

**PENGEMBANGAN SISTEM *DATA-TO-TEXT* UNTUK
MEMBANGKITKAN BERITA CUACA DENGAN PENDEKATAN
TIME-SERIES DALAM R**

SKRIPSI

Diajukan untuk Memenuhi Sebagian dari
Syarat Memperoleh Gelar Sarjana Komputer
Program Studi Ilmu Komputer



Oleh

Brahma Putra

1304811

PROGRAM STUDI ILMU KOMPUTER
DEPARTEMEN PENDIDIKAN ILMU KOMPUTER
FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PENDIDIKAN INDONESIA

2017

LEMBAR PENGESAHAN
PENGEMBANGAN SISTEM *DATA-TO-TEXT* UNTUK
MEMBANGKITKAN BERITA CUACA DENGAN PENDEKATAN
***TIME-SERIES* DALAM R**

Oleh:

Brahma Putra

1304811

DISETUJUI DAN DISAHKAN OLEH:

Pembimbing 1

Lala Septem Riza, MT. Ph.D.

NIP.197809262008121001

Pembimbing 2

Yaya Wihardi, M. Kom

NIP. 198903252015041001

Mengetahui,

Kepala Departemen Pendidikan Ilmu Komputer

Prof.Dr.Munir M.IT

NIP.196603252001121001

PERNYATAAN

Saya menyatakan bahwa skripsi yang berjudul "**Pengembangan Sistem Data-to-text untuk pembangkitan berita cuaca dengan Pendekatan Time-series dalam R**" ini sepenuhnya karya sendiri. Tidak ada plagiat dari orang lain di dalamnya dan saya tidak melakukan penyalinan atau pengutipan dengan cara-cara yang tidak sesuai etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung sanksi yang dijatuhkan kepada saya apabila ditemukan adanya pelanggaran terhadap etika keilmuan di karya ini atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Bandung, Juli 2017

Pembuat pernyataan,

Brahma Putra

1304811

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada sang maha pencipta Allah yang maha perkasa lagi mulia, berkat seluruh karunia dan hidayah yang telah diberikan-Nya, penulis dapat menyelesaikan penelitian dengan judul “**Pengembangan Sistem Data-to-text Untuk Membangkitkan Berita Cuaca dengan Pendekatan Time-series Dalam R**” ini.

Penelitian ini dilakukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Ilmu Komputer di Program Studi Ilmu Komputer, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia.

Penulis menyadari bahwa skripsi yang telah disusun ini masih belum sempurna. Maka dari itu, kritik dan saran sangat diharapkan untuk meningkatnya kualitas dari penelitian ini dan penelitian selanjutnya. Meskipun masih jauh dari kesempurnaan, penulis berharap penelitian ini dapat membawa manfaat baik bagi penulis dan pembaca.

Bandung, Agustus 2017

Penlis,

Brahma Putra

UCAPAN TERIMA KASIH

Dalam melakukan penelitian ini, penulis mendapatkan banyak dukungan, bimbingan, dan bantuan baik secara teknis, fisik, dan psikis dari berbagai pihak yang terlibat. Maka dari itu, dengan disertai rasa syukur, penulis secara pribadi mengucapkan terimakasih dan penghargaan dengan bobot tertinggi kepada:

1. Keluarga dan saudara-saudara tercinta yang selalu memberikan berbagai dukungan dan selalu mendo'akan penulis dalam menyelesaikan skripsi untuk meraih gelar sarjana komputer ini.
2. Bapak Lala Septem Riza, MT. Ph.D. selaku pembimbing I, atas seluruh jasanya dengan sabar memberikan bimbingan terbaik dan seluruh ilmu bermanfaat yang diberikan kepada penulis sehingga terwujud apa yang diharapkan dari penelitian ini.
3. Bapak Yaya Wihardi, M. Kom. Selaku pembimbing II, atas seluruh jasanya yang telah memberikan bimbingan terbaik dan seluruh ilmu bermanfaat yang diberikan kepada penulis sehingga terwujud apa yang diharapkan dari penelitian ini.
4. Bapak Eddy Prasetyo Nugroho, MT. selaku pembimbing akademik kelas C2 juga selaku kaprodi bidang ilmu komputer, yang selalu memberikan dukungan dan dorongan selama penyusunan skripsi ini.
5. Bapak Yudi Wibisono, M.T yang telah memberi banyak masukan sehingga mempermudah kelancaran penelitian ini.
6. Seluruh rekan seperjuangan mahasiswa ilmu komputer C2-2013, rekan kontrakkan, rekan angkatan, jurusan dan fakultas yang telah memberikan dukungan, motivasi, inspirasi dan bantuan lainnya sehingga dalam menyelesaikan penelitian ini penulis dapat dengan semangat dapat menyelesaiannya.
7. Seluruh elemen yang terlibat didalam proses perkuliahan hingga selesaiya penelitian ini.

Semoga seluruh kebaikan pihak yang terlibat kelak akan dilipat gandakan oleh sang pencipta Allah yang maha perkasa lagi mulia dan semoga Allah meridhoi segala aktivitas penelitian ini. *Aamiin.*

Bandung, Agustus 2017

Penulis,

Brahma Putra

ABSTRAK

Data yang diperoleh dari stasiun pemantauan cuaca dan kualitas udara yang diakuisisi dari beberapa website meteorologi merupakan data numerik, sebagian orang sulit untuk memahaminya. Penelitian ini ditujukan untuk mengembangkan sistem *Data-to-text* pembangkit berita cuaca yang memudahkan masyarakat dalam memahami informasi cuaca dan kualitas udara, berdasarkan data yang terdiri dari curah hujan, cakupan awan, suhu, kecepatan angin, arah angin, dan kualitas udara. Dalam menyampaikan informasi cuaca, sangat penting untuk menyampaikan hasil prediksi. Maka untuk mencapai hal tersebut, penulis mengembangkan sistem ini dengan menggunakan pendekatan *time-series*. Selain itu, dalam penelitian ini memanfaatkan beberapa *packages* yang tersedia dalam bahasa R untuk menghemat waktu pengembangan. Hasil eksperimen menunjukkan bahwa sistem yang dikembangkan mendapatkan rata-rata penilaian di atas sedang, dengan waktu komputasi yang baik.

Kunci: *Data-to-text, Natural Language Generation, Cuaca, Time-series*

ABSTRACT

Data which obtained from weather and air quality monitoring stations from several meteorological websites are numerical data, some people find it difficult to understand. This research is aimed to develop data-to-text System to generate weather news, so it could help people more easier to understand weather information which based on data consisting of rainfall, cloud coverage, temperature, wind speed, wind direction, and air quality. In conveying weather information, it is very important to convey prediction results. Therefore, we develop this System using the time-series approach. In addition, in this study utilizing several packages available in the R language to save development time. The experimental results show that the developed System gets the above average average, with good computation time.

Keywords: *Data-to-text, Natural Language Generation, Weather, Time-series*

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN.....	ii
KATA PENGANTAR	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
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 <i>Natural Language Processing</i>	6
2.1.1 Area <i>Natural Language Processing</i>	7
2.1.2 Terminologi <i>Natural Language Processing</i>	8
2.2 Pengertian <i>Natural Language Generation</i>	12
2.3 Pengertian dan Arsitektur <i>Data-to-text</i>	12
2.4 Penelitian Terkait <i>Data-to-text</i>	18
2.4.1 <i>Software Forecast Generator</i> (FOG)	20
2.4.2 Pembangkitan Berita Olah Raga dari <i>Twitter</i>	23
2.4.3 Perangkat Lunak <i>BabyTalk</i>	25
2.5 Pengertian dan Sejarah <i>Machine Learning</i>	26

2.5.1	<i>Supervised Learning</i>	27
2.5.2	<i>Unsupervised Learning</i>	28
2.5.3	Algoritma <i>Gradient Descent</i>	29
2.6	<i>Time-series Data</i>	30
2.7	<i>Exponential Smoothing</i>	31
2.7.1	<i>Single Exponential Smoothing</i>	32
2.7.2	<i>Single Exponential Smoothing</i>	32
2.7.3	<i>Triple Exponential Smoothing</i>	34
2.8	<i>Logika Fuzzy</i>	34
2.9	<i>R Programming</i>	36
2.9.1	Model data dalam R	38
2.9.2	Contoh kode program bahasa R	39
2.9.3	Contoh visualisasi data dalam R	41
2.9.4	<i>Package</i> alam bahasa R.....	43
BAB III	METODOLOGI PENELITIAN.....	44
3.1	Desain Penelitian.....	44
3.2	Metode Penelitian.....	47
3.3	Alat dan Bahan Penelitian	47
3.3.1	Alat Penelitian.....	47
3.3.2	Bahan Penelitian.....	48
3.4	Proses Pengembangan Perangkat Lunak	49
BAB IV	HASIL PENELITIAN DAN PEMBAHASAN	52
4.1	Pengumpulan Data	52
4.2	Pengembangan Model <i>Data-to-text</i> dengan <i>Machine Learning</i> dan <i>Time-series</i>	57
4.2.1	Model Proses <i>Signal Analysis</i>	59
4.2.2	Model Proses <i>Data Interpretation</i>	61
4.2.3	Model Proses <i>Document Planning</i>	71
4.2.4	Model Proses <i>Microplanning</i> dan <i>Realisation</i>	79
4.2.5	Pemanfaatan <i>Machine Learning</i>	87

4.2.6	Model Antarmuka Perangkat Lunak	88
4.3	Implementasi Sistem <i>Data-to-text</i>	89
4.3.1	Installasi R <i>Packages</i> yang dibutuhkan.....	89
4.3.2	Implementasi Proses <i>Signal Analysis</i>	92
4.3.3	Implementasi Proses <i>Data Interpretation</i>	102
4.3.4	Implementasi Proses <i>Document Planning</i>	107
4.3.5	Implementasi Proses <i>Microplanning</i> dan <i>Realization</i>	108
4.3.6	Implementasi <i>Machine Learning</i> untuk mengatasi missing value.....	110
4.4	Hasil Implementasi.....	111
4.4.1	<i>Output</i> sistem	111
4.4.2	Perbandingan <i>Output</i> dengan Penelitian Lain.....	113
4.5	Panduan Penggunaan Aplikasi	114
4.6	Rancangan Experimen.....	120
4.7	<i>Output</i> Hasil Eksperimen	126
4.8	Analisis dan Hasil Eksperimen.....	131
BAB V	KESIMPULAN DAN SARAN.....	137
5.1	Kesimpulan.....	137
5.2	Saran	137
DAFTAR PUSTAKA	138	
LAMPIRAN	144	

DAFTAR TABEL

Tabel 2.1 Penelitian terkait <i>Data-to-text</i> dan <i>Natural Language Generation</i>	18
Tabel 4.1 Kutipan data cuaca dari MeteoGalicia.....	55
Tabel 4.2 Kutipan data kualitas udara dari MeteoGalicia.....	56
Tabel 4.3 Detail variabel data cuaca	56
Tabel 4.4 Detail variabel data kualitas udara	57
Tabel 4.5 Klasifikasi kecepatan angin Russ Rowlet	63
Tabel 4.6 Klasifikasi untuk menginterpretasikan arah angin menurut <i>Zandlo</i> dkk.	64
Tabel 4.7 Klasifikasi cakupan awan Jeff Huby.....	65
Tabel 4.8 <i>Index</i> kualitas udara dalam <i>Pollutant Standard Index (PSI)</i>	67
Tabel 4.9 Keterangan rumus <i>Pollutant Standard Index (PSI)</i>	68
Tabel 4.10 Kutipan data kualitas udara.....	81
Tabel 4.11 <i>Index</i> kualitas udara	82
Tabel 4.12 Nilai kontras dalam proses aggregasi dengan <i>simple conjunction</i>	86
Tabel 4.13 data iklim hasil prediksi.	93
Tabel 4.14 Hasil prediksi data kualitas udara.	93
Tabel 4.15 Seluruh fungsi <i>Statistical Event Detection</i> dan keterangannya.....	97
Tabel 4.16 Hasil proses <i>Statistical Event Detection</i>	102
Tabel 4.17 Perbandingan <i>output</i> yang dibangun dengan <i>output</i> dari penelitian terkait lainnya	113
Tabel 4.18 Penilaian kualitas <i>Readability</i> teks dengan menggunakan <i>Flesch Reading Ease</i>	122
Tabel 4.19 Keterangan rentang kesesuaian kuisioner.	124
Tabel 4.20 Hasil evaluasi dengan menggunakan <i>Flesch Reading Ease</i>	132
Tabel 4.21 Hasil perhitungan waktu komputasi dengan <i>System.time()</i>	133
Tabel 4.22 Hasil evaluasi <i>experts</i>	134
Tabel 4.23 Hasil evaluasi user terkait aspek <i>comprehensability</i> dan <i>importance</i>	136

DAFTAR GAMBAR

Gambar 2.1 Contoh skema <i>input-output</i> sistem <i>Data-to-text</i>	13
Gambar 2.2 Arsitektur <i>Data-to-text</i> yang diusulkan Reiter	14
Gambar 2.3 Contoh <i>output software Forecast Generator</i> (FOG).....	21
Gambar 2.4 Diagram skematis <i>software Forecast Generator</i> (FOG)	22
Gambar 2.5 Contoh <i>output</i> ringkasan berita olah raga	24
Gambar 2.6 Perbandingan <i>output</i> sistem dan ringkasan manual.	24
Gambar 2.7 Skema <i>input-output</i> aplikasi <i>BabyTalk</i>	25
Gambar 2.8 Contoh <i>Supervised Learning</i> pada pengenalan koin.....	28
Gambar 2.9 Contoh <i>Unsupervised Learning</i> dalam pengenalan koin	29
Gambar 2.10 Contoh himpunan <i>Crisp</i> pada kasus umur	35
Gambar 2.11 Contoh himpunan <i>Fuzzy</i> pada kasus umur	36
Gambar 2.12 Logo bahasa pemrograman R.....	36
Gambar 2.13 Antarmuka R <i>Graphical User Interface</i> (RGui)	38
Gambar 2.14 Model data dalam pemrograman R	39
Gambar 2.15 Operator <i>concatenate</i> dalam R.....	39
Gambar 2.16 Menampilkan dua data pertama dalam R.....	40
Gambar 2.17 Penggunaan fungsi <i>summarize</i> dalam R.....	40
Gambar 2.18 Penggabungan data dengan <i>concatenate</i> dalam R	40
Gambar 2.19 Pembuatan <i>matriks</i> dalam R.....	41
Gambar 2.20 Contoh visualisasi grafis dalam R	41
Gambar 2.21 Contoh perulangan dalam R.....	42
Gambar 2.22 Contoh implementasi <i>decision</i> dalam R.....	42
Gambar 2.23 Contoh fungsi dalam R.....	43
Gambar 3.1 Metodologi Penelitian Pengembangan Sistem <i>Data-to-text</i> dengan Pendekatan <i>Time-series</i>	46
Gambar 3.2 Pengembangan perangkat lunak model <i>Waterfall</i>	49
Gambar 4.1 Model <i>Data-to-text</i> dengan pendekatan <i>Machine Learning</i> dan <i>Time Series</i>	58

Gambar 4.2 Model proses <i>Signal Analysis</i> prediksi dengan <i>exponential smoothing</i> dalam membangun pesan deskripsi cuaca dan kualitas udara	60
Gambar 4.3 Model proses <i>Signal Analysis</i> pencarian <i>event</i> dalam membangun pesan deskripsi cuaca	61
Gambar 4.4 Model Proses <i>Data Interpretation</i> dengan menggunakan <i>Expert Rules</i>	62
Gambar 4.5 Fungsi keanggotaan suhu	65
Gambar 4.6 Fungsi keanggotaan curah hujan	66
Gambar 4.7 Perhitungan <i>Pollutant Standard Index</i> (PSI).....	68
Gambar 4.8 Hasil perhitungan <i>Pollutant Standard Index</i> (PSI)	70
Gambar 4.9 Hasil perhitungan <i>Pollutant Standard Index</i> (PSI)	70
Gambar 4.10 Model <i>Document Planning</i>	71
Gambar 4.11 <i>Routine message</i> untuk pesan prediksi.....	72
Gambar 4.12 <i>Significant Event Message</i> untuk pesan prediksi	73
Gambar 4.13 <i>Target corpus</i> hasil prediksi.....	74
Gambar 4.14 Skema untuk pesan prediksi.....	75
Gambar 4.15 <i>Event Tree</i> untuk pesan prediksi.	75
Gambar 4.16 Target Teks untuk <i>weather summary</i>	76
Gambar 4.17 Target teks untuk <i>weather summary</i>	76
Gambar 4.18 <i>Content Determination</i> untuk <i>weather summary</i>	77
Gambar 4.19 <i>Rhetorical Structured Text</i> untuk <i>weather summary</i>	78
Gambar 4.20 Model <i>Microplanning</i> dan <i>Realisation</i> untuk proses prediksi dan <i>weather summary</i>	79
Gambar 4.21 Kutipan inistial target korpus	81
Gambar 4.22 <i>Trend Description</i> untuk mendeskripsikan persepsi perubahan data kualitas udara.	83
Gambar 4.23 <i>Progressive Change word bank</i>	84
Gambar 4.24 Contoh <i>Phrase Aggregation</i>	85
Gambar 4.25 Struktur file aplikasi D2T_Apps.	88
Gambar 4.26 Purwa rupa antarmuka sistem <i>Data-to-text Weather Prediction</i> (DWP).	89

Gambar 4.27 Perintah untuk instalasi <i>packages Shiny R</i> yang berfungsi dalam membangun antarmuka aplikasi berbasis web	90
Gambar 4.28 Perintah untuk instalasi <i>packages Smooth</i> yang berfungsi dalam membangun model prediksi berbasis <i>time-series</i> dengan algoritma <i>exponential smoothing</i>	90
Gambar 4.30 Perintah untuk instalasi <i>package xts</i> untuk mendefinisikan dataset kedalam objek <i>time-series</i> berdasarkan tanggal.....	92
Gambar 4.29 Perintah untuk instalasi <i>package gradDesc</i> untuk melakukan prediksi dengan pendekatan <i>machine Learning</i> menggunakan algoritma gradient descet.....	92
Gambar 4.31 Prediksi dengan <i>exponential smoothing</i> dalam R	93
Gambar 4.33 Memuat data kualitas udara kedalam sistem.....	94
Gambar 4.32 Memuat data kedalam sistem	94
Gambar 4.34 Mengambil data satu bulan terakhir	95
Gambar 4.35 Mencari nilai <i>maximum</i> setiap variabel.....	95
Gambar 4.36 Mencari nilai <i>minimum</i> setiap variabel	96
Gambar 4.37 Mencari nilai rata-rata	96
Gambar 4.38 Mencari nilai total setiap variabel	96
Gambar 4.39 <i>Output</i> standar statistik.....	97
Gambar 4.40 Pendekripsi hari hujan berturut-turut.....	100
Gambar 4.41 MonthlyTempMessage function	101
Gambar 4.42 Fungsi mendekripsi keadaan kualitas udara bulan terakhir	101
Gambar 4.43 <i>Pseudocode</i> proses interpretasi data.....	103
Gambar 4.44 <i>Data Interpretation</i> dalam R	103
Gambar 4.45 <i>Input</i> kecepatan angin.....	104
Gambar 4.46 <i>Input</i> arah angin.....	104
Gambar 4.47 Fungsi <i>AirQuality_Interpreter</i> untuk menginterpretasikan kualitas udara.....	105
Gambar 4.48 Fungsi untuk menginterpretasikan curah hujan dan suhu.	106
Gambar 4.49 Hasil proses <i>Data Interpretation</i>	107

Gambar 4.50 <i>Content Selection</i> saat menggabungkan pesan yang berelasi (<i>Wind State</i>)	107
Gambar 4.51 <i>Source code Trend Description</i>	108
Gambar 4.52 <i>Pseudocode Aggregation</i>	109
Gambar 4.53 <i>Referring Expression generation</i>	109
Gambar 4.54 <i>Structure Realization</i>	110
Gambar 4.55 <i>Machine Learning</i> dalam R untuk mengatasi <i>missing value</i>	110
Gambar 4.56 Hasil keluaran sistem.	111
Gambar 4.57 Jendela saat proses instalasi aplikasi <i>R for Windows</i>	115
Gambar 4.60 Instalasi <i>packages</i> ShinyR.....	116
Gambar 4.58 Jendela antarmuka R <i>Graphical User Interface</i> (RGui)	116
Gambar 4.59 Contoh lokasi penyimpanan direktori D2T_Apps	116
Gambar 4.61 Contoh <i>output</i> dari <i>console</i> jika instalasi <i>packages</i> ShinyR berhasil.	117
Gambar 4.62 Keluaran dari <i>console</i> setelah server dijalankan pada port 6632	117
Gambar 4.63 Contoh jendela browser yang muncul ketika menjalakan perintah runApp("D2T_Apps").....	118
Gambar 4.64 Isi dari direktori D2T_Apps	118
Gambar 4.65 Contoh isi dari file D2T_Machine.R	119
Gambar 4.66 Folder <i>Datasets</i> , berisi seluruh dataset yang tersedia.	120
Gambar 4.67 Cara mengganti dataset yang digunakan dalam file D2T_Machine.R	120
Gambar 4.68 Dataset eksperimen	122
Gambar 4.69 Evaluasi aspek <i>relevance</i> dan <i>truthfulness</i> oleh <i>expert</i>	125
Gambar 4.70 Evaluasi aspek <i>comprehensability</i> dan <i>importance</i> oleh <i>expert</i>	126
Gambar 4.72 Hasil keluaran dari eksperimen ke-dua dengan data real 2016-2017.	127
Gambar 4.71 Hasil keluaran dari eksperimen pertama dengan dataset <i>Modified1</i> ...	127
Gambar 4.73 Hasil keluaran dari eksperimen ke-tiga dengan data Modified2.....	128
Gambar 4.74 Hasil keluaran dari eksperimen ke-empat dengan data real 2014- 2015.....	128
Gambar 4.75 Hasil keluaran dari eksperimen ke-lima dengan data real 2015-2016	129
Gambar 4.76 Hasil keluaran dari eksperimen ke-enam dengan data real 2013-	

2014.....	129
Gambar 4.77 Hasil keluaran dari eksperimen ke-tujuh dengan data Modified 3.....	130
Gambar 4.78 Hasil keluaran dari eksperimen ke-delapan dengan data Modified 4.	130
Gambar 4.79 Hasil keluaran dari eksperimen ke-sembilan dengan data Modified 5.....	131
Gambar 4.80 Hasil keluaran dari eksperimen ke-sepuluh dengan data Modified 6.	131
Gambar 4.81 Contoh kuisioner yang diisi oleh pakar klimatologi	134
Gambar 4.82 Contoh kuisioner yang diisi oleh user.....	135

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, telah dikembangkan banyak aplikasi yang mampu menghasilkan informasi dalam bentuk text dengan *input* non-linguistik atau data numerik. Contohnya seperti yang diperkenalkan oleh (Goldberg, Driedger, & Kitteridge, 1994) yaitu *Forecast Generator* (FOG), *software* ini dapat mengkonversi peta cuaca menjadi ramalan dalam bentuk kalimat dengan menggunakan pengolah bahasa alami. Contoh lainnya dalam salah satu aplikasi pengontrol 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 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). Secara umum, berbagai contoh yang telah dipaparkan tersebut termasuk implementasi dari sistem *Data-to-text*.

Sistem *Data-to-text* (D2T) merupakan bagian dari sistem *Natural Language Generation* (NLG) yang mampu menghasilkan kalimat atau teks dari data numerik secara otomatis (Reiter, E, 2007). Karena sistem perlu mengkonversi dari data raw (data mentah), D2T harus melibatkan analisis data dan proses linguistik. Dengan mengembangkan arsitektur NLG yang dipaparkan oleh (Reiter, dkk., 2005), tahapan pembangunan sistem D2T terdiri dari empat langkah, yaitu: analisis sinyal, interpretasi data, perencanaan dokumen, *microplanning* dan realisasi (Reiter, E, 2007). Tahap ini hampir serupa dengan arsitektur NLG pada umumnya yang megandung tiga modul, diantaranya: perencanaan text, perencanaan kalimat, dan realisasi linguistik. Tiga modul tersebut terurai menjadi 6 tahap, yaitu: penentuan konten,

penentuan tulisan, pengumpulan kalimat, leksikalisasi, pembangkitan ekspresi acuan, dan realisasi linguistik.

Data-to-text telah menjadi sebuah solusi yang dapat dikatakan sukses untuk diterapkan dalam berbagai kasus, seperti pada beberapa referensi yang telah dipaparkan diawal BAB ini. Dalam penelitian ini, akan dibangun sistem *Data-to-text* yang dapat membangkitkan bahasa alami untuk menyampaikan informasi cuaca dan kualitas udara. Hal ini dilakukan untuk melengkapi dan meningkatkan kualitas informasi yang disampaikan agar lebih mudah dipahami oleh manusia, mengingat data yang diperoleh dari stasiun pemantauan merupakan data RAW (data mentah). Data yang diperoleh dari stasiun pemantauan ini, khususnya data kualitas udara merupakan data yang kurang mudah dipahami. Dikarenakan data yang diperoleh bersifat eksak, tidak semua orang dapat memahami data tersebut hanya dengan melihat hasil pemantauannya (angka). Oleh karena itu, dibutuhkan serangkaian proses yang mampu menganalisis data-data yang bersifat *numerik* dari stasiun pemantauan dan menyampaikannya dalam bentuk bahasa yang mudah dipahami oleh manusia. Semua proses yang dibutuhkan ini terliput kedalam satu arsitektur *Data-to-text*. Itulah mengapa *Data-to-text* dapat memberi manfaat yang terasa sangat signifikan untuk mempermudah pemahaman informasi.

Sumber data yang digunakan dari penelitian ini didapatkan dari salah satu stasiun pemantauan meteorologi milik lembaga bidang meteorologi dan geofisika (*GaliWeather*), data ini akan menjadi data *input* yang kemudian akan dibangkitkan hasil analisisnya dalam bahasa alami melalui serangkaian proses dalam arsitektur *Data-to-text*. Data yang digunakan meliputi data kualitas udara (seperti kandungan SO₂, NO, NO₂, NOX, CO, O₃, PM₁₀, dan sebagainya) serta data hasil pemantauan cuaca (meliputi kecepatan angin, arah angin, rata-rata suhu, rata-rata kelembaban relatif, radiasi global matahari, hembusan angin, penyinaran matahari, suhu embun, suhu tanah, suhu udara, curah hujan, tekanan barometrik, dan sebagainya). Sangat masuk akal jika

sistem *Data-to-text* dapat menjadi pelengkap dan mempermudah penyampaian informasi, serta mengurangi aktivitas yang biasanya dilakukan oleh manusia. Karena, pada era *Big Data* ini kebutuhan otomatisasi terlihat sangat meningkat dikarenakan jumlah data yang sangat besar, sangat bervariasi, dan pertumbuhannya sangat cepat tidak memungkinkan jika dikelola secara *manual* oleh manusia, apalagi untuk menyimpulkan hasil analisis.

Dalam membangkitkan berita cuaca dan kualitas udara, satu hal yang sangat penting adalah menyampaikan informasi mengenai hasil prediksi. Maka untuk mencapai hal ini, pengembangan sistem *Data-to-text* dalam penelitian ini akan menggunakan pendekatan *time-series* untuk memprediksi data. Yang akan digunakan adalah algoritma *exponential smoothing*. Hal lain yang perlu diperhatikan dalam melakukan pengembangan perangkat lunak adalah dengan memperhatikan *Development Time*, untuk dapat mempersingkat waktu pengembangan, penulis memanfaatkan beberapa *packages* yang tersedia dalam R.

1.2 Rumusan Masalah

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

1. Bagaimana pengembangan model dari sistem *Data-to-text* untuk pembangkitan berita cuaca dan kualitas udara 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 ini:

1. Pembangunan sistem *Data-to-text* dengan pendekatan *Time Series* ini hanya didasarkan pada data kualitas udara dan data cuaca yang diperoleh dari stasiun pemantauan meteorologi *Mabegondo*.
2. Pembangunan sistem *Data-to-text* ini hanya menggunakan bahasa pemrograman R.

1.4 Tujuan

Berdasarkan rumusan masalah diatas, maka tujuan penelitian dalam tugas akhir ini dirumuskan sebagai berikut.

1. Untuk melakukan pengembangan model *Data-to-text* untuk membangkitkan berita cuaca dengan 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-to-text* dan *time-series* serta penerapannya dalam membangkitkan bahasa alami untuk mendeskripsikan data kualitas udara dan cuaca.
2. Dapat menjadi salah satu alternatif dan pelengkap dalam menyampaikan hasil analisis data secara otomatis oleh sistem *Data-to-text*.
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 terdiri 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*

Komunikasi merupakan hal yang sangat penting bagi manusia, karena dengan komunikasi manusia dapat saling memahami informasi. Dalam kehidupan sehari-hari, manusia berkomunikasi dengan menggunakan bahasa alami. Bahasa alami atau *Natural Language* merupakan bahasa yang digunakan antar manusia untuk berkomunikasi, seperti bahasa indonesia dan bahasa inggris. Untuk mempermudah komunikasi antara manusia dengan mesin, atau dengan kata lain dapat saling memahami informasi, dibutuhkan sebuah sistem yang dapat mendeskripsikan informasi itu sendiri. *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 berkomunikasi 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.

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 mengambil dokumen atau email, dan menghasilkan ringkasan yang jelas dari konten tersebut. Seperti program untuk merubah dari konten paragraph yang panjang menjadi beberapa slide 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 pertanyaan 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.

2.1.2 Terminologi *Natural Language Processing*

Perkembangan NLP menghasilkan kemungkinan dari antarmuka bahasa alami menjadi *knowledge-based* dan penerjemahan bahasa natural. Ada tiga aspek utama pada 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 mempertimbangkan lebih dari hanya sekedar kalimat. Agen harus melihat lebih ke dalam konteks kalimat, keadaan dunia, tujuan dari speaker dan listener, konvensi khusus, dan sejenisnya.

Pemrosesan Bahasa Alami atau *Natural Language Processing* (NLP) adalah komponen penting dalam text mining dan subbidang dalam kecerdasan buatan dan komputasi linguistik. Hal tersebut mempelajari tentang bagaimana memahami bahasa manusia alami, dengan cara mengubah bahasa manusia (seperti dokumen teks) menjadi penyajian yang lebih formal (dalam bentuk data numeric dan simbolik) yang lebih mudah untuk dimanipulasi oleh program computer. Tujuan NLP adalah untuk melangkah melebihi manipulasi teks berbasis sintaks (yang sering kali disebut dengan ‘*word counting*’) ke pemahaman yang benar dan memproses bahasa alami yang

mempertimbangkan berbagai batasan semantik dan gramatikal dan juga konteks.

Berikut adalah beberapa tantangan yang biasanya dikaitkan dengan implementasi NLP:

- *Part-of-speech tagging.* Sangatlah sulit untuk menandai istilah-istilah dalam suatu teks yang terkait dengan bagian tertentu dari suatu naskah (misalnya kata benda, kata kerja, kata sifat, kata keterangan, dst), karena bagian dari naskah tidak hanya bergantung pada definisi istilah tetapi juga pada konteks dimana teks digunakan.
- *Text segmentation.* Beberapa bahasa tulisan, seperti bahasa mandarin, jepang, dan thai, tidak memiliki batasan kata. Dalam contoh ini, tugas text-parsing memerlukan identifikasi terhadap batasan kata, yang seringkali merupakan tugas yang sangat sulit. Tantangan serupa dalam segmentasi naskah muncul ketika menganalisa bahasa verbal, karena suara menyajikan rangkaian huruf dan kata yang bercampur satu sama lain.
- *Word sense disambiguation.* Banyak kata yang memiliki lebih dari satu arti. Memilih arti yang paling masuk akan hanya bisa dicapai dengan mempertimbangkan konteks di mana kata digunakan.
- *Syntactic ambiguity.* Tata bahasa dalam bahasa alami seringkali ambigu; artinya, ada berbagai struktur kalimat yang memungkinkan yang perlu dipertimbangkan. Memilih struktur yang paling tepat biasanya memerlukan paduan informasi kontekstual dan semantik.
- *Imperfect or irregular input.* Aksen asing atau lokal dan berbagai hambatan vokal dalam pidato dan kesalahan ketik dan tata bahasa dalam teks-teks menyebabkan pengolahan bahasa bahkan lebih sulit.
- *Speech acts.* Suatu kalimat seringkali dianggap sebagai suatu aksi oleh si pembicara. Struktur kalimatnya sendiri mungkin tidak berisi cukup informasi untuk mendefinisikan tindakan ini.

NLP telah sukses diterapkan pada berbagai pekerjaan melalui program-program ckomputer untuk memproses secara otomatis bahasa manusia yang sebelumnya hanya bisa dilakukan oleh manusia. Berikut ini adalah beberapa dari pekerjaan-pekerjaan yang paling popular dilakukan oleh NLP:

- *Information retrieval.* Ilmu untuk melakukan pencarian terhadap berbagai dokumen yang relevan, menemukan informasi tertentu didalamnya, dan menghasilkan metadata untuk isinya.
- *Information extraction.* Sejenis ‘*information retrieval*’ yang tujuannya adalah untuk mengekstrak secara otomatis informasi terstruktur, seperti data yang sudah terdefinisi dengan baik secara semantik dan secara kontekstual yang sudah terkelompok dari domain tertentu, dengan menggunakan berbagai dokumen tak-terstruktur yang bisa terbaca oleh mesin.
- *Named-entity recognition.* Juga dikenal sebagai ‘*entity identification*’ dan ‘*entity extraction*’, ini adalah sub-pekerjaan dari ‘*information extraction*’ yang bertujuan untuk menemukan dan mengklasifikasikan elemen-elemen *atomic* di dalam teks menjadi berbagai kategori yang sudah ditentukan sebelumnya, seperti nama orang, organisasi, lokasi, pernyataan waktu, kuantitas, nilai uang, persentase, dan seterusnya.
- *Questioning answering.* Pekerjaan menjawab secara otomatis suatu pertanyaan yang diajukan dalam bahasa alami; yaitu, menghasilkan jawaban bahasa manusia ketika diberi pertanyaan bahasa manusia. Untuk mendapatkan jawaban terhadap pertanyaan, program computer bisa menggunakan baik database pra-terstruktur atau kumpulan dokumen bahasa alami (suatu ‘text corpus’ seperti world wide web).

