



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1. Teori Terkait

2.1.1. *Time Series Forecasting*

Time Series secara umum merupakan rangkaian data yang diambil dalam waktu yang berurutan. Analisa *time series* berfokus pada teknik analisa yang saling keterkaitan antara data dan waktu. Sebuah *time series* suatu observasi dapat dinotasikan dengan sebuah kombinasi linear dari variabel acak (Shumway & Stoffer, 2011). Pada rumus 2.1. berikut merupakan notasi linear dari ϵ_i dalam bentuk persamaan.

$$x_t = \mu + \delta_0\epsilon_t + \delta_1\epsilon_{t-1} + \delta_2\epsilon_{t-2} + \dots + \delta_n\epsilon_{t-n}$$

Rumus 2.1. Kombinasi Linear *White Noise*

Sumber: Shumway & Stoffer, 2011

2.1.2. ARIMA (*Auto Regressive Integrated Moving Average*)

ARIMA merupakan salah satu metode peramalan *time series* yang digunakan untuk peramalan jangka pendek. Pada model ARIMA (p, d, q) ordo p menunjukkan adanya proses *auto regressive* sehingga model ARIMA dapat menjadi model AR (p), ordo d menunjukkan proses differensiasi pada data, dan q menunjukkan proses *moving average* sehingga dapat menjadi model MA (q) (Cheng, Chiang, & Chen, 2016). Pada rumus 2.2 merupakan model ARIMA (p, d, q) yang dapat dinotasikan sebagai berikut:

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) (1-L)^d y_t = \left(1 - \sum_{i=1}^q \theta_i L^i\right) \epsilon_t$$

Rumus 2.2. ARIMA

Sumber: Cheng, Chiang, & Chen, 2016

2.1.3. AR (*Auto Regressive*)

AR merupakan bentuk umum model *auto regressive* dengan ordo p pada model ARIMA $(p,0,0)$ sebagai bentuk regresi, akan tetapi tidak menghubungkan variabel bebas dengan variabel terikat (Aziz, Sayuti, & Mustakim, 2017). Bentuk model AR dinyatakan pada rumus 2.3. sebagai berikut (Cheng, Chiang, & Chen, 2016):

$$y_t = \alpha_y + \sum_{i=1}^p \alpha_i y_{t-1} + \varepsilon_t$$

Rumus 2.3. *Auto regressive*

Sumber: Cheng, Chiang, & Chen, 2016

Keterangan:

α_0 = suatu konstanta

p = parameter lag auto regresif ke- p

α_i = nilai koefisien dari y_{t-1}

ε_t = nilai kesalahan pada saat t

2.1.4. MA (*Moving Average*)

MA merupakan model lain dari model ARIMA atau disebut dengan *moving average*. *Moving Average* merupakan ordo q pada model ARIMA $(0, 0, q)$ (Aziz, Sayuti, & Mustakim, 2017) yang dapat dinyatakan pada rumus

2.4. berikut (Cheng, Chiang, & Chen, 2016).

$$y_t = \alpha_0 + \sum_{i=1}^q b_i \varepsilon_{t-1} + \varepsilon_t$$

Rumus 2.4. *Moving Average*

Sumber: Cheng, Chiang, & Chen, 2016

Keterangan:

a_0 = suatu konstanta

q = parameter lag auto regresif ke-p

b_i = nilai koefisien dari ε_{t-1}

et = nilai kesalahan pada saat t

2.1.5. ARMA (*Auto Regressive Moving Average*)

ARMA adalah proses gabungan, dimana pola dari data *time series* yang terbentuk, model terbaik digambarkan dengan menggabungkan proses *auto regressive* dan *moving average* (Aziz, Sayuti, & Mustakim, 2017). Pada rumus 2.5. merupakan bentuk umum persamaan ARMA yang dinotasikan sebagai berikut.

