

**PENGEMBANGAN SISTEM DATA-TO-TEXT (D2T) UNTUK
MEMBANGKITKAN BERITA PADA DATA UNSPECIFIC**

SKRIPSI

Diajukan untuk Memenuhi Bagian Dari
Syarat Memperoleh Gelar Sarjana Komputer
Program Studi Ilmu Komputer



Oleh:

Muhammad Ridwan

1403407

PROGRAM STUDI ILMU KOMPUTER
DEPARTEMEN PENDIDIKAN ILMU KOMPUTER
FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PENDIDIKAN INDONESIA
BANDUNG
2018

**PENGEMBANGAN SISTEM DATA-TO-TEXT (D2T) UNTUK
MEMBANGKITKAN BERITA PADA DATA UNSPECIFIC**

Oleh
Muhammad Ridwan
NIM 1403407

Sebuah Skripsi yang Diajukan untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Komputer di Fakultas Pendidikan Matematika
dan Ilmu Pengetahuan Alam

© Muhammad Ridwan 2018
Universitas Pendidikan Indonesia
Desember 2018

Hak Cipta Dilindungi Undang-Undang
Skripsi ini tidak boleh diperbanyak seluruhnya atau sebagian, dengan dicetak
ulang, difoto kopi atau cara lainnya tanpa izin dari penulis

PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi saya dengan judul “Pengembangan Sistem *Data-To-Text* (D2T) untuk Membangkitkan Berita pada Data *Unspecific*” ini beserta seluruh isinya adalah benar-benar karya saya sendiri. Saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika ilmu yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini saya siap menanggung resiko/sanksi apabila dikemudian hari ditemukan adanya pelanggaran etika keilmuan atau klaim dari pihak lain terhadap keaslian karya saya ini.

Bandung, Desember 2018
Yang Membuat Pernyataan,

Muhammad Ridwan
NIM 1403407

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah *subhanahu wa ta'ala* karena hanya dengan kehendak, berkat, serta karunia-Nya lah penulis dapat menyelesaikan skripsi yang berjudul “Pengembangan Sistem *Data-To-Text* (D2T) untuk Membangkitkan Berita pada Data *Unspecific*” ini dapat terselesaikan.

Penyusunan skripsi ini ditunjukan untuk memenuhi dan melengkapi salah satu syarat untuk penyusunan skripsi yang merupakan syarat untuk mendapatkan gelar sarjana komputer atas jenjang studi S1 pada Program Studi Ilmu Komputer Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam Universitas Pendidikan Indonesia.

Penulis menyadari bahwa dalam penyusunan proposal ini masih terdapat banyak kekurangan dan keterbatasan yang perlu disempurnakan. Oleh karena itu, penulis mengharapkan berbagai kritik dan saran dari para pecinta ilmu pengetahuan yang bersifat positif supaya skripsi ini dapat dikembangkan dengan lebih baik lagi. Penulis juga berharap skripsi ini dapat memberikan manfaat, dan menjadi amal jariyah baik untuk penulis sendiri, umumnya bagi para pengembang teknologi dan pecinta ilmu pengetahuan.

Bandung, Desember 2018

Penulis

UCAPAN TERIMA KASIH

Alhamdulillahirabilalamin, puji dan syukur kehadirat Allah *subhanahu wa ta'ala*. Yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis diberikan kelancaran dalam menyelesaikan penulisan skripsi ini. Dalam proses menyelesaikan penelitian dan penyusunan skripsi ini, peneliti banyak mendapat bimbingan, dorongan, serta bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini peneliti mengucapkan terimakasih serta penghargaan yang setinggi-tingginya, kepada:

1. Kedua orang tua serta adik penulis yaitu Bapak Sarip, Ibu Amah, dan Nurul Fitriani, yang selalu ada untuk memberikan doa dan dukungan, baik itu dukungan moral, materil maupun spiritual sehingga dapat memotivasi penulis dalam menyelesaikan skripsi ini.
2. Bapak Lala Septem Riza, M.T., Ph.D. selaku pembimbing I atas segala ilmu, tenaga, dan waktu yang dicurahkan untuk membimbing penulis demi terselesaiannya skripsi ini.
3. Ibu Enjun Junaeti, M.Si., selaku pembimbing II yang telah memberikan saran kepada penulis selama proses penyelesaian penelitian dan penulisan skripsi.
4. Bapak Eddy Prasetyo Nugroho, M.T., selaku Ketua Program Studi Ilmu Komputer.
5. Bapak Prof. Dr. H. Munir, M.IT., selaku Kepala Departemen Pendidikan Ilmu Komputer FPMIPA Universitas Pendidikan Indonesia.
6. Bapak Prof. Dr. Wawan Setiawan, M.Kom. dan Bapak *Eki* Nugraha, M.Kom., selaku dosen pembimbing akademik yang telah memberikan arahan juga bimbingan selama penulis menjalani perkuliahan.
7. Ibu Rosa Ariani Sukamto, MT. selaku dosen yang selalu membimbing, memberi arahan, motivasi, dan inspirasi bagi penulis selama penulis menjalani masa perkuliahan.
8. Bapak dan Ibu Dosen Prodi Pendidikan Ilmu Komputer dan Ilmu Komputer yang telah berbagi ilmu yang sangat bermanfaat kepada penulis.

9. Sahabat masa depan nikah yaitu Fidela, Reinaldy, Fikry, Faisal, Zulfikar, Zakka, Eagan, Agung dan Wiwi yang senantiasa memberikan dukungan, semangat, doa, canda dan tawa hingga ilmu kehidupan yang tak ternilai kepada penulis baik selama proses perkuliahan maupun selama proses penggerjaan skripsi ini.
10. Sahabat KKN Mekarwangi UPI 2017 yaitu Obin, Angga, Darryl, Sitii, Upit, Mute, Rini, Eci, Gina, dan Gita yang sudah banyak memberikan nasihat, semangat hingga pelajaran hidup dan bantuan dalam penulisan skripsi ini.
11. Ahmad Zainal Abidin selaku rekan penelitian yang telah melalui pahit manis penelitian ini bersama, serta selalu memberi dukungan, semangat dan bantuan kepada penulis.
12. Teman-teman kelas C2 2014, yang sama-sama berjuang dari awal perkuliahan dari awal hingga ke titik akhir.
13. Semua pihak yang telah membantu peneliti dalam menyelesaikan skripsi ini yang tidak dapat peneliti sebutkan satu persatu.
Semoga semua amal baik yang telah diberikan kepada penulis menjadi amal jariyah dan diganti dengan balasan yang berlipat dari Allah *subhanahu wa ta'ala*. Aamiin.

Bandung, Desember 2018

Muhammad Ridwan

PENGEMBANGAN SISTEM DATA-TO-TEXT (D2T) UNTUK MEMBANGKITKAN BERITA PADA DATA UNSPECIFIC

Oleh

Muhammad Ridwan — just.muhammadridwan@gmail.com

1403407

ABSTRAK

Sistem *Data-to-Text* menjadi salah satu pilihan untuk menerjemahkan data *non-lingistik* kedalam bentuk textual. Namun seiring dengan perkembangan teknologi, beragamnya bidang dari suatu data dan beragamnya pengguna menjadi salah satu fokus yang harus diperhatikan dalam pengembangan sistem *Data-to-Text*. Penelitian ini bertujuan untuk mengembangkan sistem *Data-to-Text* dengan masukan berupa data *unspecific*, sebagai solusi agar sistem *Data-to-Text* dapat menerima masukan berupa data dari bidang atau domain apapun, baik data tersebut memiliki identitas berupa informasi *header*, tipe data, *rule* ataupun tidak. Maka digunakan pendekatan *Fuzzy Rule* untuk menginterpretasikan data *unspecific* tersebut. Selain itu digunakan beberapa algoritma *Machine Learning* seperti *Gradient Descent*, dan analisis lainnya seperti *Exponential Smoothing*, *Knuth-Morris-Pratt*, *Statistical tools* dan *Pearson Correlation Coefficient*. Sistem yang dikembangkan dapat menghasilkan informasi berupa ringkasan data, informasi data terkini dan informasi prediksi. Pengembangan sistem dilakukan dalam bahasa pemrograman R dengan memanfaatkan beberapa *packages* yang tersedia. Eksperimen dilakukan dengan mengukur tingkat *Readibility* dari berita yang dibangkitkan, *Computation Time*, dan membandingkan hasil dengan penelitian terkait. Hasil eksperimen menunjukkan bahwa informasi yang dihasilkan terbukti merepresentasikan data yang diberikan dan dapat dipahami oleh tingkat siswa pada tingkat sekolah dasar sekalipun, serta waktu komputasi cukup baik. Sistem ini mampu menhasilkan informasi berdasarkan data meteorologi, data klimatologi, data keuangan, dan data *time series* lainnya.

Kata Kunci— *Data-to-Text; Natural Language Processing; Natural Language Generation; Machine Learning; General purpose; Unspecific Corpora; Fuzzy Rule-based; Crisp Rule-based; Time-series Analysis; Exponential Smoothing; Linear Model; Gradient Descent; Kunth-morris-pratt; Pearson Correlation Coefficient*

DEVELOPMENT OF DATA-TO-TEXT (D2T) SYSTEMS TO GENERATE NEWS BASED ON UNSPECIFIC DATA

Arranged by

Muhammad Ridwan — just.muhammadridwan@gmail.com

1403407

ABSTRACT

The Data-to-Text system is an option for translating non-linguistic data into textual form. But along with the development of technology, the diverse fields of data and the variety of users have become one of the focuses that must be considered in the development of Data-to-Text systems. This study aims to develop a Data-to-Text system with input in the form of unspecific data, as a solution so that the Data-to-Text system can receive input in the form of data from any field or domain, both data has identity in the form of header information, data types, rules or not. Then the Fuzzy Rule approach is used to interpret the unspecific data. In addition, several Machine Learning algorithms such as Gradient Descent were used, and other analyzes such as Exponential Smoothing, Knuth-Morris-Pratt, Statistical tools and Pearson Correlation Coefficient. The system developed can produce information in the form of summary data, current data information and predictive information. The development of the system was written using the R programming language by utilizing several available packages. Experiments are carried out by measuring the level of Readability of the news generated, Computation Time, and comparing the results with related research. The experimental results show that the information produced is proven to represent the data provided and can be understood by the level of students at the elementary school level though, and computing time is quite good. This system is able to produce information based on meteorological data, climatological data, financial data, and other time series data.

Keywords — *Data-to-Text; Natural Language Processing, Natural Language Generation; Machine Learning; General purpose; Unspecific Corpora; Fuzzy Rule-based; Crisp Rule-based; Time-series Analysis; Exponential Smoothing; Linear Model; Gradient Descent; Kunth-morris-pratt; Pearson Correlation Coefficient*

DAFTAR ISI

KATA PENGANTAR	i
UCAPAN TERIMA KASIH	ii
ABSTRAK	iv
<i>ABSTRACT</i>	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	4
1.3. Tujuan Penelitian.....	4
1.4. Manfaat Penelitian.....	4
1.5. Batasan Masalah.....	5
1.6. Sistematika Penulisan	5
BAB II KAJIAN PUSTAKA	7
2.1 Pengertian <i>Natural Language</i>	7
2.2 Pengertian <i>Natural Language Processing</i>	8
2.3 Pengertian <i>Natural Language Generation</i>	10
2.4 Arsitektur sistem <i>Data-to-Text</i>	12
2.4.1. Arsitektur Data-to-Text oleh Reiter (2011).....	13
2.4.2. Arsitektur Data-to-Text oleh Putra et al., (2017)	14
2.4.3. Arsitektur Data-to-Text oleh Abidin et al., (2018).....	23
2.5 Penelitian Terkait sistem <i>Data-to-Text</i>	27
2.6 <i>Machine Learning</i>	28
2.6.1. <i>Supervised Learning</i>	29
2.6.2. <i>Unsupervised Learning</i>	31
2.6.3. <i>Semi Supervised Learning</i>	32
2.6.4. <i>Reinforcement Learning</i>	32
2.7. <i>Time-series Data</i>	33
2.8. <i>Exponential Smoothing</i>	34

2.9.	<i>String Matching</i>	36
2.10.	Logika <i>Fuzzy</i>	38
2.11.	Pemrograman R.....	39
2.11.1.	Model Data dalam R	41
2.11.2.	Contoh Kode Program Bahasa R	42
2.12.	Package Dalam R	46
	BAB III METODE PENELITIAN	48
3.1.	Desain Penelitian	48
3.2.	Metode Penelitian	50
3.3.	Alat dan Bahan Penelitian.....	51
	BAB IV HASIL PENELITIAN DAN PEMBAHASAN	52
4.1.	Pengumpulan Data.....	52
4.1.1.	Data Nilai Tukar Mata Uang Asing.....	52
4.1.2.	Data Klimatologi	54
4.1.3.	Data Kualitas Udara.....	55
4.1.4.	Data Partikel Udara Kota Beijing	56
4.2.	Model Sistem <i>Data-to-text</i>	57
4.2.1.	Model Komputasi untuk <i>Unspecific Data Handling</i>	59
4.2.2.	Model Komputasi untuk <i>Signal Analysis</i>	62
4.2.3.	Model Komputasi untuk <i>Data Interpretation</i>	74
4.2.4.	Model Komputasi untuk <i>Document Planning</i>	85
4.2.5.	Model Komputasi untuk <i>Microplanning and Realisation</i>	95
4.2.6.	Hasil Keluaran Sistem	101
4.3.	Pengembangan Sistem <i>Data-to-text Unspecific News Generator</i>	101
4.3.1	Analisis Sistem D2T UNG.....	102
4.3.2	Desain Sistem D2T UNG.....	103
4.3.3	Implementasi Sistem D2T UNG	105
4.3.4	Testing Sistem D2T UNG.....	132
4.3.5	Panduan Penggunaan Aplikasi UNG	133
4.4.	Rancangan Eksperimen.....	134
4.5.	Hasil dan Pembahasan Hasil Eksperimen.....	137

4.7.1.	Hasil dan Pembahasan Hasil Eksperimen Data Nilai Tukar Mata Uang Asing.....	137
4.7.2.	Hasil dan Pembahasan Hasil Eksperimen Data Klimatologi	149
4.7.3.	Hasil dan Pembahasan Hasil Eksperimen Data Kualitas Udara....	161
4.7.4.	Hasil dan Pembahasan Hasil Eksperimen Data Partikel Udara	173
4.6.	Perbandingan dengan Penelitian Sebelumnya.....	184
BAB V KESIMPULAN DAN SARAN.....		189
5.1.	Kesimpulan.....	189
5.2.	Saran	190
DAFTAR PUSTAKA		191

DAFTAR GAMBAR

Gambar 2.1 Arsitektur D2T oleh (Reiter, 2011)	13
Gambar 2.2 Arsitektur sistem DWP (Putra <i>et al.</i> , 2017)	15
Gambar 2.3 Contoh implementasi <i>Signal Analysis</i> (Putra <i>et al.</i> , 2017)	16
Gambar 2.4 Contoh implementasi data interpretation Rainfall DWP (Putra <i>et al.</i> , 2017)	17
Gambar 2.5 Contoh Content Determination Significant Event Message DWP (Putra <i>et al.</i> , 2017).....	19
Gambar 2.6 Contoh Target Text DWP (Putra <i>et al.</i> , 2017)	19
Gambar 2.7 Contoh skema dalam bentuk <i>tree</i> berdasarkan <i>Target Text</i> DWP (Putra <i>et al</i> , 2017).....	20
Gambar 2.8 Contoh implementasi Lexicalisation tren DWP (Putra et al., 2017).21	21
Gambar 2.9 Contoh <i>Simple Conjunction Referring to Contrast Value</i> (Putra <i>et al.</i> , 2017)	21
Gambar 2.10 Contoh Referring Expression Generation (Putra <i>et al.</i> , 2017)	22
Gambar 2.11 Contoh implementasi Structure Realisation DWP (Putra <i>et al.</i> , 2017)	23
Gambar 2.12 Arsitektur D2T untuk data <i>streaming</i> (Abidin <i>et al.</i> , 2018).....	24
Gambar 2.13 <i>Realtime animation for check file</i> (Abidin <i>et al.</i> , 2018)	25
Gambar 2.14 <i>Execute D2T_Main.R</i> (Abidin <i>et al.</i> , 2018).....	26
Gambar 2.15 <i>Read and Remove Dataset in R</i> (Abidin <i>et al.</i> , 2018).....	26
Gambar 2.16 <i>Write Result JSON and csv in R</i> (Abidin <i>et al.</i> , 2018)	26
Gambar 2.17 <i>Get JSON in AJAX</i> (Abidin <i>et al.</i> , 2018)	27
Gambar 2.18 Contoh <i>Supervised Learning</i> pada pengenalan koin.....	30
Gambar 2.19 Contoh <i>Unsupervised Learning</i> dalam pengenalan koin.	32
Gambar 2.20 Pemberian prefix pada pattern 'ATATG' (Rahman, 2017)	37
Gambar 2.21 Skenario <i>Knuth-Morris-Pratt</i> pada <i>String Matching</i>	37
Gambar 2.22 Contoh himpunan <i>Crisp</i> pada kasus umur (Putra <i>et al.</i> , 2017)	38
Gambar 2.23 Contoh himpunan <i>Fuzzy</i> pada kasus umur (Putra <i>et al.</i> , 2017).....	39
Gambar 2.24 Logo bahasa pemrograman R	40
Gambar 2.25 Antarmuka R Graphical User Interface (RGui).....	41
Gambar 2.26 Model data dalam pemrograman R (Budiharto, 2013).	42

Gambar 2.27 Operator <i>concatenate</i> dalam R	42
Gambar 2.28 Menampilkan data pertama hingga ke-dua dalam R	43
Gambar 2.29 Penggunaan fungsi <i>sum</i> dalam R.....	43
Gambar 2.30 Penggunaan fungsi <i>concatenate</i> untuk <i>string</i> dalam R.....	43
Gambar 2.31 Pembuatan matriks dalam R.....	43
Gambar 2.32 Contoh <i>Visualisasi</i> grafis dalam R	44
Gambar 2.33 Contoh perulangan dalam R	45
Gambar 2.34 Contoh implementasi <i>decission</i> dalam R	45
Gambar 2.35 Contoh fungsi dalam R	46
Gambar 2.36 Contoh Instalasi <i>Package</i> dalam R	47
Gambar 2.37 Cara menggunakan <i>package</i> yang sudah diinstal	47
Gambar 3.1 Desain Penelitian Sistem D2T.....	48
Gambar 3.2 Model Linear <i>Sequential Model</i> (Pressman, 2001b)	50
Gambar 4.1 Model <i>Data-to-text</i> untuk Data <i>Unspecific</i>	58
Gambar 4.2 Penamaan <i>header</i> pada Proses <i>Unspecific Data Handler</i>	60
Gambar 4.3 Contoh penentuan <i>trend</i> pada data <i>dummy</i> untuk parameter pertama	65
Gambar 4.4 Hasil plot parameter ke-dua pada data <i>dummy</i> , warna hijau merepresentasikan kenaikan ekstrem, dan warna merah merepresentasikan penurunan ekstrem	67
Gambar 4.5 Sinyal <i>repeated event</i> ditandai dengan garis biru	69
Gambar 4.6 Hasil plot parameter ke-tiga data <i>dummy</i> , garis kuning menandakan pola data yang sama	71
Gambar 4.7 Hasil plot data <i>dummy</i> yang menunjukkan hubungan linear antara parameter pertama dan ke-dua.....	73
Gambar 4.8 <i>Linguistic variable for trend description</i> (Castillo-Ortega et al., 2014)	77
Gambar 4.9 <i>Unspecific Fuzzy Membership Function</i>	77
Gambar 4.10 Himpunan <i>fuzzy</i> dari <i>corpus</i> temperatur pada penelitian DWP (Putra et al., 2017).....	79
Gambar 4.11 Himpunan <i>fuzzy</i> Param1 yang dihasilkan oleh <i>unspecific rule generator</i>	80

Gambar 4.12 <i>Routine Message</i> untuk ringkasan Data.....	86
Gambar 4.13 <i>Significant Event Message</i> untuk ringkasan data	86
Gambar 4.14 Contoh <i>academic writing task</i> pada IELTS	88
Gambar 4.15 <i>Initial Corpus</i> untuk ringkasan data.....	89
Gambar 4.16 Skema teks untuk ringkasan data.....	89
Gambar 4.17 Struktur Pohon untuk teks ringkasan data.....	90
Gambar 4.18 <i>Routine Message</i> untuk data terkini.....	90
Gambar 4.19 <i>Significance Event Message</i> untuk data terkini	91
Gambar 4.20 <i>Initial Corpus</i> untuk pesan data terkini.....	91
Gambar 4.21 Struktur teks data terkini	92
Gambar 4.22 Pohon struktur untuk teks data terkini	92
Gambar 4.23 <i>Routine Message</i> untuk Prediksi data	92
Gambar 4.24 <i>Significant Event Message</i> untuk Prediksi data.....	93
Gambar 4.25 <i>Initial Corpus</i> untuk Prediksi Data	94
Gambar 4.26 Struktur teks untuk prediksi data	94
Gambar 4.27 Pohon struktur prediksi data.....	95
Gambar 4.28 <i>Rule</i> penginterpretasian IVL	98
Gambar 4.29 Skema untuk mendeskripsikan pesan <i>Trend Description</i>	98
Gambar 4.30 <i>ProgressiveChange Corpus</i>	99
Gambar 4.31 Contoh Phrase Aggregation (Putra <i>et al.</i> , 2017).....	99
Gambar 4.32 Struktur file sistem UNG	104
Gambar 4.34 Antarmuka sistem <i>Unspecific News Generator</i> (UNG).....	105
Gambar 4.34 Proses <i>Unspecific Data Handler</i>	106
Gambar 4.36 Fungsi <i>data description</i> dalam proses <i>Unspecific Data Handler</i> ..	107
Gambar 4.37 File <i>datadescription.csv</i> pada folder <i>Config</i>	107
Gambar 4.38 <i>Statistical Summary Function</i>	109
Gambar 4.39 <i>Trend Analysis function</i>	110
Gambar 4.40 <i>ResumeEventExtreme function</i>	111
Gambar 4.41 <i>ResumeRepeatedAnalysis function</i>	112
Gambar 4.42 <i>PredictDataset function</i>	113
Gambar 4.43 <i>MotifDiscoveryAnalysis function</i>	114
Gambar 4.44 Implementasi algoritma KMP (Rahman, 2017)	115

Gambar 4.45 Penentuan prefix pada algoritma KMP (Rahman, 2017).....	115
Gambar 4.46 <i>CorrelationAnalysis function</i>	116
Gambar 4.47 <i>CorrelationRoutineMessage function</i>	116
Gambar 4.48 CorrelationSignificantMsgContentDetermination function.....	117
Gambar 4.49 <i>Fuzzy Corpus for Data Interpretation</i>	118
Gambar 4.50 <i>DataInterpreterAdjective function</i>	118
Gambar 4.51 <i>Fuzzy Membership Function</i>	119
Gambar 4.52 <i>Crisp Membership Function</i>	120
Gambar 4.53 <i>UnspecificFuzzyGenerator Function</i>	121
Gambar 4.54 Implementasi perhitungan PSI untuk kualitas udara (Putra <i>et al.</i> , 2017)	122
Gambar 4.55 Implementasi <i>Content Determination</i> untuk <i>Repeated Event</i>	124
Gambar 4.56 Implementasi <i>Content Determination</i> untuk <i>Extreme Event</i>	125
Gambar 4.57 Implementasi <i>Content Determination</i> untuk pendekripsi <i>String Matching</i>	126
Gambar 4.58 Implementasi <i>Content Determination</i> untuk korelasi antar parameter	127
Gambar 4.59 Implementasi <i>Content Determination</i> untuk <i>Current Text</i>	127
Gambar 4.60 Implementasi <i>Lexicalisation</i> untuk format range tanggal.....	128
Gambar 4.61 Implementasi <i>Aggregation</i> untuk mengelompokan sinyal pada perbandingan data	129
Gambar 4.62 Implementasi <i>Referring Expression Generation</i> untuk menentukan intro secara <i>random</i>	130
Gambar 4.62 Implementasi <i>Referring Expression Generation</i> untuk <i>Time Descripiton</i>	131
Gambar 4.64 Implementasi <i>Structure Realisation</i>	131
Gambar 4.65 Hasil eksperimen pertama menggunakan data nilai tukar tanpa menggunakan <i>header</i> (CE_NH)	138
Gambar 4.66 Eksperimen kedua menggunakan data nilai tukar dengan menggunakan <i>header</i> (CE_WH)	139
Gambar 4.67 Eksperimen ke-tiga menggunakan data nilai tukar dengan kustomisasi <i>corpus</i> (CE_WHM).....	140

