PENGEMBANGAN SISTEM DATA-TO-TEXT UNTUK MEMBANGKITKAN BERITA INFLASI DAN INDEKS HARGA KONSUMEN DENGAN PENDEKATAN *TIME-SERIES* MENGGUNAKAN BAHASA R

PROPOSAL SKRIPSI

Diajukan untuk Memenuhi Persyaratan dan Penulisan Skripsi Akhir Studi S1

Program Studi Ilmu Komputer

Oleh

Muhammad Ridwan

1403407

PROGRAM STUDI ILMU KOMPUTER

DEPARTEMEN PENDIDIKAN ILMU KOMPUTER

FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PENDIDIKAN INDONESIA

2017

DAFTAR ISI

DAFTAR ISI ii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 4
1.4 Tujuan 4
1.5 Manfaat Penelitian 4
1.6 Sistematika Penulisan 5
BAB II KAJIAN PUSTAKA 6
2.1 Pengertian *Natural Language Processing* 6
2.1.1 Area *Natural Language Processing* 8
2.2 Pengertian *Natural Language Generation* 9
2.3 Pengertian dan Arsitektur *Data-to-text* 9
2.4 Penelitian Terkait *Data-to-text* 15
2.5 Pengertian dan Sejarah *Machine Learning* 17
2.5.1 *SupervisedLearning* 18
2.5.2 *UnsupervisedLearning* 19
2.5.3 Algoritma *Gradient Descent* 20
2.6 *Time-series* Data 21
2.7 *Autoregresif Integrated Moving Average* (ARIMA) 22
2.8 R *Programming* 32
2.8.1 Model data dalam R 34
2.8.2 Contoh kode program bahasa R 35
2.8.3 Contoh visualisasi data dalam R 37
2.8.4 *Package* alam bahasa R 39
BAB III 40
METODOLOGI PENELITIAN 40
3.1 Desain Penelitian 40

3.2 Alat dan Bahan Penelitian 42
3.2.2. Bahan Penlitian 43
3.3. Metode Penelitian. 43
3.3.1. Metode Pengumpulan Data 43
3.3.2. Metode Pengembangan Perangkat Lunak 43
DAFTAR PUSTAKA 45

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, kebutuhan manusia akan informasi semakin tinggi. Perkembangan teknologi dan ilmu pengetahuan menyebabkan ketersediaan informasi meningkat, terutama informasi berupa data non-linguistik atau data numerik. Hal tersebut bertolak belakang dengan kebutuhan user akan informasi yang mudah dipahami dan dimengerti dengan cepat, sehingga waktu yang digunakan user untuk memahami informasi yang berbentuk data numerik relatif lebih lama dibandingkan dengan informasi berupa text atau berita. Sehingga para peneliti dan pengembang berlomba-lomba untuk mengembangkan aplikasi-aplikasi yang mampu menghasilkan informasi dalam bentuk text dengan *input* data non-linguistik atau data numerik. Salah satunya yaitu aplikasi atau sistem *Data-to-text* (D2T) yang diperkenalkan oleh (Reiter, E, 2007).

Sistem D2T ini merupakan salah satu bagian dari sistem *Natural Language Generation* (NLG) yang dapat menghasilkan data tekstual dari data numerik secara otomatis. Sistem D2T ini dapat berbagai input data non linguistik mulai dari data numerik, *event logs*, maupun data yang dihasilkan dari sensor. Karena sistem D2T ini sangat erat dengan proses linguistik begitu juga proses analisis data, maka (Reiter, dkk, 2007) memaparkan bahwa setidaknya ada empat langkah dalam tahapan pembangunan sistem D2T, yaitu: analisis sinyal, intrepretasi data, perencanaan dokumen, *microplanning*, dan realisasi.

Sudah banyak implementasi sistem D2T yang menjadi solusi dalam menyediakan informasi tekstual dalam beberapa bidang. Contohnya pada bidang peramalan cuaca, yaitu aplikasi *Forecast Generator* (FOG) yang diperkenalkan oleh (Goldberg, Driedger, & Kitterdge, 1994), aplikasi tersebut dapat mengkonversi peta cuaca menjadi ramalan dalam bentuk kalimat dengan pengolahan bahasa alami. Selain itu, ada *SumTime-Mousam* yang

diperkenalkan oleh (Sripada, dkk., 2003), aplikasi ini dapat menghasilkan ramalan cuaca laut tekstual untuk rig minyak lepas pantai. Contoh lainnya yaitu pada bidang kesehatan, yaitu *BABYTALK family System* yang diperkenalkan oleh (Potret, dkk., 2009), aplikasi ini mampu membuat sebuah ringkasan peristiwa yang terjadi selama 45 menit dari sinyal psikologis kontinyu dan diskrit, seperti pengaturan peratalatan dan pemberian obat dalam bentuk kalimat. Selain itu, (Hunter, dkk., 2012) memperkenalkan sistem yang dapat menghasilkan ringkkasan dari pergantian keperawatan yang berasal dari pencatatan pasien elektronik di Neonatal Intensive Care *Unit* (NICU).

Dalam penelitian ini, akan dibangun sistem D2T yang dapat membangkitkan bahasa alami yang dapat menyampaikan informasi terkait inflasi, indeks harga konsumen, dan analisis komoditas-komoditas yang mempengaruhinya. Sehingga informasi yang dihasilkan lebih mudah dipahami dibandingkan dengan data numerik yang diperoleh langsung dari Badan Pusat Statistik (BPS). Fokus dalam penelitian ini adalah meningkatkan kualitas informasi yang disampaikan agar lebih mudah dipahami. Maka dari itu diperlukan serangkaian proses yang dapat menganalisis data numerik sehingga dapat menghasilkan pola ataupun trend dan apa saja relasi dari kedua hal tersebut yang selanjutnya dikemas dalam bentuk bahasa alami yang mudah dipahami oleh manusia. Berdasarkan hal tersebut maka D2T dirasa dapat memberikan pengaruh yang cukup signifikan dalam mempermudah pemahaman suatu informasi.

Sumber data yang digunakan dalam penelitian ini didapatkan langsung dari Badan Pusat Statistik (BPS). Beberapa data yang digunakan, meliputi data inflasi bulanan, data Indeks Harga Konsumen (IHK) per kelompok dan subkelompok, gabungan dari 82 kota, data inflasi menurut kelompok komoditi, data tingkat inflasi gabungan 82 kota, data inflasi umum, data harga yang diatur pemerintah, dan barang bergejolak inflasi indonesia. Karena pada dasarnya di era *Big Data* ini ketersediaan data semakin meningkat, mudah diakses, variatif, dan juga dinamis. Namun jika tidak didampingi dengan sebuah sistem yang

dapat mengelola data tersebut sehingga informasi yang diperoleh mudah dipahami maka akan dirasa sangat sulit jika kita harus menganalisis data tersebut secara manual. Maka tidak heran sistem D2T ini bisa menjadi suatu solusi yang dapat mempermudah dalam penyampaian dan analisis suatu informasi.

Dalam pembangunan sistem ini digunakan pendekatan *time*-series untuk mempermudah dalam analisis data, dan juga menggunakan model *Autoregresif Integrated Moving Average* (ARIMA) dalam menentukan hasil prediksi. Selain itu juga untuk mempersingkat waktu pembangunan maka digunakan beberapa package yang sudah tersedia dalam R. Penilitian ini akan menghasilkan produk akhir sebuah aplikasi berbasis *web* dengan menggunakan package *shiny* yang sudah tersesdia dalam R, sehingga konten yang dimuat bisa lebih kaya dan variatif, elemen-elemen pendukung seperti grafik, tabel, maupun diharapkan dapat mendukung dan memperkaya berita yang akan dihasilkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang ada, maka permasalah dalam skripsi ini dirumuskan sebagai berikut:

1. Bagaimana pengembangan model dari sistem *Data-to-Text* untuk membangkitkan berita tingkat inflasi, indeks harga konsumen, dan harga komoditas yang berpengaruh terhadap inflasi dengan menggunakan pendekatan *Time Series*?
2. Bagaimana proses implementasi sistem *Data-to-text* dalam R?
3. Bagaimana eksperimen dan hasil eksperimen dari sistem *Data-to-text* yang dikembangkan?

1.3 Batasan Masalah

Dalam penelitian ini, permasalahan dibatasi hal-hal berikut inil:

1. Pembangunan sisten *Data-to-text* dengan pendekatan *Time Series* ini hanya didasarkan pada data harga dan bobot komoditas, indeks harga konsumen nasional dan per kota, dan tingkat inflasi bulanan yang diperoleh dari Badan Pusat Statistik (BPS) dan Kementrian Perdagangan Republik Indonesia.
2. Pembangunan sisten *Data-to-text* ini hanya menggunakan bahasa pemrograman R.

1.4 Tujuan

Berdasarkan rumusan masalah di atas, maka tujuan penelitian dalam tugas akhir ini dirumuskan sebagai berikut:

1. Untuk melakukan pengembangan model *Data-to-text* untuk membangkitkan berita terkait tingkat inflasi, indeks harga konsumen, dan peranan komoditas yang mempengaruhinya dengan menggunakan pendekatan *Time Series.*
2. Untuk melakukan implementasi model *Data-to-text* dalam bahasa pemrograman R.
3. Untuk mengetahui kualitas sistem dengan melakukan eksperimen.

1.5 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Diharapkan dapat menambahkan pengetahuan tentang sistem *Data-totext* dan *time-series* serta penerapannya dalam membangkitkan bahasa alami untuk mendeskripsikan data inflasi dan komoditas yang mempengaruhi inflasi.
2. Dapat menjadi salah satu alternatif dan pelengkap dalam menyampaikan hasil analisis data secara otomatis oleh sistem *Data-totext.*
3. Dapat menjadi salah satu referensi dalam pembangunan sistem *Data-to-text* yang memanfaatkan bahasa pemrograman R beserta fiturnya seperti *packages.*

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini diuraikan menjadi lima bab, yaitu:

BAB I PENDAHULUAN

BAB I teridiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian yang akan dilakukan, dan sistematikan penulisan.

BAB II TINJAUAN PUSTAKA

BAB II terdiri dari beberapa kajian singkat tentang teori-teori dan konsep yang dibutuhkan dalam penelitian. Terdiri dari pembahasan mengenai *Natural Language Processing, Natural Language Generation, Data-to-text, Time-series, R Programming*, dan lain-lain.

BAB III METODOLOGI PENELITIAN

BAB III terdiri dari langkah-langkah yang akan dilakukan dalam penelitian. Terdiri dari desain penelitian, alat penelitian, dan bahan penelitian.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Berisi hasil penelitian serta analisis yang dilakukan selama penelitian. Yaitu terdiri dari pengembangan model, implementasi sistem, eksperimen dan hasil eksperimen.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan yang didapat selama penelitian dan saran-saran dalam meningkatkan kualitas dan kuantitas hasil penelitian.

LAMPIRAN

Berisi dokumen-dokumen yang menunjang keabsahan penelitian.

BAB II
KAJIAN PUSTAKA

2.1 Pengertian *Natural Language Processing*

Natural Language Processing (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa alami yang diterima oleh komputer akan melalui serangkaian proses terlebih hingga computer dapat memahami maksud dari bahasa alami tersebut. Serangkaian proses inilah yang disebut dengan NLP. *Natural Language Processing* (NLP) adalah pendekatan terkomputerisasi untuk menganalisis teks yang didasarkan pada seperangkat teori dan satu set teknologi (Liddy, 2001). NLP termasuk kedalam bidang ilmu *Artificial Intelligence* (AI) yang hanya berfokus pada pengolahan bahasa alami. Tujuan utama dari NLP adalah memberikan kemudahan komunikasi antara computer dengan manusia (*computer-human communication*).