- *Automatic summarization.* Pembuatan ringkasan terhadap dokumen teks yang dilakukan oleh program komputer yang berisi poin-poin paling penting dari dokumen asli.
- *Natural Language Generation.* Sistem mengubah informasi dari database komputer menjadi bahasa manusia yang bisa dibaca.
- *Natural language understanding.* Sistem mengubah contoh-contoh bahasa manusia menjadi penyajian yang lebih formal yang lebih mudah bagi program komputer untuk dimanipulasi.
- *Machine translation.* Penerjemahan otomatis dari satu bahasa manusia ke bahasa yang lain.
- *Foreign language reading.* Program komputer yang membantu orang yang non-native untuk membaca bahasa asing dengan pengucapan benar dan aksen pada bagian yang berbeda dengan benar.
- *Foreign language writing.* Program komputer yang membantu orang yang non-native dalam menulis dalam bahasa asing.
- *Speech recognition.* Mengubah kata-kata verbal menjadi *input* yang bisa terbaca oleh mesin. Dengan adanya sound clip dari orang yang sedang berbicara, sistem menghasilkan dikte dari teks.
- *Text-to-speech.* Disebut juga ‘*speech synthesis*’, suatu program komputer yang mengubah secara otomatis teks menjadi suara manusia.
- *Text proofing.* Program komputer yang membaca bukti copy teks yang bertujuan untuk mendeteksi dan mengoreksi kesalahan.
- *Optical character recognition.* Penerjemahan otomatis dari Gambar tulisan tangan, tulisan ketikan, atau teks cetak (biasanya diambil dengan *scanner*) menjadi dokumen-dokumen teks yang bisa diedit oleh mesin.

Keberhasilan dan popularitas *text mining* sangat bergantung pada perkembangan NLP baik dalam proses menghasilkan maupun memahami bahasa manusia. NLP memungkinkan ekstraksi berbagai *features* dari teks tak-terstruktur sehingga berbagai jenis teknik data mining bisa digunakan untuk

mengekstrak pengetahuan (pola dan keterkaitan yang baru dan bermanfaat) darinya. Dalam arti tersebut, gampangnya, *text mining* adalah kombinasi NLP dan *data mining*.

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. NLU melingkup segala topik mengenai komunikasi dari bahasa manusia ke mesin, sementara NLG melingkup komunikasi dari mesin ke bahasa manusia.

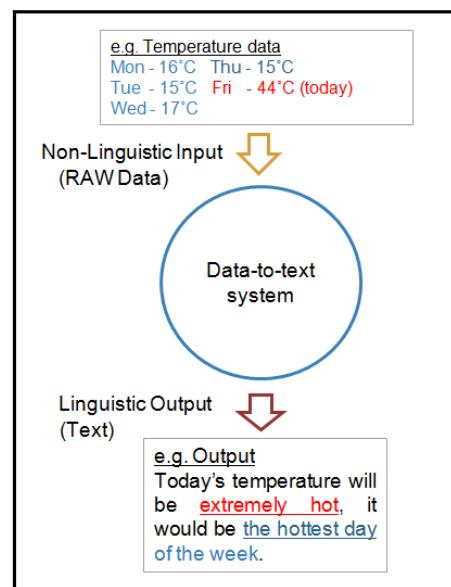
2.3 Pengertian dan Arsitektur *Data-to-text*

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. Untuk orang-orang dibidang sosial, data yang disajikan dalam bentuk teks bahkan dapat dijadikan sebagai data empiris yang disertakan dalam riset mereka. Meskipun *Data-to-text* merupakan bagian dari NLG, namun perbedaan terbesar antara sistem *Data-to-text* dengan NLG dimana *inputnya* berbasis *knowledge*, yaitu sistem *Data-to-text* harus mampu menganalisis dan menerjemahkan *input* data yang dimiliki sehingga ada *knowledge* yang dapat disampaikan dan dikemas senatural mungkin dalam bahasa alami.

Sistem *Data-to-text* telah diaplikasikan pada beberapa perangkat lunak, contohnya seperti yang diperkenalkan oleh (Goldberg, Driedger, dan Kitteridge, 1994) yaitu FOG, *software* ini dapat mengkonversi peta cuaca menjadi ramalan dalam bentuk kalimat dengan menggunakan pengolah bahasa alami. Contoh lainnya dalam salah satu aplikasi pengontrol kesehatan, yaitu

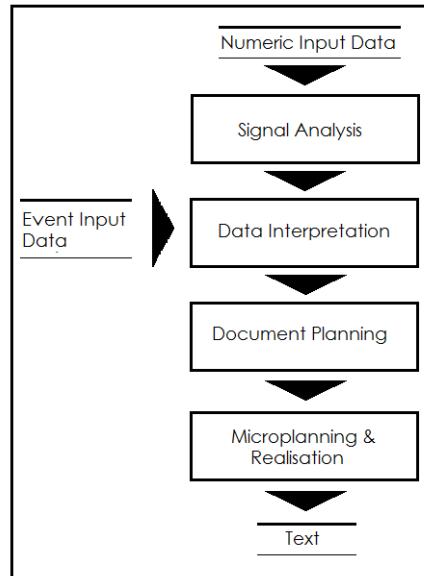
BABYTALK family Systems, diperkenalkan oleh (Potret, dkk., 2009), aplikasi ini mampu membuat sebuah ringkasan peristiwa yang terjadi selama 45 menit dari sinyal psikologis kontinyu dan diskrit, seperti pengaturan peralatan dan pemberian obat dalam bentuk kalimat. Selain itu, penerapan ini juga telah 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 menunjukkan 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 Diagram *input-output* sistem *Data-to-text*.

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



Gambar 2.1 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 diimplementasikan 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 didapatkan. 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°C. Maka dengan melalui serangkaian proses interpretasi data ini, angka 35°C diinterpretasikan 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 keempat 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 sinyal).

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

a. *Lexicalisation*

Proses *lexicalisation* adalah bagaimana melakukan pemilihan kata atau frase yang akan digunakan dalam mendeskripsikan 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). Dalam sub-bab 2.4.1 akan dijelaskan sebuah penelitian yang dilakukan oleh (Goldberg, Driedger, dan Kitteridge, 1994) megenai *software Forecast Generator* (FOG), serta dalam sub-bab 2.4.2 akan dipaparkan garis besar penelitian yang dikemukakan oleh (Nichols, Mahmud, dan Drews, 2012) mengenai pembangkitan ringkasan berita olahraga menggunakan *twitter*.

Sedangkan 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 <i>Content Selection</i>	Domain	Sumber Data
TREND			
(Boyd, 1998)	No Content Selection	Weather	Database
(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	Decompression Models	Dive	Sensor

dan Gao, 2007)			
(Turner R, dkk., 2008)	Decision Tree	Georeferenced Data	Database
(Gatt, A, dkk., 2009)	Rule-Based	Health	Sensor
(Thomas, dkk., 2010)	Document Schema	Georeferenced Data	Database
(Demir, dkk., 2011)	Rule-based	Domain Independent	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)	<i>Fuzzy-sets</i>	Weather	Database
(Gkatzia, dkk., 2016)	Rule-based	Weather	Numerical data with assigned probabilities

2.4.1 *Software Forecast Generator (FOG)*

Terkait dengan penelitian ini, sebelumnya pernah dikembangkan sebuah *software* yang dapat menghasilkan kalimat bahasa alami dengan *input* yang bersifat non-linguistik, salah satu diantaranya adalah FOG. *Forecast Generator (FOG)* merupakan sebuah *software* yang mampu mengkonversi peta cuaca menjadi ramalan dalam bentuk kalimat dengan menggunakan pengolah bahasa alami, yang dikemukakan dalam penelitian (Goldberg, Driedger, dan Kitterdge, 1994). *Software* ini memanggil *Sampler* dari setiap titik untuk mendapatkan nilai medan meteorologi yang dibutuhkan. *Sampler* menerima permintaan parameter tertentu (seperti suhu udara) pada titik sampel dan memantau interpolasi grafik untuk nilai pada titik tersebut. Kemudian mengembalikan nilai yang didapatkan dalam bentuk rangkaian waktu yang menjadi *input* bagi FOG.

1. *Output* FOG

Contoh *output* dari *software* ini dapat dilihat pada Gambar 2.3, dimana (a) menunjukkan kondisi cuaca di wilayah *Great Lakes Marine*, dan (b) menunjukkan kondisi cuaca di *Arctic Marine*, dan (c) menunjukkan kondisi cuaca di *Atlantic Marine*.

(a)

WINDS NORTHWEST 15 DIMINISHING TO LIGHT MONDAY AFTERNOON. CLOUDY WITH OCCASIONAL LIGHT SNOW. FOG PATCHES. VISIBILITIES 2 TO 5 NM IN SNOW.

(b)

BELLE ISLE
NORTHEAST GULF
NORTHEAST COAST.
GALE WARNING IN BELLE ISLE AND NORTHEAST GULF ISSUED.
GALE WARNING IN NORTHEAST COAST CONTINUED.
FREEZING SPRAY WARNING CONTINUED.
WINDS SOUTHWEST 15 TO 20 KNOTS INCREASING TO WEST GALES 35 NEAR NOON FRIDAY. SNOW BEGINNING OVERNIGHT THEN ENDING FRIDAY AFTERNOON.
VISIBILITY FAIR IN SNOW. OCCASIONAL FREEZING SPRAY OVER OPEN WATER.
TEMPERATURES MINUS 14 TO MINUS 10.
OUTLOOK FOR SATURDAY... GALE FORCE WEST WINDS BECOMING STRONG TO GALE FORCE EASTERLIES.

(c)

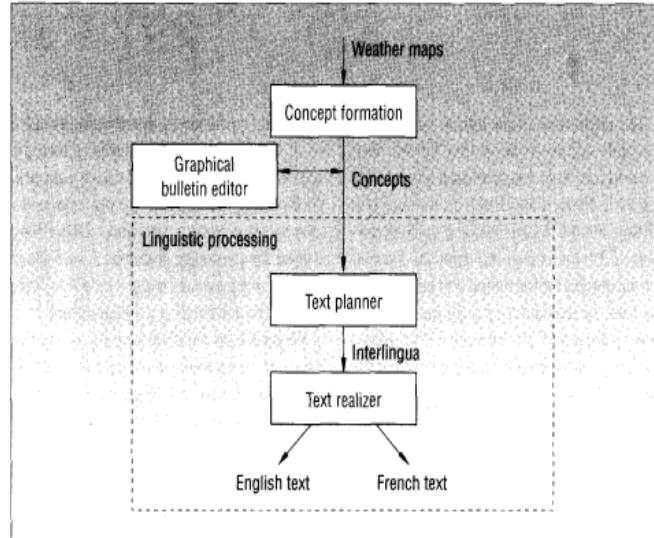
BLIZZARD WARNING ENDED.
TONIGHT.. SNOW AND BLOWING SNOW. WINDS SOUTHWESTERLY 35 KM/H OCCASIONALLY GUSTING TO 50. LOW NEAR MINUS 30.
THURSDAY.. MAINLY CLOUDY. WINDS SOUTHEASTERLY 30. VERY HIGH WINDCHILLS.
TEMPERATURE NEAR MINUS 30.
FRIDAY.. CLOUDY PERIODS. WINDS WESTERLY 25. LOW NEAR MINUS 32. HIGH NEAR MINUS 28.
PROBABILITY OF PRECIPITATION IN PERCENT 100 TONIGHT. 30 THURSDAY AND 20 FRIDAY.

Gambar 2.3 Contoh *output software Forecast Generator (FOG)*

Satu produk prediksi seperti pada Gambar 2.3 dibangkitkan dari 76 titik sampel berbeda. Dengan grafik interpolasi yang tersedia setiap jam, dan perkiraan yang berlangsung selama 48 jam, setiap rangkaian waktu akan berisi setidaknya 40 data Perkiraan, diperlukan lima parameter (angin, suhu, jarak pandang, cuaca, dan *freezing spray*), jadi ramalan mentahnya mungkin mengandung lebih dari 15.000 nilai data (76 poin, masing-masing memiliki 5 runtutan pencatatan dari 40 titik).

2. Skema FOG

FOG membagi proses pembangkitan bahasa alami dari data angka menjadi tiga bagian, yaitu: ekstraksi data, pemrosesan konseptual, dan pemrosesan linguistik. Skema dalam bentuk grafik dari tahap ini dapat dilihat pada Gambar 2.4.



Gambar 2.4 Diagram skematik *software Forecast Generator (FOG)*

(Goldberg, Driedger, dan Kitterdge, 1994)

Langkah-langkah yang diperlihatkan pada Gambar 2.4 terdiri dari:

a. Concept formation

Langkah pertama adalah menentukan area perkiraan mana yang akan dikelompokkan bersama dengan membandingkan deret waktu dari berbagai titik sampel. Untuk prakiraan laut, kriteria utamanya adalah kecepatan dan arah angin. Untuk prakiraan publik, tutupan awan dan curah hujan lebih signifikan. Jika deret waktu cukup serupa, maka FOG menggabungkan area yang ditunjukkan oleh titik sampel. area gabungan dipertimbangkan untuk memiliki beberapa titik sampel, dan pada akhir penggabungan ruang, sehingga didapatkan rangkaian waktu yang representatif. Dari beberapa sampel, FOG kemudian mengklasifikasikan data dan waktu ini-menggabungkannya untuk mengidentifikasi kondisi cuaca yang mengalami perbedaan secara signifikan.

b. *Grapichal Buletin Editor*

Grapichal Bulletin Editor menampilkan *event-event* yang diproduksi oleh ahli konseptual Sistem sebagai histogram interaktif. Peramal dapat menyesuaikannya dengan mouse untuk dengan mudah mengontrol konten produk teks tanpa memerlukan peta cuaca. Pada intinya, *Grapichal bulletin editor* digunakan agar ada pertimbangan dari sudut pandang objektif.

c. *Text Planner*

Perencanaan teks (*Text Planner*) memperkuat beberapa konsep yang tidak terstruktur dengan konsep tambahan, dan kemudian mengaturnya menjadi urutan kalimat dalam bentuk potongan. Ini menentukan urutan kalimat akhir, namun hanya memberikan spesifikasi abstrak dari isi kalimat tertentu. Spesifikasi adalah struktur interlingual, yang merupakan *input* untuk realisasi teks.

d. *Text Realization*

Dalam realisasi teks (*Text Realization*), aspek gramatikal dan leksikal diatur pada setiap kalimat untuk mendapatkan susunan teks yang tepat. Para pakar bahasa Inggris dan Prancis menggunakan gramatikal formal untuk penyusunan kalimatnya.

2.4.2 Pembangkitan berita olah raga dari *twitter*

Orang-orang dizaman teknologi ini setiap harinya melakukan *update status* untuk menyalurkan pemikiranya melalui sosial media seperti Facebook dan *Twitter*, dalam kedua platform tersebut tentu saja banyak mengandung informasi mengenai apa yang orang lakukan dan mengenai apa yang orang lihat. Dalam sebuah acara seperti acara olahraga, banyak status yang di-*update* oleh masyarakat yang mendeskripsikan dan mengekspresikan pendapat tentang acara tersebut. Dalam penelitian yang dipaparkan oleh (Nichols, Mahmud, dan Drews, 2012) ini mendeskripsikan algoritma yang mampu

membangkitkan ringkasan jurnalistik dari sebuah *event* dengan hanya menggunakan data *status update* dari *twitter*.

Gambar 2.5 memperliatkan *output* sistem yang dibangun dengan membangkitkan berita *Spike7: Boston Red Sox vs. Seattle Mariners*.

Spike 7: Boston Red Sox vs. Seattle Mariners
Josh reddick homerun puts the red sox up 5-4 on seattle. #redsox take a 5 to 4 lead and put Lackey in position to win despite sucking. You hear all those cheers following # red sox

Gambar 2.5 Contoh *output* ringkasan berita olah raga

Perbandingan antara *output* dari sistem NLG yang dibangun oleh (Nichols, Mahmud, dan Drews, 2012) dan ringkasan yang dibangun manual dapat dilihat pada Gambar 2.6. Dalam gambar tersebut, diperlihatkan perbandingan antara ringkasan manual dan ringkasan sistem.

Game	Spike	Manual Summary	Our Summary
US vs. Slovenia	1	In the first 15 mins of the soccer game between USA and Slovenia, Slovenia is leading with a goal by Birsa. Birsa scored an easy goal from midfield to the right of the goal, as USA left that shot wide open. Terrible defense by USA team, too much space left open.	Good goal for Slovenia and the USA once again starts a game terrible. Birsa gives #SVN 1-0 lead with smart shot. Howard didn't even look like he saw that one coming.
Germany vs. Serbia	3	Klose argues with referee, gets second yellow cards and is out of the game. Germany down to 10 men. 1-0 Serbia.	Germany screwed by the refs and a red card for Klose; seconds later, a pretty goal by the Serbs. yellow seems to be a very popular colour in this game.
Australia vs. Serbia	9	Serbia Australia match ends with 2-1. With a result of 1:0 between Germany and Ghana this means that Ghana and Germany will advance to the knock out rounds and serbia and australia will be out.	Australia won 2-1 on serbia, Germany won 1-0 vs Ghana, Germany and Ghana goes on to the next round. Great win by #aus but not good enough to go through. Final score #Aus 2 #Srb 1.

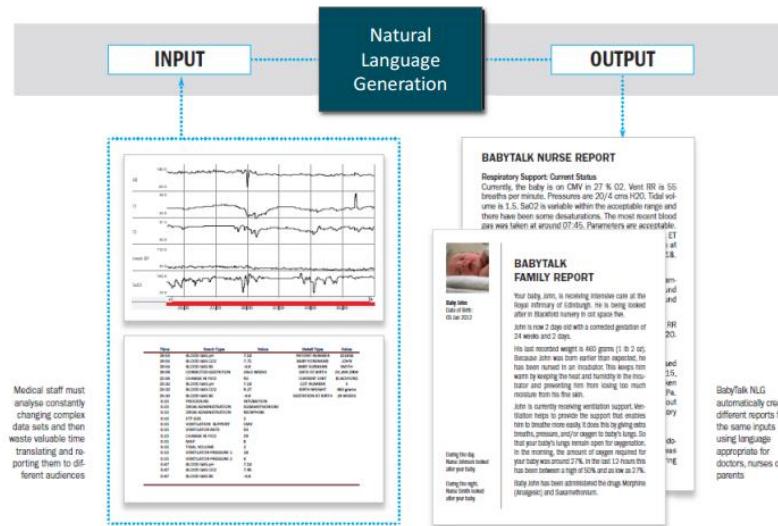
Gambar 2.6 Perbandingan *output* sistem dan ringkasan manual.

2.4.3 Perangkat Lunak BabyTalk

BabyTalk merupakan sebuah projek yang bertujuan untuk membangkitkan kesimpulan dalam bahasa alami dari data medis mengenai bayi, di *Neonatal Intensive Care Unit* (NICU). Penerapan konsep *Data-to-text* dan NLG dipaparkan dalam (Potret, dkk., 2009). *Software* ini memiliki dua tipe *input*, yaitu:

1. Sensor data numerik mengenai bayi, dimana sensor tersebut merekam informasi keadaan bayi yang diteliti seperti detak jantung, tekanan darah, dan suhu badan.
2. Rekam aksi medis dan observasi, seperti obat yang diberikan pada bayi dan hasil tes darah.

Hasil akhir dari sistem ini adalah kesimpulan dalam bentuk kalimat atau paragraf yang diperoleh dari serangkaian proses dalam arsitektur *Data-to-text*. Secara umum, Gambaran dari sistem *BabyTalk* ini dapat dilihat pada Gambar 2.7.



Dalam Gambar 2.7, *input* sistem yang berupa *signal stream* seperti pada kotak sebelah kiri Gambar akan diterjemahkan menjadi *output* berupa teks bahasa alami seperti yang ada pada kotak sebelah kanan Gambar.

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 penyelesaian dilakukan dengan menyusuri pohon permasalahan sampai mendapatkan solusinya.

Awal ditemukannya *Machine Learning* yaitu pada abad ke-20, seorang ilmuwan 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 memimpulkan mempergunakan komputer untuk memodelkan bagaimana cara berfikir

manusia. Mereka berhipotesis bahwa Mekanisme berfikir manusia dapat secara tepat dimodelkan dan disimulasikan pada komputer digital.

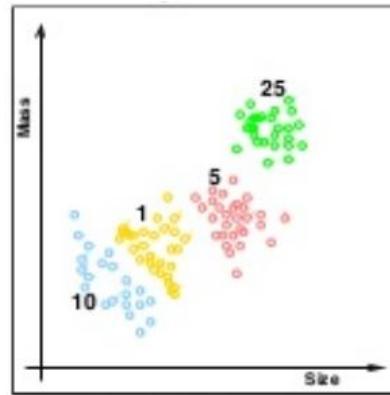
Machine Learning memiliki beberapa tipe dengan proses pembelajaran yang berbeda, tipe-tipe tersebut akan dijelaskan pada sub-bab berikutnya.

2.5.1 *Supervised Learning*

Tugas dari *Supervised Learning* terdiri dari pembangunan model yang memetakan nilai *input* pada nilai output 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 *outputnya*. Contoh algoritma dari salah satu bagian dari *Machine Learning* yaitu jaringan saraf tiruan yang menggunakan 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.

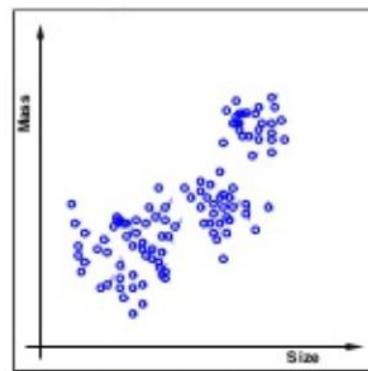
Pada Gambar 2.8, diperlihatkan bagaimana klasifikasi dari pengenalan koin, terlihat sangat jelas lokasi bagian dari tiap kelas, seperti koin dengan nilai sepuluh terpisah dipaling bawah dengan warna biru, koin dengan nilai satu yang berwarna kuning tidak bercampur dengan yang lainnya, dan seterusnya.

2.5.2 *Unsupervised Learning*

Unsupervised Learning terdiri dari pembangunan model dari *data training* dengan tidak mengandung nilai *output* (Riza, 2015). *Unsupervised Learning* merupakan pembelajaran 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.



Gambar 2.9 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 *steepest ascent*.

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 didapatkan adanya 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 fluctuation*), 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 menunjukkan 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 teknik peramalan 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 *Exponential Smoothing*

Exponential Smoothing adalah suatu prosedur yang secara terus menerus memperbaiki peramalan dengan merata-rata (menghaluskan = *smoothing*) nilai masa lalu dari suatu data runtut waktu dengan cara menurun (*exponential*). Menurut (Trihendadi, 2005) analisis *exponential smoothing* merupakan salah satu analisis deret waktu, dan merupakan metode peramalan dengan memberi nilai pembobot pada serangkaian pengamatan sebelumnya untuk memprediksi nilai masa depan.

2.7.1 Single Exponential Smoothing

Single Exponential Smoothing Atau biasa disebut sebagai *Simple Exponential Smoothing* adalah metode yang digunakan untuk peramalan jangka pendek. Model mengasumsikan bahwa data berfluktuasi di sekitar nilai mean yang tetap, tanpa trend atau pola pertumbuhan konsisten. Tidak seperti *Moving Average*, *Exponential Smoothing* memberikan penekanan yang lebih besar kepada *Time Series* saat ini melalui penggunaan sebuah konstanta *smoothing* (penghalus). Konstanta *smoothing* mungkin berkisar dari 0 ke 1. Nilai yang dekat dengan 1 memberikan penekanan terbesar pada nilai saat ini sedangkan nilai yang dekat dengan 0 memberi penekanan pada titik data sebelumnya.

Rumus untuk Simple exponential *smoothing* adalah sebagai berikut:

$$S_t = \alpha * X_t + (1-\alpha) * S_{t-1}$$

dimana:

S_t = peramalan untuk periode t.

X_t = Nilai aktual *Time Series*

F_{t-1} = peramalan pada waktu t-1 (waktu sebelumnya)

α = konstanta perataan antara 0 dan 1

2.7.2 Double Exponential Smoothing

Metode ini digunakan ketika berbentuk data trend. Ada dua metode dalam *Double Exponential Smoothing*, yaitu:

- a. Metode Linier Satu Parameter dari Brown's

Metode ini dikembangkan oleh Brown's untuk mengatasi perbedaan yang muncul antara data aktual dan nilai peramalan apabila ada trend pada polnya. Dasar pemikiran dari pemulusan eksponensial linier dari Brown's adalah serupa dengan rata-rata bergerak linier (*Linier Moving Average*), karena kedua nilai pemulusan tunggal dan ganda ketinggalan dari data yang sebenarnya bilamana terdapat unsur trend, perbedaan antara nilai pemulusan tunggal dan ganda

ditambahkan kepada nilai pemulusan dan disesuaikan untuk trend. Persamaan yang digunakan pada metode ini adalah:

$$\begin{aligned} S'_t &= \alpha_p X_t + (1 - \alpha_p) S'_{t-1} \\ S''_t &= \alpha_p S'_t + (1 - \alpha_p) S''_{t-1} \\ a_t &= S'_t + (S'_t - S''_t) = 2S'_t - S''_{t-1} \\ b_t &= \frac{\alpha_p}{1 - \alpha_p} (S'_t - S''_t) \\ F_{t+m} &= a_t + b_t m \end{aligned}$$

Dimana,

S'_t = Nilai pemulusan eksponensial tunggal

S''_t = Nilai pemulusan eksponensial ganda

α_p = Parameter pemulusan eksponensial yang besarnya $0 < \alpha_p < 1$

a_t, b_t = Konstanta pemulusan

F_{t+m} = hasil peramalan untuk periode ke depan yang diramalkan

Agar dapat menggunakan persamaan di atas, nilai S'_t-1 dan S''_{t-1} harus tersedia. Tetapi pada saat $T=1$, nilai tersebut tidak tersedia. Jadi nilai-nilai ini harus tersedia di awal.

b. Metode Dua Parameter dari Holt

Metode ini nilai trend tidak dimuluskan dengan pemulusan ganda secara langsung, tetapi proses pemulusan trend dilakukan dengan parameter berbeda dengan parameter pada pemulusan data asli. Secara matematis metode ini dituliskan pada tiga persamaan:

Dimana,

- Pemulusan total : $S_t = \alpha X_t + (1 - \alpha)(S_{t-1} + T_{t-1})$
- Pemulusan trend : $T_t = \beta(S_t - S_{t-1}) + (1 - \beta)T_{t-1}$
- Peramalan metode Holt : $F_{t+m} = S_t + T_t \times m$

S_t = Nilai pemulusan tunggal

X_t = Data sebenarnya pada waktu ke-t

T_t = Pemulusan trend

- F_{t+m} = nilai ramalan
 m = Periode masa mendatang
 α, β = konstanta dengan nilai antara 0 dan 1

2.7.3 *Triple Exponential Smoothing*

Triple Exponential Smoothing Atau metode *Winter's three parameters linear and seasonal exponential smoothing*. Ini termasuk dalam model Holt's ditambah indeks-indeks musiman dan sebuah koefisien *smoothing* untuk indeks-indeks tersebut.

Metode ini digunakan ketika data menunjukkan adanya trend dan perilaku musiman. Untuk menangani musiman, telah dikembangkan parameter persamaan ketiga yang disebut metode "*Holt-Winters*" sesuai dengan nama penemunya. Terdapat dua model *Holt-Winter* tergantung pada tipe musimannya yaitu *Multiplicative seasonal model* dan *Additive seasonal model*. Komponen musiman sering menjadi faktor yang paling penting untuk menerangkan variasi-variasi dalam variabel tak bebas selama periode satu tahun

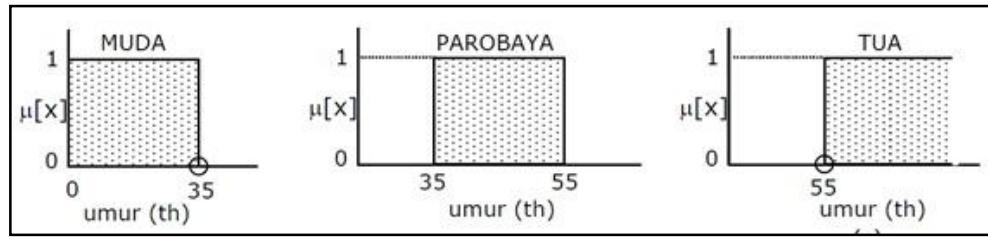
2.8 Logika *Fuzzy*

Logika *Fuzzy* adalah logika *multivalued* yang memungkinkan untuk mendefinisikan nilai menengah diantara dua logika atau evaluasi konvensional yang berbeda, seperti benar atau salah, iya atau tidak, tinggi atau rendah, panas atau dingin, dan lain-lain. Logika *Fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika *Fuzzy* adalah teori himpunan *Fuzzy*. Pada teori himpunan *Fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting.

Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dalam penalaran dengan logika *Fuzzy* tersebut. Logika *Fuzzy* dapat dianggap sebagai kotak hitam yang berhubungan antara ruang *input* menuju ruang *output*. Kotak hitam tersebut berisi cara atau metode yang dapat digunakan untuk mengolah data *input* menjadi *output* dalam bentuk

informasi yang baik. himpunan *Fuzzy* adalah himpunan yang menyatakan suatu obyek dapat menjadi anggota dari beberapa himpunan dengan nilai keanggotaan (μ) yang berbeda. untuk lebih jelasnya, perhatikan contoh dibawah:

Misalnya, variable umur dibagi 3 kategori, yaitu: Muda < 35 tahun, Parobaya $35 \leq \text{umur} \leq 55$ tahun, dan Tua > 55 tahun. Secara grafis, dapat dilihat pada Gambar 2.10.



Gambar 2.10 Contoh himpunan *Crisp* pada kasus umur

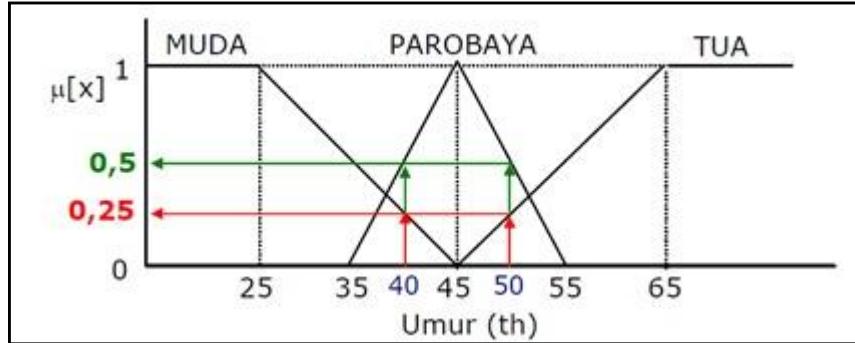
Jika menggunakan himpunan *crisp* yang dapat dilihat pada Gambar 2.10, dapat diambil kesimpulan bahwa:

- Usia 34 tahun, dikatakan Muda $\rightarrow \text{Muda}[34]=1$
- usia 35 tahun kurang 1 hari, dikatakan Muda $\rightarrow \text{Muda}[35\text{th}-1\text{hr}]=1$
- Usia 35 tahun, dikatakan Tidak Muda $\rightarrow \text{Muda}[35]=0$
- Usia 55 tahun, dikatakan Parobaya $\rightarrow \mu_{\text{Parobaya}}[55]=1$
- Usia 55 tahun lebih 1 hari, dikatakan Tidak Parobaya $\rightarrow \mu_{\text{Parobaya}}[55\text{th}+1\text{hr}]=0$ atau
- Usia 55 tahun lebih 1 hari, dikatakan Tua $\rightarrow \mu_{\text{Tua}}[55\text{th}+1\text{hr}]=1$

Dari kesimpulan diatas, himpunan crisp menyatakan umur seseorang kedalam suatu kategori secara tidak adil, karena orang yang berusia 35 tahun dikatakan parobaya, sedangkan orang yang berusia 35 tahun kurang 1 hari dikatakan tidak parobaya (karena masuk kategori muda). selisih 1 hari saja menimbulkan berbedaan kategori yang signifikan.

Himpunan *Fuzzy* digunakan untuk mengatasi hal tersebut, sehingga dengan menggunakan himpunan *Fuzzy*, seseorang dapat masuk ke dua kategori secara bersamaan, misalnya seseorang yang berusia 35 tahun kurang 1 hari dapat

masuk kategori Muda dan Parobaya sekaligus, tetapi dengan nilai keanggotaan yang berbeda. Contohnya seperti pada Gambar 2.11.