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + et - \phi_1 x_{t-1} - \phi_2 x_{t-2} - \dots + \phi_p x_{t-p} + et$$

Rumus 2.5. *Auto Regressive Moving Average*

Sumber: Cheng, Chiang, & Chen, 2016

2.1.6. ACF (*Auto Correlation Function*)

ACF adalah korelasi antara data pada periode waktu t dengan periode waktu sebelumnya t-1. Nilai tengah dan ragam dari suatu data *time series* mungkin tidak bermanfaat apabila data tersebut tidak stasioner, akan tetapi nilai maksimum dan minimum dapat digunakan untuk tujuan *plotting* (Aziz,

Sayuti, & Mustakim, 2017). Persamaan ACF pada rumus 2.6. berikut:

$$r_k = \frac{\Sigma(X_t - \bar{X})(X_{t+k} - \bar{X})}{\Sigma(X_t - \bar{X})^2}$$

Rumus 2.6. *Auto Correlation Function*

Sumber: Aziz, Sayuti, & Mustakim, 2017

2.1.7. PACF (*Partial Auto Correlation Function*)

PACF adalah auto korelasi parsial yang digunakan untuk mengukur tingkat kecerdasan antara X_t dan X_{t-k} , apabila pengaruh dari *lag time* dianggap terpisah. Satu-satunya tujuan di dalam analisis *time series* adalah untuk membantu menetapkan model ARIMA yang tepat (Aziz, Sayuti, & Mustakim, 2017). Persamaan PACF dapat di lihat pada rumus 2.7. berikut:

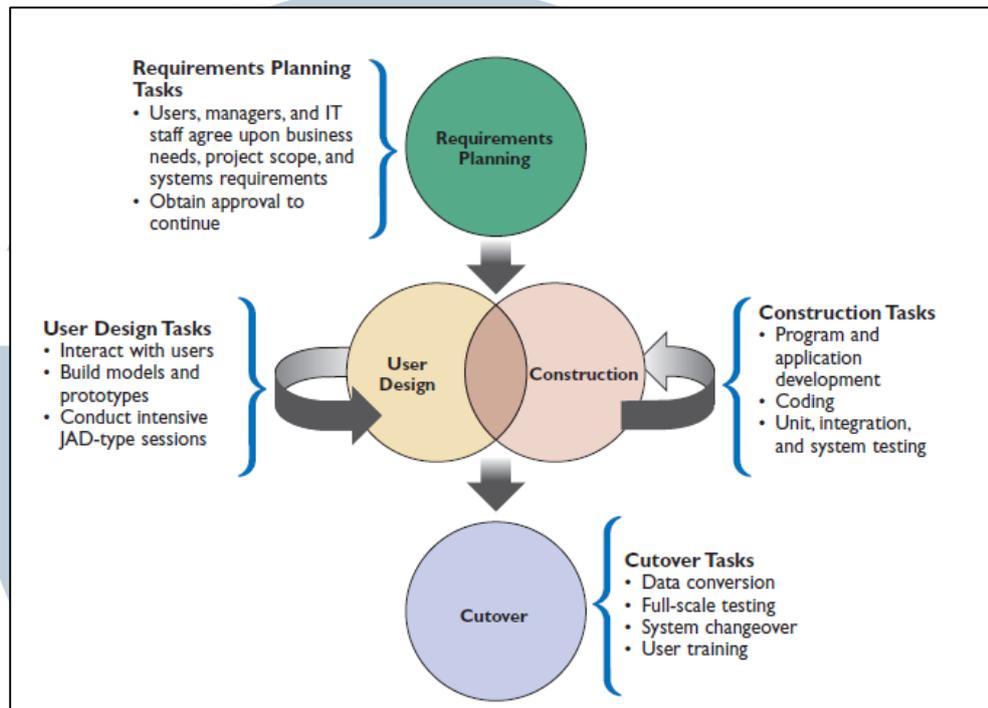
$$r_{kk} = \begin{cases} r_k - \sum_{j=1}^{k-1} r_{j-1} r_{k-j} \\ 1 - \sum_{j=1}^{k-1} r_{j-1} r_{k-j} \end{cases}$$