Ada banyak aplikasi yang telah menerapkan NLP,salah satu diantaranya adalah Chatbot. Aplikasi Chatbot dapat membuat user seolah-olah mampu berkomuniasi langsung dengan komputer dalam bahasa manusia, contoh aplikasi Chatboth adalah sim-simi. Contoh lainnya adalah aplikasi *Stemming* atau *Lemmatization*, aplikasi ini dapat melakukan pemotongan kata dalam bahasa tertentu menjadi bentuk dasar pengenalan fungsi setiap kata dalam kalimat. Contoh lainnya adalah aplikasi *Summarization* yang dapat melakukan peringkasan terhadap sebuah berita, serta aplikasi Translation Tools yang mampu menerjemahkan bahasa.

Perkembangan NLP menghasilkan kemungkinan dari interface bahasa natural menjadi knowledge base dan penterjemahan bahasa natural. Terdapat bahwa ada 3 (tiga) aspek utama pada teori pemahaman mengenai natural language:

1. *Syntax*: menjelaskan bentuk dari bahasa. Syntax biasa dispesifikasikan oleh sebuah *grammar*. *Natural language* jauh lebih daripada formal language yang digunakan untuk logika kecerdasan buatan dan program komputer
2. *Semantics*: menjelaskan arti dari kalimat dalam satu bahasa. Meskipun teori semantics secara umum sudah ada, ketika membangun sistem natural language understanding untuk aplikasi tertentu, akan digunakan representasi yang paling sederhana.
3. *Pragmatics*: menjelaskan bagaimana pernyataan yang ada berhubungan dengan dunia. Untuk memahami bahasa, agen harus mempertimbangan lebih dari hanya sekedar kalimat. Agen harus melihat lebih ke dalam konteks kalimat, keadaan dunia, tujuan dari speaker dan listener, konvensi khusus, dan sejenisnya.
4. Morfologi. Adalah pengetahuan tentang kata dan bentuknya sehingga bisa dibedakan antara yang satu dengan yang lainnya. Bisa juga didefinisikan asal usul sebuah kata itu bisa terjadi. Contoh : membangunkan –> bangun (kata dasar), mem- (prefix), -kan (suffix).
5. Fonetik. Adalah segala hal yang berhubungan dengan suara yang menghasilkan kata yang dapat dikenali. Fonetik digunakan dalam pengembangan NLP khususnya bidang *speech based system*.

2.1.1 Area *Natural Language Processing*

Dijelaskan dalam (James dan Amber, 2012), bahwa area ruang lingkup yang termasuk kedalam *Natural Language Processing* terdiri dari:

1. *Question Answering System* (QAS) atau Sistem Tanya Jawab: misalnya, anda dapat bertanya kepada komputer dimana restoran terbaik (dalam bahasa manusia).
2. *Summarization* atau Peringkasan: Area ini termasuk aplikasi yang dapat mengambill dokumen atau email, dan menghasilkan ringkasan yang jelas dari konten tersebut. Seperti program untuk merubah dari konten paragraph yang panjang menjadi beberapa silde presentasi.
3. *Machine Translation* atau mesin translasi: Dalam ruang lingkup NLP, area ini merupakan area yang menduduki urutan pertama dari sisi riset dan pengembangan. Program seperti Google Translate semakin hari semakin baik.
4. *Speech Recognition* atau Pengenalan Pembicaraan: Hal yang satu ini merupakan masalah yang paling sulit di dunia NLP. Dalam area ini ada *progress* yang bagus dalam membangun model yang dapat digunakan pada ponsel atau komputer untuk mengenali bahasa yang diucapkan yang berupa pertayaan atau perintah. Namun sayang sekali, ketika sistem *Automatic Speech Recognition* (ASR) tersebar dimana-mana, Mereka bekerja paling baik dalam domain yang didefinisikan secara sempit dan tidak membiarkan pembicara menyimpang dari masukan tertulis yang diharapkan.
5. *Documet Classification* atau Klasifikasi Dokumen: Hal yang satu ini merupakan area paling sukses dari NLP, ketika tugas untuk mengidentifikasi dimana kategori dari sebuah dokumen harus termasuk. Ini terbukti sangat bermanfaat untuk aplikasi seperti *spam filtering*, Klasifikasi artikel berita, dan ulasan film. Salah satu alasan mengapa hal ini menjadi dampak besar adalah model pembelajaran yang dibutuhkan

relative sederhana untuk melatih algoritma yang mampu melakukan klasifikasi.

2.2 Pengertian *Natural Language Generation*

Natural Language Generation (NLG) adalah bagian dari *Artificial Intelligence* (AI) yang mampu membangkitkan bahasa alami sebagai *output*. Sedangkan menurut (McDonald, 1987), NLG adalah proses menyusun teks bahasa alami untuk memenuhi tujuan komunikatif tertentu. *Natural Language Understanding* (NLU) melingkup segala topik mengenai komunikasi dari bahasa manusia ke mesin, sementara NLG melingkup komunikasi dari mesin ke bahasa manusia.

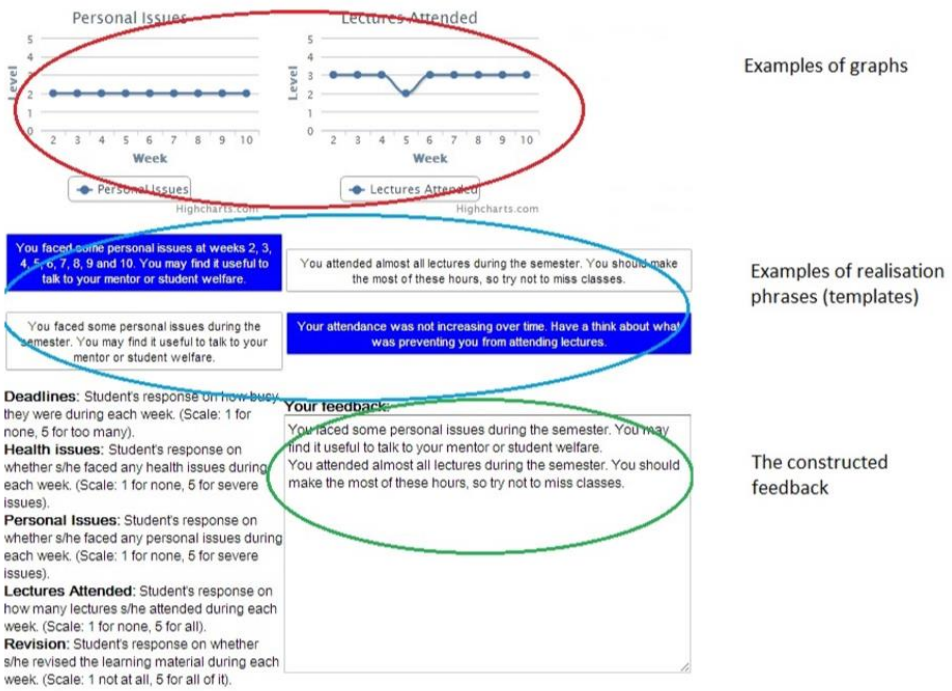
2.3 Pengertian dan Arsitektur *Data-to-text*

Sistem *Data-to-text* (D2T) merupakan sistem *Natural Language Generation* (NLG) yang mampu menghasilkan teks dari *input* data non-linguistik, seperti data sensor dan *event log* (Reiter, E, 2007). Dengan kemampuannya mendeskripsikan data non-linguistik menjadi bahasa alami, *Data-to-text* memudahkan manusia untuk memahami informasi yang disampaikan. Dikarenakan untuk memahami sebuah data yang berbentuk numerik ataupun data yang berasal langsung dari sebuah sensor diperlukan waktu dan pengetahuan yang cukup untuk memahaminya, berbeda dengan data yang bebentuk teks, pembaca akan lebih mudah memahami dan menangkap informasi jika data yang disajikan berbentuk teks. Disamping menampilkan informasi sistem D2T juga harus dapat menyampaikan *knowledge* untuk user yang menggunakannya.

Sistem D2T ini menjadi solusi di beberapa bidang yang membutuhkan keluaran berupa informasi atau report yang bentuknya tekstual. Contohnya pada bidang pendidikan, Aplikasi yang diperkenalkan oleh (Gkatzia, dkk., 2013) dapat menghasilkan *feedback* untuk siswa dari hasil kuisioner yang diedarkan setiap 1 minggu sekali. Contoh lainnya dalam salah satu aplikasi pengontrol kesehatan, yaitu *BABYTALK family Systems,* diperkenalkan oleh (Potret, dkk.,

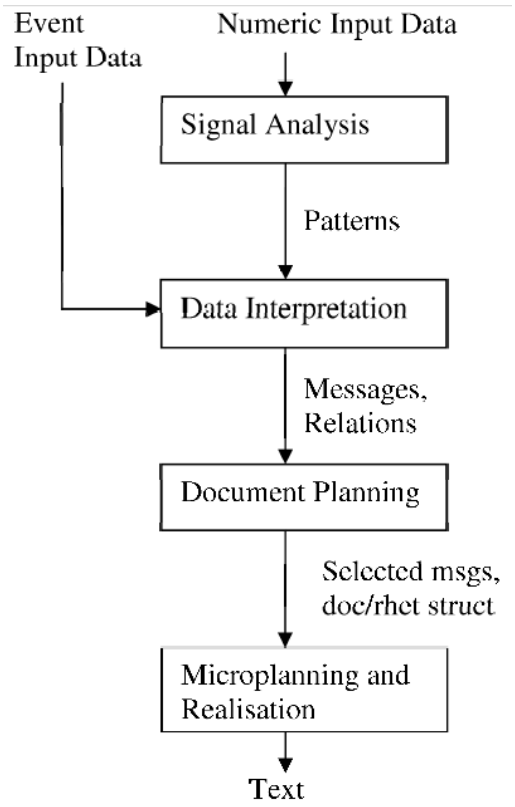
2009), aplikasi ini mampu membuat sebuah ringkasan peristiwa yang terjadi selama 45 menit dari sinyal psikologis kontiyu dan diskrit, seperti pengaturan peralatan dan pemberian obat dalam bentuk kalimat. Selain itu, penerapan ini juga telah digunakan untuk menghasilkan ringkasan dari pergantian keperawatan hanya dari sistem pencatatan pasien elektronik, di Neonatal Intensive Care *Unit* (NICU) (Hunter, dkk., 2012). Beberapa contoh diatas memperlihatkan bahwa penerapan dari *Data-to-text* dapat menjadi sebuah riset yang dapat bermanfaat bagi berbagai bidang, seperti bidang kesehatan, bidang sosial, bidang ekonomi, dan sebagainya.

Pada Gambar 2.1 dapat dilihat contoh singkat untuk menjelaskan konteks dasar dari sistem *Data-to-text.* Gambar tersebut menunjukan skema *input output* dari sistem *Data-to-text.* dapat dilihat bahwa tujuan utama sistem *Data-to-text* adalah untuk membangkitkan bahasa alami dengan *input* data mentah atau numerik.



Gambar 2.1 Contoh *input-output* sistem *Data-to-text.*

Salah satu arsitektur dari *Data-to-text* telah dipaparkan dalam (Potret, dkk., 2009) yang diterapkan didalam sebuah apliikasi yang bernama *BABYTALK.* Arsitektur *Data-to-text* dari aplikasi *BABYTALK* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Arsitektur sistem *Data-to-text* yang dikembangkan oleh Reiter. (Reiter, E, 2007)

Dari Gambar 2.2 dapat dilihat bahwa arsitektur yang diusulkan terdiri dari empat tahapan utama, tahapan-tahapan tersebut terdiri dari:

1. Analisis sinyal

Dalam arsitektur yang dalam arsitektur sistem *Data-to-text* yang telah diimpementasikan dalam (Potret, dkk., 2009) ini, hal pertama yang harus dilakukan untuk membangun sistem *Data-to-text* adalah dengan melakukan

analisis sinyal, yaitu mencoba untuk mendeteksi adanya pola sederhana dari *input* data numerik. Tujuan utama dari analisis sinyal adalah untuk mengganti data numerik tersebut menjadi sebuah pola diskrit. Hal ini menjadikan sistem dapat menggunakan pola secara simbolis (diskrit) dari pada dalam bentuk numerik.