Gambar 2.11 Contoh himpunan *Fuzzy* pada kasus umur

Sebagai contoh, seseorang yang berumur 40 tahun termasuk dalam himpunan Muda dengan $\mu_{\text{muda}}[40]=0,25$, namun dia juga termasuk dalam himpunan Parobaya dengan $\mu_{\text{Parobaya}}[50]=0,5$.

2.9 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, python, 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*)

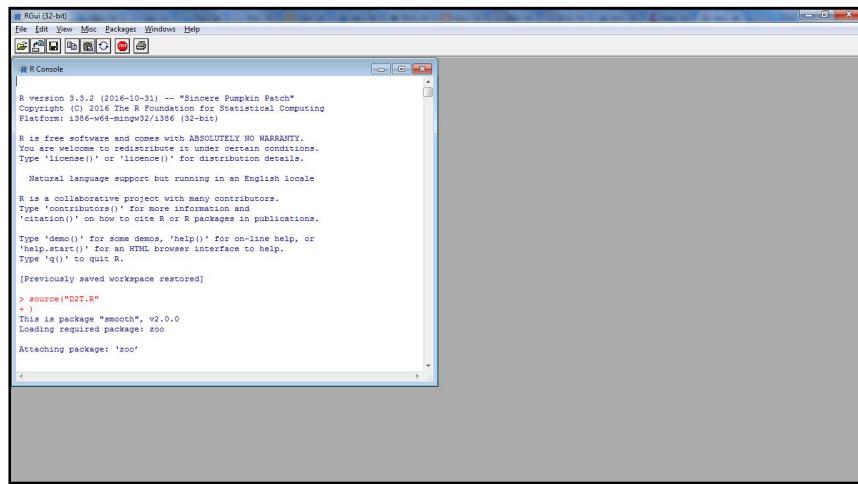
Pada saat ini analisis data membutuhkan pengoperasian yang interaktif. Apalagi jika data yang dianalisis adalah data yang bergerak. R dilengkapi dengan konektivitas ke database server, olap, maupun format data web service seperti XML, spreadsheet dan sebagainya. Sehingga apabila data set berubah hasil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

3. Berbasis S yaitu turunan dari tool statististik 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 proyek R.

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.



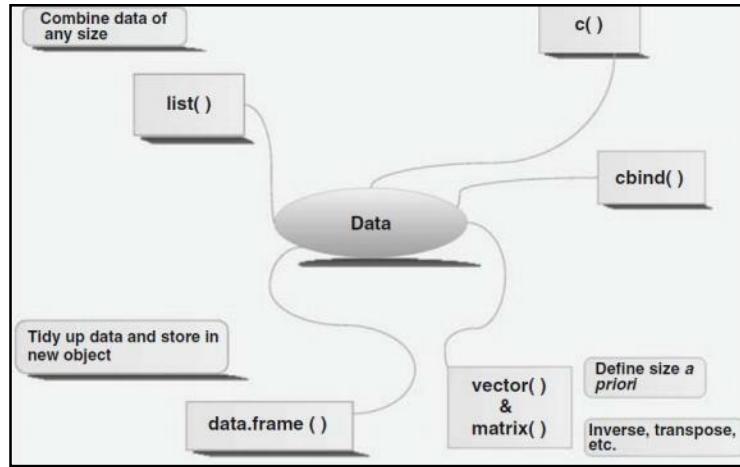
Gambar 2.13 Antarmuka R *Graphical User Interface* (RGui).

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 masing-masing.

2.9.1 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 *concatenate* (`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 *concatenate* 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.

```
> x <- c("brahma", "putra")
> x
[1] "brahma" "putra"
```

Gambar 2.18 Penggabungan data dengan *concatenate* dalam R

- Membuat *matriks*

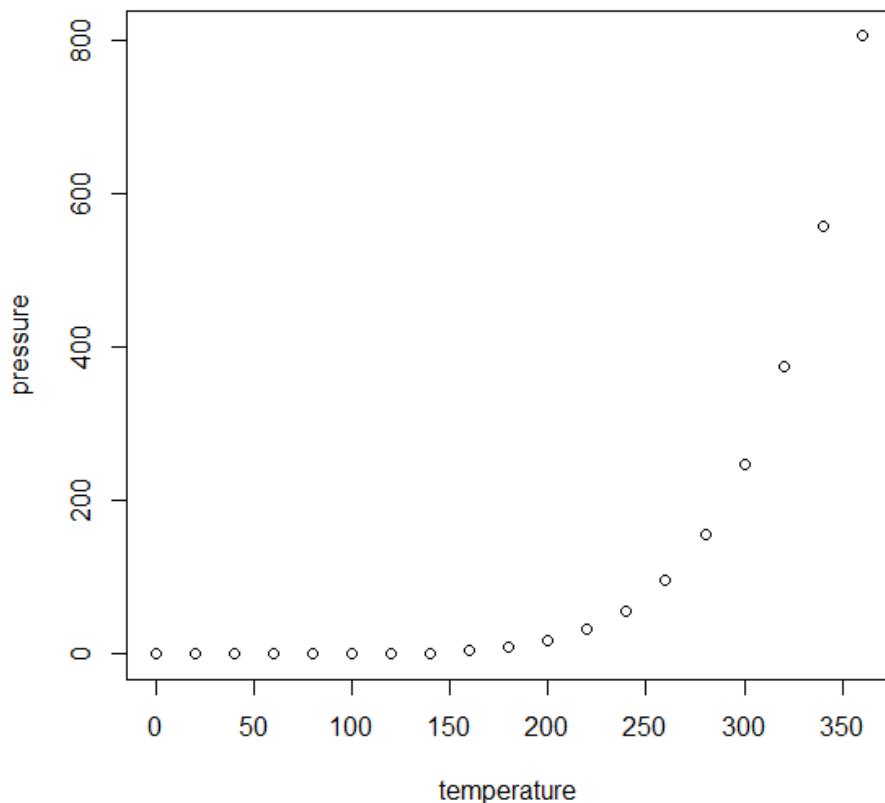
Matriks adalah data dua dimensi dimana sebagian besar fungsi-fungsi statistik dalam R dapat dianalisis dengan menggunakan bentuk matriks. Bentuk matriks ini juga banyak digunakan pada operasi fungsi-fungsi built-in untuk aljabar linear dalam R, seperti untuk penyelesaian suatu persamaan linear. Argumen 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

2.9.3 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.

```
> i<-1 ; n<-10
> for(i in 1:n){
+ print("Hello World!")
+
[1] "Hello World!"
```

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.

```
> a <- 3
> b <- 2
> if(a>b){
+ print("Hello World!")
+
[1] "Hello World!"
```

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.

```
x<-1
y<-2
pertambahan <- function(a,b){
  c <- a+b
  return(c)
}
hasil<-pertambahan(x,y)

> hasil
[1] 3
```

Gambar 2.23 Contoh fungsi dalam R

2.9.4 *Package* alam bahasa R

Secara konseptual, R *package* adalah kumpulan fungsi, objek data, dan dokumentasi yang secara koheren mendukung 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

Penelitian ini dilakukan melalui serangkaian tahapan yang dapat dilihat pada Gambar 3.1, dimana pada gambar tersebut diperlihatkan bahwa proses penelitian ini terdiri dari studi literatur, pengumpulan data, pengembangan model, implementasi aplikasi, eksperimen dan pengujian aplikasi, dan dokumentasi. Studi literatur adalah proses dimana penulis melakukan pengumpulan dan pemahaman materi yang terkait dengan arsitektur *Data-to-text*, tahapan *Natural Language Generation*, penggunaan *Machine Learning*, *Time Series*, penggunaan bahasa pemrograman R, dan penelitian terkait dengan topik ini. 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.

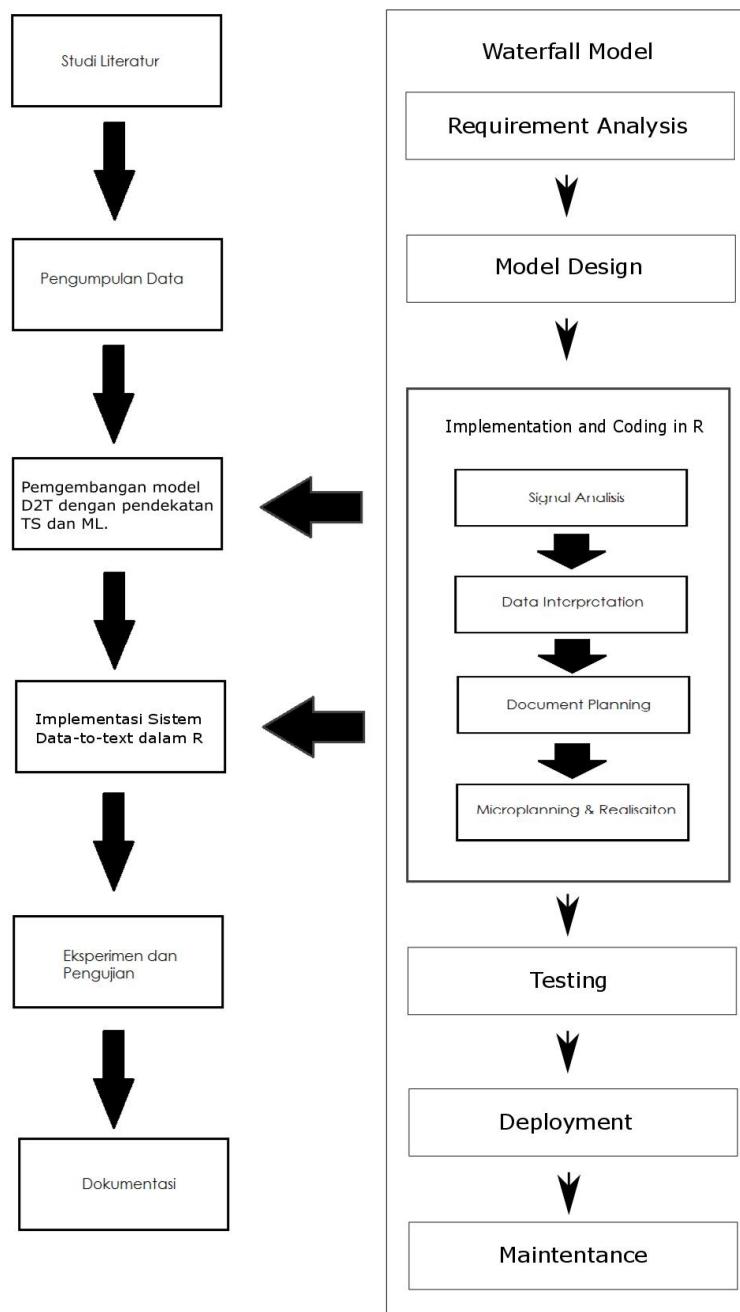
Setelah melakukan mengumpulan dan pemahaman materi pada proses studi literatur, penulis melakukan penelusuran terhadap data yang layak untuk digunakan dalam pembangkitan bahasa alami dengan sistem *Data-to-text*. Berdasarkan hasil penelusuran, penulis menarik kesimpulan bahwa data-data yang dibutuhkan terdiri dari beberapa data cuaca dan data kualitas udara dalam bentuk numerik.

Langkah selanjutnya setelah mengetahui data yang dibutuhkan adalah melakukan pengumpulan data. Data yang dikumpulkan terdiri dari data curah hujan, data cakupan awan, data suhu, data kecepatan angin, data arah angin, dan data kualitas udara. Data kualitas udara yang dibutuhkan adalah data SO₂, NO, NO₂, NOX, CO, O₃, PM₁₀, dan PM₂₅. Data-data tersebut diperoleh dari salah satu website pengelola data meteorologi dan geofisika di wilayah komUnitas otonom Galicia, Spanyol yaitu www.MeteoGalicia.gal. Untuk memperoleh data dari website tersebut, dapat mengakses halaman web tersebut dan memilih menu

“Acceso a Datos”. Data yang digunakan dalam penelitian ini adalah data yang berada di wilayah Mabegondo, kota a Coruna, daerah kom^{Unitas} otom Galicia, Spanyol. Rentang waktu yang digunakan terhitung satu tahun dari tanggal 06 Juli 2016 sampai dengan 6 Juli 2017 sebanyak 366 baris data.

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.

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.



Gambar 3.1 Metodologi Penelitian Pengembangan Sistem *Data-to-text* dengan Pendekatan *Time-series*

Gambar 3.1 menjelaskan bahwa dalam penyusunan penelitian ini terdiri dari serangkaian tahapan yang terdiri dari studi literatur hingga dokumentasi. Dalam tahap implementasi, dilakukan proses pembangunan sistem *Data-to-text* dengan arsitektur diatas.

3.2 Metode Penelitian

Ada dua proses utama yang dilakukan dalam penelitian ini yang dipaparkan dalam sub-bab 1.2.1 yang terdiri dari studi literatur dan observasi. Dalam penelitian ini, data dan informasi yang tersedia dapat meunjang penelitian. Metode yang digunakan dalam pengumpulan data adalah:

- Studi Literatur

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

- Observasi

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

3.3 Alat dan Bahan Penelitian

Untuk menunjang berbagai kebutuhan penelitian penulis menggunakan alat dan bahan untuk berlangsungnya penelitian yang terdiri dari perangkat keras, perangkat lunak dan bahan kajian serta sejumlah data sebagai berikut:

3.3.1 Alat Penelitian

Perangkat keras:

- Laptop Lenovo Seri G40-30 dengan spesifikasi sebagai berikut:
 - Processor intel Celeron N2840 @ 2.16GHz (2 CPUs)
 - Memori / Ram 2GB, DDR3

- *Hard disk 512 GB*
- Layar monitor 14 inch, WXGA
- Resolusi Layar 1366 x 768
- Dimensi Perangkat: 34.9 x 24.5 x 2.5 cm
- Tipe Grafis: Intel HD
- Speaker Dolby Advance Audio V2
- Slot: USB 3.0, USB 2.0, Headphone/Mic
- Batterai: 4-Cell External AC Adapter
- Berat: 2.1 Kg
- *2.4GHz Wireless Optical Mouse*
- *Keyboard*
- *Touch Pad*

Perangkat lunak:

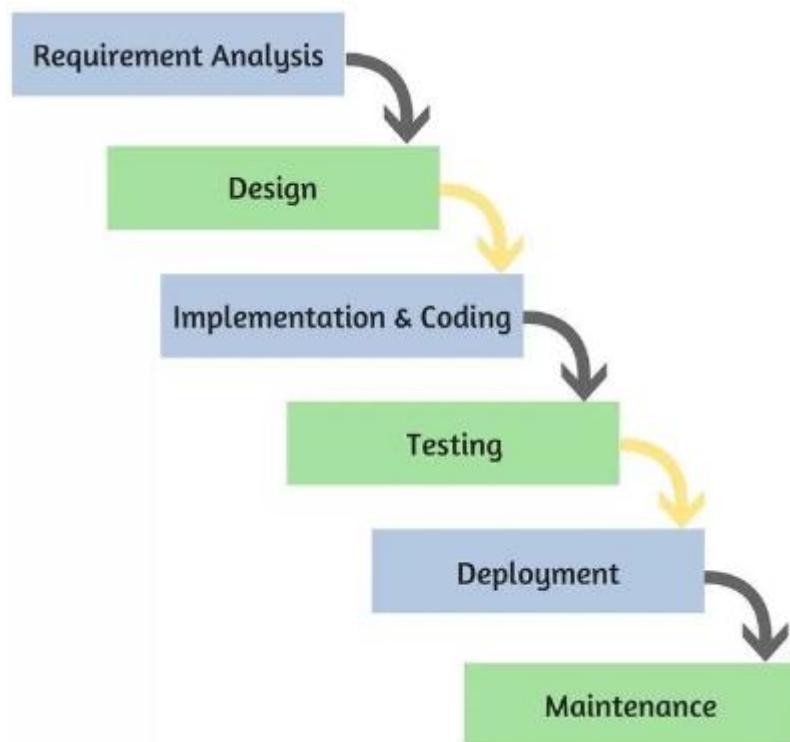
- Sistem Operasi Microsoft Windows 7, seri Ultimate x64 bit
- Microsoft Office Word 2007
- Microsoft Office Power Point 2007
- Microsoft Office Excel 2007
- Text Editor Notepad ++
- Text Editor Sublime Text, Versi: 3
- Rgui i386, Versi: 3.3.2
- Browser Mozilla Firefox
- Foxit Reader, Versi: 7.0.3.916

3.3.2 Bahan Penelitian

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 Language Generation*, *Machine Learning*, *Big Data* dan platformnya.

3.4 Proses Pengembangan Perangkat Lunak

Dalam penelitian ini, penulis mengembangkan perangkat lunak *Data-to-text* dengan tahapan pengembangan mengikuti metode *Waterfall*. Untuk diagram pengembangan perangkat lunak *Waterfall* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Pengembangan perangkat lunak model *Waterfall*

Setiap pada Gambar 3.2 terdiri dari tahapan berikut:

- *Requirement Definition*

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau study literatur. Seorang *System* analisis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan

sistem. Dokumen inilah yang akan menjadi acuan *System* analisis untuk menterjemahkan kedalam bahasa pemrograman.

- *Design System (Design sistem)*

Proses *Design* akan menterjemahkan syarat kebutuhan kesebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada: struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

- *Coding dan Unit Testing*

Coding merupakan penerjemahan *Design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menterjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *Testing* terhadap sistem yang telah dibuat tadi. Tujuan *Testing* adalah menemukan kesalahan-kesalahan terhadap *System* tersebut dan kemudian bisa diperbaiki.

- *Integration dan Testin*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, *Design* dan pengkodean maka sistem yang sudah jadikan digunakan oleh user.

- *Pemeliharaan (Operation and Maintenance)*

Perangkat lunak yang susah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau

System operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Pengumpulan Data

Penelitian ini menggunakan data klimatologi dan data kualitas udara yang diperoleh dari stasiun pemantauan milik lembaga meteorologi dan klimatologi galicia yaitu MeteoGalicia yang berlokasi di kota Mabegondo, Provinsi A Coruna, daerah kom*Unitas* otonom Galicia, Spanyol. Data yang digunakan terhitung satu tahun dari tanggal 06 Juli 2016 sampai dengan 6 Juli 2017 sebanyak 366 baris data, data tersebut diakuisisi dari situs web www.MeteoGalicia.gal. Untuk mengakses dataset yang tersedia, penulis mengunjungi situs web tersebut. Lalu, pada side-bar sebelah kanan yang berjudul *Informes Meteorológicos*, penulis memilih tautan dengan nama *Acceso a datos*, setelah itu mengikuti petunjuk yang diarahkan oleh situs web tersebut untuk mengakuisisi data. Sedangkan untuk mengakses data kualitas udara, pada *navigation bar* dalam website tersebut dipilih menu *calidade aire*. Pada *sub-navigation bar*, penulis memilih *consulta de datos*. Dalam website ini tersedia banyak data yang diperoleh dari stasiun pemantauan di masing-masing wilayah, akan tetapi setelah melakukan survey ke setiap wilayah, ditemukan bahwa stasiun pemantauan kota Mabegondo memiliki data dengan atribut paling lengkap.

Setelah melakukan serangkaian proses untuk mengumpulkan data, maka didapatkan bahwa data yang disediakan oleh MeteoGalicia dilokasi stasiun pemantauan tersebut terdiri dari data berikut:

- Tanggal Data Klimatologi dan Kualitas Udara
- Lokasi Data Klimatologi dan Kualitas Udara
- Curah Hujan per hari dengan satuan L/m^2 (Liter per meter kuadrat)
- Cakupan Awan per hari dengan satuan % (Persen)
- Rata -Rata suhu per hari dengan satuan $^{\circ}C$ (derajat celcius)

- Maksimum suhu per hari dengan satuan °C (derajat celcius)
- Minimum suhu per hari dengan satuan °C (derajat celcius)
- Suhu pengembunan per hari dengan satuan °C (derajat celcius)
- Suhu tanah per hari dengan satuan °C/10cm (derajat celcius per 10 centi meter)
- Suhu udara per hari dengan satuan °C/10cm (derajat celcius per 10 centi meter)
- Waktu pendinginan per hari dengan satuan °C/h (celcius per jam)
- Tiupan per hari dengan satuan km/h (kilometer per jam)
- Lama penyinaran per hari dengan satuan h (jam)
- *Global Daily Irradiation* per hari dengan satuan 10kJ/m^2 (10 kilo Joule per meter kuadrat)
- Insolasi per hari dengan satuan % (persen)
- Keseimbangan Air per hari dengan satuan L (liter)
- Referensi Evapotransporasi per hari dengan satuan L/m^2 (Liter per meter kubik)
- Tekanan Barometrik per hari dengan satuan hPa (hectopascal)
- Pengurangan tekanan permukaan laut
- Kelembaban relatif per hari dengan satuan % (persen)
- Kelembaban *maximum* per hari dengan satuan % (persen)
- Kelembaban *minimum* per hari dengan satuan % (persen)
- Kecepatan Angin per hari dengan satuan km/h (Kilometer per jam)
- Arah Angin per hari dengan satuan °(derajat)
- Data Karbon Monoksida (CO) per hari dengan satuan ppm (Part per Million)
- Data Nitrogen Monoksida (NO) per hari dengan satuan ppm (Part per Million)
- Data Nitrogen Dioksida (NO₂) per hari dengan satuan ppm (Part per Million)

- Data Nitrogen Oksida (NOX) per hari dengan satuan ppm (Part per Million)
- Data Ozone (O3) per hari dengan satuan ppm (Part per Million)
- Data Particulate Matter 10 mikronmeter per hari
- Data Particulate Matter 25 mikronmeter per hari
- Data Sulfur Dioksida (SO2) per hari dengan satuan ppm (Part per Million)

Berdasarkan hasil akuisisi dan analisis dari berbagai referensi, termasuk salah satunya didalam (Ramos-soto, dkk., 2015), penulis hanya menggunakan beberapa variabel data yang memungkinkan untuk digunakan dalam penelitian. Berikut adalah data yang diusulkan untuk membangkitkan bahasa alami:

- Curah Hujan per hari dengan satuan L/m² (Liter per meter kuadrat)
- Cakupan Awan per hari dengan satuan % (Persen)
- Rata -Rata suhu per hari dengan satuan °C (derajat celcius)
- Kecepatan Angin per hari dengan satuan km/h (Kilometer per jam)
- Arah Angin per hari dengan satuan °(derajat)
- Data Karbon Monoksida (CO) per hari dengan satuan ppm (Part per Million)
- Data Nitrogen Monoksida (NO) per hari dengan satuan ppm (Part per Million)
- Data Nitrogen Dioksida (NO2) per hari dengan satuan ppm (Part per Million)
- Data Nitrogen Oksida (NOX) per hari dengan satuan ppm (Part per Million)
- Data Ozone (O3) per hari dengan satuan ppm (Part per Million)
- Data Particulate Matter 10 mikronmeter per hari
- Data Particulate Matter 25 mikronmeter per hari
- Data Sulfur Dioksida (SO2) per hari dengan satuan ppm (Part per Million)

Kutipan data dari data klimatologi yang telah penulis dapatkan dari stasiun pemantauan yang disediakan didalam situs web tersebut dapat dilihat dalam Tabel 4.1, terdiri dari atribut yang dibutuhkan (temperatur rata-rata, kecepatan angin, cakupan awan, curah hujan, dan arah angin) sedangkan data kualitas udara dapat dilihat dalam Tabel 4.2, terdiri dari (karbon monoxida, nitrogen monoksida, nitrogen dioksida, nitrogen oksida, ozone, sulfur dioksida, PM10 dan PM25).

Tabel 4.1 Kutipan data cuaca dari MeteoGalicia

Date	Average Temperature °C	Wind Speed (km/h)	Cloud Coverage (%)	Rainfall (L/m ²)	Wind Direction (degree)
07/06/2016	21.3	5.47	40.8	0	315
08/06/2016	20.1	6.41	20.9	0	315
09/06/2016	19.5	7.02	27.2	0	315
10/06/2016	19.1	5.94	23.2	0	315
11/06/2016	18.7	5.44	77.5	0	180
12/06/2016	17.1	4.82	63.9	0	225
13/06/2016	16.7	6.44	31.5	0	315
....
06/07/2017	17.59	6.48	58.3	0	315

Tabel 4.1 merupakan kutipan data yang digunakan selama penelitian, untuk 366 baris data secara lengkap dapat dilihat pada lampiran.

Tabel 4.2 Kutipan data kualitas udara dari MeteoGalicia

Date	CO	NO	NO ₂	NOX	O ₃	PM ₁₀	PM ₂₅	SO ₂
06/07/2016	0.13	3	15	19	51	18	10	1
07/07/2016	0.11	1	10	10	56	14	7	1
08/07/2016	0.1	1	10	8	59	12	8	1
09/07/2016	0.11	2	10	11	57	12	7	1
10/07/2016	0.1	2	18	12	53	12	11	1
11/07/2016	0.13	9	31	29	44	12	8	1
12/07/2016	0.12	18	24	58	35	12	8	2
....
06/07/2017	0.18	32	61	90	1	0	12	7

Berdasarkan hasil analisis yang dilakukan menggunakan Microsoft Excel, detail dari setiap variabel pada data klimatologi dapat dilihat pada Tabel 4.3, sedangkan untuk data kualitas udara dapat dilihat pada Tabel 4.4.

Tabel 4.3 Detail variabel data cuaca

Nama Variabel	Satuan	Nilai Minimum	Nilai Maximum	Standar Deviasi	Nilai Rata-Rata
Curah Hujan	L/m ²	0	67.6	6.91	2.57
Suhu	°C	1	25.3	4.79	14
Kecepatan Angin	km/h	6.56	29.84	3.93	6.56
Arah Angin	°	45	315	76.9	226
Cakupan Awan	%	7.9	100	30.11	46.15

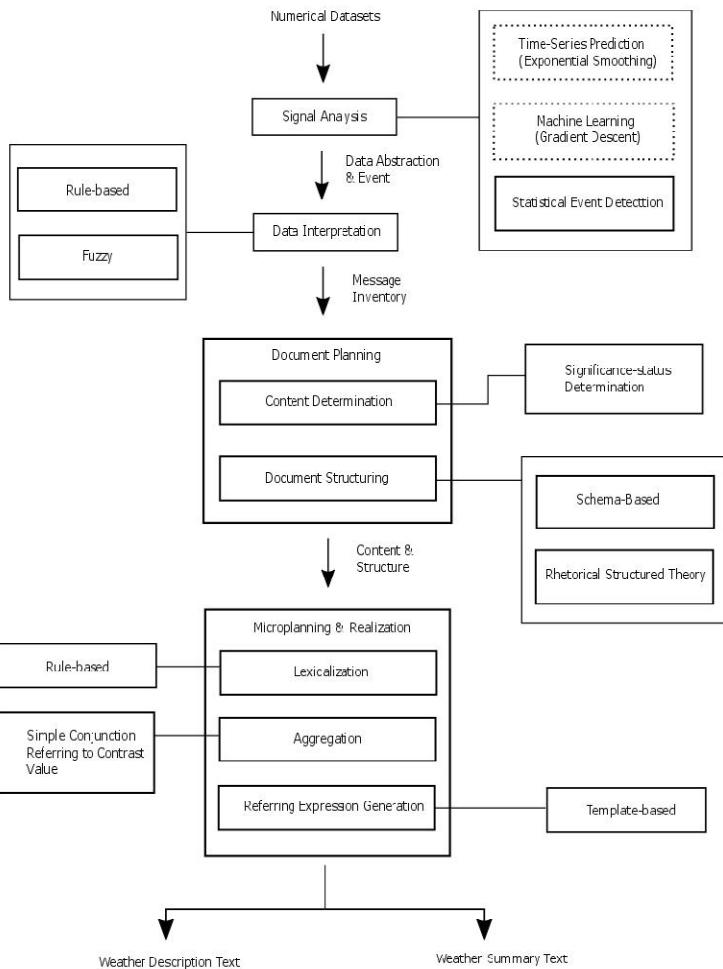
Tabel 4.4 memperlihatkan detail variabel tiap data kualitas udara yang digunakan saat penelitian ini.

Tabel 4.4 Detail variabel data kualitas udara

Nama Variabel	Satuan	Nilai Minimum	Nilai Maximum	Standar Deviasi	Nilai Rata-Rata
CO	Ppm	0	0.46	0.06	0.19
NO	Ppm	0	66	12.23	13.4
NO2	Ppm	0	74	12.51	25.87
NOX	Ppm	0	176	19.52	45.92
O3	Ppm	0	95	20.33	32.91
Per 10					
PM10	mikron meter	0	40	10.64	12.27
Per 25					
PM25	mikron meter	0	32	6.39	12.19

4.2 Pengembangan Model *Data-to-text* dengan *Machine Learning* dan *Time-series*

Model sistem *Data-to-text* yang dibangun dalam penelitian ini berdiri diatas pondasi dari arsitektur dalam sistem *Data-to-text* yang diusulkan oleh (Reiter, E, 2007), dan beberapa proses banyak mengacu pada apa yang dilakukan oleh (Schneider, 2013), (Ramos-soto, dkk., 2015), (Reiter, E, 2007) dan (Reiter dan Dale, 2000). Namun, pada proses *Signal Analysis* digunakan pendekatan *Machine Learning* dan *Time-Series*. Dalam penelitian ini, dibangkitkan dua informasi, yaitu informasi mengenai hasil prediksi dan informasi deskripsi hasil ringkasan cuaca selama satu bulan. Arsitektur yang dikembangkan pada penelitian ini dapat dilihat pada Gambar 4.1.



Gambar 4.1 Model *Data-to-text* dengan pendekatan *Machine Learning* dan *Time Series*

Dari Gambar 4.1, diperlihatkan bahwa pemanfaatan *Machine Learning* dan *time-series* berada pada tahap *Signal Analysis*. Pada sub-bab selanjutnya, akan dijelaskan mengenai deskripsi ringkasan cuaca dan kualitas udara bulanan, prediksi kualitas udara, dan pemanfaatan *Machine Learning* dalam penelitian ini. Konten pada sub-bab dibawah ini berkaitan dengan apa yang disajikan pada model *Data-to-text* yang tersedia pada Gambar 4.3.

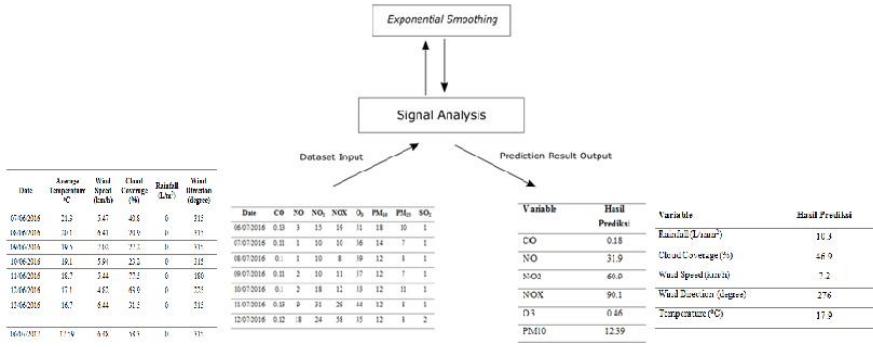
4.2.1 Model Proses *Signal Analysis*

Proses *Signal Analysis* merupakan proses analisis data, yaitu memproses data untuk menghasilkan ringkasan-ringkasan data yang akan disampaikan dalam bentuk bahasa alami. Jika dalam pengembangan *BABYTALK Family Sistem* yang dipaparkan dalam (Potret, dkk., 2009), tahapan *Signal Analysis* adalah memperoleh pattern dari *input* data numerik yang merupakan ringkasan kejadian selama 45 menit. Artinya, mereka mencari ringkasan kejadian dalam kurun waktu tertentu yang akan disampaikan. Dalam penelitian ini, seperti apa yang telah dijelaskan pada proses 4.8, penulis menyimpulkan seluruh kejadian mengenai keadaan klimatologis dan kualitas udara selama satu bulan dan melakukan prediksi untuk satu hari beikutnya.

4.2.1.1 Proses *Signal Analysis* Untuk Membangun Prediksi

Salah satu bagian dari proses *Signal Analysis* pada sistem ini dilakukan proses prediksi. Prediksi dilakukan agar informasi yang hendak disampaikan kepada pembaca lebih informatif, karena informasi yang disampaikan berisi mengenai kemungkinan-kemungkinan keadaan klimatologis yang akan terjadi dikemudian hari. Sehingga, dengan mengetahui informasi yang disampaikan, pembaca dapat mengantisipasi keadaan tertentu. Dalam melakukan prediksi, penulis menggunakan konsep prediksi *time-series* dengan algoritma *Exponential Smoothing*. *Time-series* digunakan karena dalam kasus ini, munculnya data untuk setiap variabel terjadi secara bersamaan, sehingga tidak memungkinkan untuk menggunakan konsep *Supervised learning*.

Untuk memperjelas tahapan pada proses *Signal Analysis* pada Gambar 4.2, khusus untuk pembangunan bahasa alami terkait dengan informasi prediksi, penulis membuat skematik tahap demi tahap dengan contoh data sebagai berikut. Terdiri dari detail *input*-*output* secara sekuensial berdasarkan arsitektur diatas.