Rumus 2.7. *Partial Auto Correlation Function*

Sumber: Aziz, Sayuti, & Mustakim, 2017

2.1.8. RAD (*Rapid Application Development*)

RAD (*Rapid Application Development*) merupakan teknik dalam pengembangan aplikasi yang membutuhkan waktu pengerjaan yang relatif singkat. Pada umumnya perusahaan menggunakan RAD (*Rapid Application Development*) untuk mengurangi biaya dan waktu pengembangan, dan meningkatkan probabilitas kesuksesan. RAD (*Rapid Application Development*) memberikan pengguna untuk memeriksa aplikasi secara cepat. Pengguna dapat memberikan evaluasi kepada tim atau individu. Pada gambar 2.1. merupakan tahapan yang terdapat pada metode pengembangan aplikasi RAD (*Rapid Application Development*) yang memiliki 4 tahapan (Shelly & Rosenblatt, 2012), yaitu:



Gambar 2.1. 4 Tahapan *Rapid Application Development*

Sumber: Shelly & Rosenblatt, 2012

1. *Requirements planning*

Pada fase *requirements planning*, pembuatan ruang lingkup proyek, penyusunan rencana kebutuhan sistem, perencanaan kebutuhan *user* dan mencari tujuan dari proyek. Fase *requirements planning* berakhir apabila seluruh kebutuhan telah teridentifikasi dengan lancar (Shelly & Rosenblatt, 2012).

2. *User Design*

Pada fase *user design*, seluruh tim merancang model atau *prototype* yang merepresentasikan seluruh proses sistem ketika akan dibangun, termasuk *output* dan *input*. Pada fase ini memungkinkan untuk melakukan interaksi antar *user* agar dapat memahami, mengubah, dan

menyetujui model dari sistem yang dibutuhkan (Shelly & Rosenblatt, 2012)

3. *Construction*

Pada fase *construction* berfokus pada pengembangan program dan aplikasi. Dalam fase ini *user* berpartisipasi dalam memberikan saran maupun aspirasi dalam bentuk laporan atau masukan kepada tim *developer* (Shelly & Rosenblatt, 2012).

4. *Cutover*

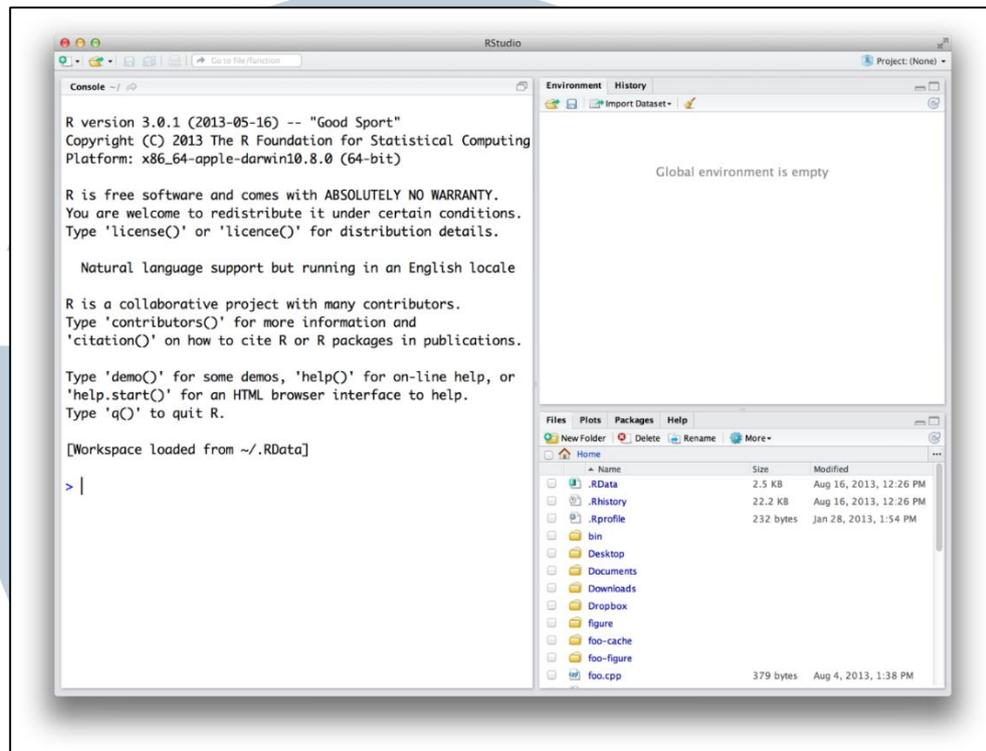
Fase *cutover* terakhir pada metode RAD adalah proses implementasi aplikasi yang telah dikembangkan, *testing*, transformasi sistem lama menjadi sistem yang baru, dan melakukan *user training*. Sistem yang baru diimplementasikan kepada *user*.