Intinya, proses analisis sinyal merupakan proses menganalisis data *input* numerik untuk menghasilkan *output* berupa informasi yang akan disampaikan. Contohnya dalam kasus pembangkitan berita cuaca, *input* yang digunakan merupakan seluruh data cuaca yang terjadi selama satu bulan, lalu data tersebut dianalisis sehingga didapatkan sinya-sinyal dari sekumpulan data tersebut seperti hujan terbesar, hari terkering, dan lain-lain.

2. Interpretasi Data

Langkah kedua setelah mendapatkan sinyal-sinyal dari proses analisis sinyal, yang harus dilakukan kemudian adalah menerjemahkan sinyal-sinyal yang telah didapatkan tersebut kedalam pesan dan menganalisis apakah ada relasi antara pesan-pesan yang didapatlan. Jadi, tujuan utama dari *Data Interpretation* ini adalah untuk memetakan pola dan *event* dasar menjadi pesan dan relasi dimana manusia membutuhkannya.

Sebagai contoh, misalnya terdapat data suhu udara selama satu minggu, dari hasil analisis sinyal didapatkan bahwa ternyata suhu paling panas pada data tersebut adalah senilai 35^{o}C. Maka dengan melalui serangkaian proses interpretasi data ini, angka 35^{o}C diiterpretasikan menjadi pesan *"extremely hot"*.

3. Perencanaan Dokumen

Langkah ketiga yang dilakukan dalam arsitektur ini adalah menentukan *event* mana yang akan disebutkan didalam teks, dan juga didalam struktur dokumen. Analisis sinyal dan *Data Interpretation* dapat menghasilkan sejumlah pesan, pola, dan *event* yang banyak, tetapi teks biasanya terbatas untuk mendeskripsikan sebagian kecil pesan. Perencanaan dokumen harus

menentukan pesan mana yang sebenarnya dapat dikomunikasikan dalam bentuk teks, pilihan ini didasarkan pada genre dan domain. Dalam langkah ini juga harus direncanakan bagaimana pesan disebutkan dalam sebuah teks yang berkaitan antara satu dengan yang lainnya.

Dalam (Reiter dan Dale, 2000) dipaparkan bahwa serangkaian proses *Document Planning* ini diantaranya adalah membagi tugas menjadi beberapa bagian berikut:

a. Content Determintation

Tahap ini melakukan pemilihan *event*atau pesan yang didapatkan, idenya adalah membagi status pesan menjadi *Routine Message* dan *Significant Event Message*. *Routine Message* merupakan pesan-pesan yang akan selalu disampaikan disetiap pembangkitan kalimat, sedangkan *Significant Event Message* adalah pesan-pesan yang hanya akan disampaikan jika dan hanya jika indikasi pembangkitan dipenuhi. Artinya, *Significant Event Message* hanya disampaikan saat kondisi tertentu.

b. Documenet Structuring

Dalam (Reiter dan Dale, 2006) dijelaskan bahwa proses menentukan bagaimana struktur pesan yang akan disampaikan. Urutan pesan-pesan ditentukan sesuai dengan relasinya masing-masing. Ada beberapa cara untuk membuat struktur dokumen, salah satunya adalah dengan menggunakan skema.

4. *Microplanning* dan Realisasi

Langkah ke-empat adalah membangkitkan bahasa alami dalam bentuk teks didasarkan pada konten dan struktur yang dipilih pada tahap perencanaan dokumen. Tahap *Microplanning* dan realisasi harus menentukan bagaimana sebenarnya mengekspresikan apa yang telah disusun pada tahap-tahap sebelumnya (Perencanaan dokumen, interpretasi data dan analisis sinya).

Dalam proses *Microplanning*, pesan-pesan yang disampaikan akan melalui serangkaian proses berikut:

a. Lexicalisation

Proses *lexicaisation* adalah bagaimana melakukan pemilihan kata atau frase yang akan digunakan dalam mendeksripsikan segala hal, contohnya mendeskripsikan relasi, tren, dan kemungkinan.

b. Aggregation

Proses *aggregation* adalah bagaimana setiap kata digabungkan menjadi frase, bagaimana frase dihubungkan menjadi kalimat, dan bagaimana kalimat digabungkan menjadi paragraf. Intinya, proses *Aggregation* adalah menghubungkan pesan yang didapat dengan menggunakan beberapa teknik. Ada beberapa teknik yang dapat dilakukan untuk proses *aggregation,* salah satu diantaranya adalah dengan menggunakan *simple conjunction.*

c. Referring Expression Generation

Proses ini berisi mengenai bagaimana sistem dapat merujuk informasi tertentu kepada sebuah subjek. Contohnya: “Bandung pada hari ini diterpa hujan badai”, sistem dikondisikan agar dapat menyampaikan bahwa informasi “hujan badai” adalah penjeasan informasi dari subjek “Bandung”.

Sedangkan tujuan dari proses *Realisation* adalah untuk menghasilkan teks aktual. Proses *Realisation* ini terdiri dari serangkaian proses berikut.

d. *Structure Realisation*

Dalam (Reiter dan Dale, 2006) Pada proses ini, setiap struktur yang telah dibuat dalam proses *dokumen planning* direalisasikan sehingga menghasilkan teks dalam bentuk aktual. Contohnya, merealisasikan struktur teks dalam bahasa pemrograman menjadi teks aktual dalam HTML, LaTeX, RTF, SABLE, dan lain-lain.

2.4 Penelitian Terkait *Data-to-text*

Penelitian terkait dengan sistem *Data-to-text* akhir-akhir ini telah menjadi perhatian tertentu bagi para peneliti, ditunjukan dengan banyaknya penelitian baru terkait dengan bidang ini (D2T dan NLG). Beberapa penelitian sejauh ini mengenai *Data-to-text* dapat dilihat pada Tabel Tabel 2.1.

Tabel 2.1 Penelitian terkait *Data-to-text* dan *Natural Language Generation*

Aplikasi / Penulis	**Metode *Content Selection***	**Domain**	**Sumber Data**
(Kukich, 1983)	Rule-Based	Market	Database
(Gong Junpeng, 2017)	Rule-Based	Sport News	Sport Bureau, Website Crawling
Gkatzia, dkk., 2016)	Rule-Based	Education	Student Feedback
(Sripada, dkk., 2001)	Two Stage model: (1) Domain Reasoner; (2) communicative reasoner	Weather, Oil Rigs	Sensor data, Numerical Data
(Sripada S, dkk., 2003)	Gricean Maxims	Weather, Gas Turbins, Heatlh	Sensor data
(Hallet, C. dkk., 2006)	Rule-Based	Heatlh	Database
(Yu, dkk., 2007)	Rules derived from corpus analysis and main knowledge	Gas Tourbines	Sensor

(Sripada dan Gao, 2007)	Decompression Models	Dive	Sensor
(Turner R, dkk., 2008)	Decision Tree	Georeferenced Data	Database
(Gatt, A, dkk., 2009)	Rule-Based	Health	Sensor
(Thomas, dkk., 2010)	Document Schema	Georeferenced Data	Datatbase
(Demir, dkk., 2011)	Rule-based	Domain Indepentent	Graph-database
(Peddington dan Tintarev, 2011) dan (Tintarev, dkk., 2016)	Threshold-based rules	Assitive Technology	Sensor
(Johnson dan Lane, 2011)	Search Algorithm	Autonomous Underwater vehicle	Sensor
(Banaee, dkk., 2013)	Rule-based	Health	Grid of sensor
(Schneider, dkk., 2013)	Rule-based	Health	Sensor
(Ramos-soto, dkk., 2015)	*Fuzzy*-sets	Weather	Database

(Gkatzia, dkk., 2016)	Rule-based	Weather	Numerical data with assigned probabilities

2.5 Pengertian dan Sejarah *Machine Learning*

Machine Learning adalah bagian dari ilmu komputer yang dapat membelajarkan komputer sehingga memiliki kemampuan untuk belajar tanpa diprogram secara eksplisit (Arthur, 1959). *Machine Learning* merupakan bagian dari kecerdasan buatan yang berfokus dalam mempelajari, mendesain, dan membuat sebuah algoritma yang memiliki kemampuan untuk belajar dari data yang ada. Agar sebuah perangkat memiliki kecerdasan, maka komputer atau mesin tersebut harus dapat belajar. Dengan kata lain, *Machine Learning* berisi tentang keseluruhan proses pembelajaran komputer atau mesin menjadi cerdas dan dapat belajar dari data. *Machine Learning* sudah ada dan mulai digunakan sejak 50 tahun yang lalu dan sudah banyak digunakan di berbagai bidang. Contohnya pada bidang ekonomi, keilmuan, industri dan sebagainya.

Salah satu implementasi *Machine Learning* yang pernah dilakukan oleh Arthur Samuel sekitar 57 tahun yang lalu yaitu pembuatan permainan catur dengan komputer. Catur dipilih karena permainan sangat mudah tetapi memerlukan strategi yang bagus. Samuel membuat permainan catur ini berdasarkan pohon penyelesaian. Pencarian penyelesain dilakukan dengan menyusuri pohon permasalahan sampai mendapatkan solusinya.

Awal ditemukannya Machine Learning yaitu pada abad ke-20, seorang ilmuan dari Spanyol, Torres y Quevedo, membuat sebuah mesin catur yang dapat mengalahkan atau melakukan skakmat pada raja lawan dengan sebuah

ratu dan raja. Perkembangan secara sistematis kemudian dimulai segera setelah diketemukannya komputer digital.

Artikel ilmiah pertama tentang Kecerdasan Buatan ditulis oleh Alan Turing pada tahun 1950, dan kelompok riset pertama dibentuk tahun 1954 di Carnegie Mellon University oleh Allen Newell and Herbert Simon. Namun bidang Kecerdasan Buatan baru dianggap sebagai bidang tersendiri di konferensi Dartmouth tahun 1956, di mana 10 peneliti muda memimpikan mempergunakan komputer untuk memodelkan bagaimana cara berfikir manusia. Mereka berhipotesis bahwa *Mekanisme berfikir manusia dapat secara tepat dimodelkan dan disimulasikan pada komputer digital*.

Machine Learning memiliki beberapa tipe dengan proses pembelajaran yang berbeda, tipe-tipe tersebut akan dijelaskan pada sub-bab berikutnya.

2.5.1 *SupervisedLearning*

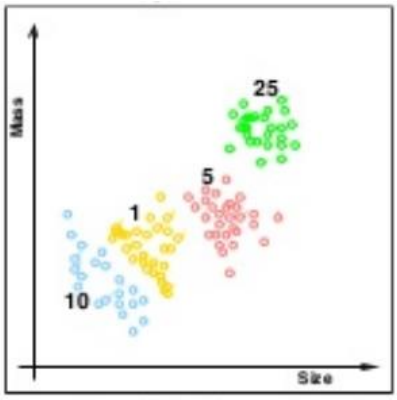
Tugas dari *Supervised Learning* terdiri dari pembangunan model yang memetakan nilai *input* pada nilai ouput dimana *data* training tersedia (Riza, 2015). *Supervised Learning* adalah *Machine Learning* yang membutuhkan label sebagai tujuan dari pelatihan data atau*data training* (Mohri, Rostamizadeh, dan Talwalkar, 2012). *Supervised Learning* merupakan suatu pembelajaran yang terawasi, dimana jika *output* yang diharapkan telah diketahui sebelumnya. Pada metode ini, setiap pola yang diberikan kedalam model *Machine Learning* telah diketahui *output*nya. Contoh algoritma dari salah satu bagian dari *Machine Learning* yaitu jaringan saraf tiruan yang mernggunakan metode *Supervised Learning* adalah hebbian (hebb rule), perceptron, adaline, boltzman, hapfield, dan backpropagation.