Gambar 4.68 Plot <i>Representative Text</i> untuk parameter US.Dollar	143
Gambar 4.69 Plot <i>Representative Text</i> untuk parameter Japan yen (JPY)	144
Gambar 4.70 Hasil plot <i>Representative Text</i> untuk parameter <i>Singapore Dollar</i> (SGD).....	145
Gambar 4.71 Hasil plot <i>Representative Text</i> untuk parameter <i>Hong Kong Dollar</i> (HKD)	146
Gambar 4.72 Hasil plot <i>Representative Text</i> untuk parameter <i>Canadian Dollar</i> (CAD).....	147
Gambar 4.73 Hasil plot <i>Representative Text</i> untuk korelasi parameter.....	148
Gambar 4.74 Hasil eksperimen ke-empat menggunakan data klimatologi tanpa menggunakan <i>header</i> (CL_NH)	150
Gambar 4.75 Hasil eksperimen ke-lima menggunakan data klimatologi dengan menggunakan <i>header</i> (CL_WH)	151
Gambar 4.76 Hasil eksperimen ke-enam menggunakan data klimatologi dengan kustomisasi <i>corpus</i> (CL_WHM)	153
Gambar 4.77 Hasil plot <i>Representative Text</i> untuk parameter CloudCoverage..	156
Gambar 4.78 Plot <i>Representative Text</i> untuk parameter Temperature	157
Gambar 4.79 Hasil plot <i>Representative Text</i> untuk parameter WindSpeed.....	158
Gambar 4.80 Hasil plot <i>Representative Text</i> untuk parameter WindDirection... <td>159</td>	159
Gambar 4.81 Hasil plot <i>Representative Text</i> untuk parameter Rainfall	160
Gambar 4.82 Hasil plot <i>Representative Text</i> untuk korelasi parameter data klimatologi	161
Gambar 4.83 Hasil eksperimen ke-tujuh menggunakan data kualitas udara tanpa menggunakan <i>header</i> (AQ_NH)	162
Gambar 4.84 Hasil eksperimen ke-delapan menggunakan data kualitas udara dengan menggunakan <i>header</i> (AQ_WH).....	164
Gambar 4.85 Hasil eksperimen ke-sembilan menggunakan data kualitas udara dengan kustomisasi <i>corpus</i> (AQ_WHM).....	166
Gambar 4.86 Hasil plot <i>Representative Text</i> untuk parameter CO	169
Gambar 4.87 Plot <i>Representative Text</i> untuk parameter NO	170
Gambar 4.88 Hasil plot <i>Representative Text</i> untuk parameter PM10	171

Gambar 4.89 Hasil plot <i>Representative Text</i> untuk korelasi parameter data kualitas udara.....	172
Gambar 4.90 Hasil eksperimen ke-sepuluh menggunakan data partikel udara tanpa menggunakan <i>header</i> (BPM_NH)	174
Gambar 4.91 Hasil eksperimen ke-sebelas menggunakan data partikel udara dengan menggunakan <i>header</i> (BPM_WH)	175
Gambar 4.92 Hasil eksperimen ke-dua belas menggunakan data partikel udara dengan kustomisasi <i>corpus</i> (BPM_WHM)	176
Gambar 4.93 Hasil plot <i>Representative Text</i> untuk parameter LWS	179
Gambar 4.94 Plot <i>Representative Text</i> untuk parameter IS	180
Gambar 4.95 Hasil plot <i>Representative Text</i> untuk parameter IR	181
Gambar 4.96 Hasil plot <i>Representative Text</i> untuk korelasi parameter data kualitas udara.....	183
Gambar 4.97 Hasil plot Representative Text untuk parameter DEWP, TEMP dan PRES	184

DAFTAR TABEL

Tabel 2.1 Penelitian terkait D2T dan NLG	27
Tabel 2.2 Contoh penggunaan <i>Exponential Smoothing</i>	35
Tabel 4.1 Data Nilai Tukar Mata Uang Asing	53
Tabel 4.2 Kutipan data klimatologi	54
Tabel 4.3 Kutipan data Kualitas Udara	56
Tabel 4.4 Kutipan data Partikel Udara Kota Beijing	57
Tabel 4.5 Contoh <i>data description</i> pada file <i>datadescription.csv</i>	61
Tabel 4.6 Contoh data <i>dummy</i> untuk kasus sederhana	63
Tabel 4.7 Hasil pendektsian sinyal <i>statistical summary</i> untuk kasus data sederhana	64
Tabel 4.8 Hasil pendektsian sinyal <i>statistical summary</i> untuk kasus data partikel udara.....	64
Tabel 4.9 Ringkasan beberapa data yang akan digunakan pada proses selanjutnya	66
Tabel 4.10 Hasil <i>signal analysis</i> untuk <i>extreme event</i> pada kasus data <i>dummy</i>	67
Tabel 4.11 Hasil pendektsian sinyal <i>Comparison</i>	68
Tabel 4.12 Hasil pendektsian sinyal <i>repeated event</i>	70
Tabel 4.13 Hasil Prediksi data untuk contoh kasus data klimatologi	70
Tabel 4.14 Hasil pendektsian sinyal <i>String Matching</i>	72
Tabel 4.15 Hasil <i>signal analysis</i> untuk korelasi paramter pada data <i>dummy</i>	73
Tabel 4.16 Hasil pendektsian sinyal menggunakan <i>Pearson Correlation</i>	74
Tabel 4.17 Contoh <i>data description</i> untuk menginterpretasikan prameter TEMP pada data partikel udara	78
Tabel 4.18 Nilai keanggotaan parameter TEMP	79
Tabel 4.19 Nilai keanggotaan dari ringkasan data untuk parameter Param1 pada data <i>dummy</i>	81
Tabel 4.20 Hasil <i>Data Interpretation</i> contoh kasus parameter Param1 pada data <i>dummy</i>	81
Tabel 4.21 Parameter Correlation Crisp Membership Function (de Vaus, 2002). .	83
Tabel 4.22 Hasil proses interpretasi data untuk sinyal korelasi parameter.	85
Tabel 4.23 Hasil <i>Content Determination Summary</i> untuk data klimatologi.....	87

Tabel 4.24 Hasil <i>Content Determination</i> untuk data terkini	91
Tabel 4.25 Hasil <i>Content Determination</i> untuk Prediksi data.....	94
Tabel 4.26 Hasil interpretasi data untuk contoh kasus kualitas udara	97
Tabel 4.27 Indeks interpretasi kualitas udara	97
Tabel 4.28 Nilai kontras dalam proses agregasi dengan Simple Conjunction (Putra <i>et al.</i> , 2017).....	100
Tabel 4.29 Hasil akhir untuk contoh kasus data partikel udara.....	101
Tabel 4.30 <i>Data description</i> untuk contoh kasus data klimatologi.	108
Tabel 4.31 <i>Data description</i> untuk contoh kasus data nilai tukar	108
Tabel 4.32 <i>Air Quality Crisp Membership Value</i>	123
Tabel 4.33 Tabel Rencana pengujian <i>blackbox</i> sistem UNG	132
Tabel 4.34 hasil pengujian <i>blackbox</i> sistem UNG.....	133
Tabel 4.35 <i>Flecs Reading Ease</i>	135
Tabel 4.36 Rancangan Eksperimen Sistem UNG	136
Tabel 4.37 Cuplikan data nilai tukar tanpa menggunakan <i>header</i> (CE_NH)	137
Tabel 4.38 Kustomisasi <i>data description</i> pada eksperimen ke-tiga	140
Tabel 4.39 Hasil evealuasi <i>Readability</i> dengan data kurs.....	141
Tabel 4.40 Hasil evaluasi <i>Computation Time</i> dengan data kurs	141
Tabel 4.41 Hasil evaluasi <i>Responsiveness</i>	142
Tabel 4.42 Cuplikan data klimatologi tanpa menggunakan <i>header</i> (CE_NH) ...	149
Tabel 4.43 Kustomisasi <i>data description</i> pada eksperimen ke-enam.....	152
Tabel 4.44 Hasil evealuasi <i>Readability</i> dengan data klimatologi.....	154
Tabel 4.45 Hasil evaluasi <i>Computation Time</i> dengan data klimatologi	154
Tabel 4.46 Hasil evaluasi <i>Unspecific Handling</i>	155
Tabel 4.47 Cuplikan data klimatologi tanpa menggunakan <i>header</i> (AQ_NH)...	162
Tabel 4.48 Kustomisasi <i>data description</i> pada eksperimen ke-sembilan	165
Tabel 4.49 Hasil evealuasi <i>Readability</i> dengan data kualitas udara.....	166
Tabel 4.50 Hasil evaluasi <i>Computation Time</i> dengan data kualitas udara.....	167
Tabel 4.51 Hasil evaluasi <i>Unspecific Handling</i>	167
Tabel 4.52 Cuplikan data kualitas udara tanpa menggunakan <i>header</i> (BPM _NH)	173
Tabel 4.53 Kustomisasi <i>data description</i> pada eksperimen ke-enam.....	176

Tabel 4.54 Hasil evaluasi <i>Readability</i> dengan data partikel udara	177
Tabel 4.55 Hasil evaluasi <i>Computation Time</i> dengan data kualitas udara.....	178
Tabel 4.56 Hasil evaluasi <i>Unspecific Handling</i>	178
Tabel 4.57 Hasil perbandingan dengan penelitian sebelumnya	184
Tabel 4.58 Perbandingan dengan penelitian sebelumnya	188

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dewasa ini, kebutuhan manusia akan informasi semakin tinggi. Perkembangan teknologi dan ilmu pengetahuan menyebabkan ketersediaan informasi meningkat, terutama informasi berupa data non-linguistik atau data numerik. Dalam merepresentasikan sebuah data numerik atau non-linguistik seringkali digunakan *chart* maupun grafik untuk merepresentasikannya, disamping itu penjelasan data maupun grafik secara singkat dalam bentuk textual dapat menjadi salah satu alternatif sehingga penyajian informasi data numerik dapat lebih mudah dipahami. Sehingga para peneliti dan pengembang berlomba-lomba untuk mengembangkan aplikasi-aplikasi yang mampu menghasilkan informasi dalam bentuk text dengan *input* data *non-linguistik* atau data numerik. Salah satunya yaitu aplikasi atau sistem *Data-to-text* (D2T) yang diperkenalkan oleh Reiter (2011).

Sistem D2T ini dapat menerima berbagai masukan berupa data non linguistik mulai dari data numerik, *event logs*, maupun data yang dihasilkan dari sensor sekalipun. Dengan kemampuannya yang dapat menerjemahkan data numerik kedalam data berbentuk textual secara otomatis, membuat sistem D2T ini menjadi salah satu bagian dari sistem *Natural Language Generation* (NLG) dimana D2T ini dapat menerjemahkan data ke dalam teks dengan mengasumsikan bahwa data yang digunakan pada dasarnya benar dan akurat (Gkatzia *et al.*, 2017). Karena arsitekturnya yang mirip dengan arsitektur sistem NLG, membuat sistem D2T ini sangat erat dengan proses linguistik dan juga proses analisis data, maka Reiter (2011) memaparkan bahwa setidaknya ada empat langkah dalam tahapan pembangunan sistem D2T, yaitu: (*signal analysis, data interpretation, document planning, microplanning* dan *realisation*).

Sudah banyak implementasi sistem D2T yang menjadi solusi dalam menyediakan informasi textual dalam beberapa bidang. Contohnya pada bidang peramalan cuaca, yaitu aplikasi *Data-to-text Weather Prediction* (DWP) yang dapat menghasilkan ringkasan cuaca selama satu bulan dan prediksi cuaca dalam

bentuk berita dari masukan berupa data klimatologi dan data kualitas udara selama satu tahun (Putra *et al.*, 2017). Lalu, ada *Forecast Generator* (FOG) yang diperkenalkan oleh (Kittredge & Driedger, 1994), aplikasi tersebut dapat mengkonversi peta cuaca menjadi ramalan dalam bentuk kalimat dengan pengolahan bahasa alami. Selain itu, ada *SumTime-Mousam* yang diperkenalkan oleh (Sripada & Reiter, 2003), aplikasi ini dapat menghasilkan ramalan cuaca laut teksual untuk rig minyak lepas pantai. Contoh lainnya yaitu pada bidang kesehatan, yaitu *BABYTALK family System* yang diperkenalkan oleh (Portet *et al.*, 2009), aplikasi ini mampu membuat sebuah ringkasan peristiwa yang terjadi selama 45 menit dari sinyal psikologis kontinyu dan diskrit, seperti pengaturan peratalatan dan pemberian obat dalam bentuk kalimat. Selain itu Hunter *et al.*, (2011) memperkenalkan sistem yang dapat menghasilkan ringkasan dari pergantian keperawatan yang berasal dari pencatatan pasien elektronik di Neonatal Intensive Care Unit (NICU). Pada bidang ekonomi, terdapat *Knowledge-Based Report Generator* yang diperkenalkan oleh (Kukich, 1983) yang dapat mengkonversi data stok produk (non-linguistik) menjadi laporan stok pada suatu pasar. Ada juga sistem D2T yang digunakan untuk menerima masukan berupa data *streaming* (Abidin *et al.*, 2018). Mengingat luasnya penerapan sistem D2T tersebut, membuat sistem ini menjadi salah satu pilihan yang dapat digunakan untuk merepresentasikan data *non-linguistik* agar lebih mudah dipahami disamping penggunaan *chart* maupun grafik, tanpa menghilangkan makna asli yang terkandung pada data tersebut.

Setidaknya ada dua masalah utama yang harus diperhatikan dalam pembangunan sebuah *corpus* pada sistem D2T. Pertama, beragamnya jenis informasi yang disimpan dalam sebuah *corpus*, sehingga aspek sumber daya menjadi salah satu faktor yang penting dan harus diperhitungkan. Kedua, beragamnya jenis user, sehingga corpus yang dibangun harus bisa mencakup berbagai kebutuhan dari setiap user (Soehn *et al.*, 2007). Maka pada penelitian ini penulis akan mengembangkan sebuah sistem D2T yang dapat menerima input berupa data *unspecific* atau data yang tidak terikat pada suatu bidang apapun, baik data tersebut bertipe *numerical* maupun *categorical*, serta memiliki identitas berupa informasi header, kategori, ataupun tidak, sehingga corpus dan keluaran yang dihasilkan bersifat umum atau *unspecific* sesuai dengan informasi yang terkandung

dalam data tersebut. Karena seringkali pembangunan sistem D2T ini hanya terdapat untuk satu bidang spesifik seperti yang sudah penulis jelaskan pada bagian sebelumnya. Hal inilah yang menjadi latar belakang pengembangan sistem D2T pada penelitian ini, sehingga sistem D2T yang dibangun diharapkan mampu menerima masukan data *non-linguistik* dalam bidang apapun, tidak terbatas pada suatu bidang spesifik, dapat menerima masukan baik data tersebut memiliki informasi berupa *header* atau tidak, serta mampu menganalisis keterkaitan setiap parameter pada data masukan.

Data yang digunakan dalam penelitian ini meliputi data nilai tukar mata uang asing terhadap rupiah (kurs), data klimatologi, data kualitas udara, dan data partikel udara Kota Beijing. Untuk mengolah data yang bersifat eksak tersebut secara manual tentu saja dibutuhkan sumber daya manusia dan waktu yang cukup untuk menganalisa dan menarik informasi dari data tersebut, sehingga dengan adanya D2T ini sumber daya tersebut bias kita minimalisir. Karena pada dasarnya di era *Big Data* ini ketersediaan data semakin meningkat, mudah diakses, variatif, dan juga dinamis. Namun jika tidak didampingi dengan sebuah sistem yang dapat mengelola data tersebut sehingga informasi yang diperoleh mudah dipahami maka akan dirasa sangat sulit jika kita harus menganalisis data tersebut secara manual. Maka tidak heran sistem D2T ini bisa menjadi suatu solusi yang dapat mengurangi sumber daya, dan mempermudah dalam penyampaian serta analisis suatu data khusunya pada data yang bersifat *unspecific*.

Untuk membangun sistem D2T yang dapat menerima masukan data *unspecific*, maka digunakan beberapa penerapan *Machine Learning* seperti *Gradient Descent*, lalu diterapkan beberapa fitur untuk analisis data seperti *Statistical Tools*, *Time-Series Analysis*, *Exponential Smoothing*, *Linear Model*, *Knuth Morris Pratt* (KMP), *Pearson Correlation*, dan metode lainnya seperti *Fuzzy Membership Function* dan *Crisp Membership Function* untuk menginterpretasikan data masukan. Selain itu, untuk mengefisiensikan *Development Time*, penulis menggunakan beberapa *package* 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 membangkitkan berita pada data *unspecific* dengan menggunakan pendekatan *Time Series* dan *Machine Learning*?
2. Bagaimana proses implementasi sistem *Data-to-text* untuk data *unspecific* dalam R?
3. Bagaimana eksperimen dan hasil eksperimen dari sistem *Data-to-text* yang dikembangkan?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penelitian ini adalah sebagai berikut:

1. Mengembangkan model sistem *Data-to-text* untuk membangkitkan berita pada data *unspecific* dengan menggunakan pendekatan *Time Series* dan *Machine Learning*.
2. Mengimplementasi model *Data-to-text* untuk data *unspecific* menggunakan bahasa pemrograman R .
3. Menganalisis kualitas sistem dengan melakukan eksperimen, dan pembahasan hasil eksperimen

1.4. Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Diharapkan dapat menambahkan pengetahuan tentang sistem *Data-to-text*, *Natural Language Processing*, *Time-Series* serta penerapannya dalam membangkitkan bahasa alami untuk mendeskripsikan data *unspecific*.
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*.

1.5. Batasan Masalah

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

1. Pembangunan sistem *Data-to-text* dengan pendekatan *Time Series* ini hanya didasarkan pada data *numerical* dan *categorical* yang berbentuk tabel dan eksak dengan format waktu “mm/dd/yyyy hh:mm”.
2. Pembangunan sistem *Data-to-text* untuk data *unspecific* ini menggunakan bahasa pemrograman R, HTML, dan JavaScript.

1.6. Sistematika Penulisan

Sistematika penulisan skripsi ini diuraikan menjadi lima bab, yaitu:

BAB I PENDAHULUAN

BAB I menjelaskan mengenai latar belakang dilakukannya penelitian, dimana pengembangan dilakukan dikarenakan model sistem *Data-to-Text* penelitian sebelumnya yaitu DWP (Putra *et al.*, 2017) tidak dapat menerima masukan berupa data *unspecific*, dan *categorical*. Bab ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian yang akan dilakukan, manfaat penelitian dan sistematikan penulisan.

BAB II TINJAUAN PUSTAKA

BAB II berisi tentang kajian pustaka yang dilakukan penulis yang ditujukan untuk menggali pengetahuan mengenai ilmu yang terkait dengan penelitian dan berujung pada pemahaman yang cukup untuk melakukan penelitian. Bab ini 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*, *Machine Learning*, *Time-series*, *String Matching*, *R Programming*, dan lainnya.

BAB III METODOLOGI PENELITIAN

BAB III terdiri dari langkah-langkah yang akan dilakukan dalam penelitian. Terdiri dari desain penelitian dari sistem D2T yang akan dikembangkan, metode pengembangan perangkat lunak D2T, serta alat dan bahan yang digunakan dalam penelitian D2T.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

BAB IV berisi pengumpulan data yang terdiri dari data kurs, data klimatologi, data kualitas udara, dan data partikel udara. Pengembangan model sistem D2T yang terdiri dari *Unspecific Data Handling, Signal Analysis, Data Interpretation, Document Planning, dan Microplanning*. Selain itu, pada bab ini dijelaskan mengenai pengembangan sistem D2T menggunakan *Linear Sequential Model*, desain eksperimen, dan hasil dan analisa hasil eksperimen yang mengukur tingkat *Readability, Computation Time*, dan analisis *Representative Text* dengan plot grafis, serta perbandingan dengan penlitian terkait.

BAB V KESIMPULAN DAN SARAN

BAB V berisi kesimpulan yang didapat selama penelitian. Selain itu, pada bab ini dipaparkan saran-saran dalam meningkatkan kualitas dan kuantitas hasil penelitian sistem D2T.

BAB II

KAJIAN PUSTAKA

2.1 Pengertian *Natural Language*

Natural Language atau bahasa alami adalah suatu bahasa yang diucapkan, ditulis, atau diisyaratkan (secara visual atau isyarat lain) oleh manusia untuk berkomunikasi. Singkatnya *Natural Language* adalah bahasa yang sering kita gunakan untuk berkomunikasi sehari-hari, seperti bahasa Indonesia, bahasa Inggris, bahasa isyarat dan bahasa lainnya sesuai letak geografisnya (Putra *et al.*, 2017). Bahasa alami menjadi topik yang hangat diperbincangkan akhir-akhir ini, banyak peneliti yang berlomba-lomba untuk menciptakan teknologi sehingga interaksi manusia dan komputer menjadi lebih mudah lagi, salah satu penerapan teknologi adalah *Natural Language Processing* (NLP) (Chowdhury, 2005). Karena erat sekali dengan kaidah atau aturan, maka setidaknya ada tiga aspek utama pada *Natural Language*, yaitu:

- a. Sintaks: menjelaskan bentuk atau struktur dari sebuah bahasa. Sintaks biasa direpresentasikan oleh sebuah *grammar* atau tata bahasa. Sebagai contoh, untuk membentuk sebuah kalimat yang valid dalam bahasa kita memakai struktur: Subjek + Predikat + Objek. *Natural language* jauh melebihi daripada *formal language* yang digunakan untuk logika kecerdasan buatan dan program komputer
- b. Semantik: menggambarkan hubungan antara sintaks dan model komputasi. Meskipun teori Semantik secara umum sudah ada, ketika membangun sistem *natural language understanding* untuk aplikasi tertentu, akan digunakan representasi yang paling sederhana.
- c. *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.

2.2 Pengertian *Natural Language Processing*

Ketika seseorang melihat atau membaca sebuah tulisan, orang tersebut akan menggunakan seluruh pengetahuan dan wawasan yang ia miliki untuk memahami tulisan tersebut. Tidak hanya sebatas tata bahasa atau *grammar*, namun lebih dari itu manusia akan mengolah informasi yang ia dapatkan dan menganalisa substansi atau konteks dari tulisan tersebut sehingga didapatkanlah pengetahuan yang baru. Maka, dikembangkan sebuah teknologi Artificial Intelligence (AI) yang berfokus pada bahasa alami, yaitu NLP. Menurut Liddy (2001) NLP adalah teknik-teknik komputasi yang didorong secara teoritis untuk menganalisa dan merepresentasikan bahasa alami pada tingkat analisis linguistik untuk mencapai pemrosesan bahasa seperti manusia. NLP hadir sebagai teknologi agar komputer dapat memahami dan memproses bahasa alami tanpa menghilangkan makna yang terkandung di dalamnya. Selain itu, NLP hadir untuk mempermudah interaksi antara manusia dan komputer. Banyak aplikasi yang menerapkan prinsip-prinsip NLP, salah satunya adalah fitur “Oke, Google” pada smartphone, pengguna kini dapat melakukan perintah-perintah seperti memutar lagu, menyetel alarm, hingga melakukan navigasi hanya cukup dengan perintah suara. Contoh lainnya adalah sentimen analisis, dimana komputer dapat menentukan sentimen dari sebuah respon yang diberikan, apakah itu berupa respon yang positif, negatif, ataupun netral. Contoh lainnya adalah aplikasi penerjemah, dimana komputer dapat menerjemahkan suatu bahasa kedalam bahasa lainnya secara otomatis.