2.1.9. RStudio

RStudio merupakan aplikasi seperti Microsoft Word namun aplikasi ini tidak digunakan untuk menulis dokumen namun dapat membantu dalam menulis dalam bahasa R. Rstudio merupakan aplikasi yang terintegrasi dengan IDE (*Integrated Development Environment*) untuk menulis R, termasuk dalam menampilkan grafik plot maupun grafik untuk kebutuhan visualisasi (Grolemund & Wickham, 2014). Pada gambar 2.2. merupakan

tampilan awal dari Rstudio yang digunakan untuk menulis bahasa R.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 2.2. User Interface RStudio

Sumber: Golemund & Wickham, 2014

2.1.10. R

R adalah *software* yang saling terintegrasi untuk melakukan manipulasi data, perhitungan data, dan visualisasi tampilan. Kelebihan yang dimiliki oleh R adalah fasilitas menyimpan data yang efektif, rangkaian operator untuk perhitungan pada *arrays*, khususnya perhitungan matriks, kumpulan seperangkat yang digunakan untuk melakukan analisis data, fasilitas untuk menampilkan dalam bentuk grafik sehingga dapat membantu *data scientist* untuk melakukan penelitian, bahasa pemrograman yang dikembangkan dengan baik, sederhana, dan efektif (Venables & Smith, 2018). R merupakan bahasa pemrograman seperti C, C++, atau UNIX. Sebelum RStudio, R dijalankan di UNIX *Terminal* (Golemund & Wickham,

2014). R sangat banyak sarana untuk mengembangkan metode analisis data interaktif. Saat ini R berkembang pesat, dan telah memiliki banyak *package* yang dapat memudahkan dalam melakukan pemrograman R. Namun, sebagian besar program yang ditulis dalam R pada dasarnya singkat, ditulis untuk satu bagian dari analisis data (Venables & Smith, 2018).

2.1.11. Shiny

Shiny merupakan *package* yang dibuat oleh tim RStudio yang dapat memudahkan untuk membuat tampilan antarmuka R yang telah memiliki *framework* untuk membuat *web browser*. Salah satu keunggulan dari Shiny ini adalah tidak perlu menggabungkan R *code* dengan HTML (*Hypertext Markup Language*) atau JavaScript karena di dalam *package* Shiny telah memiliki fitur yang secara fungsionalitas sama dengan interaktif *website*, khususnya berorientasi terhadap visualisasi data yang interaktif. Kelebihan lain yang dimiliki oleh Shiny yaitu *open-source* sehingga siapapun dapat menggunakan Shiny untuk mengembangkan aplikasi, kemudian seluruh kemampuan bahasa R juga disediakan oleh Shiny sehingga sebagai *developer* dapat menjalankan pemrograman statistik dan dapat menampilkan ke dalam bentuk *browser* yang *user friendly* (Resnizky, 2015).

2.1.12. UML (*Unified Modeling Language*)

Unified Modeling Language adalah bahasa yang banyak digunakan sebagai metode untuk memvisualisasikan dan mendokumentasikan sistem perangkat lunak. Desain pada UML menggunakan konsep desain berorientasi objek, tetapi sangat independen terhadap bahasa pemrograman tertentu dan

dapat digunakan untuk menggambarkan proses dan persyaratan bisnis umumnya.