Berikut ini adalah beberapa contoh penerapan tipe *Machine Learning*, *Supervised Learning*:

- Klasifikasi: sebuah metode untuk menyusun data secara sistematis menurut aturan-aturan yang telah ditetapkan sebelumnya (Muhammad, Irawan, dan

Matu, 2015). Dengan melakukan klasifikasi, dari data yang telah ada dapat dibuat sebuah model prediksi dengan *output* kelas.

- Regresi: Analisis regresi adalah salah satu metode statistik untuk memprediksi nilai dari satu atau lebih variabel respon/dependen dari satu set variabel prediktor/independen (Johnson dan Wichern, 1982).



Gambar 2.8 Contoh *Supervised learning* pada pengenalan koin.

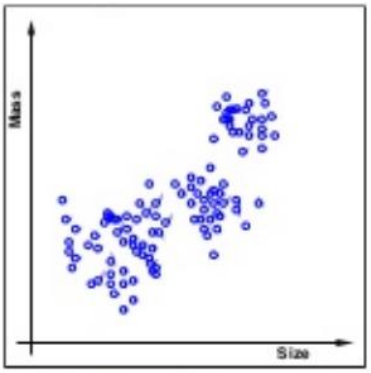
Pada Gambar 2.8, diperlihatkan bagaimana klasifikasi dari pengenalan koin, terlihat sangat jelas lokasi bagian dari tiap kelas, seperti koin dengan nilai sepuluh terpisah dipaling bawah dengan warna biru, koin dengan nilai satu yang berwarna kuning tidak bercampur dengan yang lainnya, dan seterusnya.

2.5.2 *UnsupervisedLearning*

*Unsupervised Learnin g*terdiri dari pembangunan model dari *data training* dengan tidak mengandung nilai *output* (Riza, 2015). *Unsupervised Learning* merupakan pembelajan yang tidak terawasi dimana tidak memerlukan target *output*. Teknik ini menggunakan prosedur yang berusaha untuk mencari partisi dari sebuah pola. *Unsupervised Learning* mempelajari bagaimana sebuah sistem dapat belajar untuk merepresentasikan pola *input* dalam cara yang menggambarkan struktur statistikal dari keseluruhan pola *input*. Berbeda

dari *Supervised Learning*, *Unsupervised Learning* tidak memiliki target *output* yang eksplisit atau tidak ada pengklasifikasian *input*.

Dalam *Machine Learning*, teknik *Unsupervised* sangat penting. Hal ini dikarenakan cara kerjanya mirip dengan cara bekerja otak manusia. Dalam melakukan pembelajaran, tidak ada informasi dari contoh yang tersedia. Oleh karena itu, *Unsupervised Learning* menjadi esensial. Pada metode ini tidak dapat ditentukan hasil seperti apa yang diharapkan selama proses pembelajaran, nilai bobot yang disusun dalam proses range tertentu tergantung pada nilai *output* yang diberikan. Tujuan metode *Unsupervised Learning* ini agar kita dapat mengelompokkan *Unit-Unit* yang hampir sama dalam satu area tertentu. Pembelajaran ini biasanya sangat cocok untuk klasifikasi pola. Contoh algoritma jaringan saraf tiruan yang menggunakan metode *Unsupervised* ini adalah competitive, hebbian, kohonen, *Learning Vector Quantizatio (*LVQ), dan neocognitron.



Gambar2.9 Contoh *Unsupervised Learning* dalam pengenalan koin

Salah satu contoh dari *Unsupervised Learning* adalah clustering, sistem diharapkan mampu untuk memisahkan data serupa ke dalam kelompoknya masing-masing, seperti pada Gambar 2.9, belum diketahui kelas dari masing-masing data, mesinlah yang menentukan berdasarkan kedekatannya.

2.5.3 Algoritma *Gradient Descent*

Algoritma *Gradient Descent* adalah algoritma optimasi untuk menemukan *minimum* lokal dari fungsi menggunakan *gradien descent*, diambil langkah sebanding dengan negatif dari gradien (atau perkiraan gradien) dari fungsi pada titik sekarang. Jika diambil langkah sebanding dengan gradien positif, maka akan didapatkan maksimum lokal fungsi tersebut; prosedur ini kemudian dikenal sebagai *gradient ascent*. *Gradient descent* juga dikenal sebagai steepest descent, sedangkan *gradient ascent* dikenal dengan *steepestascent*.

2.6 *Time-series* Data

Kumpulan data yang tercatat dalam periode waktu mingguan, bulanan, kuartalan, atau tahunan (Mishra dan Jain, 2014). Ada 4 faktor yang mempengaruhi data *Time Series*. Dalam data ekonomi biasanya didapatkanadanya fluktuasi atau variasi dari waktu ke waktu atau disebut dengan variasi *Time Series*. Variasi ini biasanya disebabkan oleh adanya faktor *Trend* (*trend factor*), Fluktuasi siklis (*cyclical fluktuation*), Variasi musiman (*seasonal variation*), dan pengaruh *random* (*irregular atau random influences*).

Trend adalah keadaan data yang menaik atau menurun dari waktu ke waktu. Contoh yang menunjukkan trend menaik yaitu pendapatan per-kapita, jumlah penduduk. **Variasi musiman** adalah fluktuasi yang muncul secara reguler setiap tahun yang biasanya disebabkan oleh iklim, kebiasaan (mempunyai pola tetap dari waktu ke waktu). Contoh yang menunjukan variasi musiman seperti penjualan pakaian akan meningkat pada saat hari raya, penjualan buku dan tas sekolah akan meningkat pada saat awal sekolah.

Variasi siklis muncul ketika data dipengaruhi oleh fluktuasi ekonomi jangka panjang, variasi siklis ini bisa terulang setelah jangka waktu tertentu. Variasi siklis biasanya akan kembali normal setiap 10 atau 20 tahun sekali, bisa juga tidak terulang dalam jangka waktu yang sama. ini yang membedakan antara variasi siklis dengan musiman. Gerakan siklis tiap komoditas mempunyai jarak waktu muncul dan sebab yang berbeda-beda, yang sampai saat ini belum dapat dimengerti. Contoh yang menunjukkan variasi siklis seperti industri konstruksi

bangunan mempunyai gerakan siklis antara 15-20 tahun sedangkan industri mobil dan pakaian gerakan siklisnya lebih pendek lagi. **Variasi *random*** adalah suatu variasi atau gerakan yang tidak teratur (*irregular*). Variasi ini pada kenyataannya sulit diprediksi. Contoh variasi ini dalam data *Time Series* karena adanya perang, bencana alam dan sebab-sebab unik lainnya yang sulit diduga. Total variasi dalam data *Time Series* adalah merupakan hasil dari keempat faktor tersebut yang mempengaruhi secara bersama-sama. Dalam tulisan ini hanya akan dianalisa dua variasi pertama, sedangkan dua variasi terakhir tidak dianalisa karena memang pola variasi tersebut tidak tersistem dengan baik selain membutuhkan waktu yang sangat lama untuk mendapatkan data yang panjang.

Model *Time Series* adalah suatu peramalan nilai-nilai masa depan yang didasarkan pada nilai-nilai masa lampau suatu variabel dan atau kesalahan masa lampau. Model *Time Series* biasanya lebih sering digunakan untuk suatu peramalan/prediksi. Dalam tehnik peramal an dengan *Time Series* ada dua kategori utama yang perlu dilakukan pengujian, yaitu pemulusan (*smoothing*) dan dekomposisi (*decomposition*). Metode pemulusan mendasarkan ramalannya dengan prinsip rata-rata dari kesalahan masa lalu (*Averaging smoothing past errors*) dengan menambahkan nilai ramalan sebelumnya dengan persentase kesalahan (*percentage of the errors*) antara nilai sebenarnya (*actual value*) dengan nilai ramalannya (*forecasting value*). Metoda dekomposisi mendasarkan prediksinya dengan membagi data *Time Series* menjadi beberapa komponen dari Trend, Siklis, Musiman dan pengaruh *Random*. Kemudian mengkombinasikan prediksi dari komponen-komponen tersebut (kecuali pengaruh *random* yang sulit diprediksi). Pendekatan lain untuk peramalan adalah metoda causal atau yang lebih dikenal dengan sebutan regresi. Tehnik pemulusan dan regresi akan dibahas pada sesi tulisan yang lain.

2.7 *Autoregresif Integrated Moving Average* (ARIMA)

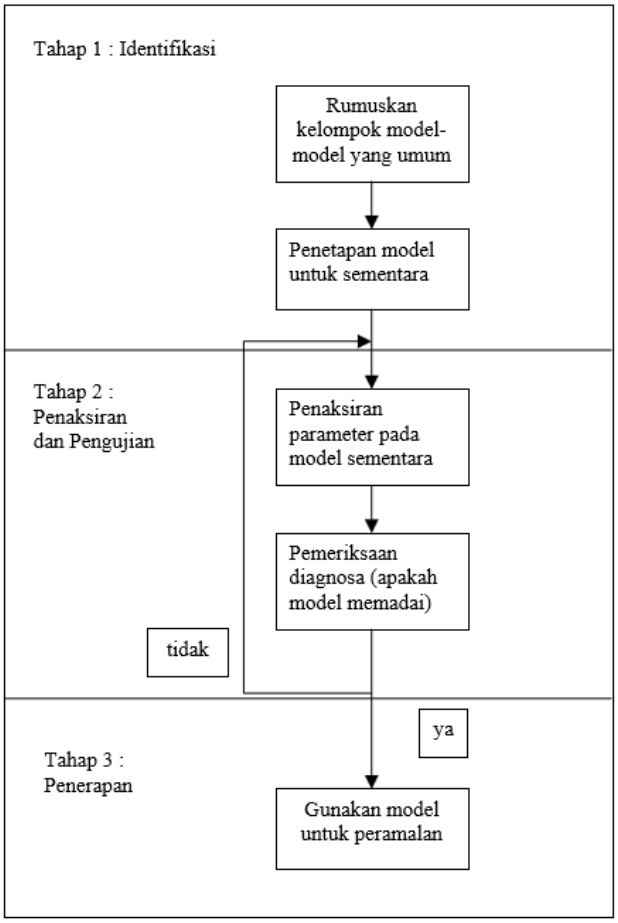
ARIMA sering juga disebut metode runtun waktu Box-Jenkins. ARIMA sangat baik ketepatannya untuk peramalan jangka pendek, sedangkan untuk

peramalan jangka panjang ketepatan peramalannya kurang baik. Biasanya akan cenderung flat (mendatar/konstan) untuk periode yang cukup panjang.

Model 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. *Autoregressive* adalah model terbaik untuk peramalan dengan waktu yang pendek (short-term forcasting). Sedangkan untuk peramalan dengan waktu yang cukup panjang (long-term forcasting) menggunakan proses autoregressive tidak begitu baik (Dickey, 1996).

ARIMA hanya menggunakan suatu variabel (univariate) deret waktu. Misalnya: variabel IHSG. Program komputer yang dapat digunakan adalah EViews, Minitab, SPSS, dll.