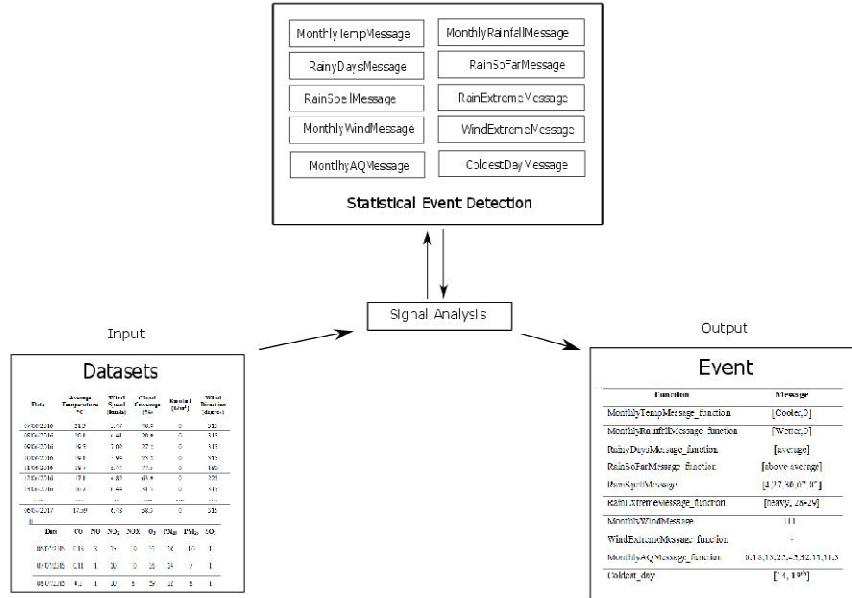
Gambar 4.2 Model proses *Signal Analysis* prediksi dengan *a exponential smoothing* dalam membangun pesan deskripsi cuaca dan kualitas udara.

Saat proses analisis, yang dilakukan adalah melakukan prediksi dengan *eksponensial smoothing* dan *outputnya* adalah nilai-nilai hasil prediksi (masih dalam bentuk numerik).

4.2.1.2 Proses *Signal Analysis* Untuk Menbangun Ringkasan Cuaca dan Kualitas Udara

Untuk mendeteksi setiap pesan yang akan disampaikan, dalam proses ini penulis melakukan analisis terhadap contoh *corpus* pembangkitan berita cuacayang ada dalam (Reiter dan Dale, 2000). Hasilnya, didapatkan sebelas pesan yang disampaikan dalam berita tersebut. Oleh karena itu penulis merumuskan pesan-pesan yang disampaikan sesuai dengan *corpus* yang dianalisis serta ditambahkan beberapa pesan terkait dengan kualitas udara. Pesan-pesan yang disampaikan diantaranya hasil dari membandingkan rata-rata temperatur bulan ini dengan rata-rata temperatur selama satu tahun, Membandingkan rata-rata curah hujan bulan ini dengan rata-rata curah hujan selama satu tahun, mengukur jumlah hari hujan selama satu bulan, membandingkan rata-rata curah hujan minggu ini dengan rata-rata curah hujan selama satu bulan, mencari urutan hujan terpanjang (hujan berturut turut), mencari nilai ekstrim dari hujan (hujan dengan klasifikasi terbesar, diatas atau sama dengan *heavy rain*), mencari rata-rata nilai kecepatan angin perbulan, mencari nilai ekstrim dari angin (hujan dengan klasifikasi terbesar, diatas atau sama dengan *Strong Gale*), mencari rata-rata nilai kualitas

udara perbulan dan mencari nilai terdingin selama satu bulan. Model dalam proses ini dapat dilihat pada Gambar 4.3.



Gambar 4.3 Model proses *Signal Analysis* pencarian *event* dalam membangun pesan deskripsi cuaca

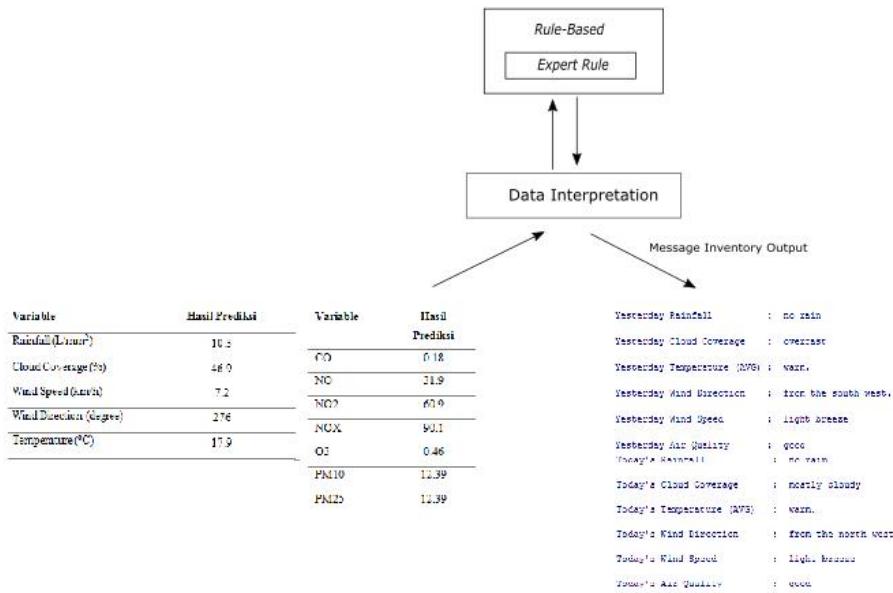
Dari Gambar 4.3 dapat dilihat bahwa proses ini menganalisis dataset untuk mendekripsi *event* terkait dengan pesan-pesan yang dicari, seperti pada kotak teratas diperlihatkan ada sepuluh pesan yang dicari. Lalu, *output* yang diharapkan dari proses ini adalah seluruh *event* yang didapatkan seperti pada kotak paling kanan pada Gambar 4.3.

4.2.2 Model Proses *Data Interpretation*

Dalam (Reiter, E, 2007) dijelaskan bahwa proses *Data Interpretation* merupakan tahap dimana sistem menganalisis pola atau *event* yang didapatkan dari hasil proses analysis sinyal, dan menerjemahkan pola atau *event* tersebut menjadi pesan-pesan yang akan disampaikan. Berdasarkan yang telah dilakukan oleh (Sripada & Gao, 2007) dalam melakukan interpretasi data dalam kasusnya mengenai detak jantung, setelah mendapatkan data detak jantung secara numerik

dalam penelitian tersebut menginterpretasikan nilai yang didapat dalam bentuk pesan teks. Mereka melakukannya dengan cara mendefinisikan pesan tertentu untuk nilai numerik tertentu. Contohnya, untuk detak jantung dibawah 100 didefinisikan dengan “*heart rate is temporarily low*”. Sedangkan didalam (Potret, dkk., 2009), proses *Data Interpretation* dilakukan 3 hal, yaitu membuat pesan, menentukan seberapa penting pesan tersebut, lalu mendeteksi relasi antar pesan. Untuk kasus mengenai data cuaca didalam (Sripada, Reiter, dan Davy, 2005), proses interpretasi data tidak melibatkan pencarian relasi diantara pola atau *event*, melainkan hanya menganalisis pola-pola umum yang dibutuhkan dalam kasus data cuaca.

Oleh karena itu, berdasarkan hasil analisis dari beberapa sumber, penulis menyimpulkan bahwa dalam kasus ini proses interpretasi data adalah mendeskripsikan *event*, pola, atau abstraksi data menjad pesan –pesan yang akan disampaikan. Beberapa hasil dari *Signal Analysis* yang akan diterjemahkan terdiri dari kecepatan angin, arah angin, curah hujan, temperatur, cakupan awan dan kualitas udara.



Gambar 4.4 Model Proses *Data Interpretation* dengan menggunakan *Expert Rules*.

Pada Gambar 4.4, diperlihatkan bahwa *input* dari *Data Interpretation* adalah abstraksi data hasil dari proses analisis sinyal analisis. Kemudian *output* yang dihasilkan adalah *Message Inventory*, yaitu keseluruhan pesan-pesan yang telah diinterpretasikan kedalam bentuk kata atau frase. Pesan-pesan ini didapatkan dengan menerjemahkan abstraksi data dengan pendekatan *rule-based* dan *Fuzzy*. Lalu, untuk memperjelas bagaimana langkah-langkah untuk menginterpretasikan seluruh proses pada tahap *Data Interpretation*, beberapa cara mengklasifikasikan pesan yang terdiri dari interpretasi kecepatan angin, interpretasi arah angin, interpretasi curah hujan, interpretasi cakupan awan, interpretasi temperatur, dan interpretasi kualitas udara.

4.2.2.1 Interpretasi Kecepatan Angin

Untuk mendeskripsikan kecepatan angin, penulis menerapkan klasifikasi yang telah ditetapkan oleh Russ Rowlet di *University of North California* dalam (Rowlet, 2001) dan memetakannya kedalam fungsi keanggotaan himpunan crisps pada Tabel 4.5.

Tabel 4.5 Klasifikasi kecepatan angin Russ Rowlet

No	Speed (km/h)	Kelas
1	$x < 2$	Calm
2	$2 \leq x < 5$	Light Air
3	$6 \leq x < 11$	Light Breeze
4	$12 \leq x < 19$	Gentle Breeze
5	$20 \leq x < 29$	Moderate Breeze
6	$30 \leq x < 39$	Fresh Breeze
7	$40 \leq x < 50$	Strong Breeze
8	$51 \leq x < 61$	Near Gale
9	$62 \leq x < 74$	Gale
10	$74 \leq x < 87$	Strong Gale
11	$88 \leq x < 102$	Storm
12	$103 \leq x < 118$	Violent Storm
13	$119 \leq x < 130$	Hurricane

4.2.2.2 Interpretasi Arah Angin

Untuk menginterpretasikan data arah angin, penulis menerapkan klasifikasi arah angin (*wind direction*) yang dimuat didalam (Zandlo, Spoden, Bouley, dan Ruschy, 2001) dengan klasifikasi seperti pada Tabel 4.6.

Tabel 4.6 Klasifikasi untuk menginterpretasikan arah angin menurut Zandlo dkk.

No	degree	class
1	348.75-11.25	North
2	11.25-33.75	North North East
3	33.75-56.25	North East
4	56.25-78.75	East North East
5	78.75-101.25	East
6	101.25-123..75	East South East
7	123.75-146.25	South East
8	146.25-168.75	South South East
9	168.75-191.25	South
10	191.25-213.75	South South West
11	213.75-236.25	South West
12	236.25-258.75	West South West
13	258.75-281.25	West
14	281.25-303.75	West North West
15	303.75-326.25	North West
16	326.25-348.75	North North West

4.2.2.3 Interpretasi Cakupan Awan

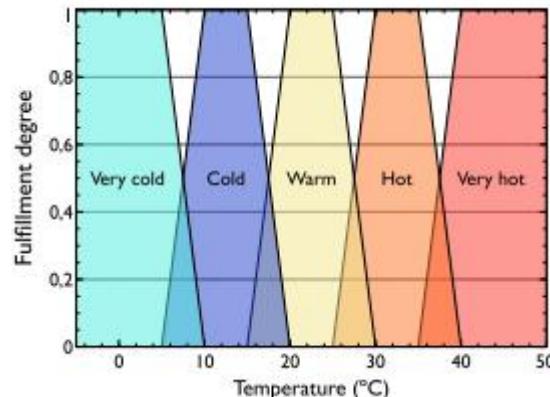
Kemudian untuk menginterpretasikan data cakupan awan, penulis merujuk pada klasifikasi yang telah dibuat oleh ahli meteorologis yang bernama Jeff Huby, didalam situs (Huby, 2010). Menurutnya, ada beberapa cara untuk melakukan *breaking down* terhadap cakupan awan, Salah satunya adalah membaginya menjadi 10 kelas yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Klasifikasi cakupan awan Jeff Huby

NO	Cakupan Awan (%)	kelas
1	0-10	Clear
2	10-20	Foggy
3	20-30	Mostly sunny
4	40-60	Partly cloudy
5	70-80	Mostly cloudy
6	80-90	Broken
7	90-100	overcast

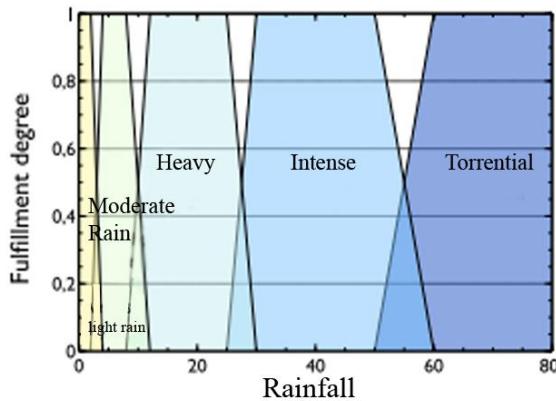
4.2.2.4 Interpretasi Temperatur

Untuk menginterpretasikan temperatur, dalam penelitian ini digunakan *Fuzzy* dengan mengacu pada *membership function* yang terdapat didalam (Ramos-soto, Alejandro, Bugarín, dan Barro, 2016). Membership function untuk menerjemahkan data temperature dapat dilihat pada Gambar 4.5.

**Gambar 4.5** Fungsi keanggotaan suhu

Sedangkan untuk menginterpretasikan curah hujan, juga merujuk pada penelitian (Ramos-soto, Alejandro, Bugarín, dan Barro, 2016). Membership function untuk menerjemahkan data curah hujan dapat dilihat pada Gambar 4.5. Namun dengan perbedaan istilah-istilah tertentu.

4.2.2.5 Interpretasi Curah hujan



Gambar 0.1 Fungsi keanggotaan curah hujan

Pada Gambar 4.6, diperlihatkan fungsi keanggotaan *Fuzzy* dari curah hujan yang terdiri dari *Light Rain*, *Moderate Rain*, *Heavy Rain*, *Intense Rain* dan *Torential Rain*. Karena fungsi keanggotaan untuk curah hujan dan suhu direpresentasikan dalam bentuk trapezoid, maka untuk menentukan *membership degree* digunakan rumus yang dapat dilihat pada tahapan *Implementasi*.

Fungsi untuk melakukan interpretasi data curah hujan terdiri *membership_partition()* dan *membership function()* yang akan dijelaskan pada tahap *Implementasi*.

4.2.2.6 Interpretasi Kualitas Udara

Untuk menginterpretasikan data kualitas udara, penulis menerapkan perhitungan *Polutant Standard Index* (PSI). PSI ditentukan dengan membagi konsentrasi masing-masing dari lima polutan udara utama, yaitu polutan atau *Particle Matter* (PM10), partikel halus (PM2.5), sulfur dioksida (SO₂), karbon monoksida (CO), ozon (O₃) dan nitrogen dioksida (NO₂) untuk jangka waktu tertentu yang ditetapkan oleh *National Ambient Air Quality Standard* (NAAQS) untuk mendapatkan angka antara 0 dan 500 (Crowder, Moore, DeRose, dan Franek, 1999). Untuk setiap polutan, sub-indeks dihitung dari fungsi linier

tersegmentasi yang mengubah konsentrasi ambien ke skala yang berkisar dari 0 sampai 500.

Breakpoint yang digunakan untuk menentukan masing-masing dari enam sub-indeks polutan tersebut tercantum pada Tabel 4.8.

Tabel 4.8 Index kualitas udara dalam *Pollutant Standard Index (PSI)*

<i>System Category</i>	<i>PSI Index</i>	PSI	PM25	PM10	SO ₂	CO	O ₃	NO ₂
Good	Good	0-50	0-12	0-50	0-80	0-5	0-118	0
Admissible	Moderate	51-100	13-55	51-15	81-365	5.1-10	119-157	0
	Unhealthy	101-200	56-150	151-350	366-800	10.1-17.0	158-235	1130
	Very Unhealthy	201-300	151-250	351-420	801-1600	17.1-34.0	236-785	1131-2260
Hazardous	Hazardous	301-400	251-350	421-500	1601-2100	34.1-46.0	786-980	2261-3000
	us	401-500	351-500	501-600	2101-2620	46.1-57.5	981-1180	3001-3750

Setiap sub-indeks i , dihitung dengan menggunakan fungsi linier tersegmentasi yang berhubungan dengan konsentrasi polutan, X_i terhadap nilai sub-indeks, I_i . Fungsi linier tersegmentasi terdiri dari segmen garis lurus yang menggabungkan koordinat diskrit (yaitu *breakpoint*). Untuk polutan i dan segmen j , koordinat titik j^{th} break diwakili oleh nilai sub-indeks $I_{i,j}$ dan konsentrasi $X_{i,j}$ memberikan pasangan yang dipesan ($X_{i,j}, I_{i,j}$). Jika konsentrasi yang diamati adalah X_i nilai sub-indeks yang sesuai I_i dihitung dengan menggunakan persamaan yang dapat dilihat pada Gambar 4.7.

$$I_i = \frac{I_{i,j+1} - I_{i,j}}{X_{i,j+1} - X_{i,j}} (X_i - X_{i,j}) + I_{i,j}$$

for $X_{i,j} \leq X_i \leq X_{i,j+1}$

Gambar 4.7 Perhitungan *Pollutant Standard Index* (PSI)

Keterangan dari setiap variabel yang ada pada Gambar 4.7, dapat dilihat pada Tabel 4.9.

Tabel 4.9 Keterangan rumus *Pollutant Standard Index* (PSI)

NO	Variabel	Keterangan
1	X_i	Nilai konsentrasi pollutant (yang diamati) ke-i.
2	$I_{i,j}$	Nilai PSI untuk polutan ke-i dan <i>breakpoint</i> ke-j (seperti yang diberikan pada Tabel).
3	$I_{i,j+1}$	Nilai PSI untuk polutan ke-i dan <i>breakpoint</i> ke-(j + 1) (seperti yang diberikan pada Tabel).
4	$X_{i,j}$	Konsentrasi untuk polutan dan <i>breakpoint</i> seperti yang diberikan pada Tabel.

Sebagai contoh bagaimana program mengimplementasikan persamaan dan keterangan diatas dalam menentukan nilai PSI, penulis akan mensimulasikan dalam notasi algoritmik bagaimana program berjalan secara tahap demi tahap.

- Langkah 1: definisikan tiap *breakpoint* bagi setiap variabel polutan (PM_{25} , PM_{10} , SO_2 , CO , O_3 , NO_2). Nilai *breakpoint* dari setiap variabel ini akan digunakan saat mengimplementasikan persamaan.
- Langkah 2 : definisikan tiap *breakpoint* untuk *Index* PSI.
- Langkah 3: Hitung nilai PSI bagi tiap variabel, dengan mengimplementasikan persamaan pada Gambar 4.7.
- Langkah 4: Setelah didapatkan nilai PSI dari tiap variabel, cari nilai PSI terbesar dari.
- Langkah 5: Setelah nilai PSI terbesar didapatkan, cari interval dari PSI *Index* yang sesuai dengan nilai ini dari setiap indeks yang tersedia.
- Langkah 6: Setelah nilai PSI *Index* didapatkan, lalu ubah kedalam kategori sistem.

Agar lebih jelas lagi, penulis mengimplementasikan langkah-langkah diatas untuk mendefinisikan satu set nilai polutan dan mencari *Index* dari kualitas udaranya. Misalnya, diketahui nilai polutan dari PM_{25} adalah $40 \mu\text{g}/\text{m}^3$. Maka penyelesaiannya adalah sebagai berikut.

- Berdasarkan Tabel 13, dilihat pada interval *breakpoint* dari variabel PM_{25} bahwa nilai 40 berada diantara 12-55.
- Oleh karena itu, Perhitungannya dilakukan pada segmen pertama, yaitu ($j = 1$). Untuk segmen ini, $X_{1,1} = 12 \mu\text{g}/\text{m}^3$ dan $X_{1,2} = 55 \mu\text{g}/\text{m}^3$ Dengan nilai indeks PSI yang sesuai dari $I_{1,1} = 50$ dan $I_{1,2} = 100$.
- Sehingga, perhitungannya dapat dilihat pada Gambar 4.8.

$$\begin{aligned}
 I_i &= \frac{I_{i,j+1} - I_{i,j}}{X_{i,j+1} - X_{i,j}} (X_i - X_{i,j}) + I_{i,j} \\
 &= \frac{100 - 50}{55 - 12} (40 - 12) + 50 \\
 &= 83
 \end{aligned}$$

Gambar 4.8 Hasil perhitungan *Pollutant Standard Index* (PSI)

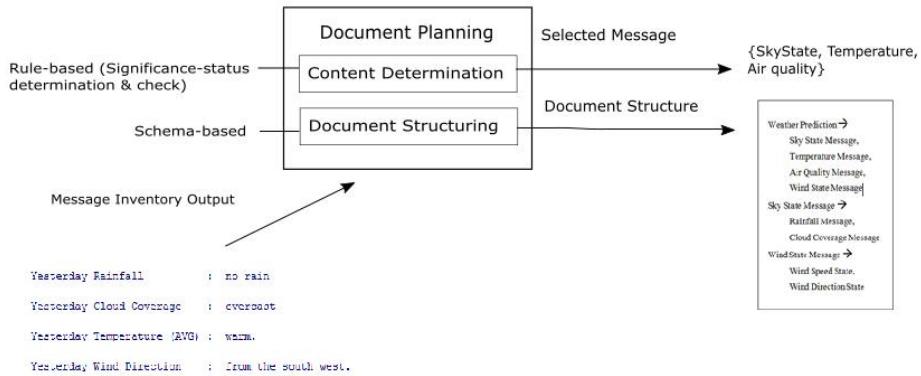
- Maka diperoleh nilai dari sub-indeks PSI bagi variabel PM25 adalah 83 seperti pada Gambar 4.9 diasumsikan bahwa seluruh nilai sub-*Index* PSI bagi setiap variabel telah didapatkan, dengan nilai I2 (PM10) = 48, I3 (SO2) = 46, I4 (CO) = 15, I5 (O3) = 45, I6 (NO2) = -*.
- Langkah berikutnya adalah mencari nilai terbesar diantara variabel yang telah diapatkan tersebut.

$$\begin{aligned}
 \text{PSI} &= \max(83, 48, 46, 15, 45, -*) \\
 \text{PSI} &= 83
 \end{aligned}$$

Gambar 4.9 Hasil perhitungan *Pollutant Standard Index* (PSI)

- Diperoleh angka 83 sebagai nilai terbesar, maka nilai ini dipetakan kedalam PSI *Index*, didapatkan *Index* “Moderate / Admissible”.

4.2.3 Model Proses *Document Planning*



Gambar 4.10 Model *Document Planning*

Proses *Document Planning* pada Gambar 4.10 memperlihatkan, bahwa *input* dari proses ini adalah *Message Inventory*, pesan-pesan masukan ini dilakukan pemilihan konten atau *Content Determination*, lalu dipetakan sesuai dengan struktur dokumen yang dibuat. Pemilihan konten menggunakan pendekatan *rule-based*, dimana pesan diberikan *Significance-status* dengan mendefinisikannya sebagai *Routine Message* atau *Significant Event Message*. *Routine message* berarti pesan yang selalu disampaikan, sedangkan *Significant Event Message* merupakan pesan yang akan disampaikan jika dan hanya jika memenuhi kondisi konstruksi. Contohnya, didalam penelitian ini, pesan terkait dengan kecepatan angin dan arah angin hanya akan disampaikan apabila kondisi angin lebih cepat atau sama dengan “*light breeze*”.

Proses *document structuring* menggunakan pendekatan *Schema-based* seperti apa yang dilakukan oleh (Turner, dkk., 2008) yaitu dengan membangun skema, baik dalam bentuk skema pada Gambar 4.12 (bagian *output* dari *document structuring*) maupun dengan membangun *Event Tree* yang akan diimplementasikan dengan pemaparan yang lebih detail pada sub-bab implementasi.

Setelah melakukan sinyal dan interpretasi data, pada tahap ketiga dalam arsitektur *Data-to-text* yang dipaparkan didalam (Reiter, E, 2007) terdiri beberapa tugas utama. Tugas-tugas tersebut terdiri dari determinasi konten dan pembuatan struktur dokumen. Pembuatan struktur dokumen yang akan disampaikan bertujuan untuk membuat teks yang koheren. Langkah-langkah perencanaan dokumen ini sama seperti yang dipaparkan didalam (Reiter dan Dale, 2000).

4.2.3.1 Document Plan untuk hasil prediksi

1. Content Determination

Proses *Content Determination* atau penentuan konten merupakan proses dimana sistem melakukan seleksi terhadap pesan apa yang akan disampaikan. Didalam (Reiter dan Dale, 2000), ide untuk menentukan determinasi konten adalah membagi pesan kedalam dua bagian, yaitu membuat “*Routine Message*” dan membuat “*Significant Event Message*”. *Routine Message* adalah pesan-pesan yang selalu dibangun setiap proses pembangkitan, apapun hasil dari proses analisis sinyal dan interpretasi data. Sedangkan “*Significant Event Message*” merupakan pesan-pesan yang hanya dibangun jika pesan tersebut memenuhi syarat konstruksinya. Misalnya, dalam kasus pembangkitan bahasa alami mengenai gempa bumi, pesan status hanya akan dibangkitkan jika status guncangan mencapai kode siaga.

Dalam membangun “*Routine Message*” dan “*Significant Event Message*” untuk membangkitkan kalimat hasil prediksi, penulis menetapkan “*Routine Message*” yang terdapat pada Gambar 4.11 dan significant event message yang terdapat pada Gambar 4.12.

```

Prediction - Routine Message →
Construct Message {
Sky State { Rainfall + Cloud Coverage } Prediction
Temperature Prediction
Air Quality Prediction}

```

Gambar 4.11 Routine message untuk pesan prediksi

```

Prediction - Significant Event Message →
IF Wind Speed >= "Strong Breeze" |
THEN Construct Message{
Wind State { Wind Speed + Wind Direction } Prediction
}

```

Gambar 4.12 Significant Event Message untuk pesan prediksi

2. Document Structuring

Proses ini merupakan pembangunan struktur dokumen yang akan disampaikan. Didalam (Reiter dan Dale, 2000), dijelaskan bahwa ada beberapa cara untuk mendapatkan struktur dokumen, dua diantaranya adalah dengan membangun skema dokumen dan penalaran eksplisit. Runtutan ide dalam membangun skema dokumen telah dijelaskan didalamnya. Yang pertama, teks diharapkan mengikuti pola konvensional, artinya dibutuhkan *Initial Corpus* untuk mengetahui bagaimana manusia membuat pesan yang terkait. Pola ini dapat didapatkan dengan menggunakan 'tata bahasa teks' yang mendikte konten dan memastikan struktur yang koheren. Pola yang didapat ini menentukan bagaimana rencana dokumen tertentu dapat dibangun dengan menggunakan skema atau pesan yang lebih kecil.

Karena dibutuhkan *Initial Corpus*, untuk menyesuaikan data yang tersedia maka penulis melakukan hipotesis terhadap *Target Corpus* atau harapan *output* berdasarkan pengamatan dari beberapa sumber, salah satunya dari (Ramos-Soto,

Pereira-Farina, Bugarin, dan Barro, 2015). *Target Corpus* yang dibuat untuk mendeskripsikan prediksi cuaca tersebut dapat dilihat pada Gambar 4.13.

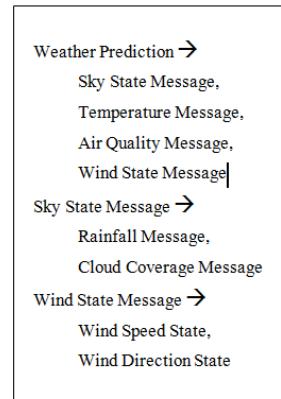
BASED ON PREDICTION RESULT, PREDICTED THAT A STORM RAIN WILL COME COVERED WITH OVERCAST CLOUD. FOLLOWED BY VERY WARM TEMPERATURE. WITH RESPECT TO THE AIR QUALITY STATE, IT WILL CHANGE PROGRESSIVELY TO BAD. WIND WILL BLOW VERY STRONG FROM THE NORTH.

Gambar 4.13 *Target corpus* hasil prediksi

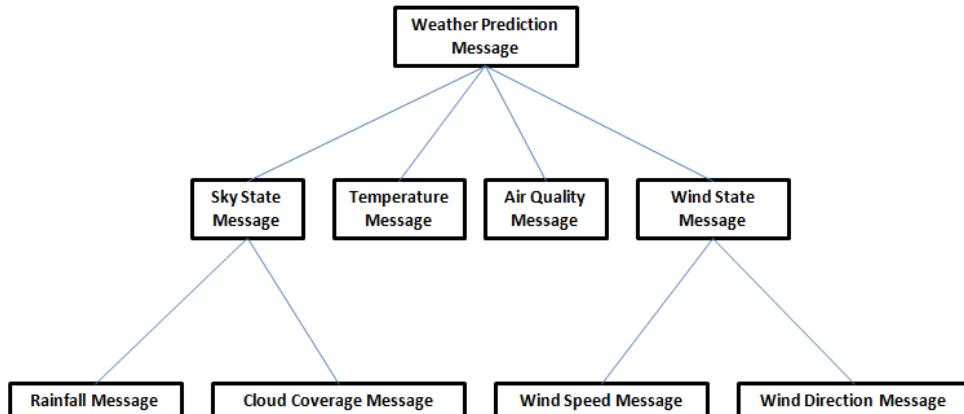
Paragraf pada Gambar 4.13 akan digunakan sebagai *Target Text*, langkah-langkah untuk mendapatkan skema dokumen dijelaskan didalam (Reiter dan Dale, 2000) adalah sebagai berikut:

1. Ambil contoh sejumlah teks yang sama, dalam kasus ini mengambil contoh teks deskripsi mengenai cuaca, dapat dilihat pada Gambar 4.13.
2. melakukan identifikasi terhadap pesan-pesan, dan mencoba menentukan bagaimana setiap pesan dapat dibangun dari data masukan
3. Mengusulkan aturan atau struktur yang menjelaskan mengapa pesan “x” ada dalam teks A tetapi tidak dalam teks B-ini mungkin lebih mudah jika pesan disusun menjadi taksonomi.
4. Diskusikan analisis ini dengan pakar.

Maka didapatkan struktur dokumen yang akan dibuat, penulis akan merepresentasikannya dalam bentuk skema seperti yang dapat dilihat pada Gambar 4.14 dan pohon struktur dekumen yang dapat dilihat pada Gambar 4.15.



Gambar 4.14 Skema untuk pesan prediksi.



Gambar 4.15 Event Tree untuk pesan prediksi.

4.2.3.2 Document Plan untuk hasil deskripsi cuaca

Dalam membuat perencanaan dokumen untuk menyampaikan pesan hasil *weather summary*, penulis menggunakan pendekatan *Re-Structured Text* (RST). Untuk mendapatkan *Initial Corpus* yang digunakan memberitakan ringkasan cuaca selama satu bulan, penulis menggunakan apa yang dipaparkan dalam (Reiter dan Dale, *Building Natural Language Generation Systems*, 2000) yang dapat dilihat pada Gambar 4.16.

THE MONTH WAS COOLER AND DRIER THAN AVERAGE, WITH THE AVERAGE NUMBER OF RAIN DAYS, BUT THE TOTAL RAIN FOR THE YEAR SO FAR IS WELL BELOW AVERAGE. ALTHOUGH THERE WAS RAIN ON EVERY DAY FOR 8 DAYS FROM 11TH TO 18TH, RAINFALL AMOUNTS WERE MOSTLY SMALL.

Gambar 4.16 Target Teks untuk *weather summary*

Didalam (Reiter dan Dale, 2000) dijelaskan bahwa dalam menentukan *Target Corpus*, yang terlebih dahulu dilakukan adalah menghilangkan informasi yang tidak tersedia dalam bentuk komputasi yang dapat dipertanggungjawabkan, proses ini disebut dengan *Simplification*.

Pada Gambar 4.16 diperlihatkan bahwa data-data yang disampaikan terdiri dari beberapa variabel yang tersedia dalam penelitian ini. Meskipun dalam penelitian ini seluruh data diatas tersedia, namun akan digunakan struktur yang sedikit berbeda, yaitu dengan menambahkan variabel kualitas udara. Sehinnga, *Target Text* yang akan disampaikan untuk mendeskripsikan ringkasan cuaca adalah sebagai berikut.

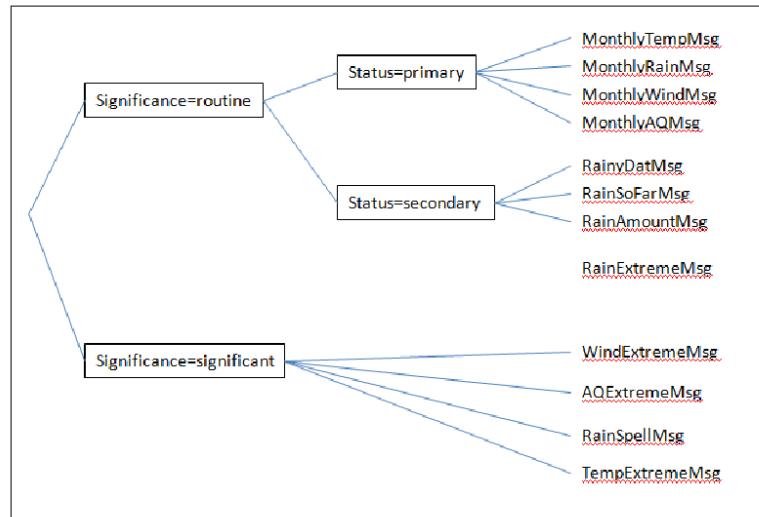
THE MONTH WAS COOLER AND DRIER THAN AVERAGE, WITH AVERAGE NUMBER OF RAIN DAYS, BUT THE TOTAL RAIN FOR THE YEAR SO FAR IS WELL BELOW AVERAGE. ALTHOUGH THERE WAS RAIN ON EVERY DAY FOR 8 DAYS FROM 11TH TO 18TH, RAINFALL AMOUNTS WERE MOSTLY SMALL. HEAVY RAIN WERE HAPPENED IN 12TH, 13TH, AND 14TH. THE MONTH WAS MOSTLY CALM, BUT STRONG WINDS BLEW ON 20TH, 24TH AND 26TH. AVERAGE AIR QUALITY WAS CALCULATED ADMISSIBLE, BUT HAZZARDOUS INDEX FOUND IN 12TH.