Liddy (2001) mengungkapkan, setidaknya ada enam istilah yang sering digunakan dalam NLP:

- a. *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 atau kata keterangan), karena bagian dari naskah tidak hanya bergantung pada definisi istilah tetapi juga pada konteks dimana teks digunakan.
- b. *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.

- c. *Word sense disambiguation*: Banyak kata yang memiliki lebih dari satu arti. Memilih arti yang paling masuk akal hanya bisa dicapai dengan mempertimbangkan konteks di mana kata digunakan.
- d. *Syntactic ambiguity*: Tata bahasa dalam bahasa alami seringkali ambigu yang artinya, ada berbagai struktur kalimat yang memungkinkan yang perlu dipertimbangkan. Memilih struktur yang paling tepat biasanya memerlukan paduan informasi kontekstual dan semantik.
- e. *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.
- f. *Speech acts*: Suatu kalimat seringkali dianggap sebagai suatu aksi oleh si pembicara. Struktur kalimatnya sendiri mungkin tidak berisi cukup informasi untuk mendefinisikan tindakan ini.

Liddy (2001) juga mengungkapkan bahwa penerapan-penerapan NLP seringkali tidak akan terlepas dari bidang-bidang berikut ini:

- a. *Information Retrieval*: Ilmu untuk melakukan pencarian terhadap berbagai dokumen yang relevan, menemukan informasi tertentu didalamnya, dan menghasilkan metadata untuk isinya.
- b. *Information Extraction* (IE): 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.
- c. *Question-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*).
- d. *Summarization*: Tingkat NLP yang lebih lanjut, menghasilkan sebuah rangkuman singkat dari sebuah dokumen (dengan teks dalam jumlah besar) dengan waktu yang cepat.
 - e. *Machine Translation*: Penerjemahan otomatis dari satu bahasa manusia ke bahasa manusia yang lain.
 - f. *Dialogue Systems*: Berkommunikasi dengan komputer layaknya dengan manusia, yang memungkinkan dikembangkan lebih lanjut mengingat potensi yang dimiliki sistem ini sangat besar.
 - g. *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.
 - h. *Natural Language Generation*: Sistem mengubah informasi dari database komputer (simbolik atau numerik) menjadi bahasa manusia yang bisa dibaca.
 - i. *Natural Language Understanding*: Sistem mengekstrak informasi yang mewakili makna dari suatu sumber teks (dokumen atau rangkuman).
 - j. *Speech Synthesis*: Perangkat yang mampu berbicara atau membaca teks.

NLP mempelajari tentang bagaimana memproses dan mengubah bahasa alami kedalam bentuk yang lebih sederhana dan terstruktur (simbol atau numerik) sehingga proses komputasi menjadi lebih mudah. Lebih dari itu, NLP bertujuan agar informasi yang kita dapatkan dari bahasa alami didapatkan juga oleh komputer, atau dengan kata lain komputer dapat memahami bahasa alami kita dengan memperhatikan batasan sintaks, semantik, gramatikal dan konteks.

2.3 Pengertian *Natural Language Generation*

Berbeda dengan *Natural Language Processing* (NLP), *Natural Language Generation* (NLG) merupakan sistem yang mampu membangkitkan bahasa alami sebagai keluaran. NLG dapat diartikan juga sebagai proses penyusunan teks bahasa alami untuk memenuhi tujuan komunikatif tertentu. Sedangkan Bateman & Zock (2012) menuturkan bahwa NLG adalah sistem yang mampu menghasilkan

informasi dalam bentuk text (linguistic) dengan berdasarkan data non-linguistic (data raw atau mentah yang terukur atau berasal dari serangkaian kejadian) agar mudah dipahami oleh manusia. Selain itu, hal yang paling membedakan antara NLP dan NLG adalah pemilihan informasi untuk keluaran, dimana sistem NLG harus memilih beberapa pilihan berupa teks yang akan disampaikan, singkatnya jika terdapat pemilihan informasi untuk keluaran berupa teks maka sistem tersebut lebih mendekati ciri-ciri sistem NLG dibandingkan dengan NLP (Reiter, 2010). Seringkali, pemilihan teks keluaran dapat mempermudah proses pada sistem NLG, contohnya ada pilihan teks keluaran sebagai berikut:

- a. Saya bertemu dengan Lira di jalan, dan pergi ke sekolah bersamanya.
- b. Saya bertemu dengan Lira di jalan, dan pergi ke sekolah bersama Lira.

Kedua pilihan tersebut jika dilihat secara makna, tentu memiliki makna yang sama. Tetapi secara kasat mata tentu kita akan memilih pilihan a, dikarenakan lebih sederhana dan tidak terkesan kaku. Namun dalam perspektif komputer, pilihan a akan lebih lama pemrosesannya dibandingkan pilihan b. Karena, komputer harus mendefinisikan konteks dari kata bersamanya pada pilihan a, tentu sistem harus memiliki pemahaman untuk mendefinisikan konteks dari kata tersebut. Berbeda dengan pilihan b, sistem akan lebih mudah mendefinisikan bahwa subjek pulang pulang bersama Lira.

Maka tak heran, salah satu indikator kesuksesan dalam membangun sistem NLG adalah adanya pengetahuan teknik (menguraikan, merepresentasikan, dan mengatur proses informasi dari input) serta pengetahuan tentang keadaan dan kendala dari pengguna (prosesor informasi yang diterima) (Bateman & Zock, 2012). Bateman & Zock (2012) juga mengungkapkan bahwa setidaknya ada tiga definisi NLG, yaitu:

- a. NLG sebagai *mapping problem*
- b. NLG sebagai *problem of choice*
- c. NLG sebagai *planning problem*

Dimana setiap definisi tersebut mewakili setiap layer dalam arsitektur utamanya. Berikut adalah penjelasan arsitektur utama dalam sistem NLG menurut (Bateman & Zock, 2012):

- a. *Macroplanning*: Penentuan konten, tujuan, dan *knowledge source* dari data mentah, lalu mengorganisir dan melakukan perencanaan untuk membangun sebuah teks keluaran.
- b. *Microplanning*: Mendeskripsikan kejadian-kejadian lalu mengelompokkan materi yang berkaitan sehingga dapat membangun informasi yang lebih terintegrasi dan ringkas.
- c. *Surface realization*: Pengkontruksian gramatikal yang akan dipilih (pemilihan sintaksis), penandaan *Part-of-Speech* (POS), penambahan atribut-atribut seperti preposisi, dan penentuan bentuk akhir sehingga menjadi kata-kata yang tersusun (morfologi).
- d. *Physical presentation*: Proses penambahan artikulasi, pungtuasi, dan layout mana yang akan digunakan.

2.4 Arsitektur sistem *Data-to-Text*

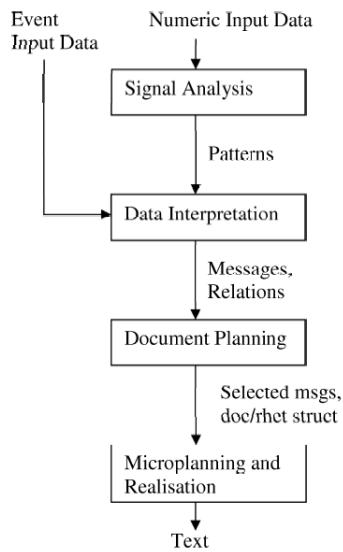
Data-to-text (D2T) adalah sistem *Natural Language Generation* (NLG) yang mampu menghasilkan teks dari input data *non-linguistic*, seperti data sensor dan serangkaian kejadian (Reiter, 2011). Seperti namanya, sistem D2T mengkonversikan data raw (data mentah) dari sensor baik berupa data numerik, data log, ataupun data *non-linguistic* menjadi sebuah teks yang mudah dipahami oleh pembaca data memuat informasi serta *knowledge* sesuai dengan data masukannya.

Berbagai pengaplikasian D2T sudah banyak dgunakan dalam berbagai bidang, seperti pada bidang peramalan cuaca misalnya, Putra *et al.* (2017) membuat sistem bernama *Data-to-text Weather Prediction* (DWP) yang mampu menghasilkan ringkasan berita klimatologis dan cuaca selama satu bulan serta memberikan informasi prediksi untuk satu hari berikutnya. Setelah itu, sistem DWP dikembangkan lagi oleh Abidin *et al.* (2018), sehingga sistem D2T yang dibangun mampu menerima masukan berupa data *streaming*. Pada bidang lain, seperti bidang kesehatan dibangun sistem seperti *BabyTalk* yang mampu menghasilkan ringkasan teks dari data neonatal selama 45 menit kemudian ringkasan tersebut digunakan sebagai bahan pendukung keputusan presentasi modalitas yang terjadi saat itu (Gatt *et al.*, 2009), selain itu terdapat sistem BT-Nurse yang mampu meringkas kejadian

selama pergantian shift keperawatan berlangsung, berdasarkan hasil rekaman medis elektronik pasien (Hunter et al., 2011). Pada bilang ekonomi, terdapat *Knowledge-Based Report Generator* yang mampu menghasilkan laporan stok berdasarkan data stok produk (*non-linguistic*) suatu pasar (Kukich, 1983). Beberapa contoh tersebut membuktikan bahwa D2T menjadi pilihan yang tepat dalam berbagai bidang.

2.4.1. Arsitektur Data-to-Text oleh Reiter (2011)

Seperti yang digambarkan pada Gambar 2.1, Reiter (2011) memaparkan bahwa setidaknya ada empat elemen utama yang diperlukan untuk membangun sebuah sistem D2T, yaitu *Signal Analysis*, *Data Interpretation*, *Document Planning*, dan *Microplanning and Realisation*. Elemen-elemen ini hampir mirip dengan arsitektur sistem NLG yang sudah dijelaskan pada sub-bab sebelumnya. Reiter (2011) memaparkan bahwasannya, perbedaan terbesar antara sistem D2T dan sistem NLG adalah sistem D2T harus menganalisis dan menginterpretasikan data masukannya, begitu juga dengan menentukan bagaimana proses penyampaianya dari segi linguistik.

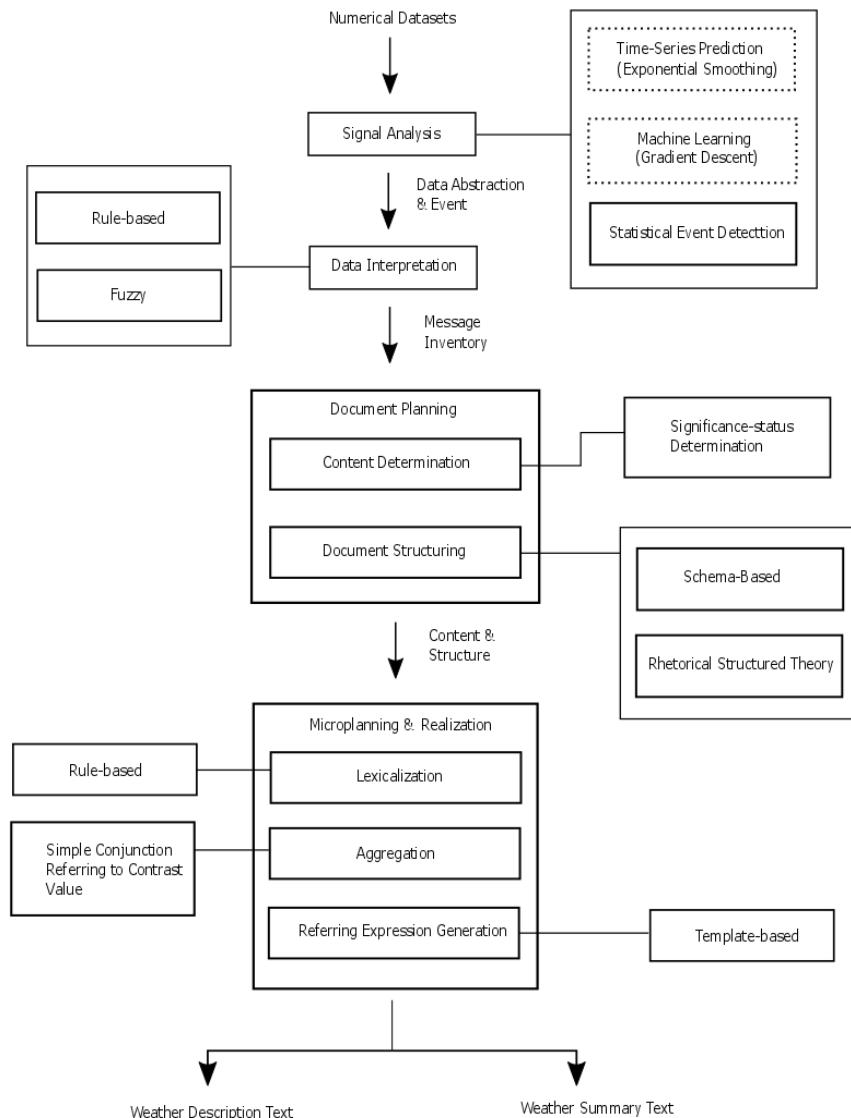


Gambar 2.1 Arsitektur D2T oleh (Reiter, 2011)

Pada gambar 2.1 Reiter (2011) memaparkan bahwa sistem D2T terdiri dari 4 fase utama, yaitu *Signal Analysis*, *Data Interpretation*, *Document Planning*, dan *Microplanning*. Dimana proses ini diterapkan pada penelitian BT45 (Portet et al., 2009) dan (Hunter et al., 2011).

2.4.2. Arsitektur Data-to-Text oleh Putra et al., (2017)

Dalam penelitian DWP, dilakukan pengembangan model dengan memodifikasi beberapa elemen, sehingga output yang dihasilkan dibagi menjadi dua kategori berupa *Weather Description Text* dan *Weather Summary Text*. Gambar 2.2 menjelaskan bagaimana sistem DWP yang dikembangkan oleh Putra *et al.*, (2017), dimana ada beberapa penambahan yaitu penggunaan *Machine Learning* yaitu *Gradient Descent* dan pendekatan *Time Series* yaitu *Exponential Smoothing*, dan juga terdapat pendekatan *Statistical Event*. Proses *Signal Analysis* menghasilkan *Data Abstraction and Event*, yang menjadi masukan bagi proses *Data Interpretation*. Proses yang diklaim oleh Reiter (2011) sebagai pembeda antara sistem D2T dan sistem NLG ini, merupakan proses untuk menginterpretasikan data masukan. DWP menggunakan 2 cara untuk menginterpretasikan datanya, yang pertama yaitu *Rule-Based* dan *Fuzzy* sehingga dihasilkan *Message Inventory* yang akan diproses pada proses selanjutnya yaitu *Document Planning*. Pada proses *Document Planning*, terdapat *Content Determination* dan *Document Structuring*. *Content Determination* atau pemilihan konten dilakukan dengan memilah seberapa pentingnya status suatu pesan atau nama lainnya yaitu *Significance Status Determination*. Setelah ditentukan seberapa pentingnya status suatu pesan, maka pesan-pesan tersebut digabungkan pada sebuah proses bernama *Document Structuring*. Setelah ditentukan statusnya, maka pesan tersebut diproses pada proses *Document Planning*, sehingga dihasilkan *Content* dan *Structure*, yang nantinya akan diproses pada tahap *Microplanning* dan *Realization*, sehingga dihasilkanlah sebuah teks yang terdiri dari *Weather Description Text* dan *Weather Summary Text*.



Gambar 2.2 Arsitektur sistem DWP (Putra *et al.*, 2017)

Tahapan pada gambar 2.2 akan dijelaskan secara terperinci pada poin-poin berikut:

a. *Signal Analysis*

Tahapan pertama dalam membangun sistem D2T adalah *Signal Analysis*. Tahapan ini bertujuan untuk mendekripsi pola-pola yang terdapat pada data masukan. Sehingga sistem dapat memproses data dengan bentuk simbolis (diskrit) dibandingkan memprosesnya dalam bentuk numerik. Namun, dalam beberapa kasus tahapan ini menjadi opsional, ketika data

masukan sudah terstruktur dalam bentuk kejadian-kejadian diskrit, contohnya seperti data rekord kejadian medis, atau *log file*. Jika data masukan sudah berbentuk simbolis (diskrit) seperti yang sudah dijelaskan sebelumnya, maka proses *Signal Analysis* ini tidak perlu dilakukan (McKeown *et al.*, 1994).

Pada dasarnya tahap *Signal Analysis* merupakan proses untuk menganalisa data masukan yang berupa numerik sehingga dihasilkan sebuah informasi berbentuk simbolis yang akan disampaikan. Contohnya untuk kasus pembangkitan berita cuaca pada DWP, data masukan berupa seluruh data klimatologi dan kualitas udara selama satu tahun, lalu data tersebut dianalisa dan dihasilkanlah sinyal-sinyal dari data tersebut seperti curah hujan terbesar, kondisi kualitas udara hari ini, serta bagaimana kondisi curah hujan dan temperatur bulan ini dibandingkan dengan curah hujan dan temperatur satu tahun penuh, dan lain-lain. Pada penmbangunan sistem DWP (Putra *et al.*, 2017) , dilakukan pendekatan *Time Series* dan pendekatan *Statistical Tools* dalam menganalisa data masukan sehingga dihasilkan sebuah pesan diskrit untuk satu bulan. Contoh penggunaan statistik (*mean, min, max, sum*) dalam bahasa R digambarkan pada pada gambar 2.3 di bawah ini.

```
...
for(i in i:n){ #Loping sebanyak jumlah parameter
  max_amt[i] <- max(LM[,i]) #Nilai Max pd parameter ke-i
  min_amt[i] <- min(LM[,i]) #Nilai Min pd parameter ke-i
  sum_amt[i] <- sum(LM[,i]) #Total nilai dari parameter ke-i
  ....
}
#Rata-rata 1 bulan terakhir untuk setiap parameter
LMmean_result <- colMeans(xLM)
....
```

Gambar 2.3 Contoh implementasi *Signal Analysis* (Putra *et al.*, 2017)

b. *Data Interpretation*

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

```
membership_check <- function(partition,v,pname,oname){
  i=1; n=length(partition)
  membership_value <- c(1)
  for(i in i:n){
    ....
    if((v<a) || (v>d)){
      membership_value[i] <- 0
    }
    if((v>=a) && (v<=b)){
      membership_value[i] <- ( (v-a) / (b-a) )
    }
    if((v>b) && (v<=c)){
      membership_value[i] <- 1
    }
    if((v>c) && (v<=d)){
      membership_value[i] <- ( (d-v) / (d-c) )
    }
  }
  i=1; biggest=0; part<-"a"
  for(i in i:n){
    ....
    if(biggest<=membership_value[i]){
      biggest <- membership_value[i]
      part<-pname[i]
    }
  }
  return (part)
}
.....
Temperature_partition <- c("very cold.", "cold.", "warm.", "hot.", "very hot.")
Temperature_interval
list(vary_cold=c(a=0,b=0,c=5,d=10), cold=c(a=5,b=10,c=15,d=20),
      warm=c(a=15,b=20,c=25,d=30),
      hot=c(a=25,b=30,c=35,d=40), very_hot=c(a=35,b=40,c=45,d=50))
Temperature_interval
membership_partition(Temperature_interval, "Temperature")
InterpretationResult_temperature
membership_check(Temperature_interval, as.double(climatePredictionResult
  ["Average.Temperature"]), Temperature_partition, " ")
.....
```

Gambar 2.4 Contoh implementasi data interpretation Rainfall DWP (Putra *et al.*, 2017)

Sebagai contoh, misalnya terdapat data suhu udara hari ini senilai 40°C. Maka dengan melalui serangkaian proses interpretasi data ini, angka 40°C diinterpretasikan menjadi pesan “*very hot*”, seperti implementasi pada DWP (Putra *et al.*, 2017) pada gambar 2.4 dengan berdasarkan pada *fuzzy membership function* (Ramos-Soto *et al.*, 2016a).

c. *Document Planning*

Langkah ke-tiga 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.

Menurut Reiter (2011) bahwa serangkaian proses *Document Planning* ini diantaranya adalah membagi tugas menjadi beberapa bagian berikut:

1. *Content Determination*

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. DWP menerapkan *Significant Event Message* dalam menentukan event suatu hujan terjadi berturut turut secara *extreme* atau tidak seperti pada gambar 2.5.

```
RainExtremeMessage function <- function(Dataset){
  if(interpreterRainfall == "no rain" || interpreterRainfall=="light rain" || 
  interpreterRainfall=="moderate rain"){
    return("x") #artinya tidak extreme
  }else{
    ..... #get repeated date
    return (date)
  }
}
```

Gambar 2.5 Contoh Content Determination Significant Event Message DWP (Putra *et al.*, 2017)

2. Document Structuring

Document Structuring adalah proses penentuan 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. Skema tersebut dibuat berdasarkan *Target Text* yang ingin dicapai atau berdasarkan penalaran (Reiter, 1996). Contoh “*Target Text*” dalam DWP dapat dilihat pada gambar 2.6.

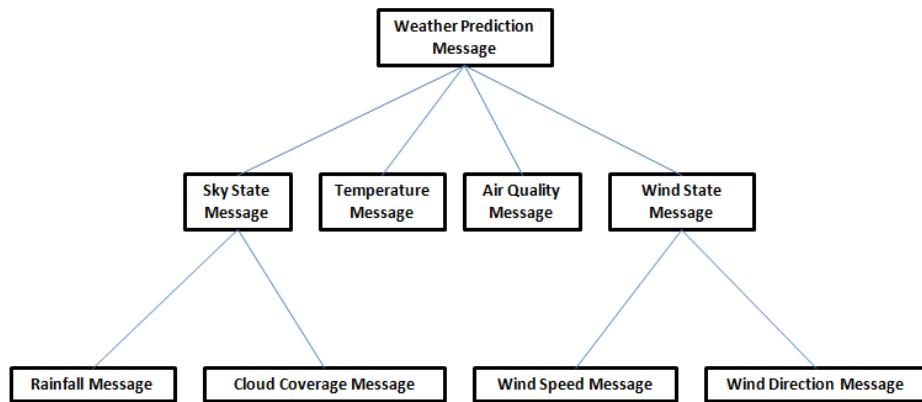
```
BASED ON PREDICTION RESULT, PREDICTED THAT A STORM RAIN WILL COME COVERED WITH OVERCAST CLOUD. FOLLOWED BY VERY WARM TEMPERATURE. WITH RESPECT TO THE AIR QUALITY STATE, IT WILL CHANGE PROGRESSIVELY TO BAD. WIND WILL BLOW VERY STRONG FROM THE NORTH.
```

Gambar 2.6 Contoh Target Text DWP (Putra *et al.*, 2017)

Reiter (1996) menjelaskan bahwa dengan berdasarkan *Target Text* pada gambar 2.6 dilakukan pembuatan skema dengan cara sebagai berikut:

- a. Ambil contoh sejumlah teks yang sama dalam *Target Text*.
- b. Identifikasi terhadap pesan-pesan yang ada, tentukan bagaimana setiap pesan dapat dibangun berdasarkan data.
- c. Mengusulkan aturan atau struktur yang menjelaskan mengapa pesan “x” ada dalam teks A tetapi tidak ada didalam teks B. Penentuan ini lebih mudah jika disusun dalam bentuk seperti taksonomi atau pohon.
- d. Diskusikan hasil analisis bersama pakar.

Dengan berdasarkan tahapan diatas, maka didapat skema dalam bentuk pohon seperti pada gambar 2.7, dimana *Weather Prediction Message* terdiri dari empat komponen, yaitu *Sky State Message*, *Temperature Message*, *Air Quality Message*, dan *Wind State Message*.