Model ARIMA terdiri dari tiga langkah dasar, yaitu tahap identifikasi, tahap penaksiran dan pengujian, dan pemeriksaan diagnostik yang digambarkan pada gambar xx. Selanjutnya model ARIMA dapat digunakan untuk melakukan peramalan jika model yang diperoleh memadai.



Gambar2.9 Contoh *Unsupervised Learning* dalam pengenalan koin

2.7.1. Model *Autoregressive*

Jika series stasioner adalah fungsi linier dari nilai-nilai lampaunya yang berurutan atau nilai sekarang series merupakan rata-rata tertimbang nilai-nilai lampaunya bersama dengan kesalahan sekarang, maka persamaan itu dinamakan model autoregressive.

Bentuk umum model ini adalah (Santoso, 2009):

$$Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + \cdots + A_p Y_{t-p} + e_t$$

Dimana :

Y_t = nilai AR yang di prediksi

Y_{t-1}, Y_{t-2}, Y_{t-n} = nilai lampau series yang bersangkutan ; nilai *lag* dari *time series*.

A_p = koefisien

e_t = residual; error yang menjelaskan efek dari variabel yang tidak dijelaskan oleh model, kesalahan peramalan dengan ciri seperti sebelumnya.

Banyaknya nilai lampau yang digunakan (p) pada model AR menunjukkan tingkat dari model ini. Jika hanya digunakan sebuah nilai lampau, dinamakan model autoregressive tingkat satu dan dilambangkan dengan AR. Agar model ini stasioner, jumlah koefisien model *autoregressive* ($\sum_{i=1}^{n} bi$) harus selalu kurang dari 1. Ini merupakan syarat perlu, bukan cukup, sebab masih diperlukan syarat lain untuk menjamin *stationarity*.

2.7.2 Model *moving average*

Jika series yang stasioner merupakan fungsi linier dari kesalahan peramalan sekarang dan masa lalu yang berurutan, persamaan itu dinamakan *moving average* model.

Bentuk umum model ini adalah (Santoso, 2009):

$$Y_t = e_t - W_1 e_{t-1} - W_2 e_{t-2} - \cdots - W_q e_{t-q}$$

Dimana :

Y_t = nilai MA yang di prediksi

$W_{1,2,q}$ = konstanta; koefisien atau bobot (*weight*)

e_t = residual; error yang menjelaskan efek dari variabel yang tidak dijelaskan oleh model.

Terlihat bahwa Y_t merupakan rata-rata tertimbang kesalahan sebanyak n periode ke belakang. Banyaknya kesalahan yang digunakan pada persamaan ini (q) menandai tingkat dari model *moving average*. Jika pada model itu digunakan dua kesalahan masa lalu, maka dinamakan model *average* tingkat 2 dan dilambangkan sebagai MA. Hampir setiap model *exponential smoothing* pada prinsipnya ekuivalen dengan suatu model ini. Agar model ini stasioner,

suatu syarat perlu (bukan cukup), yang dinamakan *invertibility condition* adalah bahwa jumlah koefisien model ($\sum_{i=1}^{n} wi$) selalu kurang dari 1. ini artinya jika makin ke belakang peranan kesalahan makin mengecil. Jika kondisi ini tak terpenuhi kesalahan yang semakin ke belakang justru semakin berperan.

Model MA meramalkan nilai Y_t berdasarkan kombinasi kesalahan linier masa lampau (lag), sedangkan model AR menunjukkan Y_t sebagai fungsi linier dari sejumlah nilai Y_t aktual sebelumnya.

2.7.3 Model Autregressive Integrated Moving Average (ARIMA)

Model *time series* yang digunakan berdasarkan asumsi bahwa data *time series* tersebut stasioner, artinya rata-rata varian (σ2) suatu data *time series* konstan. Tapi seperti kita ketahui bahwa banyak data *time series* dalam ilmu ekonomi adalah tidak stasioner, melainkan *integrated*. Jika data *time series integrated* dengan ordo 1 disebut I (1) artinya *differencing* pertama. Jika series itu melalui proses *differencing* sebanyak d kali dapat djadikan stasioner, maka series itu dikatakan non-stasioner homogen tingkat d. Seringkali proses random stasioner tak dapat dengan baik dijelaskan oleh model *moving average* saja atau *autoregressive* saja, karena proses itu mengandung keduanya. Karena itu, gabungan kedua model, yang dinamakan *Autregressive Integrated Moving Average* (ARIMA) model dapat lebih efektif menjelaskan proses itu.

Pada model gabungan ini *series* stasioner adalah fungsi dari nilai lampaunya serta nilai sekarang dan kesalahan lampaunya.

Bentuk umum model ini adalah (Santoso, 2009):

$$Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + \cdots + A_p Y_{t-p} - W_1 e_{t-1} - W_2 e_{t-2} - \ldots - W_q e_{t-q}$$

Dimana :

Y_t = nilai series yang stasioner

Y_{t-1}, Y_{t-2} = nilai lampau series yang bersangkutan

e_{t-1}, e_{t-2} = variabel bebas yang merupakan lag dari residual

W_1, W_q, A_1, A_p = koefisien model

2.7.4 Konsep Parsimoni

Pemilihan model juga menggunakan unsur seni disamping ilmu; selain itu factor parsimoni juga perlu di pertimbangkan. Parsimoni adalah konsep yang mengutamankan kesederhanaan sesuatu; dalam ARIMA. Konsep tersebut menekankan lebih baik memilih model dengan parameter sedikit daripada parameter banyak, serta mengutamakan lag yang paling sedikit. (Santoso, 2009)

2.7.5 Stasioner dan Non-stasioner

Ciri-ciri stasioner dalam *time series* adalah nilai rata-rata (*mean*) dan varian selalu konstan untuk setiap periode. Data *time series* yang tidak memeiliki tren disebut stasioner. Stasioner 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. Sebaliknya, data *time series* yang memiliki tren disebut non-stasioner. Indikasi adanya non-stasioner pada data *time series* ditunjukan dengan menurunnya koefisien auto korelasi mendekati nol (0) setelah lag 2 atau lag 3. (Rangkuti, 2005)

Hal yang perlu diperhatikan adalah bahwa kebanyakan deret berkala bersifat non-stasioner dan bahwa aspek-aspek AR dan MA dari model ARIMA hanya berkenaan dengan deret berkala yang stasioner. Jadi 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 kembali apakah stasioner atau tidak. Jika belum stasioner maka dilakukan tranformasi logaritma. (Administrator, 2009)

2.7.6 Pola autokorelasi

Setelah data runtut waktu telah stasioner, langkah berikutnya adalah menetapkan model ARIMA (p,d,q) yang sekiranya cocok (tentatif), maksudnya

menetapkan berapa p, d, dan q. Jika tanpa proses *differencing* d diberi nilai 0, jika menjadi stasioner setelah *first order differencing* d bernilai 1 dan seterusnya.

Dalam (Santoso, 2009) proses ini dilambangkan dengan ARIMA (p,d,q).

Dimana :

q menunjukkan ordo/ derajat autoregressive (AR)

d adalah tingkat proses differencing

p menunjukkan ordo/ derajat moving average (MA)

Simbol model-model sebelum ini dapat saja dinyatakan seperti berikut :

AR sama maksudnya dengan ARIMA (1,0,0),

MA sama maksudnya dengan ARIMA (0,0,2), dan

ARMA sama maksudnya dengan ARIMA (1,0,2).

Mungkin saja terjadi bila suatu series non-stasioner homogen tidak tersusun atas kedua proses itu, yaitu proses *autoregressive* maupun moving average. Jika hanya mengandung proses *autoregressive*, maka series itu dikatakan mengikuti proses *Integrated autoregressive* dan dilambangkan ARIMA (p,d,0). sementara yang hanya mengandung proses *moving average*, seriesnya dikatakan mengikuti proses *Integrated moving average* dan dituliskan ARIMA (0,d,q).

Dalam (Hanke & Wichern, 2003, p. 389) Fungsi Autokorelasi (ACF) dan Fungsi Autokorelasi Parsial (PACF) melalui korelogramnya. ACF mengukur korelasi antar pengamatan dengan jeda k, sedangkan PACF mengukur korelasi antar pengamatan dengan jeda k dan dengan mengontrol korelasi antar dua pengamatan dengan jeda kurang dari k. Untuk memilih berapa p dan q dapat dibantu dengan mengamati pola fungsi *autocorrelation* dan *partial autocorrelation (correlogram)* dari series yang dipelajari, dengan acuan sebagai berikut :

Tabel 2.1 Pola Autokorelasi dan Autokorelasi Parsial

Autocorrelation	*Partial autocorrelation*	

		ARIMA tentatif
Menuju nol setelah lag q	Menurun secara bertahap/bergelombang	ARIMA (0,d,q)
Menurun secara bertahap/bergelombang	Menuju nol setelah lag q	ARIMA (p,d,0)
Menurun secara bertahap/bergelombang (sampai lag q masih berbeda dari nol)	Menurun secara bertahap/bergelombang (sampai lag p masih berbeda dari nol)	ARIMA (p,d,q)

Sumber: (Hanke & Wichern, 2003)

Dalam praktik pola *autocorrelation* dan *partial autocorrelation* seringkali tidak menyerupai salah satu dari pola yang ada pada tabel itu karena adanya variasi *sampling*. Jika sudah terbiasa atau berpengalaman pemilihan p dan q diharapkan dekat dengan yang benar. Perhatikan bahwa kesalahan memilih p dan q bukan merupakan masalah, dan akan dimengerti setelah tahap *diagnostic checking*. Pada umumnya, analis harus mengindentifikasi autokorelasi yang secara eksponensial menjadi nol. Jika autokorelasi secara eksponensial melemah menjadi nol berarti terjadi proses AR. Jika autokorelasi parsial melemah secara eksponensial berarti terjadi proses MA. Jika keduanya melemah berarti terjadi proses ARIMA (Arsyad, 1995).

Data yang bersifat *time series* cenderung memiliki hubungan antar periode. Untuk mengetahui apakah data *time series* tersebut saling berhubungan satu sama lain, kita dapat melakukan analisis autokorelasi. Idealnya, data yang bersifat *time series* harus bebas dari pengaruh autokorelasi. Komponen yang membentuk pola tertentu pada data *time series* diakibatkan oleh pengaruh tren, kecenderungan musiman, serta ketidakajegan. Semuanya dapat dipelajari dengan menggunakan analisis koefisien autokorelasi, baik bersifat natural logs maupun berbagai senjang waktu yang berbeda (*time lags*). (Rangkuti, 2005, p. 29)

Dikemukakan *There may be some ambiguity in determining an appropriate ARIMA model from the pattern of the sample autocorrelation and partial autocorrelation. With a little practice, the analys should become more adept at identifying an adequate model.* (Hanke & Wichern, 2003). Terdapat keambiguan dalam menetukan model ARIMA yang tepat dari contoh autokorelasi dan autokorelasi parsial. Dengan banyak latihan, analis dapat menjadi lebih mahir dalam mengidentifikasi model yang memenuhi syarat.

2.7.7 Menghitung Kesalahan Peramalan

Dalam (Santoso, 2009, p. 172) penggunaan ARIMA dengan MINITAB PEMILIHAN model terbaik adalah model degan tingkat kesalahn prediksi terkecil. Acuannya adalah MS (means of square; Adalah rata-rata selisih kuadrat nilai yang diramalkan dan yang diamati). Namun dalam text book lain ada beberapa tekhnik untuk menghitung kesalahan peramalan.

Menurut (Weiers, 2011) ada beberapa teknik untuk mengevaluasi hasil peramalan, diantaranya :

- *Mean Absolute Deviation* (MAD) atau simpangan absolut rata-rata

  $MAD = \frac{\sum_{t-1}^{n}(Y_t - \hat{Y}_t)}{n}$

  MAD ini sangat berguna jika seorang analis ingin mengukur kesalahan peramalan dalam unit ukuran yang sama seperti data aslinya.

