau ketika stok tidak lagi mencukupi kebutuhan pasar. Dalam melakukan pengadaan, pemilik toko atau penjual akan membuat daftar onderdil yang membutuhkan pengadaan beserta jumlahnya masing-masing, kemudian menghubungi *supplier* atau pemasok masing-masing onderdil tersebut. Dalam menentukan jumlah pengadaan setiap onderdil, penjual dapat melakukannya dengan berbagai cara seperti berdasarkan pengalaman penjual atau berdasarkan pengamatan terhadap data-data penjualan.

3.5. Spring Framework

Spring adalah sebuah *framework* untuk bahasa pemrograman Java. Spring menyediakan berbagai fitur, diantaranya *dependency injection*, *AOP*, *Spring MVC Test*, akses basis data, dan *restful web service*. Spring digunakan karena memberikan kemudahan dalam pengaturan pengembangan proses bisnis dari sistem pendukung keputusan yang akan dibangun.

Spring Boot adalah salah satu jenis *framework* Spring. Spring Boot memberikan kemudahan seperti kemudahan dalam menjalankan aplikasi, penggunaan Maven sebagai *build manager*, serta berbagai anotasi yang mempermudah pengembangan aplikasi. Diperlukan sebuah *tool* untuk membangun aplikasi Spring Boot. Beberapa IDE (*Integrated Development Environment*) yang dapat digunakan adalah IntelliJ dan Netbean (Pivotal Software, 2018).

3.6. Vue.js

Vue.js atau Vue adalah sebuah *framework* untuk bahasa pemrograman JavaScript. Vue digunakan untuk membangun antarmuka pengguna atau *frontend*. Vue membantu membuat *code* yang lebih mudah untuk dikelola dan di test. Vue adalah progresif JavaScript *framework*. Jika *programmer* memiliki sebuah aplikasi disisi server, *programmer* dapat menambahkan Vue ke dalam sebuah bagian dari aplikasi tersebut yang membutuhkan peranan Vue.

Vue juga dapat digunakan untuk membangun proses bisnis tambahan pada sisi *frontend*. Vue memiliki *core libraries* dan ekosistem. Seperti *frontend framework* lainnya, Vue memungkinkan *programmer* untuk membagi sebuah halaman web ke dalam komponen yang dapat digunakan berulang-ulang. Masing-masing komponen tersebut memiliki HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*), dan JavaScript tersendiri (You, 2018).

3.7. Database Management Systems (PostgreSQL)

DBMS atau *Database Management Systems* adalah perangkat lunak yang digunakan untuk melakukan pengelolaan terhadap basis data. PostgreSQL adalah salah satu jenis DBMS yang sangat populer. PostgreSQL dikembangkan sejak tahun 1986 sebagai bagian dari proyek POSTGRE di *University of California* di Barkeley.

PostgreSQL adalah sistem basis data objek-relasional yang bersifat *open source*. PostgreSQL telah menjadi basis data relasional *open source* pilihan bagi banyak orang dan organisasi berkat arsitektur, keandalan, integritas data, ekstensibilitas, dan komunitas *open source* yang dimilikinya. PostgreSQL berjalan hampir pada semua sistem operasi dan telah sesuai dengan ACID sejak tahun 2001 (PostgreSQL, 2019).

3.8. Library Pmdarima 1.1.0 untuk Python

Pmdarima untuk Python adalah *library* yang dikembangkan berdasarkan fungsional yang sebelumnya sudah ada pada bahasa R. Pmdarima memiliki sebuah fungsional untuk melakukan pemilihan parameter arima paling optimal kemudian me-return model ARIMA yang sudah *fit* terhadap data *series* yang diberikan. Fungsional ini disebut dengan auto arima.

Auto arima bekerja dengan cara melakukan *differencing* tes seperti Augmented Dickey-Fuller untuk menentukan order d . Setelah mendapatkan order d , selanjutnya dilakukan proses *fitting* model terhadap data *series* pada interval nilai p dan q yang diberikan. Untuk menentukan model terbaik, auto arima menggunakan *information criterion* aic (*Akaike Information Criterion*), bic, hqic, atau oob (Smith, 2018).