Gambar 2.7 Contoh skema dalam bentuk *tree* berdasarkan Target Text DWP (Putra et al, 2017)

d. *Microplaning and Realisation*

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 (*signal analysis*, *data interpretation*, dan *document planning*).

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

1. *Lexicalisation*

Proses *lexicaisation* adalah bagaimana melakukan pemilihan kata atau frase yang akan digunakan dalam mendekripsikan segala hal, contohnya mendeskripsikan relasi, tren, dan kemungkinan. DWP menjelaskan tren yang terjadi pada kualitas udara dengan membandingkan dua baris data, yakni data ke-*n* dan data ke-(*n*-1), dimana *n* merupakan jumlah baris data, seperti pada gambar 2.8.

```

.....
TrendDesc_template <- function (IVL,data){
  if((IVL[1]=="0")&&(IVL[2]=="0")){
    TrendDesc <- change_word_bank_AQ("stable")
  }
  if(((IVL[1]=="+")&&(IVL[2]=="-"))|||((IVL[1]==-
")&&(IVL[2]=="+"))){
    TrendDesc <- change_word_bank_AQ("mediumChange")
  }
.....
  return(TrendDesc)
}
.....

```

Gambar 2.8 Contoh implementasi Lexicalisation tren DWP (Putra et al., 2017)

2. 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*, seperti dalam pada gambar 2.9.

```

.....
Contrast_lexicalisation1 <- function(msg1,msg2){
  if(msg1[2]==msg2[2]){
    return("and")
  }else{
    return("but")
  }
}
.....

```

Gambar 2.9 Contoh *Simple Conjunction Referring to Contrast Value* (Putra et al., 2017)

3. Referring Expression Generation

Proses ini berisi mengenai bagaimana sistem dapat merujuk informasi tertentu kepada sebuah subjek. Contohnya: “Suhu hari ini tergolong sangat panas”, sistem dikondisikan agar dapat menyampaikan bahwa informasi “sangat panas” adalah penjeasan informasi dari subjek “Suhu”. Salah satu contoh penerapan dengan cara *hardcode* pada kode program seperti pada gambar 2.10, dimana “*The wind for the month was*” merupakan subjek dari kalimat yang akan ditampilkan.

```
....  
MonthlyMsg4_aggregation<-function(msg1,msg2) {  
  if(msg2=="false"){  
    msg<-paste("The wind for the month was",msg1,"in average.")  
  }  
  else{  
    msg<-paste("The wind for the month was",msg1,"in average, but",msg2)  
  }  
  return(msg)  
}  
....
```

Gambar 2.10 Contoh Referring Expression Generation (Putra *et al.*, 2017)

4. Structure Realisation

Pada proses ini, setiap struktur yang telah dibuat dalam proses *dokumen planning* direalisasikan sehingga menghasilkan teks dalam bentuk aktual (Reiter, 1996). Contohnya, merealisasikan struktur teks dalam bahasa pemrograman menjadi teks aktual dalam HTML, LaTeX, RTF, SABLE, dan lain-lain, seperti pada gambar 2.11, yang merealisasikan struktur dalam bahasa R dengan menggabungkan hasil seleksi konten, kemudian menampilkan hasil ke dalam HTML dengan bantuan *package* Shiny.

```

Structure_Realization_predict <- function(){
  ....
  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)
  ....
}

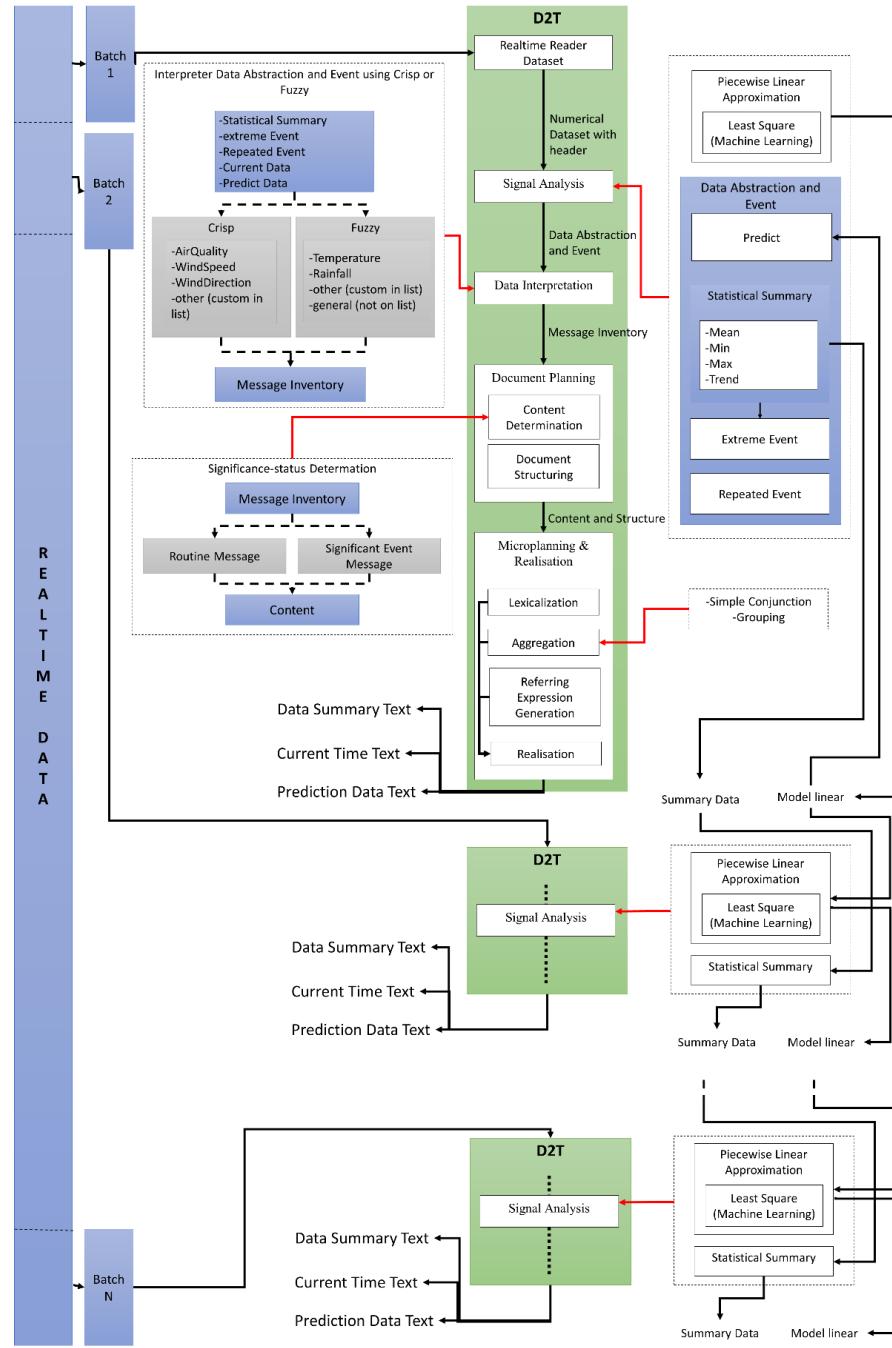
.....
library(shiny)
.....
tags$div(class="col-lg-8",
  tags$h2("Weather Prediction News"),
  tags$p(align="justify",HTML(' &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
  '),
    Prediction_Result
  ),
  tags$hr(class="style13"),
  tags$p(align="justify",HTML(' &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
  &nbsp;'),
    MonthlyMsg
  )
)
.....

```

Gambar 2.11 Contoh implementasi Structure Realisation DWP (Putra *et al.*, 2017)

2.4.3. Arsitektur Data-to-Text oleh Abidin et al., (2018)

Pada penelitian D2T yang dilakukan oleh Abidin et al., (2018), diperkenalkan sebuah sistem yang dapat menerima masukan berupa data *streaming*. Seperti yang bisa kita lihat pada gambar 2.12 terdapat perbedaan pada tahap *Signal Analysis* dan terdapatnya proses bernama *Realtime Reader Dataset*, dimana data yang masuk akan terlebih dahulu dihandle pada tahap *Realtime Reader Dataset* sebelum diproses pada tahap *Signal Analysis*. Setelah data masukan diproses pada tahap *Realtime Reader Dataset*, data masukan selanjutnya akan diproses pada tahap *Signal Analysis*, dimana data *streaming* akan diproses dan dipecah berdasarkan *batch-batch*. Sehingga proses prediksi data dilakukan menggunakan teknik Piecewise Linear Approximation (PLA) dengan metode Least Square yang diperkenalkan oleh Palpanas *et al.*, (2004).



Gambar 2.12 Arsitektur D2T untuk data *streaming* (Abidin *et al.*, 2018)

Abidin *et al.*, (2018) menuturkan bahwa pada proses ini dihasilkan model linear dari setiap batch data dan model linear untuk keseluruhan data (batch pertama hingga batch ke-N), hal ini dilakukan karena data yang digunakan merupakan data Time Series sehingga antara batch satu dan lainnya memiliki keterikatan. Penerapan PLA dilakukan untuk merepresentasikan data streaming lampau dan menggabungkannya menjadi sebuah garis linear, mengingat data yang digunakan

hanya dalam satu kali proses dan selanjutnya data terhapus untuk mengatasi masalah storage berlebihan dikarenakan kemunculan data sangat cepat dan memiliki ukuran yang tidak sedikit bila kita menggabungkan setiap data stream.

Untuk mengimplementasikan *Realtime Reader Dataset* proses ini (Abidin *et al.*, 2018) menggunakan Javascript animation dan AJAX dalam proses pengecekan dataset baru (fungsi *checkFile*), bila terdapat dataset baru, maka sinyal akan dikirimkan melalui fungsi *RunR()* dan kemudian *D2T_Main.R* akan dijalankan. Pada gambar 2.13 terlihat animasi dijalankan setiap 1000 *milisecond* untuk melakukan proses pengecekan dataset, jika terdapat dataset maka akan dikirim sinyal “*run*” kepada controller *RunR* dengan metode *post*, jika tidak terdapat dataset baru, maka cek konten yang ditampilkan apakah sesuai dengan konten terakhir yang dihasilkan oleh *D2T_Main.R* atau tidak.

```
setTimeout(checkFile,1000); //Animation Checker
function checkFile(){
    // Checking Dataset
    $.ajax({
        type: 'HEAD',
        url: 'http://localhost/D2T/DatasetsRealTime/Dataset.csv',
        success: function() {
            // This is a new dataset
            // Send signal to controller RunR for running D2T_Main
            $.ajax({
                url:"http://localhost/D2T/RunR", //the with php script
                type: "post", //request type,
                dataType: 'json',
                data: {exec: "run"}
            });
            setTimeout(checkFile, 5000);
        },
        error: function() {
            // No dataset found
            // Check last update file content
            $.ajax({
                type: 'HEAD',
                url: 'http://localhost/D2T/Result/tempTime.json',
                success: function() {
                    // Validation Content in Web
                    checktempTime();
                },
                error: function() {
                    console.log("nothing change");
                }
            });
            setTimeout(checkFile,1000);
        }
    });
}
```

Gambar 2.13 *Realtime animation for check file* (Abidin *et al.*, 2018)

Sedangkan pada gambar 2.14 terlihat bahwa sinyal “run” diproses untuk menjalankan fungsi yang dapat mengeksekusi D2T_Main.R untuk membaca dataset.

```
public function executeR() {
    // Execute D2T_Main.R
    exec  ("\".\\R-3.4.0\\bin\\Rscript.exe\" .\\D2T_Main.R 2>&1",
$output);
    // echo '<pre>', join("\r\n", $output), "</pre>\r\n";
}

public function Index()
{
    if($_POST["exec"] == "run") {
        $this->executeR();
    }
}
```

Gambar 2.14 Execute D2T_Main.R (Abidin *et al.*, 2018)

Setelah proses membaca *file* selesai, maka dataset dihapus untuk meminimalisir penggunaan *hardisk* yang besar seperti pada gambar 2.15.

```
# Read data
dataset <- read.table(file="DatasetsRealTime/Dataset.csv", sep=",",
header=TRUE)
fn <- "DatasetsRealTime/Dataset.csv"
# Delete data
if (file.exists(fn)) file.remove(fn)
```

Gambar 2.15 Read and Remove Dataset in R (Abidin *et al.*, 2018)

Sedangkan untuk menyimpan hasil, Abidin et al., (2008) menggunakan library “*jsonlite*” dengan menggunakan fungsi *toJSON()* kemudian menuliskan data pada file dengan fungsi *write()* dan *write.csv()* seperti pada gambar 2.16

```
library(jsonlite)

timeInterval <- toJSON(timeInterval)
resumeResult <- toJSON(resumeResult)
currentResult <- toJSON(currentResult)
predictResult <- toJSON(predictResult)
columnName <- toJSON(columnName)

write.csv(statisticalResume, file =
"Result/statisticalResume.csv", row.names=FALSE)
write(columnName, file='Result/columnName.json')
write(timeInterval, file='Result/timeInterval.json')
write(now, file='Result/tempTime.json')
write(resumeResult, file='Result/resumeResult.json')
write(currentResult, file='Result/currentResult.json')
write(predictResult, file='Result/predictResult.json')
```

Gambar 2.16 Write Result JSON and csv in R (Abidin *et al.*, 2018)

Kemudian untuk menampilkan data pada web, Abidin et al., (2018) menggunakan AJAX \$.getJSON() dan menampilkan hasilnya dengan append() seperti pada gambar 2.17.

```

$.getJSON('http://localhost/D2T/Result/tempTime.json', function(data) {
    $('#tempTime').val(data);
});

$.getJSON('http://localhost/D2T/Result/resumeResult.json',
function(data) {
    $('#newsResume').append("<p>&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp"+data+"</p>");
});

$.getJSON('http://localhost/D2T/Result/currentResult.json',
function(data) {
    $('#newsCurrent').append("<p>&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp"+data+"</p>");
});

$.getJSON('http://localhost/D2T/Result/predictResult.json',
function(data) {
    $('#newsPredict').append("<p>&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp"+data+"</p>");
});

```

Gambar 2.17 Get JSON in AJAX (Abidin et al., 2018)

2.5 Penelitian Terkait sistem *Data-to-Text*

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

Tabel 2.1 Penelitian terkait D2T dan NLG

Referensi	Metode Content Selection	Domain	Sumber Data
(Kukich, 1983)	Rule-Based	Market	Database
(Boyd, 1998)	No Content Selection	Weather	Database
(Sripada et al., 2001)	Two Stage model: (1) Domain Reasoner; (2) communicative reasoner	Weather, Oil Rigs	Sensor data, Numerical Data
(Sripada et al., 2003)	Gricean Maxims	Weather, Gas Turbines, Health	Sensor data
(Hallett et al., 2006)	Rule-Based	Health	Database
(Yu et al., 2007)	Rules derived from corpus analysis and main knowledge	Gas Turbines	Sensor
(Sripada & Gao, 2007)	Decompression Models	Dive	Sensor

Referensi	Metode Content Selection	Domain	Sumber Data
(Turner <i>et al.</i> , 2008)	Decision Tree	Georeferenced Data	Database
(Gatt <i>et al.</i> , 2009)	Rule-Based	Health	Sensor
(Thomas <i>et al.</i> , 2012)	Document Schema	Georeferenced Data	Database
(Demir <i>et al.</i> , 2012)	Rule-based	Domain Independent	Graph-database
(Reddington & Tintarev, 2011)	Threshold-based rules	Assitive Technology	Sensor
(Banaee <i>et al.</i> , 2013)	Rule-based	Health	Grid of sensor
(Schneider <i>et al.</i> , 2013)	Rule-based	Health	Sensor
(Ramos-Soto <i>et al.</i> , 2016b)	Fuzzy-sets	Weather	Database
(Gkatzia <i>et al.</i> , 2016)	Rule-based	Weather	Numerical data with assigned probabilities
(Putra <i>et al.</i> , 2017)	Rule-Based and Fuzzy	Weather	Numerical data
(Abidin <i>et al.</i> , 2018)	Rule-Based and Fuzzy	General	Data Streaming

2.6 *Machine Learning*

Machine Learning termasuk dalam bagian dari ilmu komputer yang dapat membelajarkan komputer sehingga memiliki kemampuan untuk belajar tanpa diprogram secara eksplisit (Samuel, 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 sehingga mesin menjadi cerdas dan dapat belajar seiring dengan berkembangnya data masukan. *Machine Learning* sudah ada dan mulai digunakan sejak 50 tahun yang lalu, pengaplikasiannya pun sudah digunakan dalam berbagai bidang seperti bidang ekonomi, keilmuan, industri dan sebagainya.

Salah satu implementasi *Machine Learning* yang pernah dilakukan oleh Arthur Samuel sekitar 59 tahun yang lalu yaitu pembuatan permainan catur dengan computer (Samuel, 1959). 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 tahun 1914, seorang ilmuan dari Spanyol, Torres y Quevedo, membuat sebuah mesin catur yang dapat mengalahkan atau melakukan skakmat pada raja lawan dengan sebuah ratu dan raja (Shannon, 1950). Perkembangan secara sistematis kemudian dimulai segera setelah diketemukannya komputer digital.

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

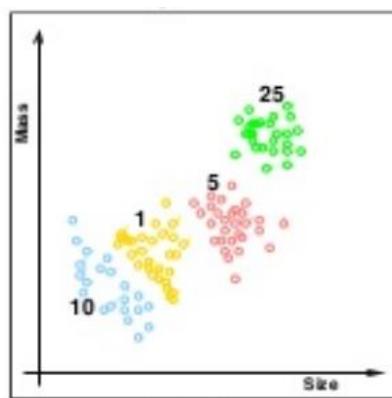
Machine Learning memiliki beberapa tipe dengan proses pembelajaran yang berbeda, tipe-tipe tersebut akan dijelaskan pada sub-bab berikutnya.

2.6.1. *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 *et al.*, 2012). *Supervised Learning* merupakan suatu pembelajaran yang terawasi, dimana jika *output* yang diharapkan telah terdapat pada daftar yang 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*:

- a. Klasifikasi: adalah sebuah metode untuk menyusun data secara sistematis menurut aturan-aturan yang telah ditetapkan sebelumnya (Athoillah *et al.*, 2015). Dengan melakukan klasifikasi, dari data yang telah ada dapat dibuat sebuah model prediksi dengan *output* kelas. Beberapa algoritma klasifikasi yang cukup terkenal adalah k-Means, SVM, EM, Naïve Bayes, dan kNN.
- b. Regresi: analisis regresi adalah salah satu metode statistik untuk memprediksi nilai dari satu atau lebih variabel respon/dependen dari satu set variabel prediktor/independen (Härdle & Simar, 2007).



Gambar 2.18 Contoh *Supervised Learning* pada pengenalan koin

Pada Gambar 2.18, 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. Beberapa contoh penerapan *Supervised Learning* yaitu pada kasus klasifikasi sentimen (Ye *et al.*, 2009), prediksi virus hepatitis B (Hospital, 2003), dan *Gait Event Detection* (Williamson & Andrews, 2000).

2.6.1.1. Algoritma Gradient Descent

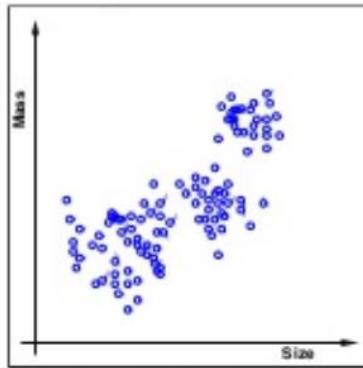
Algoritma *Gradient Descent* adalah algoritma optimasi untuk menemukan *minimum* lokal dari fungsi menggunakan *gradien descent*, diambil langkah sebanding dengan negatif dari gradien (atau perkiraan gradien) dari fungsi pada titik sekarang (Riza *et al.*, 2016). Jika diambil langkah sebanding dengan gradien positif,

maka akan didapatkan maksimum lokal fungsi tersebut; prosedur ini kemudian dikenal sebagai *gradient ascent*. Algoritma *Gradient Descent* yang digunakan dalam penelitian ini mengacu pada penerapan algoritma *Gradient Descent* pada penelitian DWP untuk mengatasi *missing value* sehingga sistem tetap berkerja tanpa galat, meskipun pada data masukan terdapat nilai NA atau *missing value* (Putra *et al.*, 2017). Dalam penelitian ini, *missing value handling* dilakukan saat pra-proses data menggunakan *package mice* dalam R.

2.6.2. *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 Quantization* (LVQ), dan neocognitron.



Gambar 2.19 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 2.19, belum diketahui kelas dari masing-masing data, mesinlah yang menentukan berdasarkan kedekatannya. Beberapa contoh penerapan *Unsupervised Learning* diantaranya sistem pendekripsi intrusi (Zanero & Savaresi, 2004), menemukan komunitas pengguna di Internet sesuai dengan kriteria (Palouras *et al.*, 2002), dan pengembangan strategi pengendalian manufaktur (Bowden & Bullington, 1996).

2.6.3. *Semi Supervised Learning*

Semi Supervised Learning adalah penggabungan dari *Supervised* dan *Unsupervised Learning*. Dimana hasil keluaran sistem ada yang termasuk dalam kategori yang sudah ditetapkan namun ada juga yang tidak. Beberapa contoh penerapannya yaitu kasus representasi kata (Turian, J., Ratinov, L., & Bengio, 2010), *Co-Tracking* (Tang *et al.*, 2007), dan identifikasi peptida (Käll *et al.*, 2007).

2.6.4. *Reinforcement Learning*

Pada *Reinforcement Learning* model yang dihasilkan terus berkembang seiring pemakaian oleh pengguna, dimana model terus menerus diperbaiki sesuai kondisi penerapan. Beberapa contoh penerapannya yaitu pada simulasi sepak bola dalam *RoboCup* (Stone *et al.*, 2005) dan penerbangan helikopter terbalik secara otonom (Ng *et al.*, 2006). Selain itu, sistem D2T pernah dikembangkan menggunakan model *Reinforcement Learning* oleh Gkatzia, Hastie, Janarthanam, & Lemon, (2013), dimana sistem yang dikembangkan mampu menghasilkan teks keluaran yang adaptif berdasarkan *feedback* dari dosen atau staff kependidikan,

adapun teks keluaran yang dihasilkan merupakan rangkuman dari perilaku mahasiswa Ilmu Komputer selama pembelajaran lab berlangsung.

2.7. Time-series Data

Kumpulan data yang tercatat dalam periode waktu mingguan, bulanan, kuartalan, atau tahunan(Mishra dan Jain, 2014). Ada 4 faktor yang mempengaruhi data *Time Series*. Dalam data ekonomi biasanya didapatkanadanya fluktuasi atau variasi dari waktu ke waktu atau disebut dengan variasi *Time Series*. Variasi ini biasanya disebabkan oleh adanya faktor *Trend (trend factor)*, Fluktuasi siklis (*cyclical 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. Teknik pemulusan dan regresi akan dibahas pada sesi tulisan yang lain.

2.8. 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. Pada penelitian ini, algoritma *exponential smoothing* digunakan untuk memprediksi nilai dari setiap parameter dengan tipe *numerical* seperti yang sudah dilakukan pada penelitian sistem DWP (Putra *et al.*, 2017).

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. Pada penelitian ini digunakan *Single Exponential Smoothing* untuk proses prediksi data seperti pada penelitian DWP (Putra *et al.*, 2017).

Rumus untuk Simple exponential *smoothing* adalah sebagai berikut:

$$F_{t+1} = \alpha A_t + (1-\alpha) F_t$$

dimana:

F_{t+1} = Peramalan untuk periode $t + 1$.

F_t = Peramalan untuk periode t (sebelumnya).

A_t = Nilai aktual Time Series

α = Monstanta perataan antara 0 dan 1

Misalnya, jika kita menggunakan contoh data seperti pada tabel 2.2, dimana A_t merupakan kolom penjualan, dan F_t merupakan hasil prediksi, dengan α bernilai 0.2, maka perhitungan *exponential smoothing* adalah sebagai berikut.