- *Mean Squared Error* (MSE) atau Kesalahan rata-rata kuadrat

  $MSE = \frac{\sum_{t-1}^{n}\left(Y_t - \hat{Y}_t\right)^2}{n}$

  Pendekatan ini menghukum suatu kesalahan yang besar karena dikuadratkan. Pendekatan ini penting karena satu teknik yang menghasilkan kesalahan yang moderat yang lebih disukai oleh suatu peramalan yang biasanya menghasilkan kesalahan yang lebih kecil tetapi kadang-kadang menghasilkan kesalahan yang

sangat besar. Pendekatan inilah yang nantinya akan muncul dalam pehitungan dengan MINITAB.

2.8 R *Programming*

Bahasa R merupakan sebuah proyek yang dirancang sebagai bahasa pemrograman yang gratis, *open source*, yang dapat digunakan sebagai pengganti dari bahasa pemrograman Splus, pada mualnya dikembangkan sebagai bahasa S di *AT&T Bell Labs*, dan sekarang dipasarkan oleh *Insightful Corporation of Seattle*, di Washington(Spector, 2004). R adalah sistem untuk komputasi statistic dan grafik. Sebagai sebuah sistem, R memiliki banyak sekali fitur. Sebagai bahasa pemrograman, R memiliki visualisasi grafik yang *high level*, antarmuka ke bahasa pemrograman lain, dan fasilitas *debugging*. Logo dari bahasa pemrograman R sendiri dapat dilihat pada Gambar 2.12.

Gambar 2.12 Logo bahasa pemrograman R

Berikut adalah kelebihan dari penggunaan bahasa R menurut (Ihaka dan Gentleman, 1996):

1. Serba guna (*versatile*)

R adalah bahasa pemrograman, sehingga tidak ada batasan bagi pengguna untuk memakai prosedur yang hanya terdapat pada paket-paket yang standar. Bahkan pemrograman R adalah berorientasi obyek dan memiliki banyak library yang sangat bermanfaat yang dikembangkan oleh kontributor. Pengguna bebas menambah dan mengurangi library tergantung kebutuhan. R juga memiliki interface pemrograman C, phyton, bahkan java yang tentu saja berkat jerih payah kontributor aktif proyek R. Jadi selain bahasa R ini cukup pintar, penggunanya pun

bisa menjadi lebih pintar dan kreatif. Beberapa analisis yang membutuhkan fungsi lanjutan memang ada yang belum tersedia dalam R. Tidak berarti R tidak menyediakan fasilitas tersebut, namun lebih karena faktor waktu. Jadi hanya menunggu waktu saja *package* lanjutan tersebut tersedia.

2. Interaktif (*interactive*)

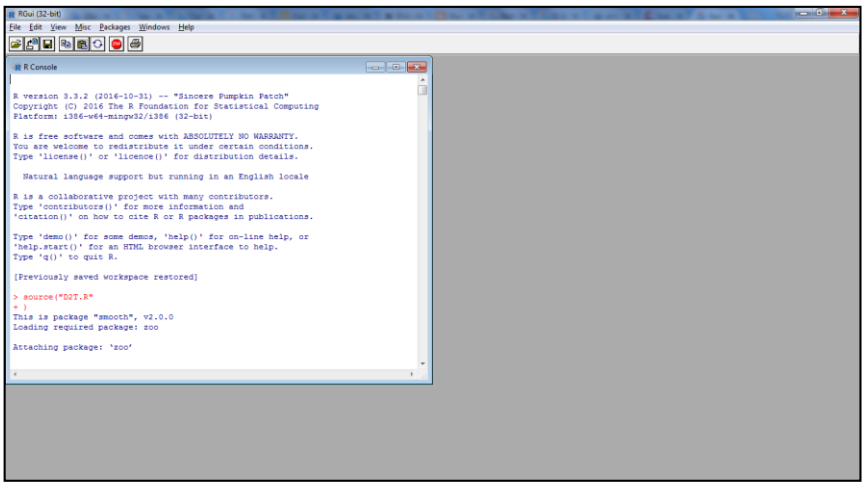
Pada saat ini analisis data membutuhkan pengoperasian yang interaktif. Apalagi jika data yang dianalisis adalah data yang bergerak. R dilengkapi dengan konektivitas ke database server, olap, maupun format data web service seperti XML, spreadsheet dan sebagainya. Sehingga apabila data set berubah hasil analisis pun dapat segera ikut berubah (*real time*).

3. Berbasis S yaitu turunan dari tool statitistik komersial S-Plus.

R hampir seluruhnya kompatibel dengan S-Plus. Artinya sebagian besar kode program yang dibuat oleh S dapat dijalankan di S-plus kecuali fungsi-fungsi yang sifatnya *add-on packages* atau tambahan yang dibuat oleh kontributor proyekR.

4. Populer.

Secara umum SAS adalah *software* statistika komersial yang populer, namun demikian R atau S adalah bahasa yang paling populer digunakan oleh peneliti di bidang statistika. Beberapa tulisan berupa jurnal statistika mengkonfirmasi kebenaran hal ini. R juga populer untuk aplikasi kuantitatif dibidang keuangan.

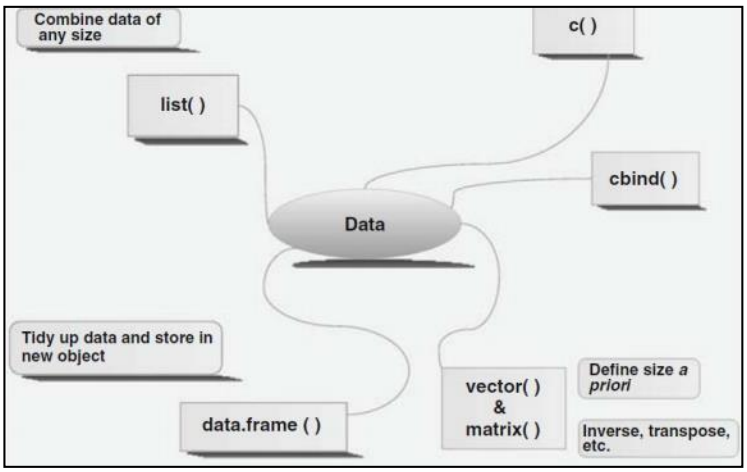


Gambar 2.13 Antarmuka R *Graphical User Interface* (RGui).

RGui merupakan tools dalam pemrograman R, antar muka tools ini dapat dilihat pada Gambar 2.13. Dalam Gambar 2.13, diperlihatkan bahwa dalam antarmuka RGui terdapat layar *console* yang berfungsi untuk memasukan perintah, terdapat *menu-bar*, *tool-bar*dan lain-lain sesuai dengan fungsinya masin-masing.

2.8.1 Model data dalam R

Menurut (Budiharto dan Rachmawati, 2013) Pada bahasa R, data dipandang sebagai suatu objek yang memiliki suatu atribut dan berbagai fungsionalitas. Sifat data ditentukan oleh type data dan mode data. Ada berbagai type data yang dikenal oleh R, antara lain vektor, matriks, list, data frame, *array*, *factor* dan fungsi *built in*. Berikut ini beberapa model data yang umum digunakan serta contoh penerapan fungsi *built in*. Untuk menyimpan data di R ada berbagai metode seperti menggunakan fungsi c(), list(), cbind() dan data.frame() seperti Gambar 2.14.



Gambar 2.14 Model data dalam pemrograman R

(Budiharto dan Rachmawati, 2013)

2.8.2 Contoh kode program bahasa R

Berikut adalah beberapa contoh kode program yang dapat dilakukan oleh bahasa pemrograman R:

- Menggabungkan data

  Untuk menggabungkan data dalam bahasa R, dapat menggunakan fungsi *concantenate* (c). contoh dari penggunaan fungsi ini dapat dilihat pada Gambar 2.15.

```
> x <- c(1,2,3,4,5)
> y <- c(6,7,8,9)
> z <- c(x,y)
> z
[1] 1 2 3 4 5 6 7 8 9
```

Gambar 2.15 Operator *concantenate* dalam R

Pada Gambar 2.16 berikut adalah contoh untuk menampilkan dua data pertama dalam vektor z yang telah dibuat.

```
> x[1:2]
[1] 1 2
```

Gambar 2.16 Menampilkan dua data pertama dalam R

Selain itu, untuk menampilkan jumlah dari seluruh elemen, dapat digunakan fungsi sum. Implementasi dari fungsi sum ini dapat dilihat pada Gambar 2.17.

```
> sum(x)
[1] 15
```

Gambar 2.17 Penggunaan fungsi *summarize* dalam R

Contoh lainnya, untuk memasukan data string, dapat dilihat pada Gambar 2.16 dibawah ini.

```
> data1 = c("aa","bb","cc","dd","ee")
> data2 = c(10,20)
>
> c(data1,data2)
[1] "aa" "bb" "cc" "dd" "ee" "10" "20"
>
```