Gambar 4.17 Target teks untuk *weather summary*.

Dari *Target Text* pada Gambar 4.17, diperoleh bahwa pesan-pesan yang disampaikan terdiri dari beberapa informasi berikut:

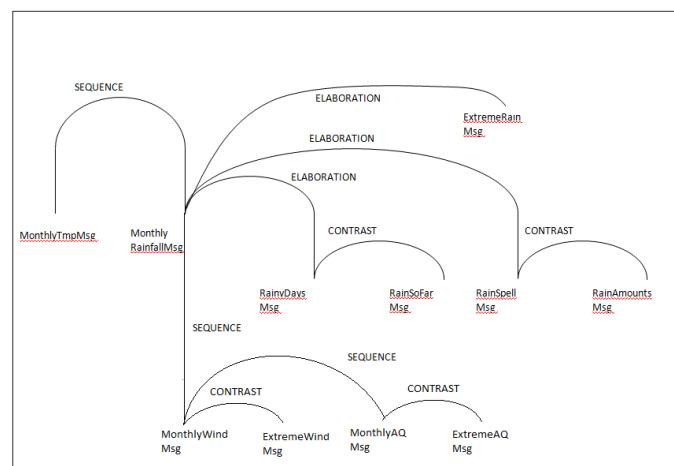
1. Content Determination

Pemilihan konten untuk ringkasan cuaca ini, penulis menetapkan tipe pesan kedalam routine message dan significant event message seperti pada Gambar 4.18.



Gambar 4.18 Content Determination untuk *weather summary*

Kemudian, untuk merepresentasikan rancangan dokumen untuk hasil dari *weather summary*, maka sistem akan menghasilkan rencana dokumen dalam bentuk *Rhetorical Structured Text* yang dapat dilihat pada Gambar 4.19.



Gambar 4.19 Rhetorical Structured Text untuk *weather summary*

Jenis relasi pada struktur RST adalah sebagai berikut:

- *Sequence*

Dua pesan dapat dihubungkan oleh hubungan *Sequence* jika keduanya memiliki atribut Message-Status=Primary

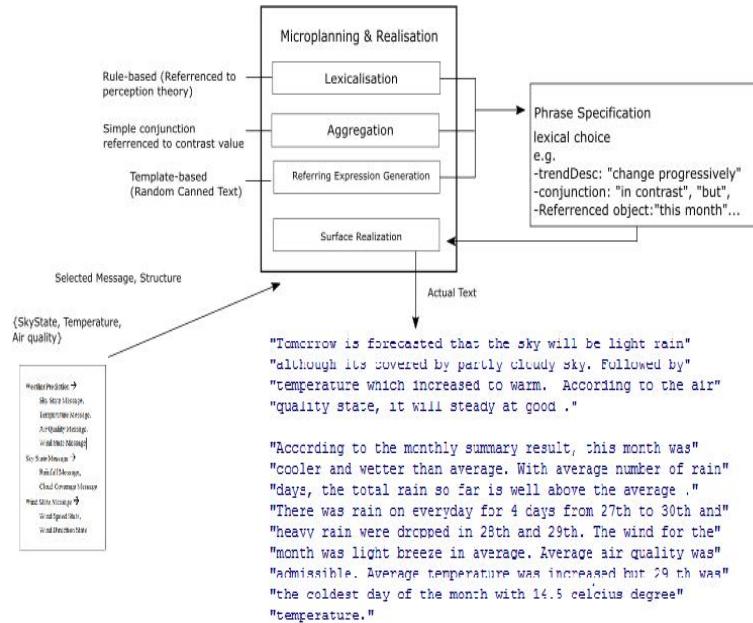
- *Elaboration*

Dua pesan dapat dihubungkan dengan hubungan *Elaboration* jika kedua memiliki topik pesan yang sama, dan pesan inti memiliki atribut Message-Status=Primary

- *Contrast*

Dua pesan bisa dihubungkan dengan Hubungan *Contrast* jika memiliki topik pesan yang sama dan kemungkinan memiliki perbedaan arah (sebanding atau berbanding terbalik)

4.2.4 Model Proses *Microplanning* dan *Realisation*



Gambar 4.20 Model *Microplanning* dan *Realisation* untuk proses prediksi dan *weather summary*

Pada proses *Microplanning* dan *Realization* seperti pada Gambar 4.20, ada tiga aspek yang perlu diperhatikan, yaitu *lexicalisation*, *aggregation* dan *referring*

expression generation. Proses *lexicalisation* dalam penelitian ini menggunakan pendekatan *rule-based*, dimana setiap pilihan leksikal dibangkitkan berdasarkan rule yang diperoleh dari berbagai sumber. Misalnya, untuk merepresentasikan persepsi perilaku perubahan data dari kualitas udara, pemilihan frase yang tepat akan melalui beberapa tahapan sehingga *outputnya* adalah frase ungkapa seperti “*change progressively*”, “*shifted*”, dan lain-lain. Sedangkan *aggregation* merupakan proses bagaimana pesan terkait dihubungkan, baik menghubungkan frase atau pun menggabungkan kalimat. Selanjutnya, *referring expression generation* merupakan bagaimana mendefinisikan subjek informasi, contohnya: “*this month....*”.

Tugas utama dari tahap *Microplanning* adalah mengemas pesan yang didapatkan sehingga dapat tersusun sebaik mungkin mengikuti struktur dokumen yang telah dibuat pada tahap sebelumnya. *Output* dari proses ini adalah *Phrase Specification* atau spesifikasi frase. Spesifikasi frase merupakan satu set instruksi untuk menghasilkan teks yang disusun secara gramatiskal. Untuk membuat spesifikasi frase, dibutuhkan tiga proses utama. Proses-proses tersebut terdiri dari *lexicalisation*, *aggregation*, dan *referring general expression*.

1. Lexicalisation

Proses *Lexicalisation* atau leksikalisis adalah menentukan kata-kata apa yang akan digunakan untuk merujuk pada konsep dan kemungkinan. Didalam (ARRIA, 2015), Aturan leksikalisis menentukan bagaimana, berdasarkan informasi kontekstual, konsep atau elemen pengetahuan lainnya harus diungkapkan dengan kata-kata. Seringkali proses ini melibatkan pilihan acak dari alternatif untuk memberi variasi dalam teks. Juga terkadang pemetaan dari konten pesan ke kata-kata dilakukan dengan cara yang sederhana dan mudah jika memungkinkan. Ada banyak cara yang berbeda agar konten yang sama dapat diungkapkan, jadi kita perlu menentukan aturan yang mengatur variasi.

Dalam proses leksikalisis ini, penulis merujuk pada apa yang telah dilakukan oleh (Ramos-soto, dkk., 2015) untuk menentukan spesifikasi frase yang

akan digunakan untuk mengekspresikan *Trend* pada variabel tertentu. Proses ini dinamakan dengan *Trend Description*. Penulis menerapkan proses mendeskripsikan *Trend Description* untuk beberapa variabel yang memungkinkan, diantaranya adalah kualitas udara dan temperatur. Untuk contoh hasil dari frase *Trend Description* dapat lihat pada kutipan dari *Target Corpus* berikut.

**based on prediction result, air quality state will
change progressively to hazzardous.**

Gambar 4.21 Kutipan instial target korpus

Frase “*change progressively*” yang ada pada Gambar 4.21 tersebut merupakan hasil dari proses *Trend Description* untuk mendeskripsikan bagaimana trend pergantian status kualitas udara menjadi “*hazzardous*”. Tahapan-tahapan untuk membangun trend description menurut (Ramos-soto, dkk., 2015) adalah sebagai berikut.

Misalnya, data yang diperoleh ada pada Tabel 4.10.

Tabel 4.10 Kutipan data kualitas udara

Air Quality Data		
Yesterday (aq1)	Today (aq2)	Tomorrow (aq3)
Admissible	Bad	Hazzardous

AQ = Himpunan Air Quality

AQI = Himpunan Air Quality dengan *Index Angka*

Pertama, menentukan himpunan variabel setiap pesan yang didapatkan, dalam kasus kualitas udara ini dibuat himpunan AQ yang berisi sebagai berikut.

AQ = {*Admissible, Bad, Hazzardous*}

Kemudian setiap anggota himpunan didalam himpunan AQ dipetakan kedalam himpunan AQI sesuai nilai *Index* nya. *Index* setiap anggota himpunan dipetakan berdasarkan urutannya didalam partisi keanggotaan yang telah ditetapkan pada proses *Data Interpretation*. Urutan tersebut dapat dilihat pada Tabel 4.11.

Tabel 4.11 *Index* kualitas udara

Partisi	Index
Good	0
Admissible	1
Bad	2
Hazardous	3

Setelah dipeatakan menurut masing-masing *Indexnya*, didapatkan himpunan *Air Quality Index* (AQI), yaitu $AQI = \{1,2,3\}$. Lalu yang selanjutnya dilakukan adalah mencari nilai *IV* (*Index Variation*) dengan rumus sebagai berikut :

$$IV = \{ iv1 = aqi2 - aqi1, iv2 = aqi3 - aqi2, aq3 \}$$

Maka didapatkan nilai *Index Variation* (IV) sebagai berikut :

$$IV = \{ iv1 = 2-1, iv2 = 3-2, iv3 = "Hazardous" \}$$

$$IV = \{ 1, 1, "Hazardous" \}$$

Setelah didapatkan nilai IV, maka yang selanjutnya dilakukan adalah memetakannya kedalam himpunan IVL dengan rules sbagai berikut:

$$\left. \begin{array}{l} iv_1 \\ \hline '+' \text{ jika } iv_1 > 0 \\ '-' \text{ jika } iv_1 < 0 \\ '0' \text{ jika } iv_1 = 0 \end{array} \right\} \begin{array}{l} \xrightarrow{\hspace{2cm}} \text{Lebih buruk / menurun} \\ \xrightarrow{\hspace{2cm}} \text{Lebih Baik / meningkat} \\ \xrightarrow{\hspace{2cm}} \text{Tidak berubah / stabil} \end{array}$$

Sehingga didapatkan $LD_{AirQuality} = \{ "+", "+", Hazardous \}$

Maka, setelah didapatkan nilai $LD_{AirQuality}$, realisasikan bentuk linguistik dengan mengasosiasikan apa yang telah didapat dengan template yang telah dibuat. Saat $LD_{AirQuality}$ didapatkan, template yang terkait dipilih dan aturan ahli (yang diberikan oleh pakar kualitas udara) memeriksa apakah kondisi cuaca yang diperoleh cukup relevan untuk disertakan sebagai komentar tambahan dalam deskripsi. Template yang dibuat dapat dilihat pada Gambar 4.22.

{AQ}	$\rightarrow \{ChangeType\} \{ChangeLabel\}$
{ChangeType}	$\rightarrow \{stable\} \{MediumChange\} $ $\{StartChange\} \{EndChange\}$ $\{ProgressiveChange\}$
{stable}	$\rightarrow \text{Maintain} \text{Maintain}$
{MediumChange}	$\rightarrow \text{Worsen} \text{Improve} \text{Worsen}$
{StartChange}	$\rightarrow \text{Worsen} \text{Maintain} \text{Improve} \text{Maintain}$
{EndChange}	$\rightarrow \text{Maintain} \text{Worsen} \text{Maintain} \text{Improve}$
{ProgressiveChange}	$\rightarrow \text{Worsen} \text{Worsen} \text{Improve} \text{Improve}$
{ChangeLabel}	$\rightarrow \text{Good} \text{Admissible} \text{Bad} \text{Hazardous}$

Gambar 4.22 *Trend Description* untuk mendeskripsikan persepsi perubahan data kualitas udara.

Dari serangkaian proses diatas dihasilkan nilai dari *Index Variasi* untuk deskripsi linguistik kualitas udara adalah : $LD_{AirQuality} = \{ "+", "+", \text{Bad}\}$. Lalu himpunan $LD_{AirQuality}$ ini disesuaikan dengan template yang telah dibuat.

Hasilnya, anggota himpunan $LD_{AirQuality}$ pertama yang berisi “+” yang berarti meningkatnya nilai *Index* sehingga dideskripsikan sebagai memburuk atau “worsen”, dan anggota himpunan $LD_{AirQuality}$ kedua pun berisi “+”. Sehingga jika diasosiasikan dengan template dalam Gambar 4.23 yang terkait dengan “WorsenWorsen” adalah $ChangeType \rightarrow \text{Progressive Change}$. Setelah didapatkan tipe deskripsi tren, lalu realisasikan frase dengan mengambil salah satu deskripsi dengan tipe progresif didalam *word bank*.

<code>{ProgressiveChange}</code> → {"change progressively to", "turn progressively to", "change steadily to", "turn continually to", ... }
--

Gambar 4.23 *Progressive Change word bank*

Selain mendeskripsikan tren kualitas udara, dalam penelitian ini juga dilakukan pendeskripsi tren untuk variabel suhu. Namun proses deskripsi akan lebih sedarhana jika dibandingkan dengan mendeskripsikan kualitas udara. Dalam membangun *Trend Description* untuk variabel suhu, penulis hanya membandingkan suhu pada hari ini dan pada hari esoknya, dideskripsikan dengan “increased” dan “decreased”.

2. *Aggregation*

Strategi agregasi menentukan bagaimana menggabungkan beberapa pesan bersama untuk menghasilkan spesifikasi frase yang sesuai dengan kalimat kompleks. Dalam (Ramos-soto, Alejandro, Bugarín, dan Barro, 2016), tahap *aggregation* serupa dengan proses *Document Structuring*, tahap ini berkaitan dengan menentukan bagaimana informasi dipetakan menjadi struktur pada tingkatan yang lebih rendah, yaitu menjadi kalimat.

Dalam penelitian ini, strategi agregasi digunakan dalam menggabungkan pesan-pesan yang memiliki keterkaitan. Dalam melakukan penggabungan pesan ini, penulis membagi proses agregasi menjadi dua bagian. Bagian tersebut terdiri dari Aggregasi frase menjadi kalimat dan agregasi kalimat menjadi paragraf. Dalam melakukan agregasi, menurut (Reiter dan Dale, 2000) diperbolehkan untuk menggunakan teknik *Simple Conjunction*, dimana penggabungan kaliman menggunakan kata-kata penghubung seperti “and”, “although”, “but”, dan lain-lain sesuai dengan kebutuhan sistem. Namun, dalam penelitian ini penulis tetap memperhatikan aspek leksikal dalam memilih kata konjungsi yang digunakan untuk menghubungkan frase dan kalimat. Pohon pengambilan kata konjungsi dalam proses agregasi dapat dilihat pada Gambar 4.24.

Sesuai dengan struktur dokumen yang telah dibuat pada tahap *Document Planning*, ada beberapa frase yang memiliki keterkaitan satu sama lain. Beberapa frase tersebut akan digabungkan menjadi kalimat. Yang pertama, agregasi frase akan dilakukan untuk memperoleh kalimat *Sky State* yang terdiri dari pesan-pesan yang mengandung informasi mengenai hujan dan cakupan awan.

**Based on prediction result, tomorrow galicia's status will be
heavy rain covered with overcast sky.**

Gambar 4.24 Contoh *Phrase Aggregation*

Pada Gambar 4.24, dapat dilihat bahwa kalimat yang dibangun terdiri dari pesan yang berisi informasi mengenai curah hujan dan keadaan cakupan awan. Kata penghubung atau *Conjunction* yang dipilih merupakan hasil dari serangkaian proses berikut-proses sebagai berikut:

Untuk memilih kata penghubung secara leksikal, sistem perlu mengetahui bagaimana hubungan antara pesan pertama dan pesan kedua. Contohnya, jika pesan pertama atau pesan curah hujan memberikan informasi “*heavy rain*” dan cakupan awan yang didapatkan adalah “*overcast*”, maka secara persepsi manusia kedua pesan ini ada didalam keadaan sebanding. Maka hasil agregasi dari contoh ini dapat dilihat pada Gambar 4.24.

Contoh lainnya, jika pesan pertama atau pesan curah hujan memberikan informasi “*no rain*” namun cakupan awan yang didapatkan adalah “*mostly cloudy*”, artinya dalam persepsi manusia kedua pesan ini ada didalam keadaan kontras. Oleh karena itu, berkat terinspirasi dari (Ramos-soto, dkk., 2015), penulis menggunakan nilai kontras untuk menentukan kata konjungsi yang akan digunakan untuk membentuk kalimat dengan cara yang sama sebagai mana (Ramos-soto, dkk., 2015) menentukan *Trend Description* untuk mengekspresikan hasil analisis tren.

Berikut adalah tahapan bagaimana penulis memilih kata konjungsi untuk menggabungkan pesan curah hujan dengan pesan cakupan awan.

- Definisikan nilai kontras antara curah hujan dan cakupan awan seperti pada Tabel 4.12.

Tabel 4.12 Nilai kontras dalam proses aggregasi dengan *simple conjunction*.

	Nilai Kontras	Partisi	Nilai Kontras
No Rain	0	Clear	0
Light Rain	0	Foggy	0
Moderate Rain	1	Mostly sunny	0
Intense Rain	1	Partly cloudy	1
Torential	1	Mostly cloudy	1
		Broken	1
		overcast	1

- Tentukan konjungsi yang akan digunakan sesuai dengan nilai kontras yang didapat.

3. Referring Expression Generations

Tahap ini berkaitan dengan pemilihan kata atau ungkapan yang memungkinkan untuk mengidentifikasi entitas. Menurut (Reiter dan Dale, 2000), dalam hal ini *Referring Expression Generations* relatif sederhana, karena dapat dilakukan secara *hard code* pada saat realisasi dokumen planning. Namun, dalam penelitian ini digunakan dengan pendekatan nilai acak. Setiap daftar *Referring Expression Generations* yang ada didalam *word bank* digenerate secara *random* untuk mewujudkan variasi text seperti yang dapat dilihat pada Gambar 4.46.

4. Structure Realization

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. Maka, dalam proses ini penulis menggabungkan pesan-pesan yang telah dibuat kedalam struktur dokumen.

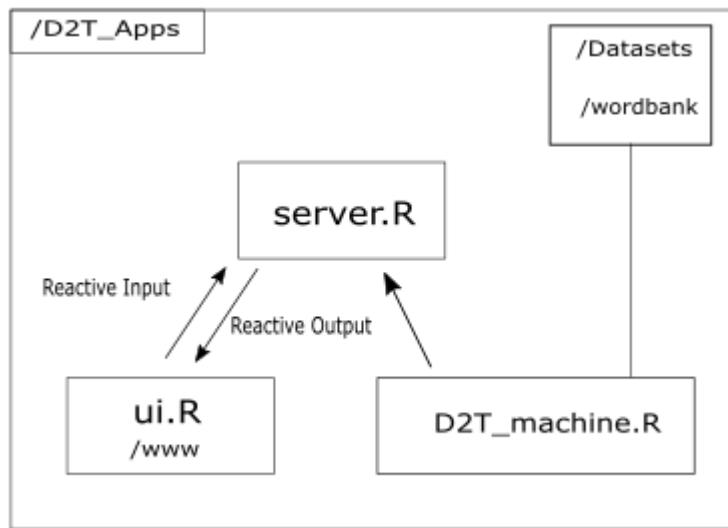
4.2.5 Pemanfaatan *Machine Learning*

Dalam (Al-Hamami dan Al-Samadi, 2013) dijelaskan bahwa setiap sistem yang dirancang menggunakan data bertujuan untuk memberikan integritas data dan kelengkapan. Beberapa masalah yang sering dihadapi saat mengambil data atau memprediksi nilai dari data yang yang disebut "*Null Values*". Nilai *Null* mewakili nilai atribut yang mungkin tidak diketahui saat ini atau mungkin tidak berlaku untuk tupel. Nilai khusus ini disebut null dan juga dikenal sebagai nilai *undefined*. Alasan di balik Nilai *Null* dapat diklasifikasikan menurut jenis *Null Value* sebagai berikut: (1) A-MARKS: Data berlaku tapi tidak diketahui, Artinya data tersebut sebenarnya ada tapi kemudian hilang, (2) I-MARKS: Data tidak dapat diterapkan; Artinya data tersebut tidak ada dalam kenyataan.

Sama halnya seperti pada ruang lingkup data cuaca dimana data yang diakuisisi tidak selalu lengkap. Misalnya, didalam situs web MeteoGalicia atau BMKG pun terkadang ada sejumlah data yang bernilai -8888 dan -9999. Dimana data tersebut memiliki artinya masing –masing, data -8888 berarti data tidak tersedia, sedangkan data -9999 berarti data tidak terjangkau. Jika sebuah sistem *Data-to-text* digunakan untuk otomatisasi penulisan ringkasan, maka sistem akan melakukan *handle* sepenuhnya dari proses analisis data hingga pembangkitan teks. Kebanyakan sistem *Data-to-text* bersifat *Real-Time*, artinya melakukan proses pembangkitan sesuai dengan tenggat waktu yang ditentukan. Misalnya, sebuah sistem *Data-to-text* akan melakukan proses pembangkitan baru apabila data baru dari sensor didapatkan. Jika data yang didapatkan dari sensor tersebut terdapat *Missing Value*, maka sistem tidak akan bekerja dengan baik, apalagi jika data tersebut adalah data “hari ini” yang didapatkan langsung dari sensor dan langsung dibangkitkan oleh sistem. Oleh karena itu, dalam sistem ini digunakan *Machine Learning* untuk mengatasi *Missing Value* dari data yang baru didapat dari sensor (data terakhir).

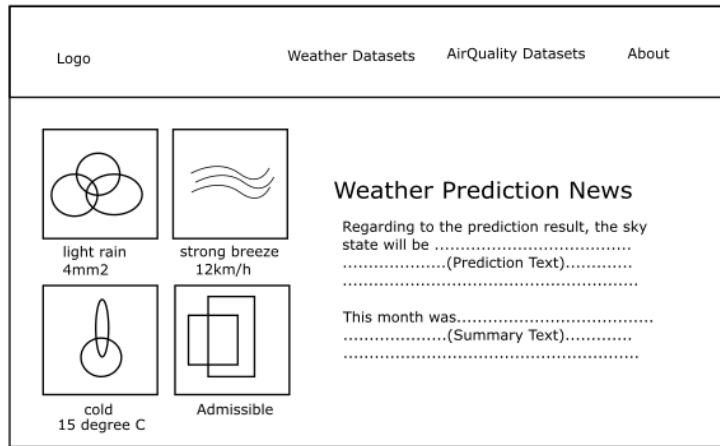
4.2.6 Model Antarmuka Perangkat Lunak

Dalam membangun antarmuka, penulis membuat sistem *Data-to-text* ini berbasis web. Agar kode program dari bahasa pemrograman R dapat terintegrasi dengan website memiliki tampilan yang lebih menarik. Untuk merealisasikan ini, penulis menggunakan sebuah *package* yang bernama *ShinyR*.



Gambar 4.25 Struktur file aplikasi D2T_Apps.

Dalam Gambar 4.25 diperlihatkan bahwa struktur file dalam aplikasi ini terdiri dari file ui.R yang berisi mengenai interface website, server.R yang berisi mengenai fungsi *reactive*, *input*, *output*, dan *session*. Sedangkan file D2T_machine.R berisi mengenai sistem *Data-to-text* dan akan menggunakan dataset dan bank kata pada folder /Datasets dan /wordbank.



Gambar 4.26 Purwa rupa antarmuka sistem *Data-to-text Weather Prediction* (DWP).

Pada Gambar 4.26 ditunjukan bagaimana antarmuka sistem akan dibangun. Antarmuka berbasis web ini akan menggunakan teknologi *bootstrap* sebagai framework css.

4.3 Implementasi Sistem *Data-to-text*

Dalam sub-bab ini, akan dipaparkan secara detail dari segi teknis mengenai pengembangan sistem *Data-to-text* dengan model yang telah dijelaskan pada sub-bab 4.2. penjelasan akan terdiri dari proses sinyal analisis, *Data Interpretation, Document Planning, Microplanning* dan ralization.

4.3.1 Installasi R Packages yang dibutuhkan

1. ShinyR

Shiny adalah *package* baru dari RStudio yang untuk membangun website interaktif berbasis R. Beberapa fitur dari ShinyR dapat dilihat pada poin-poin berikut:

- Membangun aplikasi web hanya tanpa JavaScript.
- Aplikasi hasil pengembangan ShinyR secara otomatis aka “*live*” secara dinamis mengikuti perubahan data.

- Antarmuka shiny dapat dibangun langsung dengan pemrograman R, atau dapat secara langsung dimodifikasi HTML, CSS, dan JS nya bila diibutuhkan.
- Dapat bekerja diseluruh *environtment* R (*Console* R, Rgui, ESS, StatET, Rstudio,dll).
- Memiliki default antarmuka berbasis *Bootstrap* dari *twitter*.
- *Widget* lainnya dapat menampilkan plot, Tabel, dan *output* dari objek dalam R.
- Menggunakan *reactive programming* model yang menghilangkan *event handling* yang kacau dari kode program.

Instalasi Shiny R dapat dilihat pada Gambar 4.27.

```
#1. Shiny R
install.packages(shiny) #-- server & GUI
library(shiny)
```

Gambar 4.27 Perintah untuk instalasi *packages Shiny R* yang berfungsi dalam membangun antarmuka aplikasi berbasis web

2. *Smooth*

Kumpulan fungsi *smoothing* yang digunakan untuk analisis *time-series* dan prediksi. *Package* ini meliputi model *exponential smoothing* dan SARIMA dalam bentuk beberapa fungsi simulasi. Instalasinya dapat dilihat pada Gambar 4.28.

```
#2. Smooth
install.packages(smooth) -> predicting with ex.smoothing
library(smooth)
```

Gambar 4.28 Perintah untuk instalasi *packages Smooth* yang berfungsi dalam membangun model prediksi berbasis *time-series* dengan algoritma *exponential smoothing*

3. *gradDesc*

Implementasi berbagai algoritma pembelajaran berbasis *Gradient Descent* untuk menangani tugas regresi. Algoritma *Gradient Descent* adalah: *Mini-Batch Gradient Descent* (MBGD), yang merupakan optimasi untuk menggunakan data pelatihan secara parsial untuk mengurangi beban komputasi. *Stochastic Gradient Descent* (SGD), yang merupakan optimasi untuk menggunakan data acak dalam pembelajaran untuk mengurangi beban komputasi secara drastis. *Stochastic Average Gradient* (SAG), yang merupakan algoritma berbasis SGD untuk meminimalkan langkah stokastik rata-rata. *Momentum Gradient Descent* (MGD), yang merupakan pengoptimalan untuk mempercepat pembelajaran *Gradient Descent*. *Accelerated Gradient Descent* (AGD), yang merupakan optimasi untuk mempercepat pembelajaran gradien. Adagrad, yang merupakan algoritma berbasis gradien-keturunan yang mengumpulkan biaya sebelumnya untuk melakukan pembelajaran adaptif. Adadelta, yang merupakan algoritma berbasis gradien-keturunan yang menggunakan pendekatan hessian untuk melakukan pembelajaran adaptif. RMSprop, yang merupakan algoritma berbasis gradien-keturunan yang menggabungkan kemampuan pembelajaran adaptif Adagrad dan Adadelta. Adam, yang merupakan algoritma berbasis gradien-keturunan yang berarti dan varians saat melakukan pembelajaran adaptif. Cara instalasi *packages gradDesc* dapat dilihat pada Gambar 4.29.

```
#3. Gradient Descent Packages  
install.packages(gradDescent)  
library(gradDescent)
```

Gambar 4.29 Perintah untuk instalasi *package gradDesc* untuk melakukan prediksi dengan pendekatan *machine Learning* menggunakan algoritma gradient descent

4. *xts*

Package ini menyediakan penanganan *uniform handling* bagi data *time-series* dengan cara instalasi sebagai berikut:

```
#4. Time-Series data object
install.packages(xts)
library(xts)
```

Gambar 4.30 Perintah untuk instalasi *package* *xts* untuk mendefinisikan dataset kedalam objek *time-series* berdasarkan tanggal.

4.3.2 Implementasi Proses *Signal Analysis*

1. Prediksi

Untuk implementasinya, penulis memanfaatkan dua *package* dalam R untuk melakukan prediksi, yaitu *package* *xts* dan *smooth*. *Package* *xts* digunakan untuk mendefinisikan dataset yang digunakan menjadi bentuk *time-series* yang diurutkan secara *ascending* berdasarkan format tanggal. *Package* *xts* ini dibuat oleh Jeffrey, Joshua, dan John dan terakhir dimutakhirkan di situs CRAN R pada maret 2017.

Sedangkan *package* *smooth* digunakan untuk melakukan prediksi. *Package* ini berisi kumpulan fungsi *smooth* yang digunakan untuk analisis *time-series* dan prediksi. Sampai saat ini, *package* ini berisi tentang *exponential smoothing* dan SARIMA dalam bentuk state-space dan beberapa fungsi simulasi. Berikut adalah kode program dalam R yang digunakan oleh penulis dalam memprediksi data cuaca selama satu hari kedepan.

```

i=1; n=length(dataCli)
x<-list()
y<-matrix()
for(i in i:n){
  #change dataset into time-series dataset (XTS)
  x[[i]] <- xts(dataCli[[i]],dataClimates$Date)
  #forecast with (ETS)
  y[i] <- forecast(x[[i]],h=1)$mean[1]
}

```

Gambar 4.31 Prediksi dengan *exponential smoothing* dalam R

Dari Gambar 4.31, ditunjukan bagaimana melakukan prediksi dengan menggunakan *package xts* dan *smooth* dalam bahasa R. Dimana dari data yang digunakan, dapat disimpulkan bahwa nilai keluaran dari proses analisis sinyal terdiri dari data-data yang dapat dilihat pada Tabel 4.13.

Tabel 4.13 data iklim hasil prediksi.

Variable	Hasil Prediksi
Rainfall (L/mm ²)	10.3
Cloud Coverage (%)	46.9
Wind Speed (km/h)	7.2
Wind Direction (degree)	276
Temperature (°C)	17.9

Tabel 4.13 diatas menjelaskan mengenai hasil prediksi yang telah didapatkan. Tabel tersebut memperlihatkan bahwa nilai curah hujan diperdiksi senilai 10.3 L/mm², nilai cakupan awan adalah 46.9%, kecepatan angin 7.2, arah angin 276°, dan suhu 17.9°C

Tabel 4.14 Hasil prediksi data kualitas udara.

Variable	Hasil Prediksi
CO	0.18
NO	31.9
NO2	60.9
NOX	90.1
O3	0.46
PM10	12.39
PM25	12.39

Sedangkan Tabel 4.6 menjelaskan mengenai hasil prediksi terkait dengan data kualitas udara.

1. Membangun *Weather Summary* dan *Event Detection*

Dalam membangun ringkasan cuaca yang kemudian akan dibangkitkan dalam bahasa alami, penulis menggunakan berbagai *tools* statistik dan membuat fungsi *event detection* yang terdiri dari mencari nilai *maximum*, mencari nilai *minimum*, mencari nilai rata-rata, mencari nilai total dan menemukan trend. Fungsi-fungsi ini sudah disediakan dalam bahasa pemrograman R. beberapa fungsi yang digunakan adalah *maximum*, *minimum*, *mean* dan *sum*. Berikut adalah serangkaian tahapan membangun *Weather Summary* dalam bahasa R.

- Memuat dataset cuaca

Dataset cuaca dimuat kedalam sistem seperti yang diperlihatkan pada Gambar 4.32 dan 4.33 dengan menggunakan fungsi *read.table*.

```
#----- Load and transform -----#
#Load dataset CLIMATE : Rainfall, Cloud Coverage, Temperature, Wind Speed, Wind Dir
dataClimates <- read.table(file="Datasets/2016-2017.csv", sep=",", header=TRUE)
```

Gambar 4.32 Memuat data kedalam sistem

- Memuat dataset kualitas udara

```
#Load dataset AirQuality
dataAQ <- read.table(file="Datasets/AQ_2016_2017.csv", sep=",", header=TRUE)
```

Gambar 4.33 Memuat data kualitas udara kedalam sistem

Dalam penelitian ini, dataset disimpan dalam bentuk file *Comma Separated Value* (CSV) yang kemudian dimuat kedala sistem D2T dengan perintah *read.table*.

- Mengambil data hanya satu bulan terakhir, dan satu bulan sebelumnya

```
LM <- dataClimates[(NROW(dataClimates)-30):NROW(dataClimates), ]
# MBLM <- subset(MBLM, (Date < (Sys.Date()-30)))
MBLM <- dataClimates[(NROW(dataClimates)-60):(NROW(dataClimates)-30), ]
```

Gambar 4.34 Mengambil data satu bulan terakhir

Mengambil data hanya satu bulan terakhir dan satu bulan sebelumnya dengan fungsi *subset* seperti yang diperlihatkan pada Gambar 4.34.

- Mencari nilai *maximum* dari setiap variable beserta atribut nya dengan fungsi *max* seperti pada Gambar 4.35.

```
# ----- SUMMARIZING Last Month
#Defining Max Amount-----
for(i in 1:n){
  max_amt[i] <- max(LM[i])
  max_index2 <- as.integer(which(LM[i]==max(LM[i])))
  max_index[i] <- max_index2[1]
  max_index0 <- max_index2[1]
  max_date[i] <- as.character(LM$Date[max_index0])
  col_names[i] <- colnames(LM[i])
}
Lmax_result <- data.frame(col_names, max_date, max_amt)
```

Gambar 4.35 Mencari nilai *maximum* setiap variabel.