Tabel 2.2 Contoh penggunaan *Exponential Smoothing*

Minggu	Penjualan (A_t)	Prediksi (F_t)	Keterangan
1	39	39	$F_1 = A_1 = 39$
2	44	39	$F_2 = 0.2(39) + 0.8(39) = 39$
3	40	40	$F_3 = 0.2(44) + 0.8(39) = 40$
4	45	40	$F_4 = 0.2(40) + 0.8(40) = 40$
5	38	41	$F_5 = 0.2(45) + 0.8(40) = 41$
6	43	40.40	$F_6 = 0.2(38) + 0.8(41) = 40.40$
7	39	40.92	$F_7 = 0.2(43) + 0.8(40.40) = 40.92$
		40.54	$F_8 = 0.2(39) + 0.8(40.92) = 40.54$

2.9. String Matching

String Matching merupakan salah satu teknik yang digunakan dalam *information retrieval*, dimana metode ini digunakan untuk mengambil informasi dari sesuatu yang ingin diketahui. Selain lain untuk teknik *string matching* diantaranya *pattern matching*, atau *pattern searching*, dimana teknik *string matching* ini sering digunakan untuk berbagai hal terutama dalam kemanan informasi, bioinformatika, deteksi plagiarism, pemrosesan teks dan pencocokan dokumen (Vijayarani & Janani, 2016).

Dalam bidang keamanan informasi seperti yang sudah dijelaskan sebelumnya, teknik *string matching* ini digunakan dalam bidang keamanan jaringan misalnya dalam perbaikan data paket HTTP secara *realtime*. Dimana kebutuhan akan algoritma *string matching* yang efisien untuk mereduksi data pada protokol HTTP menjadi sangat penting (Zhang *et al.*, 2015). Selain itu, bidang pengaplikasian *string matching* yang mulai hangat diperbincangkan akhir-akhir ini adalah *bioinformatics*. Dimana teknik ini digunakan untuk menganalisa sekuen DNA sehingga dapat memberikan informasi yang dibutuhkan secara tepat, seperti penelitian yang dilakukan oleh (Rahman, 2017), dimana penelitian tersebut menggunakan algoritma ini untuk mendeteksi *genomic repeats* pada sekuen DNA. Sehingga teknik *string matching* ini menjadi topik riset yang menarik dan penting dalam bidang ilmu komputer (Chen & Wu, 2016).

Pada penelitian ini, teknik *string matching* digunakan untuk mencari motif pada parameter *categorical*, dimana motif beberapa data terakhir akan dicocokkan dengan seluruh data pada parameter tersebut. Algoritma *string matching* yang digunakan dalam penelitian ini adalah algoritma *Knuth-Morris-Pratt* (KMP) yang akan dijelaskan pada sub-bab selanjutnya.

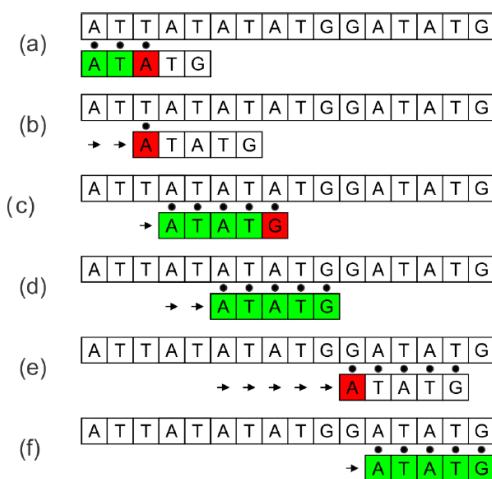
Algoritma Knuth-Morris-Pratt ditemukan oleh ketiga ilmuwan bernama Knuth, Morris dan Pratt untuk menemukan posisi string yang diberikan pada sebuah *text editing program*. Berbeda dengan algoritma pencarian *brute force* dimana pada algoritma KMP ini akan ditentukan informasi *prefix* dari pattern yang akan dicari sebelum dilakukan pencarian. Pada penelitian ini, digunakan algoritma string Matching yang digunakan pada penelitian (Rahman, 2017). Misalnya jika kita mempunyai *pattern* ‘ATATG’ yang akan dicari pada teks

‘ATTATATATGGATATG’, algoritma *brute force* akan mencocokan setiap *pattern* pada teks dan akan bergeser satu indeks untuk mencocokkan *pattern* tersebut, sehingga kompleksitas dari algoritma pencarian *brute force* ini adalah O(mn). Sedangkan Algoritma KMP akan menyimpan informasi pada pencocokan sebelumnya untuk menghindari pencocokan yang sia-sia. Praproses pada pattern dengan memberikan *prefix* dari tiap karakter pattern akan menjadi petunjuk untuk melakukan pengabaian pencocokan yang dinilai tak perlu karena sudah pasti tak cocok atau sudah pasti cocok. Sehingga didapatkan *prefix* seperti pada gambar 2.20.

Index	1	2	3	4	5
Pattern	A	T	A	T	G
Prefix	0	0	1	2	0

Gambar 2.20 Pemberian prefix pada pattern 'ATATG' (Rahman, 2017)

Gambar 2.21 adalah skenario *string matching* dengan algoritma KPM pada permasalahan yang sama seperti sebelumnya. Terlihat pada gambar 2.17 bagian (b) melakukan pergeseran dua indeks karena informasi yang teredia pada *prefix*, perhitungan pergeseran terebut ada sebagai berikut: Prefix(3 - 1) + 1 = 0 + 1 = 1, maka karakter akan langsung melakukan pengecekan dengan bergeser langsung ke indeks ke-1.



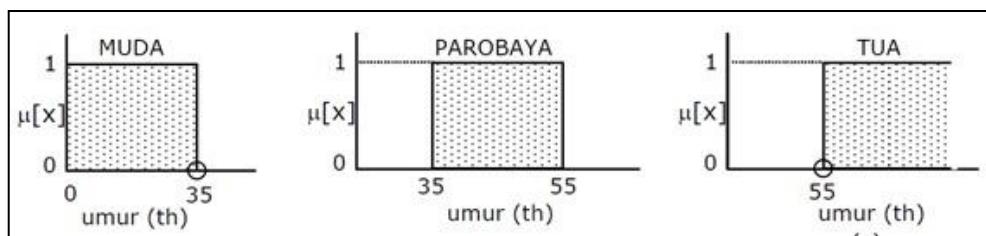
Gambar 2.21 Skenario Knuth-Morris-Pratt pada String Matching

2.10. 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 2.22.



Gambar 2.22 Contoh himpunan *Crisp* pada kasus umur (Putra et al., 2017)

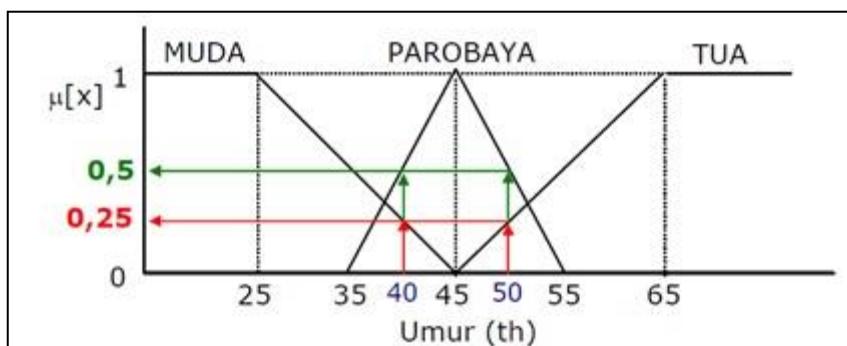
Jika menggunakan himpunan *crisp* yang dapat dilihat pada Gambar 2.10, dapat diambil kesimpulan bahwa:

- 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 → $\mu_{\text{Parobaya}}[55\text{th}+1\text{hr}] = 0$ atau
- Usia 55 tahun lebih 1 hari, dikatakan Tua → $\mu_{\text{Tua}}[55\text{th}+1\text{hr}] = 1$

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 2.23.



Gambar 2.23 Contoh himpunan *Fuzzy* pada kasus umur (Putra et al., 2017)

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 $\mu_{\text{Parobaya}}[50] = 0,5$.

2.11. Pemrograman R

Bahasa R merupakan sebuah proyek yang dirancang sebagai bahasa pemrograman yang gratis, *open source*, yang dapat digunakan sebagai pengganti dari bahasa pemrograman Splus, pada mulanya dikembangkan sebagai bahasa S di *AT&T Bell Labs*, dan sekarang dipasarkan oleh *Insightful Corporation of Seattle*, di Washington. R adalah sistem untuk komputasi statistik 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* (Spector, 2004). Logo dari bahasa pemrograman R sendiri dapat dilihat pada Gambar 2.24.



Gambar 2.24 Logo bahasa pemrograman R

Berikut adalah kelebihan dari penggunaan bahasa R (Ihaka & Gentleman, 2012):

a. 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 antarmuka pemrograman C, python, bahkan java yang tentu saja berkat usaha serta kerja keras para kontributor aktif proyek R. Jadi selain bahasa R ini cukup pintar, penggunaannya 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

b. Interaktif (*interactive*)

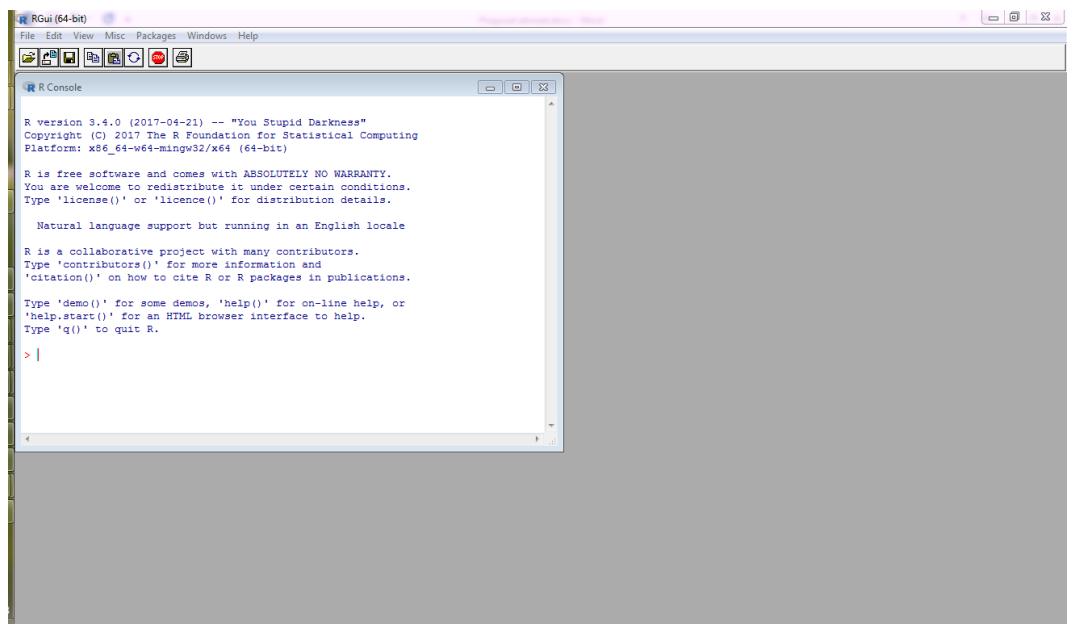
Pada saat ini analisis data membutuhkan pengoperasian yang interaktif. Apalagi jika data yang dianalisis adalah data yang bergerak. R dilengkapi dengan konektivitas ke database server, olap, maupun format data web service seperti XML, spreadsheet dan sebagainya. Sehingga apabila data set berubah hasil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

c. 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 proyekR.

d. 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.



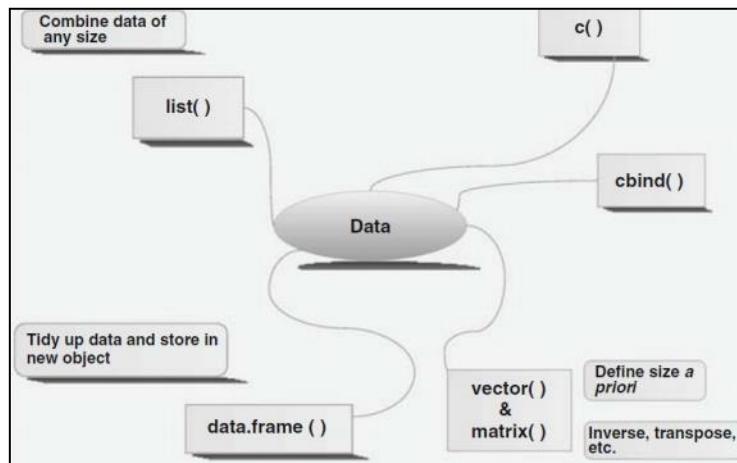
Gambar 2.25 Antarmuka R Graphical User Interface (RGui).

RGui merupakan tools dalam pemrograman R, antar muka tools ini dapat dilihat pada Gambar 2.25, 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.

2.11.1. Model Data dalam R

Pada bahasa R, data dipandang sebagai suatu objek yang memiliki suatu atribut dan berbagai fungsionalitas (Budiharto, 2013). 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 2.26.



Gambar 2.26 Model data dalam pemrograman R (Budiharto, 2013).

2.11.2. Contoh Kode Program Bahasa R

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

a. Menggabungkan data

Untuk menggabungkan data dalam bahasa R, dapat menggunakan fungsi *concatenate* (*c*). contoh dari penggunaan fungsi ini dapat dilihat pada Gambar 2.27.

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

Gambar 2.27 Operator *concatenate* dalam R.

Pada Gambar 2.28 berikut adalah contoh untuk menampilkan dua data pertama dalam vektor z yang telah dibuat.

```
x[1:2]
# output
# [1] 1 2
```

Gambar 2.28 Menampilkan data pertama hingga ke-dua dalam R.

Selain itu, untuk menampilkan jumlah dari seluruh elemen, dapat digunakan fungsi sum. Implementasi dari fungsi sum ini dapat dilihat pada Gambar 2.29.

```
sum(x)
# output
# [1] 15
```

Gambar 2.29 Penggunaan fungsi *sum* dalam R.

Contoh lainnya, untuk memasukan data string, dapat dilihat pada Gambar 2.30.

```
z <- c("muhammad", "ridwan", "UPI")
z
# output
# [1]"muhammad" "ridwan" "UPI"
```

Gambar 2.30 Penggunaan fungsi *concatenate* untuk *string* dalam R

b. Membuat matriks

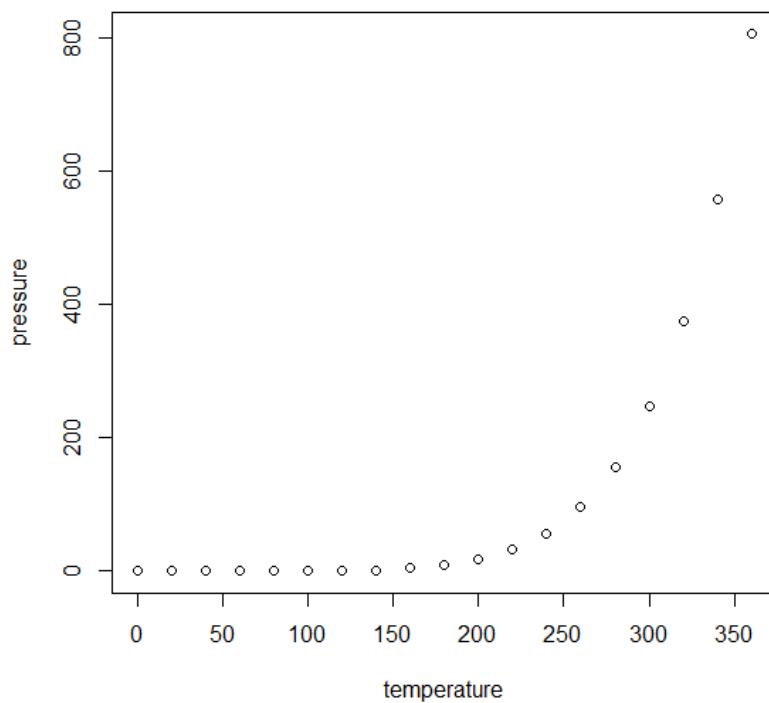
Matriks adalah data dua dimensi dimana sebagian besar fungsi-fungsi statistik dalam R dapat dianalisis dengan menggunakan bentuk matriks. Bentuk matriks ini juga banyak digunakan pada operasi fungsi-fungsi built-in untuk aljabar *linear* dalam R, seperti untuk penyelesaian suatu persamaan *linear*. Argumen yang diperlukan adalah elemen-elemen dari matriks, dan argumen optional yaitu banyaknya baris dan banyaknya kolom. Berikut contohnya ada pada Gambar 2.31.

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

Gambar 2.31 Pembuatan matriks dalam R.

c. Membuat *visualisasi* data

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 2.32.



Gambar 2.32 Contoh *Visualisasi* grafis dalam R.

d. 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 2.33.

```
i <- 1
n <- 10
for(i in i:n){
print("Hello World!")
}
# output
# [1] "Hello World!"
```

Gambar 2.33 Contoh perulangan dalam R.

e. 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 2.34.

```
i <- 1
n <- 10
if( n > i){
print("Hello World!")
}
# output
# [1] "Hello World!"
```

Gambar 2.34 Contoh implementasi *decission* dalam R.

f. Membuat fungsi

Dalam pemrograman terstruktur, salah satu hal yang penting adalah membuat fungsi. Dalam bahasa R, contoh pembuatan fungsi dapat dilihat pada Gambar 2.35.

```
Penjumlahan <- function(a,b){  
  X <- a+b  
  return(X)  
}  
Penjumlahan(1,2)  
# output  
# [1] 3
```

Gambar 2.35 Contoh fungsi dalam R.

2.12. Package Dalam 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 & Gentleman, 2012). Bahkan, sampai Desember 2018 bini banyak *packages* yang terpublish dalam *cran-r project* lebih dari 12800 *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 BioconductorProject (<http://www.bioconductor.org/>) (Riza *et al.*, 2016).

Dalam penelitian ini digunakan beberapa package, diantaranya:

1. *Shiny, package* ini berfungsi untuk menampilkan hasil keluaran dari konsol R kedalam bentuk web. Dimana pada penelitian ini konfigurasi *package shiny* terdapat dalam file *app.R*
2. *Data.Table, package* ini digunakan dalam proses pembacaan data, dimana sistem dapat melakukan proses pembacaan data dengan mode *force* atau sistem akan melakukan pembacaan dataset dan dapat membedakan mana dataset yang memiliki *header* maupun tidak.

3. *Smooth* dan *greybox*, *package* ini digunakan dalam proses prediksi data, dimana proses prediksi data dilakukan menggunakan algoritman *exponential smoothing*.
4. *Xts*, *package* ini berfungsi sebagai konverter data, dimana data masukan yang masih berbentuk tabel akan dikonversikan menjadi *time-series* sebelum nantinya dilakukan proses prediksi.
5. *Corrplot*, *package* ini berfungsi sebagai visualisator tabel matriks korelasi parameter.
6. *Mice*, *package* ini berfungsi sebagai *Missing Value Handling* dimana data akan diproses dan digunakan beberapa algoritman yang dapat memproses nilai NA dari data tersebut, seperti *random forest*,

Proses instalasi *package* pada R ini cukup mudah, dimana pengguna cukup mengetikan perintah `install.package("nama package")` pada konsol. Pastikan koneksi internet sudah terhubung sebelum melakukan proses instalasi *package*, untuk penjelasan lebih lengkapnya dapat dilihat pada gambar 2.36.

```
install.packages("Shiny")
```

Gambar 2.36 Contoh Instalasi *Package* dalam R

Setelah proses instalasi selesai, proses selanjutnya adalah meng-*load package* yang sudah diinstal, caranya pengguna cukup mengeksekusi perintan `library("nama package")` seperti pada gambar 2.37.

```
library(Shiny)
```

Gambar 2.37 Cara menggunakan *package* yang sudah diinstal

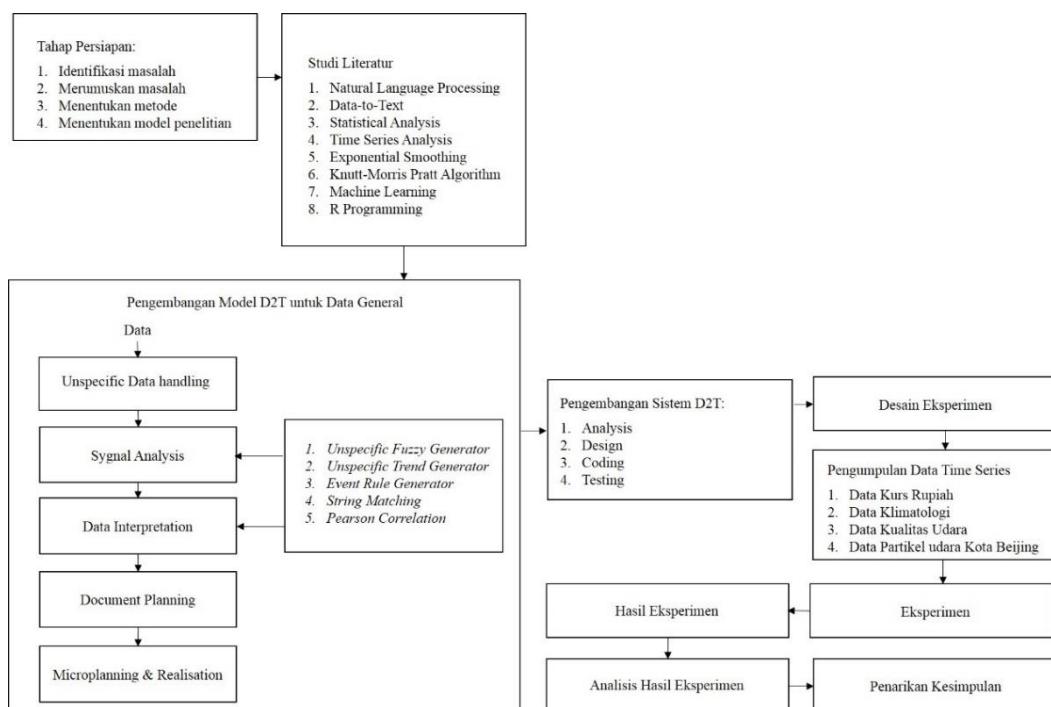
BAB III

METODE PENELITIAN

Pada bab ini akan dijelaskan mengenai metodologi penelitian, mulai dari desain penelitian, metode penelitian, alat penelitian dan data penelitian.

3.1. Desain Penelitian

Desain penelitian adalah kerangka kerja yang digunakan untuk melakukan penelitian. Pada bagian ini penulis akan memaparkan kerangka kerja dari mulai penelitian sampai selesai. Desain penelitian yang digunakan dalam pembangunan sistem *Data-to-Text* untuk data *unspecific* dengan pendekatan *Machine Learning* digambarkan pada gambar 3.1.