Gambar2.18 Penggabungan data dengan *concantenate* dalam R

- Membuat *matriks*

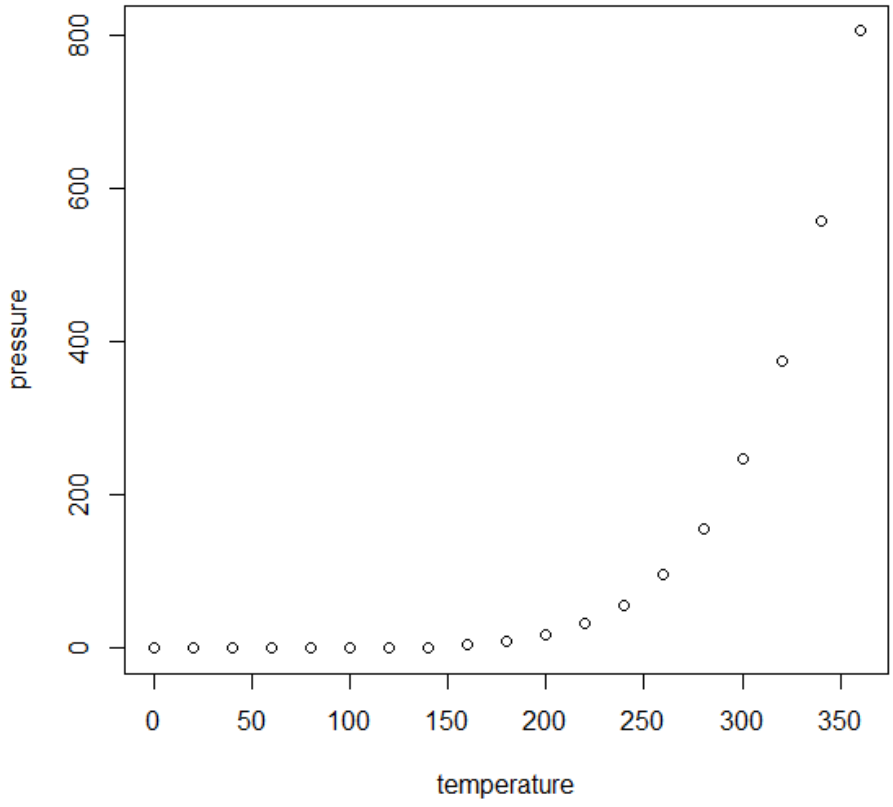
Matriks adalah data dua dimensi dimana sebagian besar fungsi-fungsi statistik dalam R dapat dianalisis dengan menggunakan bentuk matriks. Bentuk matriks ini juga banyak digunakan pada operasi fungsi-fungsi built-in untuk aljabar linear dalam R, seperti untuk penyelesaian suatu persamaan linear. Argumen yang diperlukan adalah elemen-elemen dari matriks, dan argumen optional yaitu banyaknya baris dan banyaknya kolom. Berikut contohnya ada pada Gambar 2.19.

```
> m <- matrix(c(1,2,3,4,5,6), nrow=2, ncol=3)
> m
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Gambar 2.19 Pembuatan *matriks* dalam R

2.8.3 Contoh visualisasi data dalam R

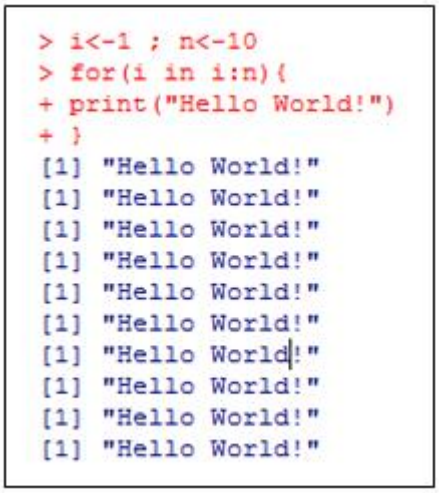
Salah satu keunggulan dalam bahasa pemrograman R adalah visualisasi data dapat disajikan dengan mudah. Data yang berhasil dientri atau diimport dari aplikasi lain selayaknya divisualisasikan pada grafik untuk analisa. Sebagai contoh, kita dapat menggunakan data dari R yaitu variabel pressure, dengan command "*plot(pressure)"* maka akan menghasilkan grafik seperti pada Gambar 2.20.



Gambar 2.20 Contoh visualisasi grafis dalam R

- Membuat perulangan

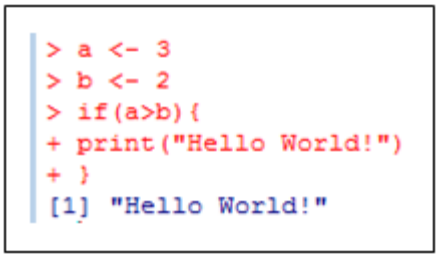
Salah satu cara yang paling populer hampir diseluruh bahasa pemrograman dalam melakukan perulangan adalah fungsi FOR. Contoh implementasi fungsi FOR dalam bahasa R dapat dilihat pada Gambar 2.21.



Gambar 2.21 Contoh perulangan dalam R

- Membuat *decision*

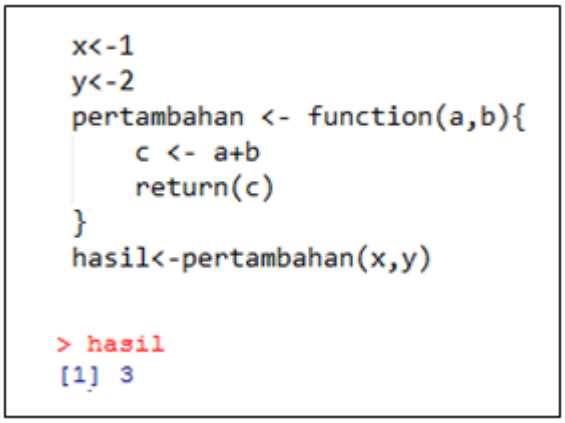
Membuat *decision*dalam dunia pemrograman adalah hal yang paling utama. Dalam bahasa R, membuat decision identik dengan bagaimana bahasa C melakukannya. Dapat dilihat pada Gambar 2.22.



Gambar 2.22 Contoh implementasi *decision* dalam R

- Membuat Fungsi

  Dalam pemrograman terstruktur, salah satu hal yang penting adalah membuat fungsi. Dalam bahasa R, contoh pembuatan fungsi dapat dilihat pada Gambar 2.23.



Gambar 2.23 Contoh fungsi dalam R

2.8.4 *Package* alam bahasa R

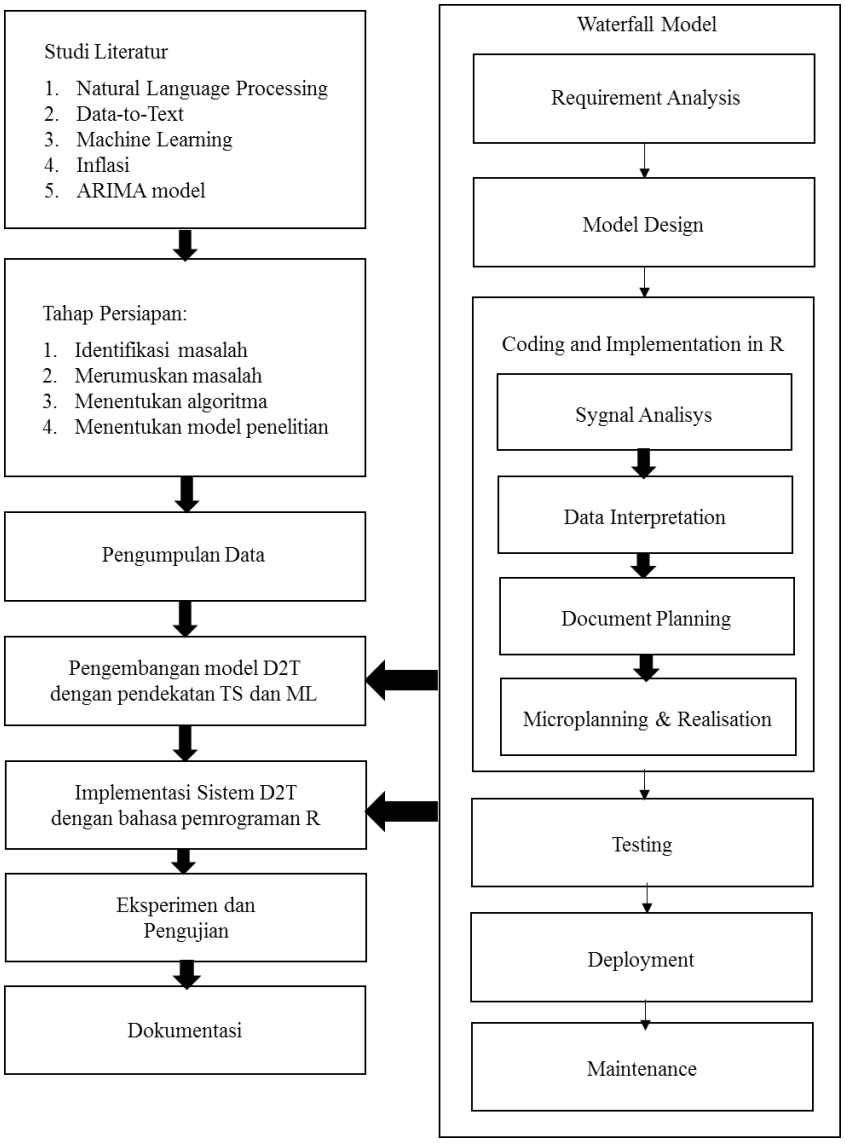
Secara konseptual, R *package* adalah kumpulan fungsi, objek data, dan dokumentasi yang secara koheren mendukun operasi analisis data. R adalah bahasa pemrograman *open-source* dan lingkungan analisis yang mengandung lebih dari 8000 *packages* untuk statistik, *bio-informatics*, visualisasi, *Machine Learning*, ekonomi, dan lain-lain. (Ihaka dan Gentleman, 1996). Bahkan, sampai tahun 2017 banyak *packages* yang terpublish dalam *cran-r project* sebanyak 11191 *packages*. Agar mudah digunakan dan untuk menjaga kualitasnya serta untuk terus mempertahankannya, kebanyakan R *package* disimpan di repositori berikut: Jaringan Arsip R Komprehensif (CRAN, http://cran.r-project.org/) dan Bioconductor Project. (Http://www.bioconductor.org/) (Riza, dkk., 2016).

BAB III

METODOLOGI PENELITIAN

3.1 Desain Penelitian

Desain penelitian adalah kerangka kerja yang digunakan untuk melakukan penelitian. Pada bagian ini penulis akan memaparkan kerangka kerja dari awal hingga akhir penelitian. Desain penelitian digambarkan pada gambar 3.1.



Gambar 3.1 Metodologi Penelitian Pengembangan Sistem *Data-to-Text* dengan pendekatan *Time-Series.*

Gambar 3.1 menjelaskan proses penelitian, penjelasannya adalah sebagai berikut:

1. Tahap persiapan adalah tahap awal dari penelitian, tahap ini dimulai dari identifikasi masalah, kemudian merumuskan masalah, lalu mencari metode atau algoritma apa yang sesuai untuk menyelesaikan masalah yang telah ditemukan. Kemudian yang terakhir adalah mendesain atau menentukan metode penelitiannya.
2. Studi literatur merupakan bagian dari tahap persiapan. Studi literatur dilakukan dengan memperlajari dan memahami teori yang akan digunakan untuk melakukan penelitian. Dalam penelitian ini penulis melakukan pengumpulan dan pemahaman materi terkait *Data-to-Text,* tahapan Natural Language Generation, Penggunaan *Machine Learning, Time Series,* Inflasi, ARIMA model, penggunaan bahasa pemrograman R, dan penelitian-penelitian yang terkait topik tersebut. Pengumpulan dan pemahaman materi dalam studi literatur ini penulis dapatkan dari beberapa media, seperti jurnal, buku, situs web, video, dan lain-lain. Selain mengumpulkan materi, penulis melakukan latihan terhadap beberapa tools yang akan digunakan seperti mempelajari bahasa R dan ShinyR.
3. Langkah selanjutnya adalah melakukan pengumpulan data. Data yang dikumpulkan terdiri dari data inflasi bulanan, data Indeks Harga Konsumen (IHK) per kelompok dan subkelompok gabungan dari 82 kota, data inflasi menurut kelompok komoditi, data tingkat inflasi gabungan 82 kota, data inflasi umum, data harga yang diatur pemerintah, dan barang bergejolak inflasi indonesia. Data-data tersebut diperoleh dari website Badan Pusat Statistik Indonesia yaitu, www.bps.go.id.
4. Setelah melakukan. pengumpulan data, maka langkah selanjutnya adalah pengembangan model. Dalam pengembangan model, digunakan arsitektur sistem utama dari (Reiter, E, 2007). Setelah model dikembangkan, langkah selanjutnya adalah melakukan implementasi sistem, yaitu merealisasikan model yang telah dibangun dengan melakukan Coding dalam bahasa pemrograman R.

5. Dalam pengujian aplikasi, penulis melakukan eksperimen sebanyak tujuh kali dengan data yang berbeda, lalu dari hasil eksperimen tersebut dilakukan beberapa evaluasi. Evaluasi yang pertama adalah mengevaluasi kualitas teks dengan menggunakan aplikasi NIST dan BLEU. Lalu untuk mengevaluasi kualitas prediksi, summary dan korelasinya dengan teks, dilakukan evaluasi oleh Human Forecaster atau Expert. Cara mengevaluasi dari expert adalah dengan cara mempresentasikan program yang dibangun, lalu expert tersebut mengisi penilaian kuisioner mengenai relevansi dan truthfulness. Sedangkan untuk mengevaluasi penyampaian informasi kepada pengguna, penulis menggunakan 10 orang untuk diberikan kuisioner mengenai sistem ini. Selain beberapa aspek tersebut, pada sistem ini juga dilakukan evaluasi mengenai waktu komputasi sistem.

3.2 Alat dan Bahan Penelitian

Bagian ini menjelaskan secara detail alat dan bahan yang digunakan untuk melakukan penelitian.

3.2.1. Alat Penelitian

1. Perangkat Keras (*Hardware*) yaitu laptop ASUS X550DP dengan spesifikasi:
   - *Processor* AMD APU A10-5750M
   - *Random Access Memory* (RAM) 8 GB
   - VGA AMD Radeon HD 8650Gb + HD 8670M
   - *Harddisk Drive* 1 TB
2. Perangkat Lunak (*Software*) sebagai berikut:
   - Sublime Text 3
   - Sistem Operasi Elementary OS 0.4.1 Loki 64 bit dan Windows 8.1 64 bit
   - *Web Browser Google Chrome*
   - Microsoft Excel 2013.
   - Rgui i386, Versi: 3.3.2
   - R studio.

3.2.2. Bahan Penlitian

Beberapa bahan penelitian yang digunakan yaitu seluruh informasi yang mengandung sumber kajian materi baik berupa jurnal, buku, *e-book, e-journal*, dan website mengenai *Data-to-text, Natural Language Processing, Natural Languige Generation, Machine Learning, Big Data* dan platformnya

3.3. Metode Penelitian.

Adapun metode yang dilakukan dalam penelitian ini dibagi kedalam dua bagian, yaitu metode pengumpulan data dan metode pengembangan perangkat lunak.

3.3.1. Metode Pengumpulan Data

Dalam penelitian ini, data dan informasi yang tersedia dapat meunjang penelitian. Metode yang digunakan dalam pengumpulan data adalah:

1. Studi Literatur

   Dengan mempelajari metode-metode mengenai evaluasi, dan mempelajari cara mengolah parameter melalui studi literatur seperti jurnal, buku, dan sumber lain di internet yang relevan dengan penelitian ini.

2. Observasi

   Observasi dilakukan dengan cara melakukan pembangunan kalimat deskripsi dengan menggunakan *Natural Language Processing.*

3.3.2. Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak dilakukan dengan metode *waterfall*. Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (sequential linier). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau urut dimulai dari analisis, desain, pengkodean, pengujian dan tahap support (Sukamto & Shalahuddin, 2011). Penulis menggunakan metode *modern waterfall* seperti pada gambar 3.2 agar jika suatu saat ada kesalahan pada salah satu tahap, bisa dikembalikan ke tahap sebelumnya. Berikut pengertian dari tahap-tahap pada model *waterfall* pada gambar 3.2 menurut Ian Sommerville (2011) :

1. *Requirments Analysis and Definition* (Analisis)

   Analisis adalah tahap menentukan aplikasi atau *software* seperti apakah yang akan dibuat. Analisis 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. Analisis ini terdiri dari analisis kebutuhan dan analisis pembuatan sistem.

2. *System and Software Design* (Desain)

   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. Desain terdiri dari desain database, desain arsitektur system, dan desain antarmuka (*user interface*)