UML menyediakan berbagai *tools* untuk mendesain UML, seperti *use case diagram* dan *sequence diagram*. Seorang sistem analis dapat menggunakan UML untuk mewakili kegunaan sistem informasi dari sudut pandang pengguna (Shelly & Rosenblatt, 2012).

2.1.12.1. Use Case Diagram

Use case diagram memungkinkan seorang analis untuk menjelaskan skenario yang mungkin akan terjadi yang dilakukan oleh aktor dan sistem. *Use Case* mendeskripsikan apa yang sistem bisa lakukan namun tidak dapat menjelaskan secara detail seperti struktur data, algoritma, dan sebagainya. Detail dari *use case* dijelaskan pada diagram yang lain seperti *class diagram* atau *activity diagram*. *Use case* merupakan konsep fundamental dari banyak metode pengembangan aplikasi berorientasi objek. *Use case* merepresentasikan apa yang pengguna inginkan terhadap sistem (Gomaa, 2011).

2.1.12.2. Activity Diagram

Activity Diagram sering terlihat sebagai bagian dari pandangan secara fungsional dari sebuah sistem karena *activity diagram* dapat menjelaskan logika dari sebuah proses maupun sebuah fungsi. Setiap proses dijelaskan secara berurutan dan pengelolaan keputusan yang dapat dijelaskan. Alur pada *activity diagram* berfokus pada penyelesaian pada sebuah kegiatan. *Activity*

diagram juga dapat menggambarkan proses paralel yang terjadi pada beberapa kegiatan (Ainapure, 2009)

2.1.12.3. Class Diagram

Class Diagram merupakan sebuah diagram yang menunjukkan beberapa susunan *class*, tampilan antarmuka, dan kolaborasi dalam setiap hubungannya. *Class diagram* menggambarkan hubungan antara *class*, *package*, relasi, asosiasi dan lain sebagainya (Swain, 2010).

Sebuah *class diagram* memiliki komponen umum seperti pada *diagram* pada umumnya. Komponen utama pada *class diagram* memiliki 3 inti yaitu, nama *class*, atribut, dan metode.

2.1.13. Black Box Testing

Black Box Testing merupakan sebuah metode pengujian yang mengacu kepada *software* yang diperlakukan seperti kotak hitam, dengan pengujian *black box* aplikasi atau *source code* tidak dilakukan pengecekan seluruhnya. Responden yang terlibat dalam *black-box testing* hanya mengetahui beberapa *input* dan *output* yang diharapkan tanpa harus memahami bagaimana proses yang terjadi di dalam aplikasi (Chopra, 2018).

2.1.14. Simple Exponential Smoothing

Simple exponential smoothing merupakan metode peramalan yang menggunakan sedikit perhitungan. Metode ini digunakan ketika pola data berbentuk horizontal seperti tidak ada variasi pada siklus atau tren nyata dalam data historis (Nazim & Afthanorhan, 2014).

2.1.15. Waterfall Development Method

Model *waterfall* merupakan proses pengembangan *software* yang dilakukan secara berurutan. *Waterfall* mendefinisikan beberapa fase yang berturut-turut yang harus diselesaikan satu demi satu dan pindah ke fase berikutnya hanya ketika fase sebelumnya telah sepenuhnya selesai (Bassil, 2012).

2.1.16. Prototyping Development Method

Metodologi *system development* mengarah kepada *framework* yang digunakan struktur, perencanaan, dan kontrol dari sebuah proses pengembangan sistem informasi. Salah satu method *system development* yang umum digunakan yaitu *prototyping method*. *Prototyping method* merupakan sebuah metode yang berfokus membuat aplikasi dalam skala kecil mengikuti perubahan yang terus berkembang hingga mode *prototype* tersebut mencapai tujuan dari *system requirements* (Departement of Health and Public Services, 2008).



2.2. Penelitian Terdahulu

Pada tabel 2.2. berikut merupakan penelitian terdahulu yang dijadikan landasan teori dalam pembuatan penelitian ini mengenai prediksi menggunakan metode ARIMA.