Gambar 3.1 Desain Penelitian Sistem D2T

Langkah-langkah penelitian yang dilakukan meliputi:

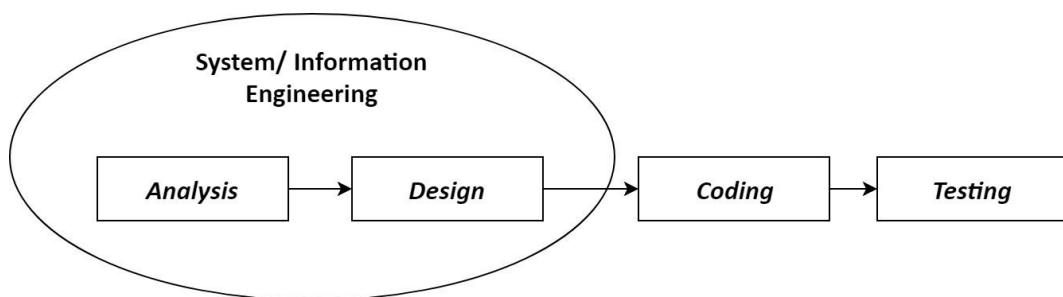
1. Tahap Persiapan adalah tahap awal dari penelitian, tahap ini dimulai dari identifikasi masalah, kemudian merumuskan masalah, lalu menentukan algoritma atau metode yang akan digunakan untuk menyelesaikan masalah tersebut, dalam kasus ini adalah pengembangan sistem D2T pada data *unspecific*.
2. Studi literatur merupakan kegiatan dimana peneliti melakukan tahap pembelajaran materi yang terkait dengan *Natural Language Processing*, *Natural Language Generation*, *Data-to-Text*, data *Time-Series*, pendekatan *Machine Learning* pada *Data-to-Text*, penggunaan bahasa pemrograman R, serta penelitian terkait dengan topik ini. Dalam mempelajari tentang bahasan di atas penulis mempelajari dari beberapa sumber, seperti buku, jurnal, juga internet, ataupun bahan bacaan lainnya yang didapat dari berbagai sumber.
3. Pengembangan Model D2T ini mengacu pada model yang dikembangkan oleh Reiter (2011) dan Putra et al., (2017) dengan sedikit modifikasi dimana ditambahkannya model proses *unspecific data handling*. Untuk proses utamanya terdiri dari *signal analysis*, *data interpretation*, *document planning*, hingga *microplanning & realisation*.
4. Setelah menentukan model sistem D2T yang akan dibangun, tahap selanjutnya adalah pengembangan sistem. Pengembangan sistem ini dilakukan dalam beberapa tahap sesuai dengan metode pengembangan *linear sequential model* yang akan dijelaskan pada sub-bab selanjutnya.
5. Setelah program selesai dibuat, tahap selanjutnya adalah menentukan desain eksperimen. Pada tahap ini dibuat sebuah rancangan untuk menguji coba sistem sesuai dengan tujuannya dan dilakukan penilaian berdasarkan aspek *Readability*, *Computation Time*, *Unspecific Handling*, dan perbandingan teks dengan grafis, serta perbandingan dengan penelitian terkait.
6. Data yang akan digunakan dalam penelitian ini adalah data *time series* dengan interval data per jam, harian, dan bulanan. Selain itu juga digunakan beberapa data dengan parameter *categorical* dan *numerical*. Sehingga diambil beberapa contoh data yang termasuk pada kategori tersebut, yakni

data nilai tukar rupiah terhadap mata uang lain (kurs) dari situs KEMENDAG, data klimatologi, data kualitas udara, dan data partikel udara Kota Beijing.

7. Eksperimen dilakukan dengan menggunakan keempat data yang sudah disebutkan sebelumnya. Pada setiap data tersebut, dilakukan simulasi sebanyak tiga kali, dimana simulasi pertama data akan dihilangkan *header* nya terlebih dahulu, lalu simulasi kedua akan digunakan data lengkap dengan *header* nya, dan simulasi ke-tiga data akan diproses dengan kustomisasi *corpus* dan *data description*.
8. Setelah melakukan eksperimen, langkah selanjutnya adalah menganalisis hasil eksperimen dengan menggunakan rancangan atau desain eksperimen yang sudah dijelaskan sebelumnya, lalu membandingkan hasil keluaran sistem dengan penelitian sebelumnya dan ditutup dengan langkah terakhir yaitu penarikan kesimpulan.

3.2. Metode Penelitian

Dalam penelitian ini, dilakukan pengembangan perangkat lunak menggunakan model *Linear Sequential*. *Linear Sequential* mengusulkan sebuah pendekatan kepada pengembangan perangkat lunak yang sistematik dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analis, desain, kode, pengujian, dan pemeliharaan. Berikut adalah proses gambaran dari *Linear Sequential Model* gambar 3.2.



Gambar 3.2 Model Linear *Sequential Model* (Pressman, 2001b)

1. *System / Informartion engineering* Merupakan bagian dari sebuah sistem terbesar yang mana dalam penggerajannya dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem dan mengalokasikannya ke dalam pembentukan perangkat lunak.

2. Analisis perangkat lunak merupakan tahap menganalisis hal-hal yang diperlukan dalam pembentukan sebuah perangkat lunak.
3. Desain merupakan beberapa langkah proses yang berfokus pada empat buah atribut yang berbeda dari program, yakni struktur data, arsitektur perangkat lunak, representasi antarmuka, dan sebuah algoritma.
4. *Coding* dilakukan untuk menerjemahkan pembuatan desain ke dalam bentuk yang bisa dimengerti oleh mesin. Sehingga komputer bisa merepresentasikan ke dalam bentuk perangkat lunak.
5. Tes merupakan langkah paling akhir yang dikerjakan, sebuah pengetesan pada perangkat lunak yang sudah melalui beberapa tahap dan dapat dipakai oleh user, dalam tes juga dapat dilakukan pengecekan apakah perangkat lunak yang dibuat sudah sesuai.

3.3. Alat dan Bahan Penelitian

Penelitian ini menggunakan seperangkap laptop yang dilengkapi perangkat lunak pendukung, dengan spesifikasi perangkat keras sebagai berikut:

1. Prosesor AMD APU A-10 5750M
2. Kartu Grafis AMD Radeon HD 8670M
3. *Random Access Memory* (RAM) 8 GB
4. *Hard Disk Drive* 1 TB
5. Monitor 16.5 inci dengan resolusi 1366x768 piksel

Adapun perangkat lunak yang digunakan adalah:

1. *Sistem Operasi Microsoft Windows 8.1 64-bit*
2. *RStudio*
3. *R i386 v3.5.0*
4. Google Chrome 64-bit v67.0.3396.99
5. *Microsoft Excel 2013*

Alat-alat penelitian tersebut digunakan untuk mengembangkan aplikasi yang nantinya akan digunakan untuk melakukan penelitian dan eksperimen.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini penulis menguraikan hasil dari implementasi dan uji coba sistem. Bab ini terdiri dari pengumpulan data, pengembangan model, pengembangan sistem, rancangan eksperimen, hasil eksperimen, analisis hasil eksperimen, perbandingan dengan penelitian terkait.

4.1. Pengumpulan Data

Data yang menjadi masukan untuk sistem ini adalah adalah data dengan jenis *time series* yang bersifat eksak yang berbentuk tabel khususnya data dengan ekstensi file .csv, atau format data tabel lainnya. Sistem ini mampu memproses data dengan tipe *numerical* maupun *categorical* sebagai masukannya. Data yang menjadi masukan haruslah berjenis *time-series* dengan format penulisan *date time* nya yaitu “mm/dd/yyyy hh:mm”, dimana penempatan parameter *date time* ini berada pada kolom pertama.. Selain itu juga digunakan beberapa data dari penelitian sebelumnya, guna untuk membandingkan hasil dari sistem yang akan dikembangkan dengan sistem yang dikembangkan sebelumnya. Sehingga digunakan beberapa contoh data yang memenuhi kriteria yang sudah dijelaskan sebelumnya yakni data historis nilai tukar mata uang asing terhadap rupiah (kurs), data klimatologi, data kualitas udara, dan data partikel udara Kota Beijing. Untuk lebih lengkapnya, pemaparan data secara detil akan disampaikan pada sub-bab selanjutnya.

4.1.1. Data Nilai Tukar Mata Uang Asing

Data diperoleh dari situs Kementerian Perdagangan Republik Indonesia (<http://www.kemendag.go.id>). Data berisikan nilai tukar mata uang asing terhadap mata uang rupiah atau kurs dengan rentang per bulan, mulai dari 1 Januari 2001 sampai dengan 1 Oktober 2018 dengan jumlah data sebanyak 215 baris. Untuk mengakses data yang tersedia, penulis mengunjungi situs Kemenag RI lalu memilih menu statistik yang berada pada bagian bawah situs, lalu penulis memilih menu nilai tukar rupiah dan memilih menu export to excel, atau bisa diakses melalui

alamat (www.kemendag.go.id/id/economic-profile/economic-indicators/exchange-rates).

Setelah melakukan serangkaian proses untuk mengakses data, data yang didapatkan berbentuk tabel dengan jumlah baris sebanyak 211 baris, dan jumlah atribut sebanyak 10 atribut, dengan nilai atribut mewakili nilai tukar mata uang tersebut terhadap rupiah. Berikut adalah atribut-atribut yang terdapat pada data tersebut:

- Waktu data
- Dolar Amerika (USD)
- Yen Jepang (JPY)
- Poundsterling Inggris (GBP)
- Franc Swiss (CHF)
- Dolar Singapore (SGD)
- Rigit Malaysia (MYR)
- Dolar Hongkong (HKD)
- Dolar Australia (AUD)
- Dolar Canada (CAD)

Untuk lebih lengkapnya data dapat dilihat pada lampiran, namun penulis mengutip beberapa data yang disajikan pada tabel 4.1, seperti yang sudah dijelaskan sebelumnya data berisikan beberapa atribut yang merepresentasikan nilai tukar mata uang asing terhadap rupiah.

Tabel 4.1 Data Nilai Tukar Mata Uang Asing

Date	Time	USD	JPY	GBP	CHF	...	CAD
01/01/2001	00:00	9.45000	8.13149	13.81498	5.74364	...	6.28451
02/01/2001	00:00	9.83500	8.45297	14.17963	5.85644	...	6.43274
03/01/2001	00:00	10.40000	8.37000	14.85227	6.01400	...	6.60907
04/01/2001	00:00	11.67500	9.42066	16.74548	6.76774	...	7.56988
05/01/2001	00:00	11.05800	9.21733	15.76486	6.21098	...	7.15591
...
06/01/2018	00:00	14.404	13.03707	18.83468	14.4423	...	10.86439
07/01/2018	00:00	14.413	12.98294	18.91203	14.5925	...	11.03346
08/01/2018	00:00	14.711	13.25614	19.14198	15.18951	...	11.30
09/01/2018	00:00	14.929	13.14462	19.52714	15.28281	...	11.46622
10/01/2018	00:00	15.227	13.45914	19.35428	15.152	...	11.6103

4.1.2. Data Klimatologi

Data ini diperoleh dari stasiun pemantauan milik lembaga meteorologi dan klimatologi Galicia yaitu MeteoGalicia yang berlokasi di kota Mabegondo, Provinsi A Coruna, daerah komUnitas otonom Galicia, Spanyol. Data ini berisikan data klimatologi dengan rentang per hari mulai dari tanggal 06 Juli 2016 sampai dengan 6 Juli 2017 sebanyak 365 baris data. Data ini merupakan data yang digunakan pada penelitian sebelumnya, yaitu DWP yang dibahas pada bab 2 sebelumnya (Putra *et al.*, 2017). Parameter-parameter yang terdapat pada data ini diantaranya:

- Waktu data
- Cakupan awan rata-rata per hari dengan satuan % (Persen)
- Suhu rata-rata per hari dengan satuan °C (derajat celcius)
- Kecepatan angin rata-rata per hari dengan satuan km/h (Kilometer per jam)
- Arah angin rata-rata per hari dengan satuan °(derajat)
- Curah hujan rata-rata per hari dengan satuan L/m² (Liter per meter kuadrat)

Parameter tersebut merupakan hasil analisis pada penelitian DWP sebelumnya (Putra *et al.*, 2017). Untuk lebih lengkapnya data dapat dilihat pada lampiran dan untuk kutipan datanya bisa dilihat pada tabel 4.2.

Tabel 4.2 Kutipan data klimatologi

DateTime	CloudCoverage	Temperature	WindSpeed	WindDirection	Rainfall
07/06/2016 00:00	40.8	21.3	5.47	315	0
07/07/2016 00:00	20.9	20.1	6.41	315	0
07/08/2016 00:00	27.2	19.5	7.02	315	0
07/09/2016 00:00	23.2	19.1	5.94	315	0
07/10/2016 00:00	77.5	18.7	5.44	180	0
...
07/02/2017 00:00	12.6	18.9	7.34	315	0
07/03/2017 00:00	13.3	21.8	4.86	315	0
07/04/2017 00:00	18.7	24	6.59	225	0
07/05/2017 00:00	81.1	19.2	8.35	225	0
07/06/2017 00:00	58.3	17.9	6.48	315	0

4.1.3. Data Kualitas Udara

Data ini hampir mirip dengan data klimatologi sebelumnya, data ini diperoleh dari stasiun pemantauan milik lembaga meteorologi dan klimatologi Galicia yaitu MeteoGalicia. Data ini berisikan nilai-nilai partikel yang terkandung dalam udara dengan rentang per hari mulai dari tanggal 06 Juli 2016 sampai dengan 6 Juli 2017 sebanyak 365 baris data. Parameter-parameter yang terdapat pada data ini diantaranya:

- Waktu Data
- 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)

Data ini juga digunakan pada penelitian DWP sebelumnya (Putra *et al.*, 2017). Tabel 4.3 merupakan kutipan data yang digunakan selama penelitian, untuk 366 baris data secara lengkap dapat dilihat pada lampiran.

Tabel 4.3 Kutipan data Kualitas Udara

DateTime	CO	NO	NO2	NOX	O3	PM10	PM25	SO2
07/06/2016 00:00	0.13	3	15	19	51	18	10	1
07/07/2016 00:00	0.11	1	10	10	56	14	7	1
07/08/2016 00:00	0.1	1	8	8	59	12	8	1
07/09/2016 00:00	0.1	2	10	11	57	12	7	1
07/10/2016 00:00	0.11	2	10	12	53	12	11	1
...
07/02/2017 00:00	0.17	2	35	37	10	14	20	7
07/03/2017 00:00	0.17	23	43	78	1	11	14	9
07/04/2017 00:00	0.17	31	42	90	1	0	14	9
07/05/2017 00:00	0.18	32	41	90	1	0	12	7
07/06/2017 00:00	0.18	32	61	90	1	0	12	7

4.1.4. Data Partikel Udara Kota Beijing

Data ini hampir mirip dengan data kualitas udara, data ini diperoleh dari stasiun pemantuan milik lembaga meteorologi dan klimatologi kota Beijing. Data ini berisikan nilai-nilai partikel yang terkandung dalam udara dengan rentang per jam selama dua tahun penuh mulai dari tanggal 01 Januari 2010 00:00 sampai dengan 31 Desember 2011 23:00 sebanyak 17521 baris data. Parameter-parameter yang terdapat pada data ini diantaranya:

- Waktu Data
- Data Konsentrasi Particulate Matter 2.5 mikronmeter ($\mu\text{g}/\text{m}^3$)
- DEWP: Dew Point atau titik embun
- TEMP: Temperature atau suhu
- PRES: Pressure atau tekanan (hPa)
- cbwd: Combined wind direction atau arah angin
- Iws: Cumulated wind speed atau kecepatan angin (m/s)
- Is: Cumulated hours of snow atau akumulasi jam salju
- Ir: Cumulated hours of rain atau akumulasi jam hujan

Tabel 4.4 merupakan kutipan data yang digunakan selama penelitian, untuk 17521 baris data secara lengkap dapat dilihat pada lampiran.

Tabel 4.4 Kutipan data Partikel Udara Kota Beijing

Date Time	PM2.5	DEWP	TEMP	PRES	CBWD	LWS	IS	IR
01/01/2010 00:00	138	-21	-11	1021	NW	1.79	0	0
01/01/2010 01:00	125	-21	-12	1020	NW	4.92	0	0
01/01/2010 02:00	249	-21	-11	1019	NW	6.71	0	0
01/01/2010 03:00	210	-21	-14	1019	NW	9.84	0	0
01/01/2010 04:00	98	-20	-12	1018	NW	12.97	0	0
...
01/31/2011 19:00	54	-15	-1	1025	NE	2.68	0	0
01/31/2011 20:00	71	-13	-1	1025	cv	0.89	0	0
01/31/2011 21:00	129	-8	-1	1026	SE	5.81	0	0
01/31/2011 22:00	145	-7	-2	1027	SE	10.73	0	0
01/31/2011 23:00	101	-7	-2	1027	SE	13.86	0	0

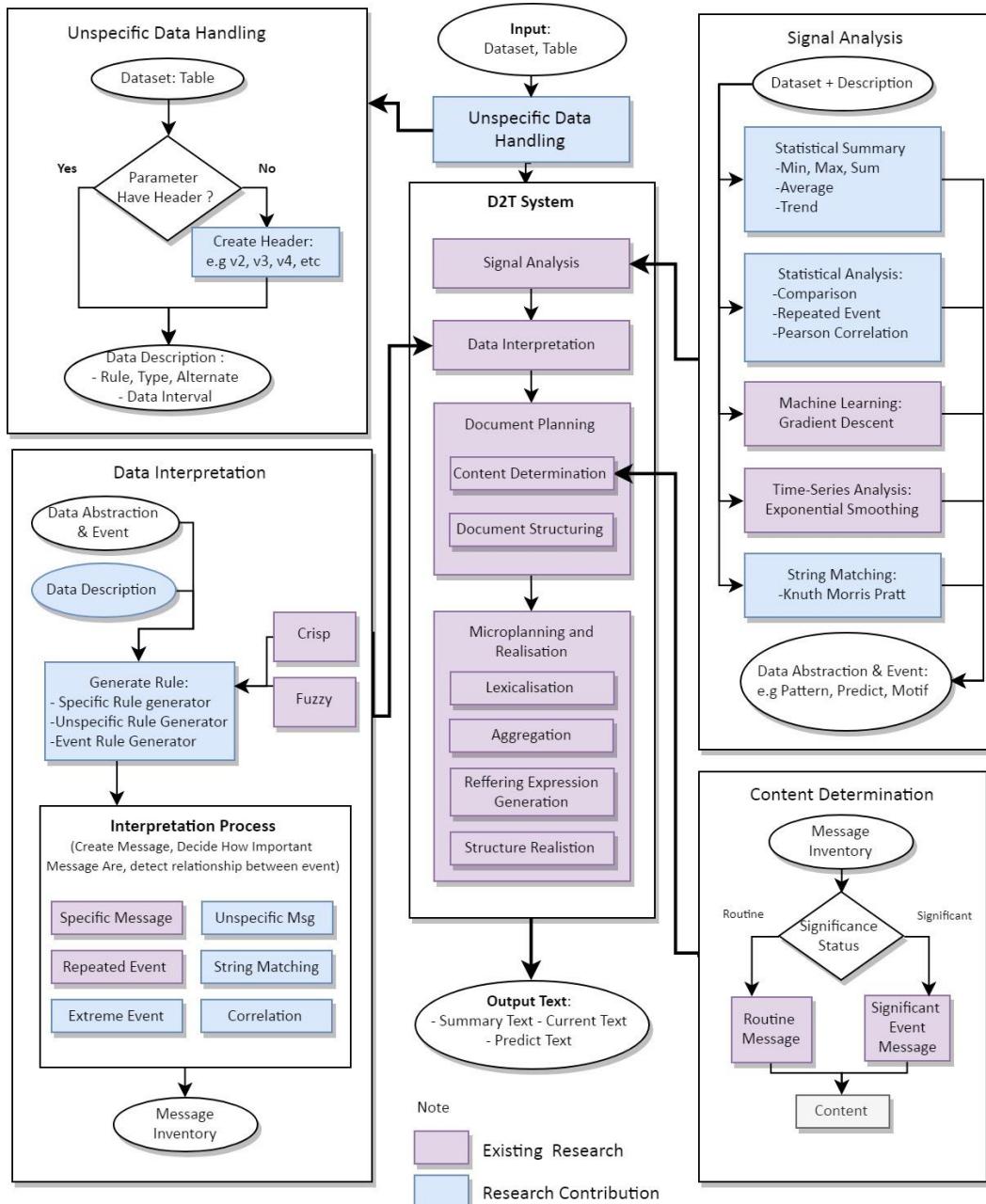
4.2. Model Sistem *Data-to-text*

Model sistem *Data-to-Text* (D2T) pada penelitian ini mengacu pada arsitektur sistem D2T yang dikembangkan oleh Reiter (2011), dan beberapa bagian yang dikembangkan oleh Putra (2017). Dengan menggunakan konsep *Fuzzy Rule-based* dan *Machine Learning* pada bagian *Signal Analysis* dan *Data Interpretation*, sistem yang dibangun dapat menerima masukan berupa data *unspecific* atau data yang berasal dari bidang apapun baik data tersebut mempunyai informasi berupa *header*, tipe data, *rule* maupun tidak.

Keluaran yang dihasilkan oleh sistem ini berupa teks yang terdiri dari tiga bagian, bagian pertama merupakan deskripsi dari ringkasan data selama periode tertentu, bagian ke-dua merupakan deskripsi dari baris terakhir pada n buah data, bagian ke-tiga merupakan informasi dari hasil prediksi untuk baris data ke $n+1$.

Model penelitian pada gambar 4.1 memperlihatkan perbedaan yang terdapat pada penelitian ini yaitu terdapatnya bagian *Unspecific Data Handler* dan pengaplikasian *Machine Learning* dan *Fuzzy Rule-based* pada bagian *Signal Analysis* dan *Data Interpretation*. Saat pertama kali data masuk, data akan diproses pada bagian *Unspecific Data Handler*, bagian ini akan mengelompokan apakah data memiliki informasi berupa *header* atau tidak, lalu sistem akan secara otomatis melihat apakah sudah terdapat *data description* untuk setiap parameter sesuai dengan *header* parameter tersebut atau tidak, jika tidak terdapat *data description*

maka sistem akan membuat *data description* dengan nilai *default* sehingga sistem dapat memproses data *unspecific* yang terdapat pada dataset.



Gambar 4.1 Model *Data-to-text* untuk Data *Unspecific*

Perbedaan lain terdapat pada bagian *Signal Analysis*, pada bagian ini digunakan *String Matching*, *Machine Learning*, *Statistical Analysis*, dan *Time-Series Analysis* untuk mendapatkan pola diskret yang akan digunakan sebagai masukan untuk tahap *Data Interpretation*. Pola yang sudah didapatkan pada tahap *Signal Analysis* ini akan diproses pada bagian *Data Interpretation*, dimana data

akan diinterpretasikan sesuai tipe *Rule-based* yang terdapat pada *data description*, ada dua pilihan *Rule-based* untuk menginterpretasikan data yaitu, *Fuzzy* dan *Crisp*. Lalu dilanjutkan dengan serangkaian proses sehingga dihasilkan teks berita, serangkaian proses tersebut dapat dilihat secara utuh pada gambar 4.1.

Sub-proses dari model sistem D2T yang akan dibangun akan dibahas pada sub-bab selanjutnya.

4.2.1. Model Komputasi untuk *Unspecific Data Handling*

Proses *Unspecific Data Handling* bertujuan untuk pengolahan dan pemrosesan data, tahapan ini bertujuan untuk menentukan interpretasi data dan penggunaan rule untuk data *unspecific* sebelum digunakan untuk proses selanjutnya. Suatu data dikatakan *unspecific* jika data tersebut tidak memiliki cara pentinterpretasian. Jika pada penelitian sebelumnya, sistem D2T hanya mampu menerima masukan berupa data yang terbatas pada suatu bidang atau *specific*, seperti data klimatologi (Putra *et al.*, 2017), keuangan, dan lain-lain. Maka pada penelitian ini, sistem D2T yang dikembangkan mampu menerima berbagai inputan data dari bidang apapun atau data *unspecific*, bahkan untuk data yang tidak dapat dikenali ataupun yang tidak memiliki *header* dan cara penginterpretasian sekalipun. Untuk lebih jelasnya, input, proses, dan output pada tahap ini adalah sebagai berikut:

- Input : *Dataset*
- Proses : *Unspecific Data Handling*
- Output : *Dataset with Header, Data Description*.