- Mencari nilai *minimum* dari setiap variable beserta atribut nya dengan fungsi *min* seperti pada Gambar 4.36.

```
#Defining Min Amount-----
for(i in i:n){
  min_amt[i] <- min(LM[i])
  min_index2 <- as.integer(which(LM[i]==min(LM[i])))
  min_index[i] <- min_index2[1]
  min_index0 <- min_index2[1]
  min_date[i] <- as.character(LM$Date[min_index0])
  col_names[i] <- colnames(LM[i])
}
LMmin_result <- data.frame(col_names, min_date, min_amt)
```

Gambar 4.36 Mencari nilai *minimum* setiap variabel

- Mencari nilai rata-rata dari setiap variable dengan fungsi *colMeans* seperti pada Gambar 4.37.

```
#Defining Mean Amount-----
xMBLM <- MBLM[2:6]
MBLMmean_result <- colMeans(xMBLM)
```

Gambar 4.37 Mencari nilai rata-rata

- Mencari nilai total dari setiap variable dengan menggunakan fungsi *sum* seperti pada Gambar 4.38.

```
#Defining Sum Amount-----
i=2
n=length(LM)
col_names <- c("");
sum_amt <- c("");
for(i in i:n){
  sum_amt[i] <- sum(LM[,i])
  col_names[i] <- colnames(LM[i])
}
LMsum_result <- data.frame(col_names, sum_amt)
```

Gambar 4.38 Mencari nilai total setiap variabel

LAST MONTH			LAST 2 MONTH		
col_names	max_date	max_amt	col_names	max_date	max_amt
Cloud.Coverage	2017-06-26	93.6	Cloud.Coverage	2017-06-11	100
Average.Temperature	2017-06-20	25.3	Average.Temperature	2017-05-24	23
Wind.Speed	2017-06-28	13.39	Wind.Speed	2017-06-05	14.51
Wind.Direction	2017-06-17	315	Wind.Direction	2017-05-18	315
Rainfall	2017-06-28	17.1	Rainfall	2017-05-26	16.9
col_names	min_date	min_amt	col_names	min_date	min_amt
Cloud.Coverage	2017-06-17	9.5	Cloud.Coverage	2017-06-07	8.4
Average.Temperature	2017-06-29	14.5	Average.Temperature	2017-05-18	12.4
Wind.Speed	2017-06-19	4.18	Wind.Speed	2017-05-21	4.28
Wind.Direction	2017-06-19	45	Wind.Direction	2017-05-23	135
Rainfall	2017-06-17	0	Rainfall	2017-05-20	0
col_names	mean_amt	col_names	mean_amt		
Cloud.Coverage	43.24	Cloud.Coverage	47.0034482758621		
Average.Temperature	19.605	Average.Temperature	17.0206896551724		
Wind.Speed	7.069	Wind.Speed	6.34379310344828		
Wind.Direction	276.75	Wind.Direction	294.827586206897		
Rainfall	2.38				
col_names	sum_amt	col_names	sum_amt		
Cloud.Coverage	864.8	Cloud.Coverage	1363.1		
Average.Temperature	392.1	Average.Temperature	493.6		
Wind.Speed	141.38	Wind.Speed	183.97		
Wind.Direction	5535	Wind.Direction	8550		
Rainfall	47.6	Rainfall	44		

Gambar 4.39 Output standar statistik

Nilai yang didapatkan dari proses ini dapat dilihat pada Gambar 4.39. Setelah nilai-nilai standar didapatkan, maka langkah selanjutnya adalah membuat fungsi-fungsi yang dapat mengidentifikasi masing-masing pesan yang akan disampaikan. Pesan-pesan ini didapatkan dari hasil analisis corpus dalam (Reiter dan Dale, 2000). Pesan-pesan tersebut terdiri dari: rata-rata suhu,rata-rata Hujan, Jumlah hari hujan, Jumlah total nilai hujan, Hujan berturut-turut, Nilai hujan ekstrem, rata-rata angin, nilai ekstrem angin, rata-rata kualitas udara, tren suhu, dan nilai suhu terendah.

Tabel 4.15 Seluruh fungsi *Statististical Event Detection* dan keterangannya.

Fungsi	Deskripsi	Input	Output
MonthlyTempMeasurement_function	Membandingkan rata-rata temperatur bulan ini dengan rata-rata temperatur selama satu tahun	Rata-rata suhu per bulan, Rata-rata suhu pertahun	Hasil perbandingan, Index
MonthlyRainfallMessage_function	Membandingkan rata-rata curah hujan bulan ini dengan rata-rata curah hujan selama satu tahun	Rata-rata curah hujan per bulan, Rata-rata curah	Hasil perbandingan, Index

		hujan pertahun	
RainyDaysMessage_function	Mengukur jumlah hari hujan selama satu bulan	Data curah hujan selama satu bulan	Deskripsi jumlah curah hujan
RainSoFarMessage_function	Membandingkan rata-rata curah hujan minggu ini dengan rata-rata curah hujan selama satu bulan	Rata-rata curah hujan satu bulan, rata-rata curah hujan satu hari	Hasil perbandingan, <i>Index</i>
RainSpellMessage	Mencari urutan hujan terpanjang (hujan berturut turut)	Curah hujan selama satu bulan	Jumlah urutan terbesar, hari pertama hujan, hari terakhir hujan
RainExtremeMessage_function	Mencari nilai ekstrim dari hujan (hujan dengan klasifikasi terbesar, diatas atau sama dengan <i>heavy rain</i>)	Curah hujan selama satu bulan	Deskripsi, tanggal nilai ekstrim
MonthlyWindMessage	Mencari rata-rata nilai kecepatan angin perbulan	Kecepatan angin selama satu bulan	Rata-rata kecepatan angin satu bulan
WIndextremeMessage_function	Mencari nilai ekstrim dari Angin (hujan dengan klasifikasi terbesar, diatas atau sama dengan <i>Strong Gale</i>)	Kecepatan angin selama satu bulan	Deskripsi, tanggal nilai ekstrim
MonthlyAQMessage_function	Mencari rata-rata nilai kualitas udara perbulan	Kualitas udara selama satu bulan	Rata-rata kualitas udara
Coldest_day	Mencari nilai terdingin selama satu bulan	Temperature selama satu bulan	Temperatur terdingin, tanggal

Tabel 4.15 menjelaskan setiap fungsi yang digunakan sesuai dengan deksripsi, *input*, dan *outputnya*. Seluruh fungsi yang berperan dalam *Event Detection* berjumlah 10, penerapan fungsi dari beberapa *event* dalam R dapat dilihat pada penjelasan berikut.

Untuk mendeteksi data dengan informasi hujan berturut-turut, penulis membuat fungsi yang bernama *RainSpellMsg_function*, dengan *input* adalah data cuaca dan kualitas udara selama satu bulan. Untuk menggambarkan fungsi ini, penulis menggunakan *Coding* dalam bahasa R yang dapat dilihat pada Gambar 4.40.

Nama Fungsi	: RainSpellMessage
Input	: LM (Data 1 Bulan terakhir)
Output	: SpellLength (Panjang urutan hujan) In_day (Hari pertama dalam urutan hujan) Out_day (Hari terakhir dalam urutan hujan) In_mnth (Bulan pada hari pertama) Out_mnth (Bulan pada hari terakhir)
<pre> RainSpellMessage<-function(LM){ x<-as.matrix(LM[,"Rainfall"]) n<-length(x);j<-1;j<-0;k<-1;l<-0;status=0 seq<-matrix("0",nrow=n,ncol=3) for(i in 1:n){ if(x[i]==0){ status=0 if(i!=1){ if(x[i-1]>0){ out_date <- as.character(LM[i-1,"Date"]) seq[j,3] <- out_date } } } else if((x[i]>0)&&(status==0)){ status=1 j<-j+1 seq[j,1] <- as.integer(seq[j,1]) + 1 in_date <- as.character(LM[i,"Date"]) seq[j,2] <- in_date } else if((x[i]>0)&&(status==1)){ seq[j,1] <- as.integer(seq[j,1]) + 1 } if(i==n){ if((x[i]!=0)&&(status==1)){ out_date <- as.character(LM[i,"Date"]) seq[j,3] <- out_date } } print(j) } max_index<-which.max(seq[,1]) in_spell<-substrRight(seq[max_index,2],2) out_spell<-substrRight(seq[max_index,3],2) in_mnth<-substr(seq[max_index,2],6,7) out_mnth<-substr(seq[max_index,2],6,7) in_ordinal <- ordinal_indicator(in_spell) out_ordinal <- ordinal_indicator(out_spell) output <- (c(max_index,in_spell,out_spell,in_mnth,out_mnth)) return(output) } </pre>	

Gambar 4.40 Pendekripsi hari hujan berturut-turut

Gambar 4.40 menunjukkan fungsi yang digunakan untuk mendekripsi hujan berturut turut, maksudnya adalah jika didalam satu bulan terdapat hujan yang

berturut-turut (terjadi dalam setiap harinya), berita dengan jumlah urutan terbanyak akan ditampilkan kedalam berita.

```

Input : Rata - rata per tahun, Rata - rata bulan ini
Output: [status perbedaan, Index]

MonthlyTempMessage_function <- function(YearMean_average, MonthMean_average){
  yearmean <- as.double(YearMean_average["Average.Temperature"])
  monthmean <- as.double(MonthMean_average["Average.Temperature"])
  if(yearmean<monthmean){
    message<- c("cooler",0)
  }
  else if(yearmean==monthmean){
    message<- c("stable",1)
  }
  else if(yearmean>monthmean){
    message<- c("warmer",2)
  }
  return(message)
}

```

Gambar 4.41 MonthlyTempMessage function

Gambar 4.41 merupakan fungsi untuk mendeteksi kondisi temperatur bulan ini dan perbandingannya dengan rata-rata tahun ini. Untuk fungsi-fungsi lainnya, dapat dilihat pada lampiran. Hasil dari serangkaian *Event Detection* ini adalah sebagai berikut.

```

MonthlyAQMessage_function <- function(AQData){
  MonthAQ <- AQData[(NROW(AQData)-30):NROW(AQData),]
  monthmean <- colMeans(MonthAQ)
  msg<-AirQuality_interpreter(monthmean)
  return(msg)
}

```

Gambar 4.42 Fungsi mendeteksi keadaan kualitas udara bulan terakhir

Pada Gambar 4.42, fungsi *MonyhlyAQMessage_function* berfungsi sebagai pendeteksi kondisi kualitas udara selama satu bulan terakhir. Fungsi ini menghitung rata – rata kualitas udara selama satu bulan.

Serangkaian fungsi diatas merupakan beberapa contoh implementasi, untuk lebih lengkapnya dapat dilihat kode program dibagian lampiran. Lalu, hasil dari seluruh fungsi yang didapatkan berada pada Tabel 4.16. pada Tabel tersebut diperlihatkan *output* dari masing-masing fungsi yang dibuat.

Tabel 4.16 Hasil proses *Statistical Event Detection*.

Function	Message
MonthlyTempMessage_function	[Cooler,0]
MonthlyRainfallMessage_function	[Wetter,0]
RainyDaysMessage_function	[average]
RainSoFarMessage_function	[above average]
RainSpellMessage	[4,27,30,07,07]
RainExtremeMessage_function	[heavy, 28-29]
MonthlyWindMessage	[11]
WIndextremeMessage_function	-
MonthlyAQMessage_function	0.18,13,25,45,32,11,11,3
Coldest_day	[14, 19 th]

4.3.3 Implementasi Proses *Data Interpretation*

Dalam impelementasinya, setiap nilai dari hasil proses sinyal analisis yang berkaitan dengan kecepatan angin, dimasukan kedalam fungsi interpretasi data yang dibuat oleh penulis dengan *input* berupa interval (klasifikasinya), nilai, nama variabel, dan nama dari setiap kelas. Source code dalam bahasa R yang telah dibuat oleh penulis untuk melakukan interpretasi data dapat dilihat pada Gambar 4.43.

```

Pseudocode Fungsi Data Interpreter
Input : - Partisi Variable (contoh : interval light wind, a>0 dan b<2)
        - Nilai ( contoh : 20 )
        - Nama State ( contoh :
Fungsi Data Interpreter ( masukan Partisi Variabel, Nilai, Nama
State, Partisi )
        Hitung jumlah Partisi Variabel petakan ke n
        FOR dari i sampai n, i = 1
            Nilai variabel "a" dari setiap Partisi Variabel
dipetakan ke a
            Nilai variabel "n" dari setiap Partisi Variabel
dipetakan ke b
            IF Nilai >= a AND Nilai <= b
                THEN Hasil adalah Nama Partisi ke-i
Return nilai Hasil

```

Gambar 4.43 Pseudocode proses interpretasi data

Dalam Gambar 4.43 diperlihatkan *pseudocode* dari *Rule-Based Data Interpretation*, dimana data diterjemahkan oleh klasifikasi yang telah ditetapkan para pakar yang dijelaskan pada sub-bab pengembangan model.

```

#Function : Data Interpreter -> translate numerical event
#Input -> -Interval : CRISP Set Membership Function (eg. )
#           -val : Event Value (eg. 24)
#           -name : Data Name (eg. Rain
#           -pname : partition name
Data_Interpreter <- function(interval,val,name,pname){
  n=length(interval) ; i=1;
  for(i in i:n){
    ;
    a<-interval[[i]][["a"]]
    b<-interval[[i]][["b"]]
    if((val>=a)&&(val<=b)){
      result <- pname[i]
    }
  }
  return(result)
}

```

Gambar 4.44 Data Interpretation dalam R

Bagi masukan fungsi *Data_interpreter* (Gambar 4.44) yang pertama adalah *interval*, seperti yang diperlihatkan pada Gambar 4.45, yaitu batas atas dan batas bawah setiap kelas dalam klasifikasinya masing-masing. Sedangkan masukan kedua adalah *val* yaitu nilai yang akan diinterpretasikan, masukan ketiga

adalah *name* untuk kata tambahan sesuai domain data, contoh: *wind*, *sky*, dll. *Input* terakhir adalah *pname* yang berisi nama dari setiap kelas / partisi.

```
WindSpeed_partition <- c("calm", "light air", "light breeze",
                         "gentle breeze", "moderate breeze", "fresh breeze",
                         "strong breeze", "near gale", "gale", "strong gale",
                         "storm", "violent storm", "Hurricane"
                         )
WindSpeed_interval <- list(calm=c(a=0,b=2), light_air=c(a=2,b=5),
                           light_breeze=c(a=6,b=11), gentle_breeze=c(a=12,b=19),
                           moderate_breeze=c(a=20,b=29), fresh_breeze=c(a=30,b=39),
                           strong_breeze=c(a=40,b=50), near_gale=c(a=51,b=61),
                           gale=c(a=62,b=74), strong_gale=c(a=74,b=87),
                           storm=c(a=88,b=102), violent_storm=c(a=103,b=118),
                           hurricane=c(a=119,b=130))
InterpretationResult_windSpeed <- Data_Interpreter(
  WindSpeed_interval,
  as.double(climatePredictionResult[["Wind.Speed"]]),
  "wind",
  WindDirection_partition)
```

Gambar 4.45 Input kecepatan angin

```
# #-----INPUT Wind Direction
WindDirection_partition <- c("from north", "from north north east", "from north east",
                           "from east north east", "from east", "from east south east",
                           "from south east", "from south south east", "from south",
                           "from south south west", "south west", "from west south west",
                           "from west", "from west north west", "from north west",
                           "from north north west", "from north")
WindDirection_interval <- list(
  N=c(a=0,b=11.25), NNE=c(a=11.25,b=33.75), NE=c(a=33.75,b=56.25),
  ENE=c(a=56.25,b=78.25), E=c(a=78.75,b=101.5), ESE=c(a=101.5,b=123.75),
  SE=c(a=123.75,b=146.25), SSE=c(a=146.25,b=168.75), S=c(a=168.75,b=191.25),
  SSW=c(a=191.25,b=213.75), SW=c(a=213.75,b=236.25), WSW=c(a=236.25,b=258.75),
  W=c(a=258.75,b=281.25), NW=c(a=281.25,b=303.75), NW=c(a=303.75,b=326.25),
  NWW=c(a=326.25,b=348.75), FN=c(a=348.75,b=360)
)
InterpretationResult_windDirection <- Data_Interpreter(WindDirection_interval,
  as.double(climatePredictionResult[["Wind.Direction"]]),
  "", WindDirection_partition)
```

Gambar 4.46 Input arah angin

Gambar 4.46 merupakan contoh lain dari *input* untuk fungsi *Data_Interpreter()*, dimana pada Gambar tersebut diisikan partisi dan interval untuk mengklasifikasikan arah angin.

Untuk menginterpretasikan kualitas udara, penulis menggunakan formula *Pollutant Standard Index* (PSI) seperti apa yang telah dijalaskan pada sub-bab pengembangan model. Untuk implementasinya dalam R, dapat dilihat pada Gambar 4.47.

```

#menginterpretasikan kualitas udara dengan Polutant Standard Inndex (PSI)
AirQuality_interpreter <- function (dataset){
  #menghitung sub-index value dari variabel PM25
  #b2 = pollutant concentration breakpoint 2, a2 = PSI Index breakpoint 2
  #b1 = pollutant concentration breakpoint 1, a1 = PSI Index breakpoint 1
  a2<-0; a1<-0; b2<-0; b1<-0;
  PM10_PSI_value<-0; CO_PSI_value<-0;
  NO2_PSI_value<-0; SO2_PSI_value <-0;
  O3_PSI_value <- 0;
  #define each sub-index interval of PM25
  PM25<-as.double(dataset["PM25"]);
  if((PM25>0)&&(PM25<=12)){
    b2<-12; b1<-0; a2<-50; a1<-0;
  }
  else if((PM25>12)&&(PM25<=55)){
    b2<-55; b1<-12; a2<-100; a1<-51
  }
  else if((PM25>55)&&(PM25<=150)){
    b2<-150; b1<-56; a2<-200; a1<-101
  }
  else if((PM25>150)&&(PM25<=250)){
    b2<-250; b1<-151; a2<-300; a1<-201
  }
  else if((PM25>250)&&(PM25<=350)){
    b2<-350; b1<-251; a2<-400; a1<-301
  }
  else if((PM25>350)&&(PM25<=500)){
    b2<-500; b1<-351; a2<-500; a1<-401
  }
  PM25_PSI_value <- ((a2-a1)/(b2-b1))*(PM25-b1)+a1
  PSI_data <- c(PM25_PSI_value,PM10_PSI_value,SO2_PSI_value,CO_PSI_value,O3_PSI_value)
  PSI_value <- as.integer(max(PSI_data))

  print(dataset)
  print(PM10_PSI_value)
  print(PM25_PSI_value)
  print(CO_PSI_value)
  print(O3_PSI_value)
  print(SO2_PSI_value)
  print(SO2)
  print("PSI PSI")
  print(PSI_value)
  if((PSI_value>0)&&(PSI_value<=50)){
    return("good")
  }
  else if((PSI_value>50)&&(PSI_value<=100)){
    return("admissible")
  }
  else if((PSI_value>100)&&(PSI_value<=250)){
    return("bad")
  }
  else{
    return("hazardous")
  }
}

```

Gambar 4.47 Fungsi *AirQuality_Interpreter* untuk menginterpretasikan kualitas udara

Khusus untuk menginterpretasikan data curah hujan dan suhu, penulis mengutip dari (Ramos-soto, dkk., 2016), dimana partisi curah hujan dan suhu yang tersedia adalah dalam bentuk *Fuzzy*. Oleh karena itu, untuk menginterpretasikan keduanya, penulis membuat fungsi yang dapat dilihat pada Gambar 4.48. Pada Gambar tersebut, dibuat dua fungsi yang terdiri dari *membership_partition()* dan *membership_check()*, dimana fungsi dari *membership_partition()* adalah memetakan memetakan setiap interval *input* kedalam bentuk partisi *Fuzzy*. Sedangkan *membership_check()* merupakan fungsi yang menghitung nilai derajat keanggotaan nilai *x* berdasarkan *membership_partition*.

```

----- Data Interpreter with Fuzzy Sets -----#
#defining Membership Partition ( Fuzzification Process ) - Trapezoidal
membership_partition <- function(variables,name){
  matrix_x_graph <- list()
  y<-as.matrix(c(1,6,6,1))
  n=length(variables)
  # print(n)
  maxX<-variables[[n]][["d"]]
  i=1;
  for(i in 1:n){
    title <- paste(name," Membership Function")
    if(i==1){
      # plot(variables[[i]],y,type="l",lwd=1,main=title,xlim=c(0,maxX),yaxt="n",col="red")
    }else{
      # lines(variables[[i]],y,lwd=1,col="red")
    }
  }
  return(variables)
}

#check membership-----
membership_check <- function(partition,v,pname,oname){
  i=1; n=length(partition)
  membership_value <- c(1)
  for(i in 1:n){
    a<-partition[[i]][["a"]]
    b<-partition[[i]][["b"]]
    c<-partition[[i]][["c"]]
    d<-partition[[i]][["d"]]

    if((v>a)&(v<d)){
      membership_value[i] <- 0
      #print("uhuh")
    }
    if((v>a)&(v<=b)){
      membership_value[i] <- ( (v-a) / (b-a) )
      #print("xyz")
    }
    if((v>b)&&(v<=c)){
      membership_value[i] <- 1
      #print("abc")
    }

    if((v>c)&&(v<=d)){
      membership_value[i] <- ( (d-v) / (d-c) )
      #print("abc")
    }
    # print(membership_value[i])
  }
  #check highest membership result
  i=1; biggest<-0; part<-"a"
  for(i in 1:n){
    if(is.nan(membership_value[i])){
      membership_value[i] <- 0
    }
    if(biggest<-membership_value[i]){
      biggest <- membership_value[i]
      part<-pname[i]
    }
  }
  part <- paste(part,oname)
  mval <- as.matrix(membership_value)
  return (part)
}

```

Gambar 4.48 Fungsi untuk menginterpretasikan curah hujan dan suhu.

Setelah melalui serangkaian proses interpretasi data, didapatkan keluaran seperti yang dapat dilihat pada Gambar 4.49. Pada Gambar tersebut diperhatikan bahwa hasil yang dihasilkan dalam bentuk teks dalam bahasa alami yang mewakili nilai data.

Output pada Gambar 4.49 terdiri dari data kemarin, data hari ini, dan data hasil prediksi yang diinterpretasikan. Dari hasil prediksi diperoleh pesan *light rain, partly cloudy, warm, from the west. Light breeze, dan bad.*

----- Data Interpretation Output -----		
----- Yesterday -----		
Yesterday Rainfall	:	no rain
Yesterday Cloud Coverage	:	overcast
Yesterday Temperature (AVG)	:	warm
Yesterday Wind Direction	:	from the south
Yesterday Wind Speed	:	light breeze
Yesterday Air Quality	:	admissible
----- Predicted -----		
Predicted Rainfall	:	light rain
Predicted Cloud Coverage	:	partly cloudy
Predicted Temperature (AVG)	:	warm
Predicted Wind Direction	:	from the west
Predicted Wind Speed	:	light breeze
Predicted Air Quality	:	bad

Gambar 4.49 Hasil proses *Data Interpretation*

4.3.4 Implementasi Proses *Document Planning*

Pada sub-bab pengembangan model *Document Planning* telah dijelaskan bahwa proses ini terdiri dari *Content Determination* dan *Document Structuring*. Salah satu contoh yang diterapkan dalam penelitian ini adalah *Content Selection* seperti yang bisa dilihat pada Gambar 4.50.

```
Wind_relation <- function(status, direction){
x<-("")  
  x<- paste("the wind will blow",status,direction)
  if((status=="calm")||(status=="light air")||(status=="light breeze")|||  
    (status=="gentle breeze")||(status=="moderate breeze")||(status=="fresh breeze")){
      return("")
    }
  else{
    return(x)
  }
}
```

Gambar 4.50 *Content Selection* saat menggabungkan pesan yang berelasi (*Wind State*)

Dari Gambar 4.50 diperlihatkan bagaimana pesan *Wind State* hanya akan disampaikan apabila hasil interpretasi lebih tinggi dari *gentle breeze*. Untuk *content selection* pada pesan lainnya, digunakan cara yang sama seperti pada Gambar 4.50.

Sedangkan proses document structuring adalah bagaimana menganalisis urutan teks dapat mengalir seperti yang dijelaskan pada sub-bab pengembangan model. Untuk realisasinya, dapat dilihat pada sub-bab *Structure Realisation*.

4.3.5 Implementasi Proses *Microplanning* dan *Realization*

Salah satu contoh aspek leksikalisa yang diterapkan dalam penelitian ini adalah menyampaikan persepsi perubahan data kualitas udara secara leksikal. Tahapannya telah dijelaskan secara detail pada sub-bab pengembangan model. Untuk contoh implementasinya dalam R, diperlihatkan pada Gambar 4.51. terdiri dari fungsi *LD_Compare()*, *TrendDesc_template*, dan *change_word_bank_AQ()*.

<pre># Source = Ramos LD_Compare <- function (data, value){ #map data-set i=1; n=length(data); index_data<-matrix(); for(i in 1:n){ j=1; m=length(AQ_val) for(j in 1:m){ x<-colnames(value[j]) #cat("x : ",x," data: ",data[i],"\n") if(data[i]==x){ index_data[i]<-value[j] } } } #Compute Index Variation i=1; n=length(data); IV<-c(0,0); for(i in 1:(n-1)){ if(i<1){ IV[i]<-((index_data[[i+1]])-(index_data[[i]])) } else{ IV[i]<-index_data[[i]] } } #Apply Rules i=1; IVL<-c(0,0) for(i in 1:n){ if(IVL[i]>0){ IVL[i]+"<" } else if(IVL[i]<0){ IVL[i]+">" } else{ IVL[i]+"<0" } } x<-TrendDesc_template(IVL) return(x) }</pre>	<pre>TrendDesc_template <- function (IVL,data){ if((IVL[1]=="")&&(IVL[2]=="")){ TrendDesc <- change_word_bank_AQ("stable") } if(((IVL[1]=="+")&&(IVL[2]=="-")) ((IVL[1]=="-")&&(IVL[2]=="+"))){ TrendDesc <- change_word_bank_AQ("mediumChange") } if(((IVL[1]=="+")&&(IVL[2]=="0")) ((IVL[1]=="-")&&(IVL[2]=="0"))){ TrendDesc <- change_word_bank_AQ("startChange") } if(((IVL[1]=="0")&&(IVL[2]=="+")) ((IVL[1]=="-")&&(IVL[2]=="-"))){ TrendDesc <- change_word_bank_AQ("endChange") } if(((IVL[1]=="+")&&(IVL[2]=="+")) ((IVL[1]=="-")&&(IVL[2]=="-"))){ TrendDesc <- change_word_bank_AQ("progressiveChange") } return(TrendDesc) } change_word_bank_AQ <- function (fragmentCode){ if(fragmentCode=="stable"){ x<-as.integer(runif(1,1)) if(x==1){ return("stay stable at") } if(x==2){ return("keep stable at") } if(x==3){ return("still stable at") } if(x==4){ return("steady at") } if(x==5){ return("constant at") } } }</pre>
--	--

Gambar 4.51 Source code *Trend Description*

Untuk implementasi dari proses agregasi, penulis menerapkan apa yang telah dibuat pada proses pengembangan model yaitu dengan teknik *Simple Conjunction Referring to Contrast Value. Pseudocode* dari proses ini dapat dilihat pada Gambar 4.52 yang menjelaskan secara detail bagaimana melakukan agregasi dengan teknik ini.

```

Function : Sky_State Aggregation with simple conjunction
Function name : Sky_Agg
Input : Sky message, Cloud Message
Output : Aggregated phrase as Sky State

Proporsionate ← Word_bank of Conjunction 1
    {"Covered with","Covered By","Followed By"..}
Contrast      ← Word_bank of Conjunction 2
    {"Although its covered by","Even though its
covered by"..}
Assign_Contrast Value (for each partition of Rain_state)
    {no_rain=0, Light_rain=0, moderate_rain=1, intense_rain=1,
torrential_rain=1}
Assign_Contrast Value (for each partition of Cloud_state)
    {clear =0, foggy =0, mostly_sunny=0, partly_cloudy=1,
mostly_cloudy=1,
broken=1, overcast =1}
IF Contrast_Value of Rain_State Is equal with Contrast_Value of
Cloud_State
THEN Conjunction is random(Proporsionate)
ELSE Conjunction is random(Contrast)

Sky_State Message ← Sky Message + Conjunction + Cloud Message
Return (Sky_state message)

```

Gambar 4.52 Pseudocode Aggregation

Untuk agregasi beberapa pesan lainnya, digunakan cara yang sama seperti pada Gambar 4.52, untuk melihat beberapa fungsi lainnya dapat dipantau pada halaman Lampiran.

Selanjutnya, pada tahapan *Referring Expression Generation* digunakan teknik *Random Canned-Text*, dapat dilihat pada Gambar 4.53. Untuk mendefinisikan subjek, penulis membuat sebuah bank-frase yang tersedia pada direktori “D2T_Apps/wordbank”. Untuk melakukan *Referring Expression Generation* pada pesan lain, digunakan cara yang sama seperti pada Gambar 4.53. untuk melihat beberapa fungsi lainnya dapat dipantau pada halaman Lampiran.

```

MonthlyIntro_lex<-function(){
  MI <- as.matrix(read.table(file="wordbank/Monthly_referring_expression.csv", sep=",", header=TRUE))
  n=length(MI); i=1
  random_value <- as.integer(runif(1,1,n+1))
  return(MI[random_value])
}
MonthlyIntro <- MonthlyIntro_lex()

```

Gambar 4.53 Referring Expression generation

Sedangkan proses *Structure Realization* seperti yang sudah dijelaskan pada proses pengembangan model adalah merealisasikan hasil analisis pada tahap *Document Structuring*, yaitu memetakan pesan pada setiap struktur yang telah dibuat seperti pada Gambar 4.54.

```
Structure_Realization_predict <- function(InterpretationResult_rainfall, InterpretationResult_cloudCoverage){
  Rain_State <- InterpretationResult_rainfall
  Cloud_State <- InterpretationResult_cloudCoverage
  Sky_State <- Sky_Agg(Rain_State,Cloud_State)
  Sky_Intro <- Prediction_Intro()
  Sky_Sentence <- paste(Sky_Intro,Sky_State)
  Temperature_Intro <- Temperature_Intro()
  Temperature_Sentence <- paste(Temperature_Intro, Temperature_State)
  AQ_Sentence <- AQ_Description
  Wind_Sentence <- Wind_Description
  Prediction_Result <- paste(Sky_Sentence,Temperature_Sentence, AQ_Sentence,Wind_Sentence)
  return(Prediction_Result)
}
Prediction_Result<-Structure_Realization_predict(InterpretationResult_rainfall,InterpretationResult_cloudCoverage)

MonthlyMsg<- paste(MonthlyIntro,MonthlyMsg1,MonthlyMsg2,MonthlyMsg3,MonthlyMsg4,MonthlyMsg5,MonthlyMsg6)
```

Gambar 4.54 Structure Realization

4.3.6 Implementasi *Machine Learning* untuk mengatasi missing value

Pada proses ini, digunakan *Machine Learning* untuk mengatasi *missing value* pada data baru. Sebagai asumsinya, dalam sistem ini missing value yang diatasi adalah pada data *Today*. Untuk implementasinya dapat dilihat pada Gambar

```
if(Rainfall == -9999 || Rainfall == -8888){
  #Rainfall Prediction based on cloud, temperature, and wind
  datasetRainfall <- data.frame(c(dataCli["Cloud.Coverage"],
                                    dataCli["Average.Temperature"],
                                    dataCli["Wind.Speed"],
                                    dataCli["Wind.Direction"],
                                    dataCli["Rainfall"])) #output
}

## Predict Using GD Model
## get z-factor Data
dataSet <- datasetRainfall
## do variance scaling to dataset
featureScalingResult <- minmaxScaling(dataSet)
## split dataset
splitedDataSet <- splitData(featureScalingResult$scaledDataSet)
## built model using GD
model <- GD(splitedDataSet$dataTrain)
## separate testing data with input only
dataTestInput <- data.frame(y1,y2,y3)
## predict testing data using GD model
prediction <- prediction(model,dataTestInput)
#----- Prediction Finished -----
}
```

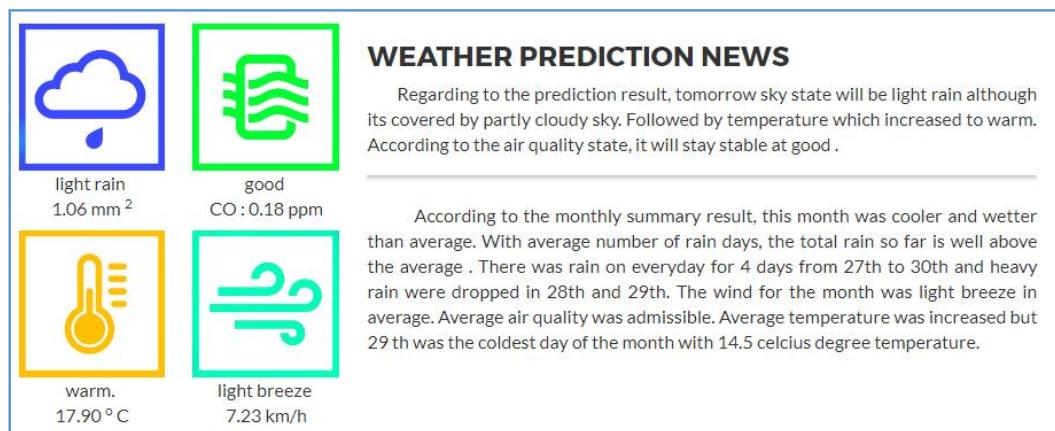
Gambar 4.55 Machine Learning dalam R untuk mengatasi missing value.