Tabel 2.2. Penelitian Terdahulu

Judul Jurnal	Hasil Penelitian
<p><i>Intermittent Demand Forecasting in a Tertiary Pediatric Intensive Care Unit.</i></p> <p>Peneliti Chen Yang Cheng, Kuo Liang Chiang, Meng Yin Chen</p> <p>Lokasi New York</p> <p>Tahun 2016</p> <p>Nama Jurnal Journal of Medical System</p> <p>Volume 40</p>	<p>Pada penelitian ini menggunakan model ARIMA karena sering digunakan untuk kebutuhan <i>demand forecasting</i>. Hasil penelitian menunjukkan metode <i>demand forecasting</i> dapat digunakan pada tingkat yang lebih baik. Oleh karena itu, pusat kesehatan dapat menggunakan sistem untuk mencegah persediaan medis agar tidak kelebihan maupun kekurangan. Penelitian ini menunjukkan ARIMA dapat digunakan sebagai metode untuk menghitung persediaan untuk kebutuhan medis seperti obat-obatan.</p>
<p><i>A new method of large-scale short term forecasting of agricultural commodity prices: illustrated by the case of agricultural markets in Beijing.</i></p> <p>Peneliti Haoyang Wu, Huaili Wu, Minfeng Zhu, Weifeng Chen, Wei Chen</p> <p>Lokasi Guangzhou</p> <p>Tahun 2017</p> <p>Nama Jurnal Journal of Big Data</p> <p>Volume 4</p>	<p>Pada penelitian ini menggunakan ARIMA untuk menentukan prediksi harian pada suatu komoditas agrikultur. Penelitian ini menunjukkan dapat memprediksi dalam waktu yang cukup singkat sehingga dapat mendukung penelitian ini dalam melakukan penelitian untuk kebutuhan prediksi vaksin yang ditentukan secara jangka waktu yang singkat.</p>

Judul Jurnal	Hasil Penelitian
<p data-bbox="312 349 732 524"><i>Application of time series analysis in modelling and forecasting emergency department visits in a medical centre in Southern Taiwan.</i></p> <p data-bbox="312 566 695 707">Peneliti Wang Chuan Juang, Sin Jih Huang, Fong Dee, Pei Wen Huang, Ren Wann Shue</p> <p data-bbox="312 750 419 815">Lokasi London</p> <p data-bbox="312 857 411 922">Tahun 2017</p> <p data-bbox="312 965 501 1030">Nama Jurnal BMJ Open</p> <p data-bbox="312 1072 427 1137">Volume 7</p>	<p data-bbox="748 349 1359 929">Kunjungan <i>Emergency Departement</i> bulanan ditunjukkan dengan mengikuti ARIMA model (0, 0, 1). Model ini dapat dianggap memadai untuk memprediksi kunjungan ED di masa mendatang di rumah sakit dan dapat dipekerjakan untuk membantu dalam proses pengambilan keputusan. Penelitian ini memberikan wawasan baru tentang alasan kepadatan penduduk di UGD dan dapat membantu dalam pengembangan lebih lanjut dari model ARIMA (0, 0, 1) untuk memperkirakan alasan untuk kepadatan di <i>ermergency department</i>. Penelitian ini menggunakan ARIMA dikarenakan akurasi penggunaan model memiliki tingkat akurasi yang cukup tinggi.</p>

Berdasarkan penelitian terdahulu maka penelitian ini akan menggunakan metode ARIMA (Cheng, Chiang, & Chen, 2016) (Wang, Sin, Fong, Pei, & Ren, 2017) (Haoyang, Huaili, Minfeng, Weifeng, & Wei, 2017) sebagai solusi untuk permasalahan stok vaksin yang mengalami kekurangan atau kelebihan yang terjadi, serta dapat memprediksi permintaan stok vaksin untuk kebutuhan tahun 2018 yang akan diimplementasikan ke dalam bentuk sistem prediksi vaksin berbasis aplikasi R.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A