3. *Implementation and Unit Testing* (*Coding*)

   Coding adalah tahap proses implementasi dari desain, 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* (Testing)

   Proses testing atau pengujian dilakukan pada logika internal untuk memastikan semua pernyataan sudah diuji. 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* (Pemeliharaan)

   Dalam tahapan ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

DAFTAR PUSTAKA

ARRIA. (2015). *The Automated Description of Digital Data.* ARRIA.

Arthur, S. L. (1959). *Some Studies in Machine Learning Using the Game of Checkers.* IBM Journal of Research and Development.

Banaee, H., Ahmed, M., & Louthfi, A. (2013). Towards NLG for Physiological Data Monitoring with Body Area Networks. *14th European Workshop on Natural Language Generation (ENLG)* .

Belz, A. (2007). Probabilistic Generation of Weather Forecast Texts. *Natural Language Technology Group* .

Boyd, S. (1998). TREND: A System for Generating Intelligent Descriptions of Time-Series Data. *IEEE International Conference on Intelligent Processing Systems*

Budiharto, W., & Rachmawati, R. (2013). *Pengantar Praktis Pemrograman R untuk Ilmu Komputer.* Jakarta: Halaman Moeka Publishing.

Crowder, J., Moore, J., DeRose, L., & Franek, W. (1999). *Air Pollution Field Enforcement.* California: Research Triangle Park.

Demir, S., Carberry, S., & McCoy, K. (2011). Summarizing Information Graphics Textually. *Computational Linguistics* , 38(3):527 - 574.

Gatt, A., Potret, F., Reiter, E., & Hunter, J. (2009). From Data to Text in the Neonatal Intensive Care UnitL Using NLG Technology for Decision Support and Information Management. *AI Communication* , 22: 1533186.

Gkatzia, D., Lemon, O., & Rieser, V. (2016). Natural language generation enhances human decision-making with uncertain information. *54th Annual Meeting of the Association for Computational Linguistic (ACL)* .

Goldberg, E., Driedger, N., & Kitterdge, R. (1994). Using natural-language processing to produce weather forecast. *IEEE Expert,* , 9(2), 45 - 53.

Hallet, C., Power, R., & Scott, D. (2006). Summarisation and visualisation of e-health data repositories. *UK E-Scienc All-Hands Meeting* .

Huby, J. (2010). *cloud coverage*. Retrieved juli 19, 2017, from the weaterh prediction: http://www.theweatherprediction.com/habyhints/189/

Hunter, J., Freer, Y., Gatt, A., Reiter, E., Sripada, S., Sykes, C., et al. (2012). Automatic generation of natural language nursing shift summaries in neonatal intensive care: BT-Nurse. *Artificial Intelligence in medicine* , 56(3), 157 - 172.

Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of computational and graphical statistics* , 299-314.

Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics 5 (3)* , 299-314.

James, P., & Amber, S. (2012). *Natural Language Annnotation for Machine Learning.* Cambridge: O'REILLY.

Johnson, N., & Lane, D. (2011). Narrative monologue as a first step towards adavanced mission debrief for AUV operator situational awareness. *15th Internationa Conference on Advanced Robotics* .

Johnson, R., & Wichern, D. (1982). *Applied multivariate statistical analysis.* New Jersey: Englewood Cliffs.

Liddy, E. D. (2001). *Natural Language Processing.* New York: Syracuse University.

McDonald. (1987). Natural Language Generation. *Encyclopedia of Artificial Intelligence* , 642-655.

Mishra, N., & Jain, E. (2014). Time Series Data Analysis for Forecasting – A Literature Review. *Internationa Journal of Modern Engeneering Research* , Vol 4. 1-5.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Fondations of Machine *Learning*. MIT Press.

Muhammad, A., Irawan, I. M., & Matu, E. (2015). STUDY COMPARISON OF SVM-, K-NN- AND BACKPROPAGATION-BASED CLASSIFIER. *Journal of Computer Science and Information* , 1.

Nichols, J., Mahmud, J., & Drews, C. (2012). Summarizing Sport *Event*s Using Twitter. *IBM Research - Almaden* .

Peddington, J., & Tintarev, N. (2011). Automatically generating stories from sensor data. *6th International Conference on Intelligent user interfaces(IUI)* .

Potret, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Frer, Y., et al. (2009). Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence* , 173(7-8), 789-816.

Pressman, R. S. (2009). *Software Engineering: A Practitioner's Approach.* New York: McGraw-Hill.

Ramos-soto, A., Bugarin, A., & Barro, S. (2016). On the role of linguistic description of data in the buildings of natural language generation system. *Fuzzy Sets and System* , 285,31-35.

Ramos-Soto, A., Pereira-Farina, M., Bugarin, A., & Barro, S. (2015). On the role of linguistic description of data in the building of natural language generation system. *Fuzzy Sets and System* , 1-8.

Ramos-soto, Alejandro, Bugarín, A., & Barro, S. (2016). *Fuzzy* Sets Across the Natural Language Generation Pipeline. *Progress in Artificial Intelligence 5.4* , 261-276.

Ramos-soto, Bugarin, A., Barro, S., Gallego, N., Rodriguez, C., Fraga, I., et al. (2015). Automatic Generation of Air Quality *Index* Textual Forecast Using a Data-To-Text Approach . *CiTIUS* , 164-174.

Reiter, E. (2007). An Architecture for data-to-text systems. *Proceedings of the Eleventh European Workshop on Natural Language Generation* , 97-104.

Reiter, E., & Dale, R. (2000). Cambridge University press.

Reiter, E., & Dale, R. (2000). *Building Natural Language Generation System.* Cambridge: Cambridge University Press.

Reiter, E., & Dale, R. (2000). *Building natural language generation systems.* Cambridge university press.

Reiter, E., Sripada, S., Hunter, J., Yu, J., & Davy, I. (2005). Choosing wrds in computer-generated weather forecast. *Artificial Intelligence* , 167(1-2), 137-169.

Riza, L. S. (2015). *Data Science and Big Data Processing in R: Representations and Software.* Granada: Universidad de Granada.

Riza, L., Nasrulloh, I., Junaeti, E., Zain, R., & Nandiyanti, A. (2016). gradDescentR: An R *Package* Impementing *Gradient Descent* and Its Variants for Regression Tasks. *1st International Conference on Information Technology, Information System and Electrical Engeneering (ICITISEE)* .

Rowlet, R. (2001, Mei 31). *Beaufort Scales (Wind Speed)*. Retrieved Juli 2017, 19, from University of North Carolina: https://www.unc.edu/~rowlett/units/scales/beaufort.html

Schneider, A., Vauldry, P., Mort, A., Mellish, C., Reiter, E., & Wilson, P. (2013). MIME - NLG in Pre-hospital Care. *14th European Workshop in Pre-hospital Care* .

Spector, P. (2004). An Introduction to R. *Statistical Computing Facility* .

Sripada, S., & Gao, F. (2007). Summarizing Dive Computer Data: A Case Study in Integrating Textual and Graphical Presentations of Numerical Data. *MOG 2007 Workshop on Multimodal Output Generation* .

Sripada, S., & Gao, G. (2007). Summarizing dive computer data: A case study in integrating textual and graphical presentations of numerical data. *Workshop on multimodal output generaiton (MOG)* .

Sripada, S., Reiter, E., & Davy, I. (2005). SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator. *Artificial Intelligence 167* , 137-169.

Sripada, S., Reiter, E., Hunter, J., & Yu, J. (2001). A two-stage model for content determination. *8th European workshop on Natural Language Generation* .

Sripada, S., Reiter, E., Hunter, J., & Yu, J. (2003). Generating english summaries of time. *9th ACM International Conference on Knowledge Discovery and Data Mining (KDD)* .

Thomas, K., Sripada, S., & Noordzij, M. (2010). Atlas.txt Exploring linguistic grounding technique for communicating spatial information to blind users. *Universal Access in the Informational Society* .

Tintarev, N., Reiter, E., Black, R., & Waller, A. R. (2016). Personal storytelling: Using Natural Language Generation for children with complex communication needs, in the wild... *International Journal of Human-Computer Studies* , (92-93):1-16.

Trihendadi, C. (2005). *SPSS 13.0 Analisis Data Statistik.* Yogyakarta: Andi.

Turner, R., Sripada, S., Reiter, E., & Davy, I. (2008). Using spatial reference frame to generate grounded textual summaries of georeferenced data. *5th Natural Language Generation Conference (INLG)* .

Yu, J., Reiter, E., Hunter, J., & Mellish, C. (2007). Choosing the content of textual summaries of large time-series data sets. *Journal Natural Language Engeneering* .

Zadeh, L. (2001). From Computing With Numbers to Computing With Words-From Manipulation of Measurements to Manipulations of Perceptions. *In Human and Machine Perception 3* , 1-25.

Zadeh, L.A. (1996). *Fuzzy* logic = computing with words. *IEEE transactions on Fuzzy systems* , 103-111.

Zandlo, J., Spoden, G., Bouley, P., & Ruschy, D. (2001). *Analysis of Snow Climatology*. Retrieved juli 2017, 19, from University of Minnesota: http://climate.umn.edu/snow_fence/components/winddirectionanddegreeswithouttable3.htm