4.4 Hasil Implementasi

Setelah dilakukan serangkaian proses diatas, maka didapatkan hasil dari sistem *Data-to-text*. Hasil ini telah terintegrasi dengan tampilan website dengan memanfaatkan *package Shiny R. User Interface* yang dibangun menggunakan *framework css bootstrap*. Konten paragraf yang dihasilkan terdiri dari hasil prediksi dan hasil deksripsi cuaca dan kualitas udara. Pada sub-bab dibawah ini akan dipaparkan hasil keluaran sistem, perbandingan *output* dengan penelitian lain, dan validasi keluaran. Dalam sub-bab 4.4.1 akan dijelaskan mengenai *output* murni dari sistem, dan pada sub-bab 4.4.2 akan dijelaskan mengenai perbandingan *output* pada peneitian ini dengan penelitian yang lain.

4.4.1 *Output* sistem

Pengembangan aplikasi pada penelitian ini menghasilkan sistem yang mampu membangkitkan berita cuaca dalam bentuk teks, hasil yang murni didapatkan dari serangkaian proses pada sistem ini dapat dilihat pada Gambar 4.56.



Gambar 4.56 Hasil keluaran sistem.

Dari Gambar 4.56, dapat dilihat bahwa didapatkan *output* dari sistem terdiri dari informasi prediksi dan hasil deskripsi. Paragraf pertama pada *output* tersebut menjelaskan data hasil prediksi yang dapat dilihat pada keempat nilai disamping kiri pada Gambar 4.56. Dari gambar tersebut memperlihatkan bahwa

hujan diprediksi akan terjadi dengan nilai curah hujan 1.06mm^2 . Dengan melalui proses *Data Interpretation*, maka nilai 1.06mm^2 ini diterjemahkan kedalam frase “*Light Rain*” atau hujan ringan. Selain itu, didapatkan status awan dengan frase “*Partly Cloudy Sky*”. Menurut perencanaan dokumen yang dibuat, untuk mendapatkan kalimat *Sky State*, pesan hujan dan pesan awan digabungkan dalam tahap *Aggregation* dan *Lexicalisation*. Setelah itu, dalam tahap *Referring Expression Generation* dipilih template didalam *word bank* dengan teknik *Random Generation*. Pada kalimat “*light rain although its partly cloudy sky*”, kata penghubung atau *Conjunction* dengan menggunakan *although* dipilih berdasarkan teknik “*Aggregation reference to contrast value*”, dimana *Contrast Value* merupakan nilai yang akan dibandingkan dalam menentukan frase yang akan dibangkitkan untuk menggabungkan kedua pesan tersebut, lebih lengkapnya dapat dilihat pada sub-bab pengembangan model.

Pada kalimat kedua, dihasilkan frase “*temperature which increased to warm*”. Kata “*warm*” diperoleh dari hasil prediksi dengan *exponential smoothing* yang menghasilkan nilai 17.90°C , lalu melalui proses *Data Interpretation* dengan menggunakan *Fuzzy* didapatkan kata tersebut. Lalu, kata “*increased*” didapatkan melalui proses *Lexicalisation*, dimana proses ini memilih kata yang tepat untuk digunakan dalam mengekspresikan persepsi pergerakan data. Lalu, pada kalimat selanjutnya terdapat frase “*will stay stable at good*”. Kata “*good*” didapatkan melalui proses *Data Interpretation* khusus untuk kualitas udara dimana memiliki fungsi tersendiri, karena untuk menerjemahkan data kualitas udara diperlukan perhitungan khusus. Maka, penulis menggunakan salah satu perhitungan standar internasional yaitu *Pollutant Standard Index* (PSI). Informasi tentang *Wind* atau angin tidak disampaikan karena pada Gambar 4.56 dapat dilihat bahwa hasil prediksi dari nilai *Wind Speed* adalah 7.23km/h , dimana jika nilai tersebut diinterpretasikan, menghasilkan frase “*light breeze*”.

Pada paragraf kedua, perencanaan dokumen dihasilkan dengan membangun struktur kalimat mengikuti *Rhetorical Structured Theory* seperti

yang dapat dilihat pada sub-bab pengembangan model. Pesan pertama yang disampaika adalah keadaan bulan ini yaitu “*cooler and wetter*”, pesan ini menjelaskan bahwa rata-rata suhu pada bulan ini lebih dingin dari biasanya, dan curah hujan yang turun lebih banyak dari biasanya. Pada kalimat berikutnya dijelaskan jumlah hari hujan dan bagaimana kondisi hujan pada minggu terakhir. Lalu, disampaikan pesan hujan berturut-turut, artinya serangkaian hujan yang terus-menerus terjadi dalam deret waktu tertentu. Dalam Gambar 4.56, dijelaskan bahwa selama satu bulan terakhir hujan berturut-turut terjadi selama 4 hari dari tanggal 27 sampai dengan tanggal 30. Untuk menyampaikan nilai extrem, didapatkan keadaan extrem terbesar dengan kategori “*Heavy Rain*” “*Heavy Rain*” yang turun pada tanggal 28 dan 29. Setelah itu, sistem menginformasikan rata-rata kecepatan angin, kualitas udara, dan temperatur terdingin. Semua informasi ini dihasilkan dengan membuat fungsi yang dapat dilihat pada sub-bab pengembangan model dan implemetasi.

4.4.2 Perbandingan *output* dengan penelitian lain

Pada sub-bab ini, penulis membandingkan hasil dari keluaran sistem yang telah dibangun dari sistem *Data-to-text Weather Prediction* (DWP) dengan keluaran dari penelitian yang dilakukan oleh (Ramos-soto, dkk., 2015), (Goldberg, Driedger, dan Kitterdge, 1994) dan (Gkatzia, Lemon, dan Rieser, 2016).

Tabel 4.17 Perbandingan *output* yang dibangun dengan *output* dari penelitian terkait lainnya

DWP Output (Ramos-soto, dkk., 2015) Output (Goldberg, Driedger, dan Kitterdge, 1994) (Gkatzia, Lemon, dan Rieser, 2016)	<p>Regarding to the prediction result, tomorrow sky state will be light rain although its covered by partly cloudy sky. Followed by temperature which decreased to warm. According to the air quality state, it will start to change to good.</p> <p>According to the monthly summary result, this month was cooler and wetter than average. With average number of rain days, accordingly the total rain so far is well below the average. There was rain on everyday for 7 days from 02nd to 08th and intense rain was dropped in 06th. The wind for the month was light breeze in average. Average air quality was admissible. Average temperature was increased but 05 th was the coldest day of the month with 13.3 celcius degree temperature.</p> <p>With respect to the air quality state, it will be variable although is expected to improve to good, favored by the wind during the coming days</p> <p>Winds northwest 15 diminishingto light monday afternoon. Cloudy with occasional light snow. Fog patches. Visibilities 2 to 5 nm in snow. Belle isle. Northeast gulf northeast coast. Gale warning in belle isle and northeast gulf issued. Gale warning in northeast coast continued. Freezing spray warning continued. Winds southwest 15 to 20 knots increasing to west gales 35</p> <p>-Light rian showers are likely -Sunny intervals with rain being possible – less likely than not. -Sunny with rain being unlikely</p>
--	--

Dari Tabel 4.17 dapat dilihat bahwa kelebihan aplikasi ini jika dibandingkan dengan *output* lainnya adalah jumlah konten yang lebih banyak, hal ini dapat terjadi karena selama proses pengembangan penulis mengutip beberapa sumber dan menentukan pesan yang akan disampaikan. *Output* dari (Ramos-soto, et al., 2015) terdiri dari pesan kualitas udara dan kecepatan angin, sedangkan (Goldberg, Driedger, & Kitterdge, 1994) terdiri dari pesan-pesan angin dan salju secara spasial. Dalam (Gkatzia, Lemon, & Rieser, 2016) *output* yang disampaikan hanya terkait dengan keadaan langit.

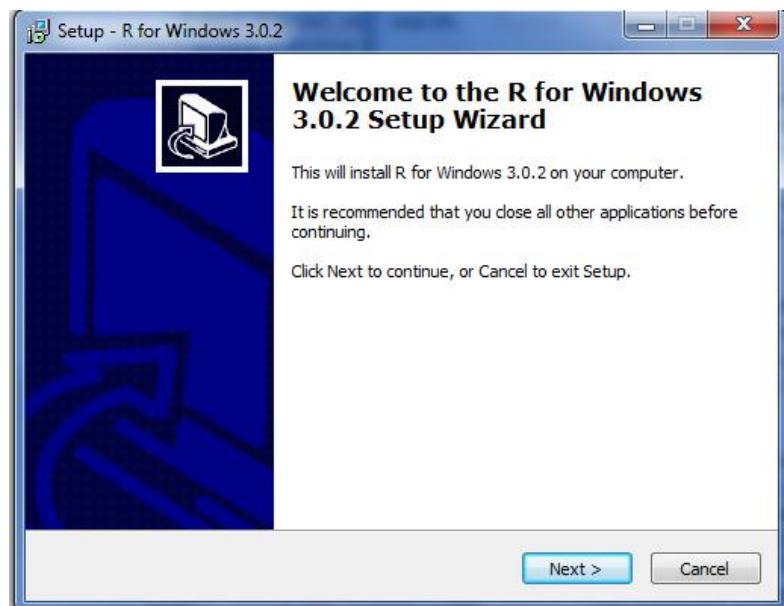
4.5 Panduan Penggunaan Aplikasi

Untuk mempermudah penggunaan aplikasi, penulis memaparkan tahapan penggunaan dalam poin-poin berikut:

1. Install RGui atau Rstudio

R dapat diperoleh di CRAN-archive yaitu *The Comprehensive R Archive Network* di alamat <http://cran.r-project.org>. Pada server CRAN ini dapat didownload file instalasi *binary* dan *source code* dari *R-base System* dalam sistem operasi Windows (semua versi), linux, dan Macintosh.

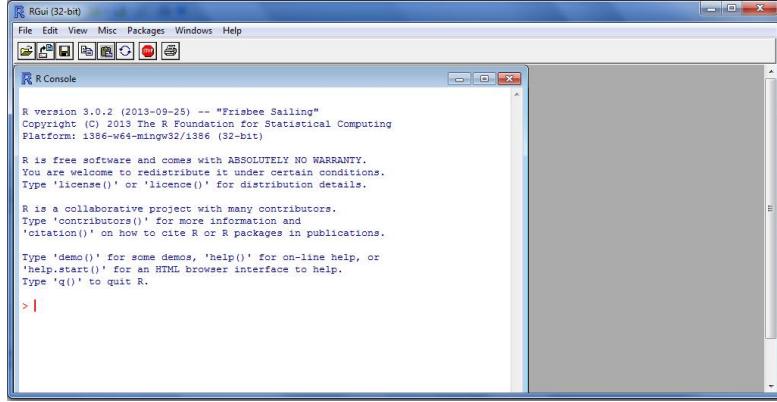
- a. Setelah download file **R-3.0.2-win.exe**, Klik dua kali (*Single click*) file **R-3.0.2-win.exe**. maka akan muncul tampilan seperti pada Gambar 4.57.



Gambar 4.57 Jendela saat proses instalasi aplikasi *R for Windows*.

- b. Setelah itu, lanjutkan jalannya proses instalasi dengan mengikuti *Wizard* dan menggunakan pilihan-pilihan **default instalasi**.
- c. Langkah terakhir jika instalasi R telah selesai adalah melakukan pengecekan atau pengujian apakah program R dapat berjalan dengan

baik. Lakukan klik dua kali pada **shortcut R** di **Desktop** atau pada **Start Menu**. Jika instalasi berlangsung dengan baik, maka jendela program R akan terbuka seperti yang terlihat pada Gambar 4.59.



Gambar 4.58 Antarmuka RGui.

2. Untuk membuka Apps dengan Website interfacenya,
 - a. Masuk ke parent directory dari "D2T_Apps" seperti pada Gambar 4.60.

Location: D:/Data-to-text/ (parent directory of "D2T_Apps")			
📁 Backup	7/24/2017 9:39 AM	File folder	
📁 CSV	7/8/2017 1:54 AM	File folder	
📁 D2T_Apps	8/5/2017 1:01 PM	File folder	
📁 D2T_Apps - backup	7/31/2017 7:40 PM	File folder	
📁 Data-to-text	7/9/2017 8:28 AM	File folder	
📁 R-Packages	7/8/2017 2:32 AM	File folder	
🇷 .RData	7/28/2017 5:03 AM	R Workspace	
📄 .Rhistory	7/28/2017 5:03 AM	RHISTORY File	

Gambar 4.59 Direktori D2T_Apps

- b. Install *Packages* ShinyR → `install.packages("shiny")`, seperti pada Gambar 4.60.

```
#1. Shiny R
install.packages(shiny) #-- server & GUI
library(shiny)
```

Gambar 4.60 Instalasi *packages* ShinyR

```
> install.packages("shiny")
Installing package into 'C:/Users/Brahma/Documents/R/win-library/3.3'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
also installing the dependency 'httpuv'

  There is a binary version available but the source version is later:
    binary source needs_compilation
shiny 1.0.3  1.0.4          FALSE

trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.3/httpuv_1.3.5.zip'
Content type 'application/zip' length 928826 bytes (907 KB)
downloaded 907 KB

package 'httpuv' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/Brahma/AppData/Local/Temp/RtmpYZgtRb downloaded_packages
installing the source package 'shiny'

trying URL 'https://cloud.r-project.org/src/contrib/shiny_1.0.4.tar.gz'
Content type 'application/x-gzip' length 2329723 bytes (2.2 MB)
downloaded 2.2 MB

* installing *source* package 'shiny' ...
** package 'shiny' successfully unpacked and MD5 sums checked
```

Gambar 4.61 Contoh *output* dari *console* jika instalasi *packages* ShinyR berhasil

Contoh keluaran *console* dapat dilihat pada Gambar 4.61 setelah dilakukan instalasi Shiny.

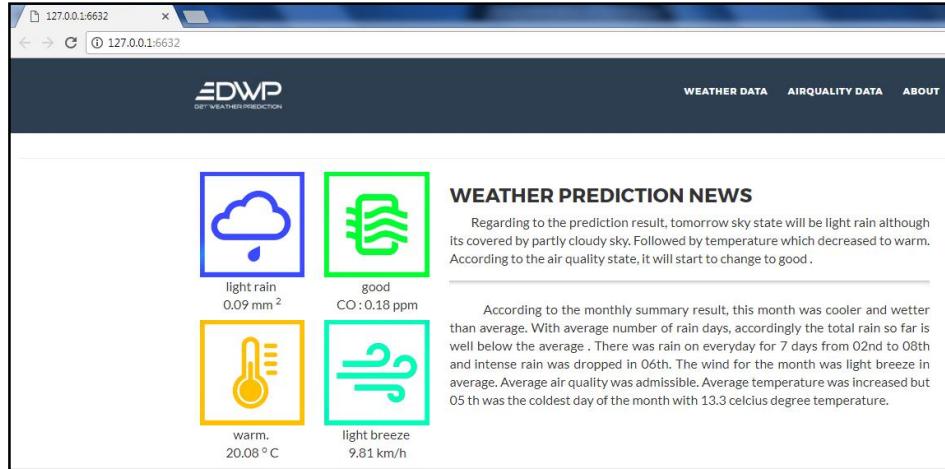
- c. Buka dengan cara → runApp("D2T_Apps"), maka akan muncul jendela dari *default browser* menayangkan aplikasi R yang telah dibuat berbasis website. Sedangkan yang akan muncul di *console* adalah seperti yang ada pada Gambar 4.62.

```
> runApp ("D2T_Apps")

Listening on http://127.0.0.1:6632
Warning: package 'smooth' was built under R version 3.3.3
This is package "smooth", v2.0.0
Warning: package 'gradDescent' was built under R version 3.3.3
Warning: package 'xts' was built under R version 3.3.3
Loading required package: zoo
Warning: package 'zoo' was built under R version 3.3.3
```

Gambar 4.62 Keluaran dari *console* setelah server dijalankan pada port 6632

Sedangkan Gambar 4.63 menunjukkan jendela browser yang terbuka apabila aplikasi sudah dijalankan. Terdiri dari *output* sistem dan beberapa fitur web seperti menubar, footer, dan lain-lain.



Gambar 4.63 Contoh jendela browser yang muncul ketika menjalakan perintah
runApp("D2T_Apps")

3. Untuk membuka *Machine* nya saja, dilakukan langkah sebagai berikut.

- Masuk ke directory dari D2T_Apps

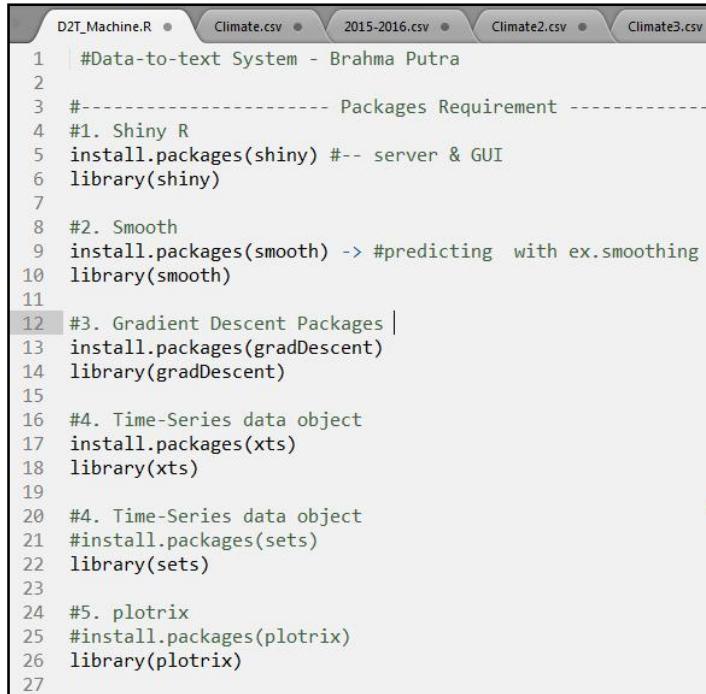
Location: D:/Data-to-text/D2T_Apps			
Backups	8/8/2017 11:26 PM	File folder	
Datasets	8/5/2017 2:26 PM	File folder	
wordbank	8/5/2017 2:55 AM	File folder	
www	7/30/2017 8:24 PM	File folder	
D2T_Machine.R	8/11/2017 8:04 AM	R File	
server.R	8/4/2017 7:47 PM	R File	
ui.R	8/7/2017 12:30 AM	R File	

Gambar 4.64 Isi dari direktori D2T_Apps

- buka dengan cara perintah: *Source("D2T_Machine.R")*

4. Untuk mengganti Datasets, lakukan langkah-langkah berikut.

a. buka file "D2T_Machine.R"



```

1 #Data-to-text System - Brahma Putra
2
3 #----- Packages Requirement -----
4 #1. Shiny R
5 install.packages(shiny) #-- server & GUI
6 library(shiny)
7
8 #2. Smooth
9 install.packages(smooth) -> #predicting with ex.smoothing
10 library(smooth)
11
12 #3. Gradient Descent Packages |
13 install.packages(gradDescent)
14 library(gradDescent)
15
16 #4. Time-Series data object
17 install.packages(xts)
18 library(xts)
19
20 #4. Time-Series data object
21 #install.packages(sets)
22 library(sets)
23
24 #5. plotrix
25 #install.packages(plotrix)
26 library(plotrix)
27

```

Gambar 4.65 Contoh isi dari file D2T_Machine.R

Pada Gambar 4.65 diperhatikan bagaimana isi dari file "D2T.Machine.R".

b. Mengubah nama datasets cuaca & kualitas udara dengan cara mengganti nama datasets yg dibutuhkan, Datasets tersedia dalam direktori "D2T_Apps/Datasets/".

Location: D:/Data-to-text/D2T_Apps/Datasets			
2013-2014.csv	8/5/2017 10:31 AM	Microsoft Office E...	11 KB
2014-2015.csv	7/28/2017 6:23 AM	Microsoft Office E...	12 KB
2015-2016.csv	8/5/2017 10:28 AM	Microsoft Office E...	12 KB
2016-2017.csv	7/10/2017 5:08 PM	Microsoft Office E...	12 KB
AQ_2013_2014.csv	8/5/2017 1:41 PM	Microsoft Office E...	19 KB
AQ_2014_2015.csv	8/5/2017 10:19 AM	Microsoft Office E...	22 KB
AQ_2015_2016.csv	8/5/2017 10:26 AM	Microsoft Office E...	14 KB
AQ_2016_2017.csv	7/11/2017 6:16 AM	Microsoft Office E...	13 KB
AQ_Modified1.csv	8/4/2017 11:18 PM	Microsoft Office E...	13 KB
AQ_Modified2.csv	8/4/2017 7:52 PM	Microsoft Office E...	14 KB
AQ_Modified3.csv	8/4/2017 7:52 PM	Microsoft Office E...	14 KB
AQ_Modified4.csv	8/5/2017 2:03 PM	Microsoft Office E...	13 KB
AQ_Modified5.csv	8/5/2017 2:15 PM	Microsoft Office E...	12 KB
AQ_Modified6.csv	8/5/2017 2:26 PM	Microsoft Office E...	10 KB
Modified_1.csv	8/5/2017 2:25 AM	Microsoft Office E...	12 KB
Modified_2.csv	8/4/2017 5:44 AM	Microsoft Office E...	13 KB
Modified_3.csv	8/5/2017 10:41 AM	Microsoft Office E...	9 KB
Modified_4.csv	8/5/2017 2:01 PM	Microsoft Office E...	10 KB
Modified_5.csv	8/5/2017 2:14 PM	Microsoft Office E...	10 KB
Modified_6.csv	8/5/2017 2:25 PM	Microsoft Office E...	9 KB

Gambar 4.66 Folder *Datasets*, berisi seluruh dataset yang tersedia.

Pada Gambar 4.66, diperlihatkan beberapa datasets yang disimpan dalam bentuk CSV, untuk memuatnya kedalam sistem sehingga sistem dapat berjalan, pengguna harus mengganti nama database seperti pada teks yang diberi garis merah pada Gambar 4.67.

```
#----- Load and transform -----#
#Load dataset CLIMATE : Rainfall, Cloud Coverage, Tempreature, Wind Speed, Wind Direction
dataClimates <- read.table(file="Datasets/2015-2016.csv", sep=",", header=TRUE)

#transform "Date" Variable from integer to date
dataClimates <- transform(dataClimates, Date = as.Date(as.character(Date), "%d/%m/%Y"))

#Load dataset AirQuality
dataAQ <- read.table(file="Datasets/AQ_2016_2017.csv", sep=",", header=TRUE)

#transform "Date" Variable from integer to date
dataAQ <- transform(dataAQ, Date = as.Date(as.character(Date), "%d/%m/%Y"))
#-----#
```

Gambar 4.67 Cara mengganti dataset yang digunakan dalam file
D2T_Machine.R

4.6 Rancangan Experimen

Berdasarkan (Belz, 2007), serangkaian cara untuk melakukan evaluasi kalimat bahasa alami yang dibangkitkan oleh sistem dilakukan dengan mengevaluasi kualitas teks dengan menggunakan *NIST* dan *BLEU*, mengevaluasi hasil text dengan *Expert* dibidang terkait, mengevaluasi waktu komputasi, serta mengevaluasi waktu pengembangan perangkat lunak.

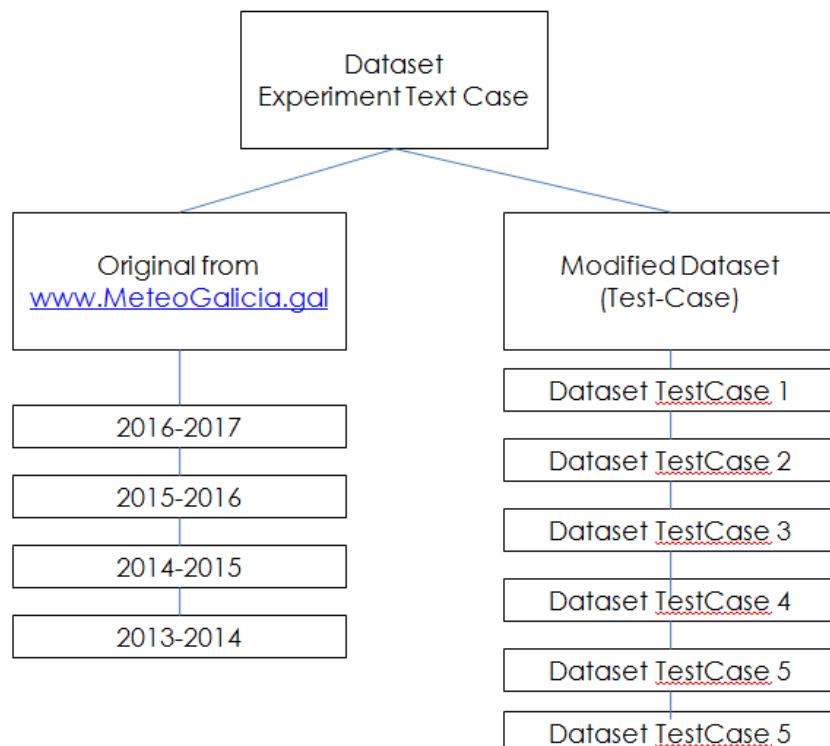
Sedangkan dalam (Ramos-soto, dkk., 2015) pengujian dilakukan dengan 20 kali pembangkitan bahasa alami dengan *input* test-case yang berbeda. Setiap test case yang dibangun dievaluasi oleh *expert* dan diberi nilai kepuasan. Nilai tersebut didapatkan dengan cara memberi pertanyaan terkait dengan *relevance* dan *truthfullness*. Sehingga, didapatkan nilai rata-rata dari seluruh test-case dan dapat diambil kesimpulan dari nilai tersebut.

Dengan menyimpulkan cara eksperimen dari dua contoh diatas, maka penulis membuat skenario eksperimen untuk penelitian ini terdiri dari enam aspek berikut:

1. Aspek *Readability*, yaitu mengevaluasi kualitas teks dengan menggunakan penilaian *Flesch Reading Ease*. Dievaluasi oleh *System datayze*.
2. Aspek *Computation Time*, yaitu mengevaluasi waktu komputasi sistem dengan menggunakan perintah *System.time()*.
3. *Relevance*, yaitu mengevaluasi kesesuaian konten dengan apa yang biasa disampaikan oleh pakar, nilai oleh *expert* dari BMKG dan AMG.
4. *Truthfulness*, yaitu mengevaluasi kesetujuan pakar atas konten yang disampaikan sertakorelasi konten dengan data, nilai oleh *expert* dari BMKG dan AMG.
5. *Comprehensibility*, yaitu mengevaluasi kemudahan penyampaian informasi, dinilai oleh *end-user*.

6. *Importance*, yaitu mengevaluasi seberapa penting informasi yang disampaikan, dinilai oleh *end-user*.

Sepuluh data test-case dipersiapkan untuk digunakan, dataset tersebut terdiri dari 4 data aktual yaitu data yang diperoleh dari www.MeteoGalicia.gal dengan rentang waktu yang berbeda, dan 6 data kondisional yaitu data yang dimodifikasi oleh penulis untuk mengidentifikasi kelemahan sistem. Pembagian ini dapat dilihat pada Gambar 4.68.



Gambar 4.68 Dataset eksperimen

1. Penilaian terhadap aspek *Readability*: Melakukan evaluasi terhadap kualitas text dari setiap *output* yang dihasilkan oleh sistem, dengan menggunakan metode *Flesch Reading Ease*. Metode ini digunakan untuk menguji tingkat *readability* dari sebuah teks untuk menunjukkan tingkat kesulitan untuk dimengerti. Pada Tabel 4.18. Pada proses ini digunakan tools yang tersedia pada website www.datayze.com.

Tabel 4.18 Penilaian kualitas *Readability* teks dengan menggunakan *Flesch Reading Ease*.

Score	School Level	Notes
100.00–90.00	5th grade	<i>Very easy to read. Easily understood by an average 11-year-old student.</i>
90.0–80.0	6th grade	<i>Easy to read. Conversational English for consumers.</i>
80.0–70.0	7th grade	<i>Fairly easy to read.</i>
70.0–60.0	8th & 9th grade	<i>Plain English. Easily understood by 13- to 15-year-old students.</i>
60.0–50.0	10th to 12th grade	<i>Fairly difficult to read.</i>
50.0–30.0	College	<i>Difficult to read.</i>
30.0–0.0	College Graduate	<i>Very difficult to read. Best understood by university graduates.</i>

2. Mengevaluasi hasil komputasi, yaitu dengan menghitung waktu komputasi dari setiap eksperimen. Dari sepuluh dataset yang dibangkitkan, dihitung waktu komputasinya dengan menggunakan perintah *System.time()*.
3. Penilaian terhadap aspek *Relevance* dan *Truthfulness*: Mengevaluasi kualitas prediksi, summary, dan korelasinya dengan teks. Pada tahap evaluasi ini penulis mengutip apa yang dilakukan oleh (Ramos-soto, dkk., 2015), dimana memberikan dua pertanyaan kepada *expert* mengenai untuk mengidentifikasi kualitas sistem *Data-to-text*. Pertanyaan tersebut terdiri dari:
 - Terkait aspek *Relevance*: dalam skala 1 -5, berapa nilai relevansi antara informasi yang disampaikan oleh sistem dengan nilai informasi yang disampaikan oleh *expert*?

- Terkait aspek *Truthfullness*: dalam skala 1-5, seberapa besar tingkat kesetujuan anda dengan hasil deskripsi dari (a) prediksi; dan (b)*summary*.

Untuk mencaai hal ini, untuk mempermudah sistem penilaian, penulis menggunakan kuisioner yang diisi oleh *expert* terkait dengan hasil eksperimen. Dalam evaluasi ini, diberikan sebelas pilihan jawaban yang mewakili kesesuaian konten dengan *user* atau *expert*.

Tabel 4.19 Keterangan rentang kesesuaian kuisioner.

Nilai	Deskripsi
0	TS
0.5 - 1	SKS
1-2	KS
2-3	HCS
3-3.5	CS
3.5-4	HS
4-4.5	S
4.5 – 5	SS

Pada Tabel 4.19, terdapat rentang nilai yang menunjukkan tingkat kesesuaian menggunakan *Likert Scale* (skala Likert) (Bertam, 2015), dengan keterangan sebagai berikut: Tidak Sesuai (TS), Sangat Kurang Sesuai (SKS), Kurang Sesuai (KS), Hampir Cukup Sesuai (HCS), Cukup Sesuai (CS), Hampir Sesuai (HS), Sesuai (S), dan Sangat Sesuai (SS).

D2T System Evaluation Target: Expert

Experiment Datasets:
 WeatherDatasets1(Modified_1.csv)
 AirQualityDatasets1(AQModified_1.csv)

WEATHER PREDICTION NEWS

Regarding to the prediction result, tomorrow sky state will be heavy rain covered with partly cloudy sky. Followed by temperature which increased to warm. Based on the air quality forecast result, it will change progressively to hazardous. the wind will blow near gale from the west.

heavy rain 9.87 mm ²	hazardous CO : 26.66 ppm
warm. 22.73 °C	near gale 57.56 km/h

This month summary showed that the month was cooler but drier than average. With high number of rain days, the total rain so far is well below the average. There was rain on everyday for 7 days from 24th to 30th and torrential rain were dropped in 07th, 10th and 17th. The wind for the month was strong breeze in average, but storm were conducted in 14th, 24th, and 31st. Average air quality was hazardous. Average temperature was increased but 21 st was the coldest day of the month with 11 celcius degree.

1. Terkait aspek *Relevance* : dalam skala 1 - 5, berapa nilai relevansi antara informasi yang disampaikan oleh sistem dengan nilai informasi yang disampaikan oleh expert?

0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5

*pilih salah satu (v)

- Terkait aspek *Truthfulness* : dalam skala 1 - 5, seberapa besar tingkat kesetujuan anda terhadap hasil deskripsi dari (a) prediksi; dan (b)summary, (serta korelasinya dengan data).
 - a) untuk hasil prediksi

0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5

*pilih salah satu (v)

 - b) untuk hasil *summary*

0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5

*pilih salah satu (v)

*detail data ada pada halaman selanjutnya jika dibutuhkan

Gambar 4.69 Evaluasi aspek *relevance* dan *truthfulness* oleh *expert*.

Contoh dari apa yang dinilai oleh *expert* berada pada Gambar 4.70 yaitu berupa kuisioner dengan tiga petanyaan terkait *relevance* dan *truthfulness*.

4. Mengevaluasi aspek *comprehensability* dan *Importance* dengan memberikan kuisioner terhadap *user* yang bukan *expert* mengenai isi deskripsi, yang terdiri dari:

- Dalam skala 1-5, seberapa besar anda memahami informasi tersebut?
- Dalam skala 1-5, apakah informasi yang disajikan berguna?

D2T System Evaluation Target: User

Experiment Datasets:
WeatherDatasets1(Modified1.csv)
AirQualityDatasets1(AQModified_1.csv)

WEATHER PREDICTION NEWS

Regarding to the prediction result, tomorrow sky state will be heavy rain covered with partly cloudy sky. Followed by temperature which increased to warm. Based on the air quality forecast result, it will change progressively to hazzardous. the wind will blow near gale from the west.

heavy rain 9.87 mm ²	hazardous CO : 26.66 ppm
warm 22.73 °C	near gale 57.56 km/h

This month summary showed that the month was cooler but drier than average. With high number of rain days, the total rain so far is well below the average . There was rain on everyday for 7 days from 24th to 30th and torrential rain were dropped in 07th, 10th and 17th. The wind for the month was strong breeze in average, but storm were conducted in 14th, 24th, and 31st. Average air quality was hazzardous. Average temperature was increased but 21 st was the coldest day of the month with 11 celcius degree.