Pertama-tama data yang masuk akan dilihat apakah dataset yang masuk setiap parameternya memiliki *header* atau tidak. Jika semua atau beberapa parameter pada suatu dataset ada yang tidak memiliki *header*, sistem akan memberi *header* pada parameter tersebut secara otomatis. Khusus untuk *header* pada parameter atau kolom pertama dari dataset, sistem akan mengubah *header* menjadi “*Date*”**Time**. Sedangkan untuk parameter selanjutnya yang tidak memiliki *header*, sistem akan menamai *header* dengan nama *v2*, *v3*, *v4*, dan seterusnya. Untuk lebih lengkapnya, bisa dilihat pada gambar 4.2 .



			test1	...	test2				DateTime	v2	v3	test1	v4	...	test2
01/01/2001 0:00	9.45000	8.13149	13.81498	5.74364	...	6.28451			01/01/2001 0:00	9.45000	8.13149	13.81498	5.74364	...	6.28451
02/01/2001 0:00	9.83500	8.45297	14.17963	5.85644	...	6.43274			02/01/2001 0:00	9.83500	8.45297	14.17963	5.85644	...	6.43274
03/01/2001 0:00	10.40000	8.37000	14.85227	6.01400	...	6.60907			03/01/2001 0:00	10.40000	8.37000	14.85227	6.01400	...	6.60907
04/01/2001 0:00	11.67500	9.42066	16.74548	6.76774	...	7.56988			04/01/2001 0:00	11.67500	9.42066	16.74548	6.76774	...	7.56988
05/01/2001 0:00	11.05800	9.21733	15.76486	6.21098	...	7.15591			05/01/2001 0:00	11.05800	9.21733	15.76486	6.21098	...	7.15591
...
03/01/2018 0:00	13.75600	12.90553	19.36503	14.38763	...	10.64665			03/01/2018 0:00	13.75600	12.90553	19.36503	14.38763	...	10.64665
04/01/2018 0:00	13.87700	12.71895	19.11904	14.04698	...	10.80554			04/01/2018 0:00	13.87700	12.71895	19.11904	14.04698	...	10.80554
05/01/2018 0:00	13.95100	12.83973	18.55344	14.11117	...	10.83194			05/01/2018 0:00	13.95100	12.83973	18.55344	14.11117	...	10.83194
06/01/2018 0:00	14.40400	13.03707	18.83468	14.44230	...	10.86439			06/01/2018 0:00	14.40400	13.03707	18.83468	14.44230	...	10.86439
07/01/2018 0:00	14.41300	12.98294	18.91203	14.59250	...	11.03346			07/01/2018 0:00	14.41300	12.98294	18.91203	14.59250	...	11.03346

Setelah proses penamaan

Setelah proses penamaan

Gambar 4.2 Penamaan header pada Proses *Unspecific Data Handler*

Setelah penamaan *header*, proses selanjutnya yaitu penentuan *data description*. *Data description* merupakan aturan atau cara bagaimana suatu parameter akan diolah dan diproses dalam sistem nantinya, contoh *data description* ini dapat dilihat pada tabel 4.5. *Data description* disimpan pada file *datadescription.csv* yang terdapat pada folder *Config*. Terdapat 3 jenis atribut *data description* yang disimpan, yaitu:

1. Atribut *Type*, atribut ini akan menentukan bagaimana suatu parameter diproses. Terdapat dua kategori yaitu, *numerical* dan *categorical*. Untuk parameter dengan tipe *numerical*, sistem akan melakukan proses analisis data menggunakan *Statistical Tools* seperti *min*, *max*, *average*, *Statistical analysis*, penerapan *Machine Learning*, *time-series analysis* dan lainnya. Untuk parameter dengan tipe *categorical*, sistem hanya akan melakukan pemrosesan *Repeated Event* dan penerapan *String Matching* menggunakan algoritma *Knuth Morris Pratt* (KMP) pada parameter tersebut.
2. Atribut *Rule*, atribut ini menentukan bagaimana nantinya suatu parameter akan diinterpretasikan. Untuk parameter dengan *rule fuzzy*, parameter tersebut akan diinterpretasikan menggunakan *Fuzzy membership function*. Sama halnya untuk parameter dengan *rule crisp*, maka parameter tersebut akan diinterpretasikan menggunakan *Crisp membership function*.
3. Atribut *Alternate*, atribut ini dapat digunakan untuk mengubah nama parameter pada teks keluaran nantinya.

Tabel 4.5 Contoh *data description* pada file *datadescription.csv*

ColName	Type	Rule	Alternate
AirQuality	numeric	crisp	Air Quality
WindSpeed	numeric	crisp	Wind Speed
WindDirection	numeric	crisp	Wind Direction
CloudCoverage	numeric	crisp	Cloud Coverage
Temperature	numeric	fuzzy	NA
Rainfall	numeric	fuzzy	NA
USD	NA	NA	U.S. Dollar
JPY	NA	NA	Japan Yen
GBP	NA	NA	Great British Pounds
CHF	NA	NA	Confoederatio Helvetica Franc
SGD	NA	NA	Singapore Dollar
MYR	NA	NA	Malaysian Ringgit
HKD	NA	NA	Hong Kong dollar
AUD	NA	NA	Australian Dollar
CAD	NA	NA	Canadian Dollar

Data description ini bersifat opsional, pengguna dapat mengubah atau menambahkan *data description* untuk suatu parameter pada file *datadescription.csv* di dalam folder *Config*. Namun jika tidak ditemukan *data description* untuk suatu parameter pada dataset, maka sistem akan secara otomatis melakukan pemrosesan data secara *unspecific* sesuai tipe datanya, lalu diinterpretasikan menggunakan *Unspecific Fuzzy membership function*. Untuk menambahkan atau mengubah *data description*, dapat dilakukan dengan langkah-langkah berikut ini:

1. Buka file *datadescription.csv* pada folder *Config*.
2. Pengguna dapat mengubah atau menambahkan *data description* dengan memasukan atau mengubah nama parameter, tipe data, *rule*, dan *alternate* pada file tersebut. Jika pengguna hanya mengisi beberapa *data description* saja, maka pengguna dapat menuliskan nilai NA untuk attribut *data description* yang kosong atau tidak diisi.
3. Untuk *data description rule*, jika pengguna memasukan *data description* tersebut, maka pengguna harus menambahkan file pada folder *Corpus* dengan nama file sesuai dengan nama parameter yang diikuti dengan

Adjective.csv. Seperti, *AirQualityAdjective.csv*, *TemperatureAdjective.csv*, dan *RainfallAdjective.csv*.

4.2.2. Model Komputasi untuk *Signal Analysis*

Proses *Signal Analysis* merupakan tahapan awal dari sistem D2T, tahapan ini bertujuan untuk mendeteksi dan menganalisis pola-pola diskret yang terdapat pada dataset (Reiter, 2011). Selain itu, pada proses *Signal Analysis* ini dihasilkan juga ringkasan data seperti pada sistem *BABYTALK Family Sistem* yang dipaparkan oleh (Portet *et al.*, 2009), sistem tersebut melakukan peringkasan data untuk mencari ringkasan kejadian selama 45 menit, dan juga pada sistem DWP yang menghasilkan ringkasan data klimatologi selama satu bulan dari data klimatologi selama satu tahun (Putra *et al.*, 2017). Selain peringkasan data, pada tahap ini sistem juga melakukan analisis untuk mencari pola kejadian seperti *String Matching*, *Repeated Event*, *Extreme Event*, analisis untuk data ke-*n* (data terakhir), dan prediksi data untuk data ke-*n+1*. Setelah melakukan serangkaian analisis pada tahap ini, hasil dari analisis tersebut akan disimpan dan digabungkan dalam bentuk *Data Abstraction and Event*. Untuk lebih jelasnya, input, proses, dan output pada tahap ini adalah sebagai berikut:

- Input : *Dataset with description*
- Proses : *Signal Analysis*
- Output : *Data Abstraction and Event*

Sub-proses pada tahap ini akan dibahas pada sub-bab berikutnya.

a. Proses *Signal Analysis* untuk *Statistical Summary*

Mengacu pada penelitian sebelumnya, untuk proses *Signal Analysis* pada sistem DWP dilakukan serangkaian analisis dan peringkasan data sehingga didapatkan tiga point utama, yakni untuk setiap parameternya dilakukan analisis menggunakan *Statistical Tools* untuk mendapatkan sinyal bulanan dari masing-masing parameter (*Monthly_Message*), khusus untuk parameter *Rainfall* dilakukan analisis pencarian perulangan suatu pola (*Repeated Event*), lalu analisis *Extreme Event* untuk parameter *Rainfall*, *WindSpeed*, dan *Temperature* (Putra *et al.*, 2017). Penelitian ini juga menggunakan beberapa analisis yang dilakukan pada sistem DWP

seperti yang sudah dijelaskan sebelumnya, namun untuk proses *Repeated Event* dan *Extreme Event* dilakukan pada proses yang berbeda. Sehingga pada tahap *Statistical Summary* ini, hanya dilakukan analisis sinyal menggunakan *Statistical Tools* dan ditambahkannya analisis sinyal untuk menentukan *trend* pada setiap parameter. Ringkasan ini akan digunakan sebagai dasar pemrosesan data untuk proses analisis yang lainnya. Semua analisis yang ada dalam proses peringkasan data ini hanya diterapkan pada parameter dengan tipe *numerical*.

Data statistik yang dicari pada proses ini berupa nilai minimum (nilai, tanggal, indeks data), nilai maksimum (nilai, tanggal, indeks data), jumlah nilai parameter, nilai rata-rata, dan representasi *trend* untuk setiap parameter dengan tipe *numerical* yang ada, sehingga jika ada n buah parameter bertipe *numerical*, maka sinyal yang dihasilkan sebanyak $n*10$ buah, dengan 1 kolom tambahan yang memuat nama parameter.

Tabel 4.6 Contoh data *dummy* untuk kasus sederhana

Date	Time	Param1	Param2	Param3
01/01/2019	00:00	0.13	15	High
01/02/2019	00:00	0.11	10	Low
01/03/2019	00:00	0.1	8	Medium
01/04/2019	00:00	0.1	10	Medium
01/05/2019	00:00	0.12	10	Medium
01/06/2019	00:00	0.1	18	Very Low
01/07/2019	00:00	0.13	31	Low
01/08/2019	00:00	0.12	24	High
01/09/2019	00:00	0.15	27	Medium
01/10/2019	00:00	0.16	26	High
01/11/2019	00:00	0.16	24	Very Low
01/12/2019	00:00	0.18	27	Very High
01/13/2019	00:00	0.17	21	Very High
01/14/2019	00:00	0.25	35	Low
01/15/2019	00:00	0.14	14	Medium
01/16/2019	00:00	0.14	14	Medium
01/17/2019	00:00	0.15	15	Medium
01/18/2019	00:00	0.14	16	Very Low
01/19/2019	00:00	0.17	23	Low
01/20/2019	00:00	0.19	30	High

Sebagai contoh, penulis menggunakan data *dummy* seperti yang ditunjukkan pada tabel 4.6 dimana data tersebut dipilih dikarenakan pada data tersebut memuat semua contoh-contoh analisis yang akan dipaparkan lebih jelas pada sub-bab selanjutnya. Sehingga setelah dilakukan analisis *Statistical Summary* pada data *dummy* pada tabel 4.6 tersebut maka

diperoleh hasil analisis seperti pada tabel 4.7, terlihat pada tabel tersebut, setelah didapatkan hasil dari *Statistical Tools* berupa nilai tertinggi dan nilai terendah bersama waktu terjadi dan nilainya, dan pemrosesan statistik lain seperti rata-rata dan *trend*.

Tabel 4.7 Hasil pendektsian sinyal *statistical summary* untuk kasus data sederhana

ColName	MaxDate	Max Value	Max Index	Min Date	Min Value	Min Index	Sum Value	Average	Trend
Param1	01/14/2019 00:00	0.25	14	01/03/2019 00:00	0.1	3	2.91	0.145 5	+
Param2	01/14/2019 00:00	35	14	01/03/2019 00:00	8	3	398	19.9	+

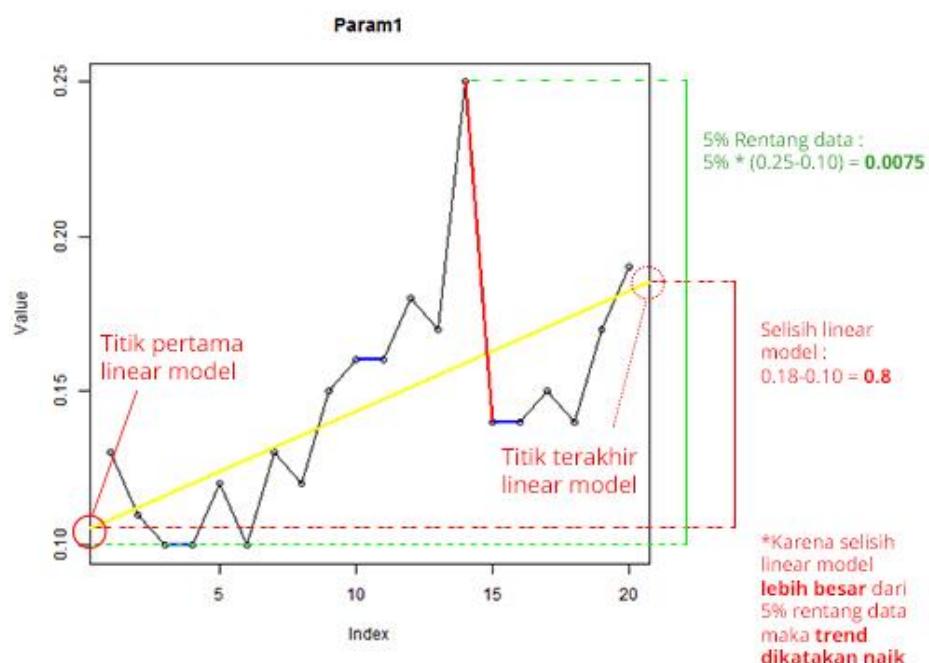
Sebagai contoh lainnya jika digunakan data partikel udara pada yang tabel 4.4 lalu dilakukan proses peringkasan data, maka didapatkan sinyal-sinyal yang kemudian disimpan dalam sebuah variabel bertipe *data frame* dengan 7 buah baris sesuai jumlah parameter dengan tipe *numerical*, dan 10 kolom sesuai banyaknya sinyal-sinyal yang disimpan, untuk lebih jelasnya dapat dilihat pada tabel 4.8.

Tabel 4.8 Hasil pendektsian sinyal *statistical summary* untuk kasus data partikel udara

ColName	MaxDate	Max Value	Max Index	Min Date	Min Value	Min Index	Sum Value	Average	Trend
pm2.5	02/14/20 10 01:00	980	1058	09/21/20 10 03:00	1	6316	176816 4	100.9 2	0
DEWP	07/29/20 10 17:00	28	5034	12/15/20 10 03:00	-28	8356	32649	1.86	+
TEMP	07/05/20 10 15:00	41	4456	01/05/20 10 02:00	-19	99	211972	12.10	+
PRES	01/27/20 11 10:00	1045	9395	06/07/20 11 15:00	993	1254	178148 39.5	1016. 83	0
Iws	01/01/20 11 04:00	585.6	8765	01/07/20 10 01:00	0.45	146	478522 .16	27.31	0
Is	01/03/20 10 21:00	27	70	01/01/20 10 00:00	0	1	1077	0.06	0
Ir	09/18/20 10 08:00	36	6249	01/01/20 10 00:00	0	1	3578	0.20	0

Dalam menentukan *trend* sebuah parameter, digunakan *Linear Model* dan simbol untuk merepresentasikan apakah *trend* dari parameter tersebut menaik (+), menurun (-), atau justru konstan (0). Namun perlu diketahui, dalam penentuan *trend* ini diterapkan *Minimum Treshold* sebesar 5%, dimana sebuah parameter akan tetap dikatakan konstan jika selisih titik terakhir dan titik pertama model linear tidak melebihi 5% dari rentang

keseluruhan data. Sebagai contoh penulis merepresentasikan parameter pertama data *dummy* pada tabel 4.6 menggunakan plot yang ditampilkan pada gambar 4.3, pada gambar tersebut terlihat selisih model linear sebesar 0.8 dan 5% rentang data sebesar 0.0075, karena selisih linear model lebih besar dari 5% rentang data dan selisih linear model bernilai positif maka *trend* dari parameter tersebut dikatakan menaik. Begitu juga sebaliknya, jika selisih linear model lebih besar dari 5% rentang data dan selisih linear model bernilai negatif maka *trend* dari suatu parameter dikatakan menurun. Namun, jika selisih model linear suatu parameter tidak melebihi dari 5% rentang data, maka parameter tersebut dikatakan konstan.



Gambar 4.3 Contoh penentuan *trend* pada data *dummy* untuk parameter pertama

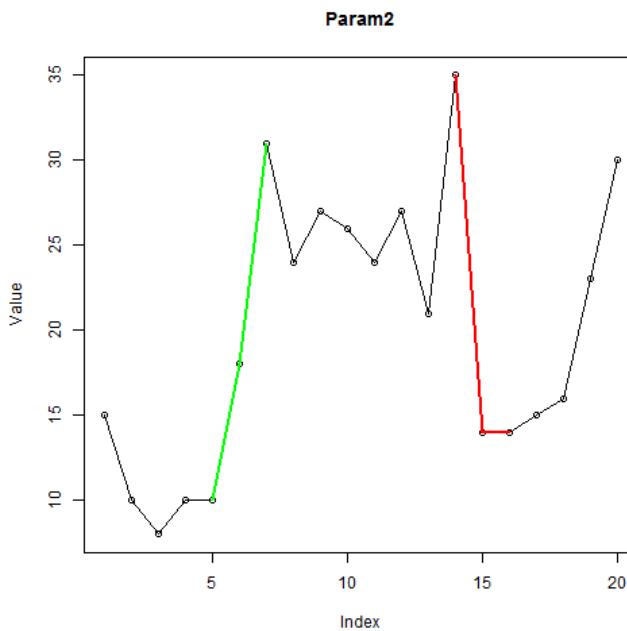
Selain itu juga pada proses *Statistical Summary* ini diambil beberapa data yang akan diproses pada proses pendekripsiannya selanjutnya. Data yang diambil adalah rata-rata setiap parameter atau *resume data*, data ke-dua terakhir atau *last data* dan data paling terakhir atau *current data*. Sehingga jika kita menggunakan data *dummy* pada tabel 4.6, didapatkan ringkasan data seperti yang ditampilkan pada tabel 4.9 berikut.

Tabel 4.9 Ringkasan beberapa data yang akan digunakan pada proses selanjutnya

	Param1	Param2
Resume data	0.1455	19.9
Last (n-1) data	0.17	23
Current data	0.19	30

b. Proses *Signal Analysis* untuk *Extreme Event*

Proses ini bertujuan untuk mencari pola-pola perubahan yang ekstrem baik berupa kenaikan ataupun penurunan pada suatu parameter dengan tipe *numerical*. Contohnya ketika suhu hari ini sangat panas, tiba-tiba sehari kemudian berubah menjadi sangat dingin, inilah perubahan ekstrem yang dimaksudkan. Untuk melakukan proses pencarian nilai ekstrem, pertama-tama setiap kenaikan dan setiap penurunan suatu parameter dijumlahkan terlebih dahulu. Setelah didapatkan jumlah kenaikan atau penurunannya, langkah selanjutnya adalah menyimpan nilai kenaikan dan penurunan tertinggi bersama indeks data dimana kenaikan atau penurunan tersebut berawal dan berakhir (dalam bentuk numerik). Sinyal *extreme event* ini dapat dilihat pada gambar 4.4, dimana gambar tersebut merupakan hasil plot dari parameter ke-dua pada data *dummy* pada tabel 4.6, pada gambar tersebut kita dapat melihat ada garis berwarna hijau dan merah yang merupakan *extreme event*, dimana garis berwarna hijau merepresentasikan kenaikan ekstrem dan garis berwarna merah merepresentasikan penurunan yang ekstrem.



Gambar 4.4 Hasil plot parameter ke-dua pada data *dummy*, warna hijau merepresentasikan kenaikan ekstrem, dan warna merah merepresentasikan penurunan ekstrem

Hasil dari *signal analysis* untuk *extreme event* berupa *data frame* seperti pada tabel 4.10 berisikan nilai kenaikan dan penurunan tertinggi bersama indeks-indeksnya. Sebuah data akan dikatakan termasuk pada *extreme event* jika kenaikan atau penurunannya melebihi 65% dari interval data. Misalkan pada parameter ke-dua pada data *dummy* yang digambarkan pada gambar 4.4 terdapat kenaikan sebesar 21 (kenaikan dari nilai 10 menuju 31), jika dibandingkan dengan interval parameter tersebut sebesar 27 (didapatkan dari 35-8), maka kenaikan 21 lebih besar dibandingkan 65% interval data sebesar 17.55 (didapatkan dari $27 \times 65\%$) sehingga kenaikan dikatakan ekstrim, hal ini berlaku juga untuk penurunan.

Tabel 4.10 Hasil *signal analysis* untuk *extreme event* pada kasus data *dummy*

Column Name	Inc Value	IncStart Index	IncEnd Index	IncInterp rater	Dec Value	DecStart Index	DecEnd Index	DecInter preter
Param1	0.08	13	14	normal	-0.11	14	16	extreme
Param2	21	5	7	extreme	-21	14	16	extreme

c. Proses *Signal Analysis* untuk *Comparison*

Proses pendekripsi sinyal *comparison* ini bertujuan untuk membandingkan data terakhir dengan beberapa data terakhir sesuai dengan interval data dari data masukan. Untuk data dengan interval per jam maka data terakhir akan dibandingkan data pada hari sebelumnya atau data 24 jam terakhir. Untuk data dengan interval harian maka data akan dibandingkan dengan data seminggu sebelumnya, atau data ke-7 terakhir. Untuk data dengan interval per minggu, akan dibandingkan dengan data pada satu bulan sebelumnya, begitu juga dengan data dengan interval perbulan akan dibandingkan dengan data setahun terakhir pada bulan yang sama. Sedangkan untuk data dengan interval per tahun maka akan dibandingkan dengan data kuarter sebelumnya. Jika data terakhir lebih besar dari pada data sebelumnya maka akan dihasilkan sinyal “higher than”, jika lebih kecil dibandingkan data sebelumnya maka akan dihasilkan sinyal “lower than”, sedangkan jika sama maka akan dihasilkan sinyal “equal with”.

Hasil dari pendekripsi ini berupa vector berisikan status ataupun sinyal dari hasil *signal analysis*, sehingga jika digunakan data *dummy* pada tabel 4.6, akan didapatkan hasil seperti pada tabel 4.11 berikut. Karena interval data *dummy* adalah per hari, maka data terakhir akan dibandingkan dengan data 1 minggu sebelumnya (data ke-7 terakhir), sehingga parameter pertama dikatakan *higher than* karena nilai data terakhir (20 januari 2019) adalah 0.19 dan data satu minggu sebelumnya (13 januari 2019) adalah 0.17, begitu juga dengan parameter ke-dua dikatakan *higher than* karena nilai data terakhir sebesar 30 dan data satu minggu sebelumnya sebesar 21.

Tabel 4.11 Hasil pendekripsi sinyal *Comparison*

Param1	Param2
Higher than	Higher than

d. Proses *Signal Analysis* untuk *Repeated Event*

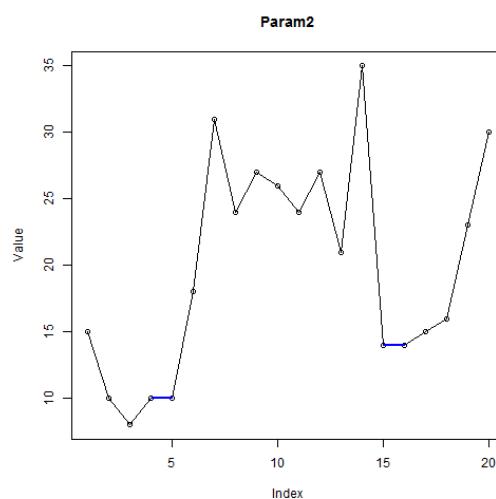
Proses pendekripsi sinyal *repeated event* ini bertujuan untuk mencari parameter yang nilainya tidak mengalami perubahan selama periode tertentu (konstan). Proses ini tidak jauh beda dengan proses

pendeksiian *Trend* dimana jika ditemukan sebuah parameter dengan nilai sama atau tidak berubah berturut-turut sebanyak lebih dari dari 10% jumlah data maka termasuk ke dalam kategori *Repeated Event*. Proses ini dilakukan untuk parameter dengan tipe *numerical*.