WEATHER DATASETS

Date	Humidity	Cloudiness	Wind Speed	Wind Direction	Temp
2020-01-01	40%	0%	0	N	10
2020-01-02	45%	0%	0	N	12
2020-01-03	50%	0%	0	N	14
2020-01-04	55%	0%	0	N	16
2020-01-05	60%	0%	0	N	18
2020-01-06	65%	0%	0	N	20
2020-01-07	70%	0%	0	N	22
2020-01-08	75%	0%	0	N	24
2020-01-09	80%	0%	0	N	26
2020-01-10	85%	0%	0	N	28
2020-01-11	90%	0%	0	N	30
2020-01-12	95%	0%	0	N	32
2020-01-13	100%	0%	0	N	34
2020-01-14	100%	100%	1	S	36
2020-01-15	95%	100%	2	S	38
2020-01-16	90%	100%	3	S	40
2020-01-17	85%	100%	4	S	42
2020-01-18	80%	100%	5	S	44
2020-01-19	75%	100%	6	S	46
2020-01-20	70%	100%	7	S	48
2020-01-21	65%	100%	8	S	50
2020-01-22	60%	100%	9	S	52
2020-01-23	55%	100%	10	S	54
2020-01-24	50%	100%	11	S	56
2020-01-25	45%	100%	12	S	58
2020-01-26	40%	100%	13	S	60
2020-01-27	35%	100%	14	S	62
2020-01-28	30%	100%	15	S	64
2020-01-29	25%	100%	16	S	66
2020-01-30	20%	100%	17	S	68
2020-01-31	15%	100%	18	S	70

AIRQUALITY DATASETS

Date	PM2.5	PM10	SO2	NO2	CO	NO	NO2	NO	NO2	NO	NO2
2020-01-01	10	20	5	10	10	5	10	10	10	10	10
2020-01-02	12	22	6	12	12	6	12	12	12	12	12
2020-01-03	14	24	7	14	14	7	14	14	14	14	14
2020-01-04	16	26	8	16	16	8	16	16	16	16	16
2020-01-05	18	28	9	18	18	9	18	18	18	18	18
2020-01-06	20	30	10	20	20	10	20	20	20	20	20
2020-01-07	22	32	11	22	22	11	22	22	22	22	22
2020-01-08	24	34	12	24	24	12	24	24	24	24	24
2020-01-09	26	36	13	26	26	13	26	26	26	26	26
2020-01-10	28	38	14	28	28	14	28	28	28	28	28
2020-01-11	30	40	15	30	30	15	30	30	30	30	30
2020-01-12	32	42	16	32	32	16	32	32	32	32	32
2020-01-13	34	44	17	34	34	17	34	34	34	34	34
2020-01-14	36	46	18	36	36	18	36	36	36	36	36
2020-01-15	38	48	19	38	38	19	38	38	38	38	38
2020-01-16	40	50	20	40	40	20	40	40	40	40	40
2020-01-17	42	52	21	42	42	21	42	42	42	42	42
2020-01-18	44	54	22	44	44	22	44	44	44	44	44
2020-01-19	46	56	23	46	46	23	46	46	46	46	46
2020-01-20	48	58	24	48	48	24	48	48	48	48	48
2020-01-21	50	60	25	50	50	25	50	50	50	50	50
2020-01-22	52	62	26	52	52	26	52	52	52	52	52
2020-01-23	54	64	27	54	54	27	54	54	54	54	54
2020-01-24	56	66	28	56	56	28	56	56	56	56	56
2020-01-25	58	68	29	58	58	29	58	58	58	58	58
2020-01-26	60	70	30	60	60	30	60	60	60	60	60
2020-01-27	62	72	31	62	62	31	62	62	62	62	62
2020-01-28	64	74	32	64	64	32	64	64	64	64	64
2020-01-29	66	76	33	66	66	33	66	66	66	66	66
2020-01-30	68	78	34	68	68	34	68	68	68	68	68
2020-01-31	70	80	35	70	70	35	70	70	70	70	70

1. Dalam skala 1 – 5, tentukan seberapa besar anda memahami informasi didalam teks tersebut.

0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5

*pilih salah satu (v)

2. Dalam skala 1 – 5, apakah informasi yang disajikan dalam teks diatas berguna bagi anda?

0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5

*pilih salah satu (v)

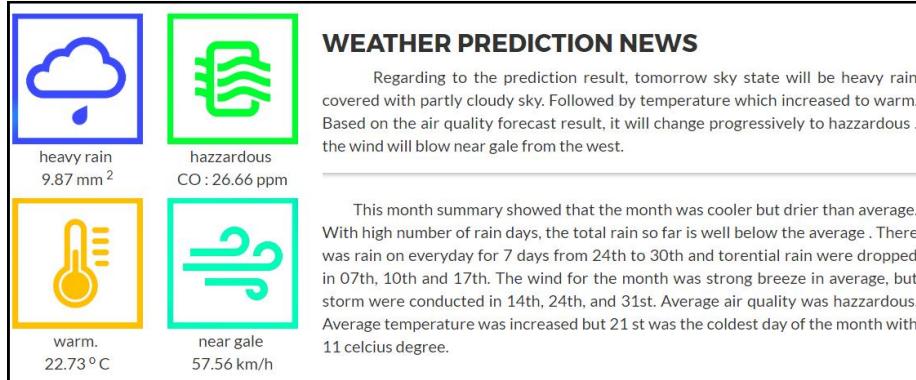
Gambar 4.70 Evaluasi aspek *comprehensability* dan *importance* oleh *expert*.

Sedangkan bentuk kuisioner yang diisi oleh *end-user* berada pada Gambar 4.71 terkait dengan aspek *Comprehensability* dan *Importance*.

4.7 *Output eksperimen*

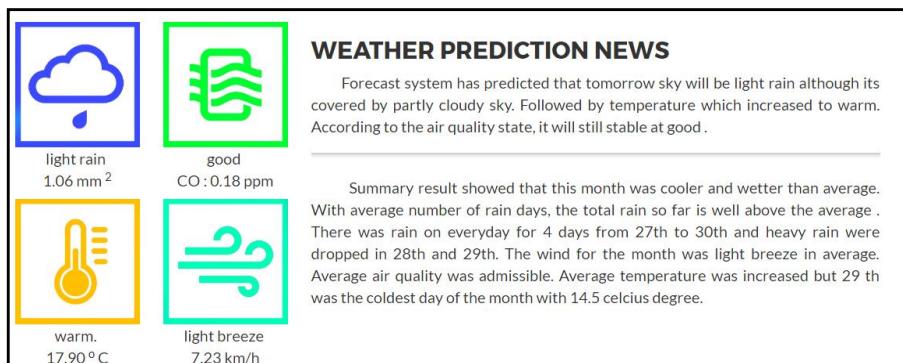
Menurut sub-bab sebelumnya, dalam penelitian ini dibangkitkan hasil eksperimen dari sepuluh dataset, dimana daftar dari setiap dataset tersebut disertakan didalam lampiran.

Untuk eksperimen pertama, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “Modified1.csv” untuk data cuaca, dan “AQModified_1.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.71.



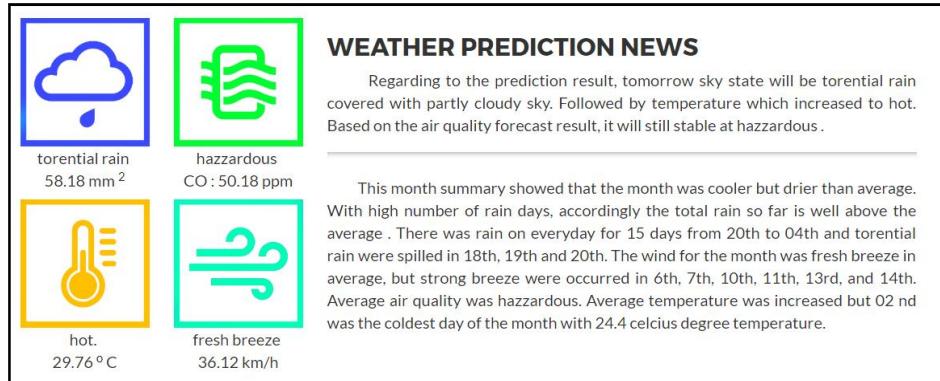
Gambar 4.71 Hasil keluaran dari eksperimen ke-dua dengan data *Modified 2*.

Untuk eksperimen kedua, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “2015-2016.csv” untuk data cuaca, dan “AQ_2016_2017.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.72.



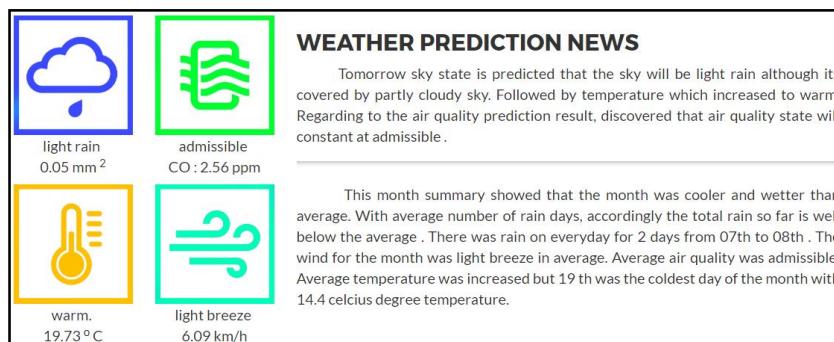
Gambar 4.72 Hasil keluaran dari eksperimen ke-dua dengan data real 2016-2017

Untuk eksperimen ketiga, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “Modified2.csv” untuk data cuaca, dan “AQModified_2.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.73.



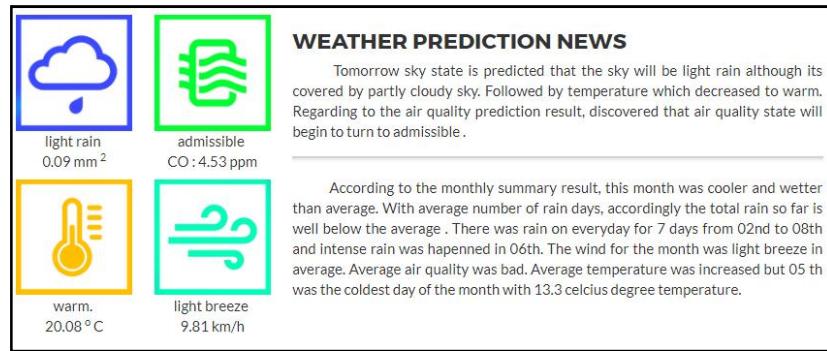
Gambar 4.73 Hasil keluaran dari eksperimen ke-tiga dengan data Modified2

Untuk eksperimen keempat, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “2014-2015.csv” untuk data cuaca, dan “AQ_2014_2015.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.74.



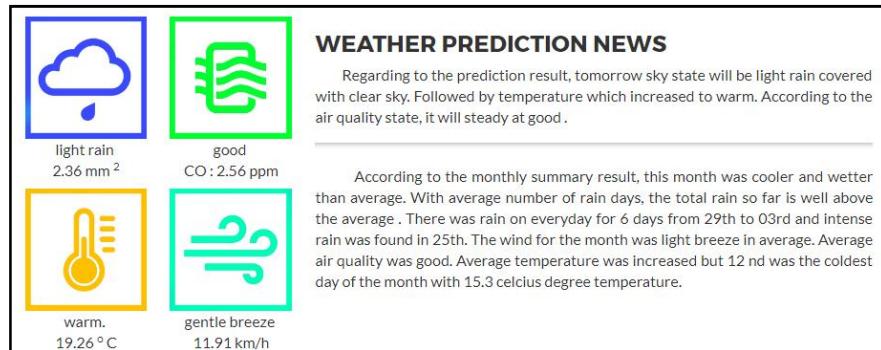
Gambar 4.74 Hasil keluaran dari eksperimen ke-empat dengan data real 2014-2015

Untuk eksperimen kelima, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “2015-2016.csv” untuk data cuaca, dan “AQ_2015_2016.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.75.



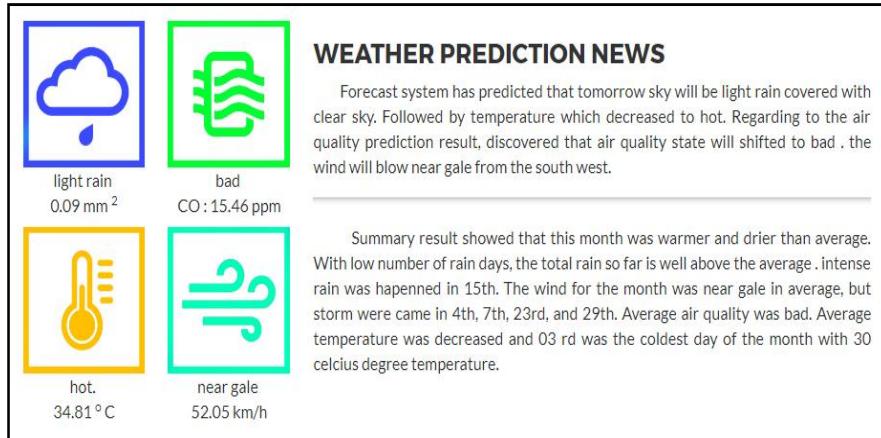
Gambar 4.75 Hasil keluaran dari eksperimen ke-lima dengan data real 2015-2016

Untuk eksperimen keenam, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “2013-2014.csv” untuk data cuaca, dan “AQ_2014_2015.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.76.



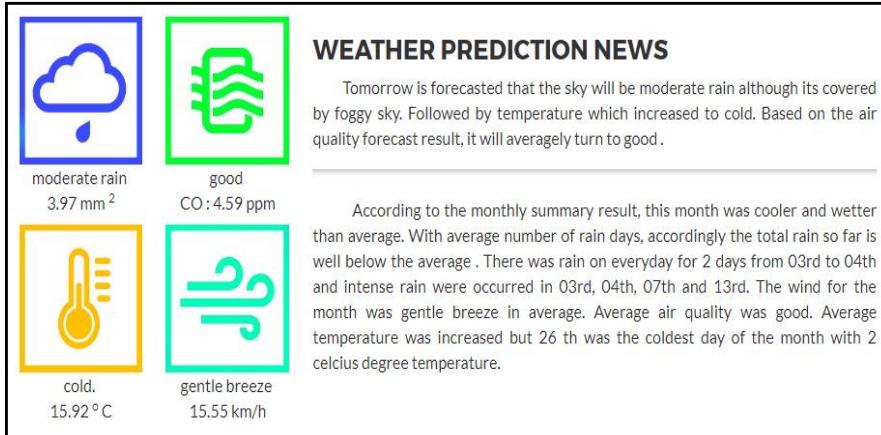
Gambar 4.76 Hasil keluaran dari eksperimen ke-enam dengan data real 2013-2014

Untuk eksperimen ketujuh, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “Modified3.csv” untuk data cuaca, dan “AQModified_3.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.77.



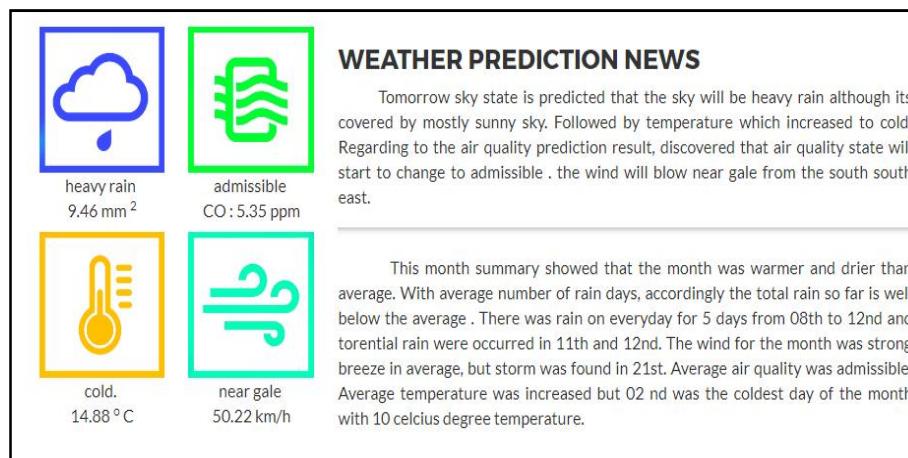
Gambar 4.77 Hasil keluaran dari eksperimen ke-tujuh dengan data *Modified 3*.

Untuk eksperimen kedelapan, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “Modified4.csv” untuk data cuaca, dan “AQModified_4.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.78.



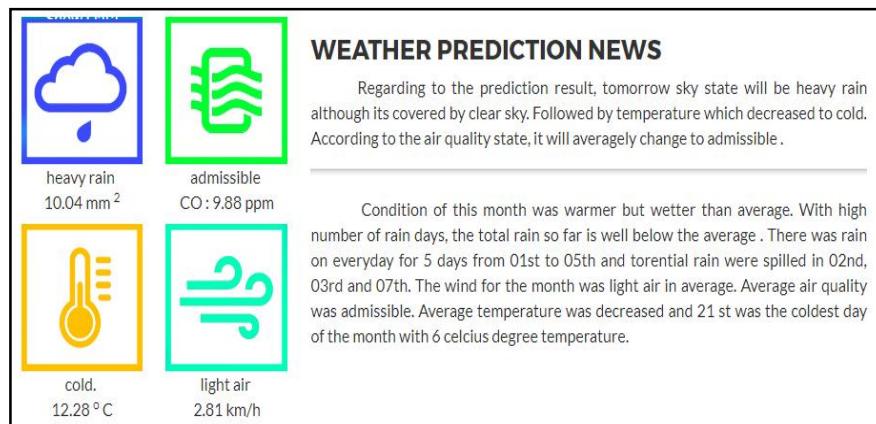
Gambar 4.78 Hasil keluaran dari eksperimen ke-delapan dengan data *Modified 4*.

Untuk eksperimen kesembilan, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “Modified5.csv” untuk data cuaca, dan “AQModified_5.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.79.



Gambar 4.79 Hasil keluaran dari eksperimen ke-sembilan dengan data *Modified 5*.

Untuk eksperimen kesepuluh, digunakan dataset pada website yang dimodifikasi. Data ini terletak didalam direktori D2T_Apps/Datasets, dengan nama “Modified6.csv” untuk data cuaca, dan “AQModified_6.csv” untuk data kualitas udara. Setelah dibangkitkan, dapat dilihat hasilnya pada Gambar 4.80.



Gambar 4.80 Hasil keluaran dari eksperimen ke-sepuluh dengan data *Modified 6*.

4.8 Analisis dan Hasil Eksperimen

- Hasil analisis aspek *readability*

Eksperimen pertama, dilakukan untuk mengetahui tingkat *readability* pada text yang dihasilkan. Pengujian dilakukan dengan membangkitkan bahasa alami dari dataset yang berbeda seperti pada Gambar 4.68. Lalu, setiap text yang dihasilkan dievaluasi dengan menggunakan metode *Flesch Reading Ease*. Untuk eksperimen ini, penulis menggunakan tools yang disediakan oleh *Readability Analyzer*, tersedia pada www.datayze.com. Hasil dari pengujian ini dapat dilihat pada Tabel 4.20.

Tabel 4.20 Hasil evaluasi dengan menggunakan *Flesch Reading Ease*.

	<i>Data-Test-case</i>	<i>Flesch Score</i>
1	DT1	67.24
2	DT2	65.69
3	DT3	59.57
4	DT4	64.81
5	MT1	64.3
6	MT2	66.32
7	MT3	71.02
8	MT4	68.5
9	MT5	64.53
10	MT6	65.39
Average		65.73

Dari Tabel 4.20 dapat dilihat bahwa *Flesch Score* tertinggi berada pada percobaan ke tujuh, dengan score 71,02. Sedangkan score terendah berada pada percobaan ke lima yaitu sebesar 64,3. Kesimpulan dari aspek *readability* ini, rata-rata dari *Flesch Score* adalah 65,73. Jika dilihat dari Tabel 4.1.1, nilai rata-rata dari score yang didapat berada pada interval 60-70. Artinya, kualitas *readability* dari teks yang dihasilkan dari sistem ini yaitu: “*Plain English, Easily understood by 13- to 15-year-old students*”.

2. Hasil analisis aspek *Computation Time*

Untuk mengevaluasi waktu komputasi, setiap eksperimen dihitung waktu komputasinya dengan menggunakan fungsi *System.time()*, cara untuk menggunakan fungsi ini adalah dengan memetakan file R kedalam fungsi *System.time()*. Misalnya *System.time(source("D2T_Machine.R"))*. Untuk hasil analisis waktu komputasi dapat dilihat pada Tabel 4.21.

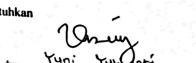
Tabel 4.21 Hasil perhitungan waktu komputasi dengan *System.time()*

	<i>Data-Test-case</i>	<i>Running Time(s)</i>
1	DT1	2.24
2	DT2	2.95
3	DT3	3.72
4	DT4	2.54
5	MT1	2.46
6	MT2	3.24
7	MT3	3.59
8	MT4	2.96
9	MT5	2.41
10	MT6	4.03
Average		3.01

Tabel 4.21 menunjukkan bahwa waktu komputasi tertinggi / terlama berada pada percobaan ke 10, dengan waktu komputasi 4.03 detik. Sedangkan waktu komputasi tercepat berada pada percobaan kesatu yaitu sebesar 2.24 detik. Kesimpulan dari aspek *Computation Time* ini, rata-rata dari waktu komputasi adalah 3.01 detik.

3. Hasil analisis aspek *Relevance* dan *Truthfulness*

Untuk mengevaluasi aspek *Relevance* dan *Truthfulness*, seperti pada subbab sebelumnya, diberikan kuisioner yang dinilai oleh *expert*. Penilaian ini diisi pada tanggal 17 juli 2017 di BMKG Bandung. Contoh pengisian kuisioner dapat dilihat pada Gambar 4.81.

Target_Expert																															
D2T System Evaluation Experiment Datasets WeatherDatasets1(2016-2017_2.csv) AirQualityDatasets1(AQ_2016_2017.csv)																															
WEATHER PREDICTION NEWS  light rain 1.06 mm ²																															
 good CO: 0.18 ppm																															
 17.90°C																															
 light breeze 7.23 km/h																															
Summary result showed that this month was colder and wetter than average. With average number of rainy days, the total amount of rain is well above the average. There were no sunny days on average. Wind speeds from 27th to 30th and heavy rains were dropped in 28th and 29th. The wind for the month was light breeze in average. Average air quality was admissible. Average temperature was increased but 29 th was the coldest day of the month with 14.5 celsius degree.																															
1. Terkait aspek <i>Relevance</i> : dalam skala 1 - 5, berapa nilai relevansi antara informasi yang disampaikan oleh sistem dengan nilai informasi yang disampaikan oleh <i>expert</i> ?																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0.5</td><td>1</td><td>1.5</td><td>2</td><td>2.5</td><td>3</td><td>3.5</td><td>4</td><td>4.5</td><td>5</td> </tr> <tr> <td colspan="10" style="text-align: center;"><input checked="" type="checkbox"/> *pilih salah satu (v)</td> </tr> </table>											0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	<input checked="" type="checkbox"/> *pilih salah satu (v)									
0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5																					
<input checked="" type="checkbox"/> *pilih salah satu (v)																															
• Terkait aspek <i>Truthfulness</i> : dalam skala 1 – 5, seberapa besar tingkat kesetujuan anda terhadap hasil deskripsi dari (a) prediksi; dan (b)summary. (serta korelasinya dengan data).																															
a) untuk hasil prediksi																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0.5</td><td>1</td><td>1.5</td><td>2</td><td>2.5</td><td>3</td><td>3.5</td><td>4</td><td>4.5</td><td>5</td> </tr> <tr> <td colspan="10" style="text-align: center;"><input checked="" type="checkbox"/> *pilih salah satu (v)</td> </tr> </table>											0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	<input checked="" type="checkbox"/> *pilih salah satu (v)									
0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5																					
<input checked="" type="checkbox"/> *pilih salah satu (v)																															
b) untuk hasil summary																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>0.5</td><td>1</td><td>1.5</td><td>2</td><td>2.5</td><td>3</td><td>3.5</td><td>4</td><td>4.5</td><td>5</td> </tr> <tr> <td colspan="10" style="text-align: center;"><input checked="" type="checkbox"/> *pilih salah satu (v)</td> </tr> </table>											0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	<input checked="" type="checkbox"/> *pilih salah satu (v)									
0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5																					
<input checked="" type="checkbox"/> *pilih salah satu (v)																															
*detail data ada pada halaman selanjutnya jika dibutuhkan																															
 Yunie Tulusanti Forecaster																															

Gambar 4.81 Contoh kuisioner yang diisi oleh pakar klimatologi

Hasil dari penilaian ini dapat dilihat pada Tabel 4.22.

Tabel 4.22 Hasil evaluasi *experts*

	<i>Data-Test-case</i>	<i>Q1(Relevance)</i>	<i>Q2(Truthfulness)</i>
1	DT1	4	4
2	DT2	4	4
3	DT3	4	4.25
4	DT4	3.5	4
5	MT1	4	4
6	MT2	4	4
7	MT3	4	4.5
8	MT4	3.5	4
9	MT5	4	4
10	MT6	4	4
Average		3.9	4.07

4. Hasil evaluasi aspek *comprehensability*

Untuk mengevaluasi aspek *comprehensability* dan *importance*, dalam penelitian ini digunakan evaluasi berbasis user (*user-based*). Artinya, setiap hasil eksperimen akan dinilai oleh *user* terkait dengan dua pertanyaan berikut:

1. Dalam skala 1-5, seberapa besar anda memahami informasi tersebut?
2. Dalam skala 1-5, apakah informasi yang disajukan berguna

. Penilaian ini diisi pada tanggal 18 juli 2017, diperoleh data evaluasi yang dapat dilihat pada Tabel 4.23. Sedangkan untuk contoh kuisioner yang diisi oleh *end-user* dapat dilihat pada Gambar 4.83.

Target User

D2T System Evaluation

Experiment Datasets:
WeatherDatasets1(Modified_2.csv)
AirQualityDatasets1(AQModified_2.csv)

	
torrential rain 58.18 mm ²	hazardous CO : 50.18 ppm
	
hot. 29.76°C	fresh breeze 33.12 km/h

WEATHER PREDICTION NEWS

Regarding to the prediction result, tomorrow sky state will be torrential rain covered with partly cloudy sky. Followed by temperature which increased to hot. Based on the air quality forecast result, it will still stable at hazardous.

This month summary showed that the month was cooler but drier than average. With high number of rain days, accordingly the total rain so far is well above the average. There was rain on everyday for 15 days from 2nd to 16th. The month was torrential rain especially in 1st, 19th and 20th. The month was fresh breeze in average, but strong breeze were occurring in 6th, 7th, 10th, 11th, 13th, and 14th. Average air quality was hazardous. Average temperature was increased but 02 nd was the coldest day of the month with 24.4 celsius degree temperature.

WEATHER DATASETS

Date	Temperature	Humidity	Pressure	Wind Speed	Wind Direction	Rainfall
2023-01-01	29.76	50.18	1012	33.12	180	58.18
2023-01-02	29.76	50.18	1012	33.12	180	58.18
2023-01-03	29.76	50.18	1012	33.12	180	58.18
2023-01-04	29.76	50.18	1012	33.12	180	58.18
2023-01-05	29.76	50.18	1012	33.12	180	58.18
2023-01-06	29.76	50.18	1012	33.12	180	58.18
2023-01-07	29.76	50.18	1012	33.12	180	58.18
2023-01-08	29.76	50.18	1012	33.12	180	58.18
2023-01-09	29.76	50.18	1012	33.12	180	58.18
2023-01-10	29.76	50.18	1012	33.12	180	58.18
2023-01-11	29.76	50.18	1012	33.12	180	58.18
2023-01-12	29.76	50.18	1012	33.12	180	58.18
2023-01-13	29.76	50.18	1012	33.12	180	58.18
2023-01-14	29.76	50.18	1012	33.12	180	58.18
2023-01-15	29.76	50.18	1012	33.12	180	58.18
2023-01-16	29.76	50.18	1012	33.12	180	58.18
2023-01-17	29.76	50.18	1012	33.12	180	58.18
2023-01-18	29.76	50.18	1012	33.12	180	58.18
2023-01-19	29.76	50.18	1012	33.12	180	58.18
2023-01-20	29.76	50.18	1012	33.12	180	58.18
2023-01-21	29.76	50.18	1012	33.12	180	58.18
2023-01-22	29.76	50.18	1012	33.12	180	58.18
2023-01-23	29.76	50.18	1012	33.12	180	58.18
2023-01-24	29.76	50.18	1012	33.12	180	58.18
2023-01-25	29.76	50.18	1012	33.12	180	58.18
2023-01-26	29.76	50.18	1012	33.12	180	58.18
2023-01-27	29.76	50.18	1012	33.12	180	58.18
2023-01-28	29.76	50.18	1012	33.12	180	58.18
2023-01-29	29.76	50.18	1012	33.12	180	58.18
2023-01-30	29.76	50.18	1012	33.12	180	58.18
2023-01-31	29.76	50.18	1012	33.12	180	58.18

AIRQUALITY DATASETS

Date	PM2.5	PM10	NO2	SO2	CO	O3
2023-01-01	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-02	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-03	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-04	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-05	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-06	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-07	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-08	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-09	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-10	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-11	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-12	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-13	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-14	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-15	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-16	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-17	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-18	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-19	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-20	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-21	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-22	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-23	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-24	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-25	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-26	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-27	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-28	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-29	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-30	0.20	0.20	0.20	0.20	0.20	0.20
2023-01-31	0.20	0.20	0.20	0.20	0.20	0.20

1. Dalam skala 1 – 5, tentukan seberapa besar anda memahami informasi didalam teks tersebut.

0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
						✓				

*pilih salah satu (v)

2. Dalam skala 1 – 5, apakah informasi yang disajikan dalam teks diatas berguna bagi anda?

0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
					✓					

*pilih salah satu (v)

Andi .RF
Andi .RF

Gambar 4.82 Contoh kuisioner yang diisi oleh user.**Tabel 4.23** Hasil evaluasi user terkait aspek *comprehensability* dan *importance*.

	Data-Test-case	Q1(Comprehensability)	Q2(Importance)
1	DT1	4	4.5
2	DT2	4	4.5
3	DT3	3.5	3.5
4	DT4	4.5	5
5	MT1	3.5	3.5
6	MT2	3	2.5
7	MT3	3.5	4
8	MT4	3.5	3
9	MT5	3.5	3.5
10	MT6	2	2.5
	Average	3.5	3.65

Kesimpulan dari hasil eksperimen diatas, dari aspek *readability* 65.73 dimana jika dilihat dalam Tabel klasifikasinya, tergolong kedalam *Plain English* yang bisa dipahami oleh siswa dengan umur 13-15 tahun keatas. Sedangkan dari aspek *Computation Time*, diperoleh rata-rata 3.01 detik. Lalu dari aspek *Acceptability* memperoleh nilai rata-rata 3.5, *importance* dengan rata-rata 3.65, lalu nilai *relevance* 3.9, dan *Truthfulness* 4.07.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan serangkaian proses penelitian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Pengembangan sistem *Data-to-text* dengan pendekatan *time-series* sangat bermanfaat terutama dalam kasus pembangkitan cuaca dan kualitas udara, karena dengan menggunakan pendekatan *time-series* dapat menambah jumlah informasi yang disampaikan, yaitu informasi hasil prediksi dengan *exponential smoothing*. Machine *Learning* juga bermanfaat untuk mengatasi *missing value* pada data sehingga kualitas dari output yang dihasilkan dapat terjaga.
2. Dalam implementasinya, penggunaan bahsa R dalam membangun sistem *Data-to-text* adalah pilihan yang tepat. Alasannya, karena R memiliki banyak *package* yang sangat bermanfaat selama proses pengembangan terutama untuk analisis data. Kekurangannya, dalam proses pengembangan ini *User Interface* kurang fleksibel hanya bisa dibangun dengan *package* tertentu, (contoh:shinyR).
3. Dari hasil eksperimen, kualitas *readability* yang diperoleh dengan metode *Flesch Reading Ease* menyatakan bahwa teks yang dihasilkan merupakan *Plain Teks* yang dapat dipahami bahkan oleh siswa di umur 13-15 tahun. Sedangkan kualitas *relevance* dan *truthfulness* telah dinilai oleh *expert* dari AMG dan BMKG Bandung memperoleh hasil baik. Pada aspek *acceptability* dan *Importance* dari *user* didapatkan nilai cukup.

5.2 Saran

Diharapkan penelitian selanjutnya dalam pembangunan sistem *Data-to-text* dapat menggunakan teknik yang lebih baik dan mutakhir, seperti *Fuzzy sets*, *GMLP*, *CTP* dan lain-lain.

DAFTAR PUSTAKA

- Al-Hamami, A., & Al-Samadi, S. (2013). Enhancement of Missing Values Prediction and Estimation Using Data Mining Algorithm. *Journal of Emergin Trends in Computing and Information Science* .
- 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., & Kitteridge, 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 Annotation 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. *International Journal of Modern Engineering Research*, Vol 4. 1-5.
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations 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 Events 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

LAMPIRAN