Berikut beberapa contoh kasus pada penentuan *Repeated Event*:

- a. Pada parameter X dalam data Y dengan jumlah 30 baris data (bulanan), terdapat nilai yang sama secara berturut-turut pada baris 5 sampai baris 10 (5 baris), maka sinyal tersebut merupakan sinyal *repeated event* karena 5 baris lebih besar dari 3 ($10\% * 30$ baris).
- b. Pada parameter X dalam data Z dengan jumlah 366 baris data (tahunan), terdapat 20 baris data dengan nilai yang sama secara berturut-turut, karena 20 baris tidak lebih besar dari ($10\% * 366$ baris) maka sinyal tersebut bukan *repeated event*.

Seperti yang sudah dijelaskan sebelumnya, sinyal *repeated event* merupakan keadaan suatu paramater yang nilainya konstan atau tetap untuk beberapa periode waktu. Jika kita plot parameter ke-dua data *dummy* pada tabel 4.6, akan didapatkan sinyal *repeated event* pada data ke-empat hingga data ke-lima, dan pada data ke-lima belas hingga data ke-enam belas, pada gambar 4.5 sinyal *repeated event* tersebut ditandai dengan garis berwarna biru.



Gambar 4.5 Sinyal *repeated event* ditandai dengan garis biru

Hasil dari *signal analysis* untuk *repeated event* ini berupa *list* yang berisikan nama kolom, banyaknya perulangan pada data, indeks dimana perulangan dimulai (dalam bentuk vektor), dan indeks dimana perulangan berhenti (dalam bentuk vektor). Jika kita menggunakan data *dummy* pada tabel 4.6 maka akan didapatkan hasil *signal analysis* untuk *repeated event* seperti pada tabel 4.12.

Tabel 4.12 Hasil pendektsian sinyal *repeated event*

Column Name	Rep Value	Start	End
Param1	3	3 10 15	4 11 16
Param2	2	4 15	5 16

e. Proses *Signal Analysis* untuk Prediksi Data

Salah satu bagian dari proses *Signal Analysis* yaitu dilakukannya proses prediksi data. Proses ini hanya diterapkan untuk parameter dengan tipe data *numerical*. Proses ini bertujuan agar informasi akan disampaikan kepada pembaca lebih informatif. Selain itu juga, informasi yang disampaikan berisi kemungkinan-kemungkinan yang akan terjadi kedepannya. Dalam melakukan prediksi, diterapkan konsep prediksi *time-series* dengan algoritma *Exponential Smoothing* seperti penelitian DWP sebelumnya (Putra *et al.*, 2017).

Dengan menggunakan data *dummy* pada tabel 4.6 setelah dilakukan proses prediksi data sehingga didapatkan hasil pada tabel 4.13 , seperti yang kita lihat pada tabel tersebut, data yang disimpan berupa hasil prediksi dengan menggunakan algoritma *Exponential Smoothing*. Data yang disimpan pada proses ini masih berbentuk numerik, yang nantinya akan diproses dan dijelaskan lebih mendetail pada bagian *Data Interpretation*.

Tabel 4.13 Hasil Prediksi data untuk contoh kasus data klimatologi

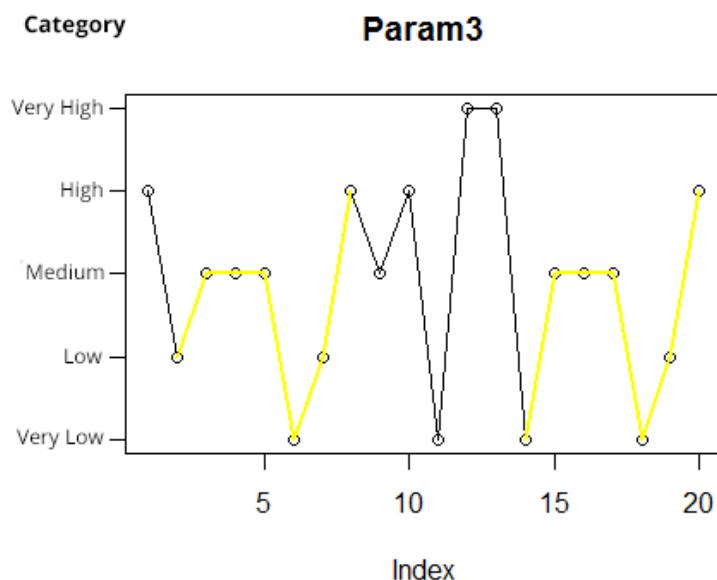
Param1	Param2
0.1807813	29.9992986

f. Proses *Signal Analysis* untuk *String Matching*

Proses pendektsian *string matching* ini bertujuan untuk mencari pola yang sama pada sebuah parameter dengan tipe *categorical*,

pendeksiian pola yang sama ini dilakukan dengan mengambil beberapa baris data terakhir sebagai acuan lalu mencocokannya dengan seluruh data pada dataset. Proses pendeksiian *string matching* ini menggunakan algoritma KMP dalam R seperti pada penelitian (Rahman, 2017).

Banyaknya data yang akan dicocokkan bergantung pada interval data yang menjadi masukan. Pengguna dapat menentukan sendiri berapa banyak data yang diambil untuk acuan nantinya. Namun untuk nilai *defaultnya*, jika interval data masukan adalah per jam, maka diambil data atau pola acuan yang diambil sebanyak 6 baris atau data 6 jam terakhir. Untuk data dengan interval per hari maka data yang diambil sebanyak 7 baris atau data seminggu terakhir. Selebihnya, untuk data dengan interval bulanan atau tahunan, maka data yang diambil sebanyak 4 baris data terakhir. Setelah diambil beberapa data terakhir, proses selanjutnya adalah mencari pola tersebut pada seluruh dataset.



Gambar 4.6 Hasil plot parameter ke-tiga data *dummy*, garis kuning menandakan pola data yang sama

Gambar 4.6 menampilkan hasil plot dari parameter ke-tiga dari data *dummy* pada tabel 4.6, garis kuning pada gambar tersebut menandakan pola yang sama, atau hasil dari *signal analysis* untuk *string matching*. Hasil dari *signal analysis* untuk *string matching* ini berupa *list* yang didalamnya

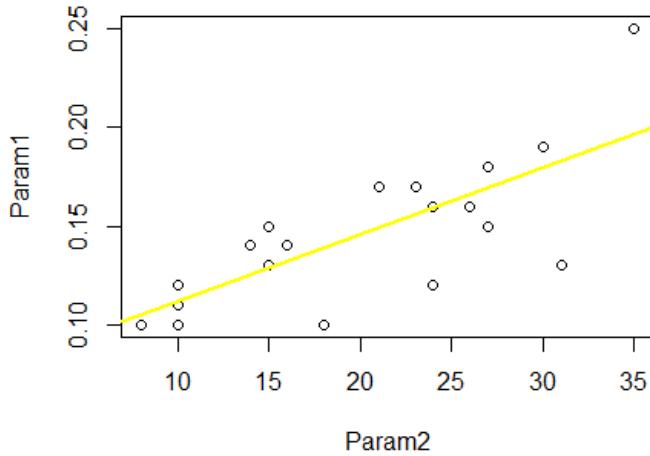
berisikan jumlah pola yang sama, dan vektor yang berisikan indeks dari pola-pola yang sama tersebut. Seperti pada tabel 4.14, tabel tersebut berisikan hasil *signal analysys* untuk *string matching* menggunakan data *dummy* pada tabel 4.6, data yang disimpan berupa total pola yang terdapat pada parameter *categorical*, lalu indeks dari pola tersebut dalam bentuk vektor. Misalnya pada tabel tersebut ditemukan satu pola yang sama dengan pola data seminggu terakhir, ditemukan pola yang sama seperti dari data kedua hingga ke-delapan.

Tabel 4.14 Hasil pendektsian sinyal *String Matching*

Column Name	Total	Pattern Index
Param3	1	2

g. Proses *Signal Analysis* untuk Menentukan Korelasi Parameter

Proses pendektsian sinyal ini bertujuan untuk mendapatkan bagaimana hubungan linear dari suatu parameter dengan parameter lainnya. Misalnya, apakah kenaikan atau penurunan pada suatu parameter akan berdampak pada parameter lainnya atau tidak. Dalam menentukan korelasi antar parameter, digunakan perhitungan korelasi Pearson (Karl Pearson, 1900) , dimana data akan dipetakan kedalam matriks berukuran n,n dimana n merupakan banyaknya parameter *numerical* dalam suatu data. Matriks tersebut berisikan nilai dengan rentang 0 sampai 1 yang merepresentasikan hubungan linear antara satu parameter dengan parameter lainnya. Semakin tinggi nilai yang didapat, menandakan hubungan linear parameter tersebut sangat kuat. Jika didapatkan hasil positif, maka angka tersebut merepresentasikan hubungan yang sebanding, kenaikan suatu parameter berdampak pada kenaikan parameter lannya, begitu juga sebaliknya. Namun jika hasil yang didapatkan bernilai postif maka, nilai tersebut merepresentasikan hubungan yang bertolak belakang antara parameter tersebut, kenaikan pada suatu parameter akan berdampak penurunan pada parameter lainnya, begitu juga penurunan suatu parameter akan berdampak pada kenaikan parameter lainnya.



Gambar 4.7 Hasil plot data *dummy* yang menunjukkan hubungan linear antara parameter pertama dan ke-dua

Misalkan jika digunakan data *dummy* pada tabel 4.6 dan dilakukan dan dilakukan plot pada parameter pertama dengan parameter ke-dua pada data *dummy tersebut* maka akan didapatkan hasil plot seperti pada gambar 4.7. Dimana pada gambar 4.7 kita dapat melihat hubungan antara parameter pertama dengan parameter ke-dua secara linear, dimana didapatkan nilai korelasi parameter antar kedua parameter tersebut sebesar 0.744 atau termasuk pada korelasi yang cukup kuat. Hasil dari *signal analysis* untuk korelasi parameter ini disimpan dalam bentuk matriks seperti yang terdapat pada tabel 4.15 berikut.

Tabel 4.15 Hasil *signal analysis* untuk korelasi paramter pada data *dummy*.

	Param1	Param2
Param1	1.0000	0.7449012
Param2	0.7449012	1.0000

Contoh lainnya yaitu jika menggunakan data partikel udara pada tabel 4.4 lalu dilakukan pendektsian sinyal untuk mencari korelasi antar parameter maka didapatkan hasil pada tabel 4.16. Dapat kita lihat, tabel tersebut menampilkan nilai korelasi antar parameter, sebagai contoh parameter TEMP memiliki nilai korelasi -0.8095 terhadap parameter PRES, yang berarti kedua parameter tersebut memiliki keterkaitan yang cukup

kuat. Dimana ketika kenaikan terjadi pada parameter TEMP atau *Temperature* maka akan terjadi penurunan pada parameter PRES atau *Pressure*, hal ini sejalan dengan hukum alam, dimana ketika suhu semakin panas maka udara akan memuoi lalu naik karena lebih ringan, akibat udara yang naik tersebut maka tekanan akan turun.

Tabel 4.16 Hasil pendekripsi sinyal menggunakan *Pearson Correlation*

	PM2.5	DEWP	TEMP	PRES	LWS	IS	IR
PM2.5	1.0000	0.2776	0.0377	-0.1905	-0.2632	-0.0130	-0.0506
DEWP	0.2776	1.0000	0.8431	-0.7687	-0.3112	-0.0410	0.1220
TEMP	0.0377	0.8431	1.0000	-0.8095	-0.1780	-0.0997	0.0500
PRES	-0.1905	-0.7687	-0.8095	1.0000	0.2120	0.0629	-0.0592
IWS	-0.2632	-0.3112	-0.1780	0.2120	1.0000	0.0237	-0.0372
IS	-0.0130	-0.0410	-0.0997	0.0629	0.0237	1.0000	-0.0098
IR	-0.0506	0.1220	0.0500	-0.0592	-0.0372	-0.0098	1.0000

4.2.3. Model Komputasi untuk *Data Interpretation*

Pada proses ini dilakukan penerjemahan sinyal-sinyal dan *event* yang dihasilkan pada bagian *Signal Analysis* menjadi bentuk frasa atau kata-kata yang akan disampaikan (Reiter, 2011). Proses penerjemahan ini dilakukan dengan menggunakan logika *Fuzzy* dan *Crisp* seperti yang dilakukan pada penelitian DWP sebelumnya (Putra *et al.*, 2017). Penentuan penggunaan logika *Fuzzy* dan *Crisp* ini ditentukan sesuai dengan tipe penginterpretasian yang tersimpan pada *data description* seperti yang sudah dijelaskan pada tahap *Unspecific Data Handler*. Dimana sinyal-sinyal baik berupa pola maupun event yang dihasilkan pada *Signal Analysis* akan direpresentasikan ke dalam bentuk pesan-pesan yang dapat dipahami manusia.

Setidaknya ada tiga hal yang dilakukan dalam proses *Data Interpretation* ini yakni, *create message* atau penerjemahan suatu data ke dalam bentuk bahasa alami, *decide how important message are* atau menentukan seberapa pentingnya suatu pesan ditampilkan, dan *detect relationship between event* atau mencari hubungan setiap *event*. Contoh penerapan proses *create message* ini seperti yang dilakukan oleh Sripada & Gao (2007) dalam penelitiannya untuk kasus detak jantung, pesan *BRADYCARDIA* akan ditampilkan jika data detak jantung yang didapatkan dari proses analisis bernilai dibawah 100, maka data tersebut akan

didefinisikan menjadi sebuah kalimat “*heart rate is temporarily low*”. Begitu juga untuk penerapan *decide how important message are*, dalam penelitian Portet et al., (2007) pesan *BRADYCARDIA* akan dianggap penting ketika pesan tersebut berlangsung lebih dari 5 menit. Sedangkan untuk penerapan *detect relationship between event*, Portet et al., (2007) akan melihat hubungan kasualitas dari beberapa data, misalnya kenaikan oksigen pada darah akan berakibat pada penurunan kadar CO₂ pada darah. Pada penelitian ini, proses penginterpretasian untuk data *specific* ini dilakukan dengan mengacu pada nilai-nilai yang sudah ditentukan oleh *expert*, *Fuzzy Membership Function* digunakan untuk kasus data dengan tipe penginterpretasian *Fuzzy*, dan *Crisp Membership Function* untuk kasus data dengan tipe penginterpretasian *Crisp* (Putra et al., 2017).

Pada penelitian ini, jika sebuah parameter tidak dikenali oleh sistem atau tidak terdapatnya cara penginterpretasian data pada *data description* dalam file *datadescription.csv*, maka sistem akan secara otomatis menggunakan *Unspecific Membership Function* untuk penginterpretasian datanya. Karena tidak semua pesan akan disampaikan, maka selain menerjemahkan data kedalam bentuk teks, pada tahap ini dilakukan penentuan seberapa pentingnya pesan tersebut disampaikan, hal ini serupa dengan penelitian yang dilakukan oleh (Portet et al., 2007). Contohnya dalam penginterpretasian sinyal *Repeated Event*, jika ada parameter yang memiliki nilai berulang maka sistem hanya akan menampilkan pesan untuk parameter tersebut saja, dan secara otomatis akan mengabaikan parameter lain yang tidak mempunyai nilai berulang di dalamnya. Begitu juga jika tidak terdapat parameter yang mempunyai nilai berulang di dalamnya, maka sistem hanya akan menampilkan pesan bahwa tidak ditemukannya nilai yang berulang sekali saja, dibanding dengan menampilkan pesan untuk semua parameternya. Sehingga *input*, proses, dan *output* pada tahap ini adalah sebagai berikut:

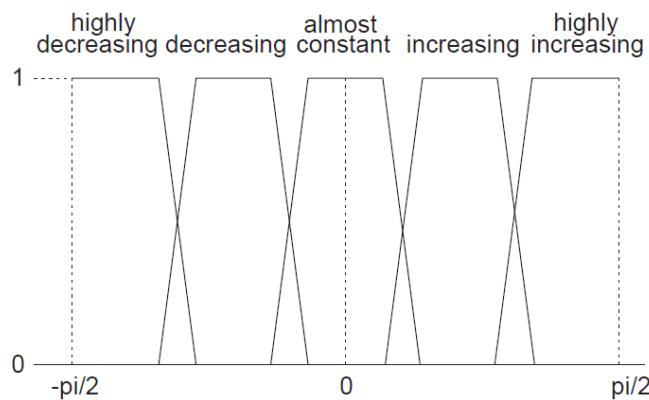
- *Input: Data Abstraction & Event, Data description.*
- Proses: *Data Interpretation*
- *Output: Message Inventory*

Sub-proses dari *Data Interpretation* ini akan dibahas pada sub-bab selanjutnya:

a. *Generate Rule* untuk Interpretasi

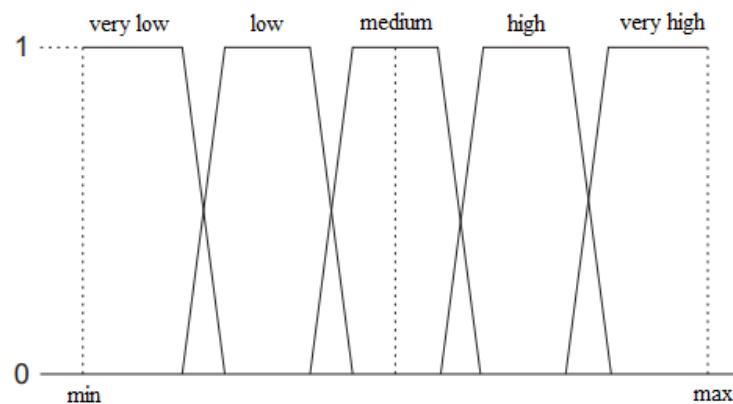
Pada proses ini dihasilkan *rule* yang akan digunakan untuk menginterpretasikan suatu data dalam bentuk numerikal ke dalam bentuk pesan atau bahasa yang mudah dipahami manusia (Reiter, 2011). Pada umumnya, proses dalam menginterpretasikan data ini dibagi menjadi dua, yakni menginterpretasikan data menggunakan logika *crisp* atau menggunakan logika *fuzzy*. Dalam penelitian ini dikembangkan juga *specific corpus* yang berasal dari sistem DWP untuk *Weather Condition* (Putra *et al.*, 2017). Beberapa *corpus* yang diadaptasi dari penelitian DWP diantaranya, *corpus* untuk cakupan awan, temperatur, kecepatan angin, arah angin, dan curah hujan yang disimpan pada folder *Corpus*.

Selain itu dalam penelitian ini dikembangkan juga interpretasi data untuk kasus data umum atau data *unspecific*, dimana sistem akan tetap menginterpretasikan sebuah parameter walau parameter tersebut tidak mempunyai cara pentinterpretasian yang *specific*. Sehingga ketika data dengan jenis apapun dimasukkan, sistem akan tetap berjalan seperti meskipun yang menjadi masukan merupakan data yang tidak mempunyai *header*. Proses untuk menginterpretasikan data *unspecific* ini menggunakan logika *fuzzy* dalam menginterpretasikan datanya. Dimana himpunan *fuzzy* yang digunakan mengacu pada penelitian penentuan trend oleh Castillo-Ortega *et al.*, (2014). Pada penelitian tersebut, Castillo membagi dan menentukan himpunan *fuzzy* untuk *trend* menjadi lima wilayah yang sama besar berdasarkan nilai pi (nilai rata-rata data) seperti pada gambar 4.8.



Gambar 4.8 *Linguistic variable for trend description* (Castillo-Ortega et al., 2014)

Peneliti melakukan modifikasi pada himpunan tersebut untuk menginterpretasikan data *unspecific*, dimana himpunan fuzzy pada gambar 4.8 diubah batas atas dan bawahnya menjadi nilai maksimum dan nilai minimum dari suatu parameter. Dimana daerah di antara rentang nilai maksimum dan nilai minimum dibagi menjadi lima bagian yang sama besar. Yang kemudian untuk *corpus* yang digunakan adalah *very low*, *low*, *medium*, *high*, *very high* seperti yang dilakukan oleh Fallah-Ghalhary et al., (2009) dalam penelitiannya. Sehingga himpunan *fuzzy* yang digunakan untuk data *unspecific* dapat dilihat pada gambar 4.9.



Gambar 4.9 *Unspecific Fuzzy Membership Function*

b. Proses Interpretasi untuk *Data Specific* dan *Data Unspecific*

1. *Data Specific*

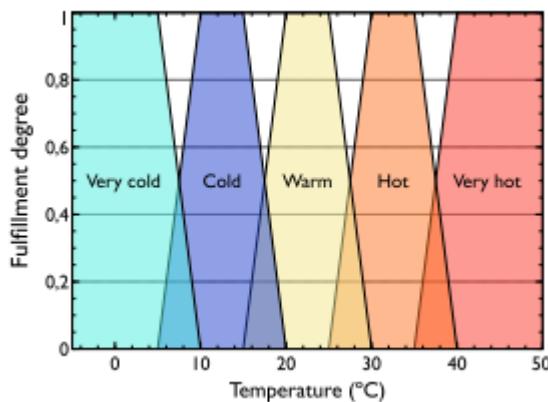
Setelah menentukan *rule* untuk memproses suatu data ataupun event, proses selanjutnya adalah menerjemahkan suatu data menggunakan *rule* yang telah dibuat untuk menghasilkan pesan dalam bentuk bahasa alami. Singkatnya, proses interpretasi pesan *specific* ini bertujuan untuk mengubah pesan yang masih berbentuk numerik, *pattern*, atau simbol menjadi bahasa alami, atau bahasa sehari-hari menggunakan *corpus* dan *rule* yang sudah didefinisikan oleh *expert* maupun *user* sebelumnya. Proses interpretasi pesan *specific* dilakukan untuk parameter yang memiliki cara penginterpretasian atau atribut rule pada *data description*. Dalam penelitian ini dikembangkan juga *specific corpus* yang berasal dari sistem DWP untuk kasus *Weather Condition* (Putra *et al.*, 2017). Beberapa *corpus* yang diadaptasi dari penelitian DWP diantaranya *corpus* untuk temperatur, cakupan awan, kecepatan angin, arah angin, dan curah hujan yang disimpan pada folder *Corpus*.

Contohnya dalam menginterpretasikan nilai temperatur, langkah pertama adalah mendefinisikan nama kolom, atribut *rule* pada *data description* seperti pada tabel 4.17. Sehingga jika terdapat parameter dengan nama kolom *Temperature* maka akan dilakukan serangkaian analisis untuk tipe data *numerical* dan proses interpretasi menggunakan *fuzzy rule based*.

Tabel 4.17 Contoh *data description* untuk menginterpretasikan parameter TEMP pada data partikel udara

ColName	Type	Rule	Alternate
Temperature	numeric	fuzzy	NA

Setelah didefinisikan cara penginterpretasianya, langkah selanjutnya adalah memasukan *corpus* berbentuk *csv* pada folder *Corpus/fuzzy*. Dalam kasus ini *corpus* yang digunakan adalah *corpus* temperatur pada penelitian DWP (Putra *et al.*, 2017), sehingga didapatkan himpunan fuzzy seperti pada gambar 4.10.



Gambar 4.10 Himpunan *fuzzy* dari *corpus* temperatur pada penelitian DWP (Putra *et al.*, 2017)

Langkah selanjutnya adalah menkonversi himpunan *fuzzy* pada gambar 4.10 kedalam bentuk *rule* menggunakan *specific rule generator* pada proses *generate rule* seperti yang sudah dijelaskan pada sub-bab sebelumnya. Setelah didapatkan *rule* untuk menginterpretasikan data, langkah terakhir adalah proses penginterpretasian nilai prediksi parameter *temperature* tersebut, misalkan parameter tersebut bernilai -6 deraja celcius, maka nilai tersebut akan diinterpretasikan menggunakan *rule* yang sudah dibuat, sehingga didapatkan nilai keanggotaan parameter seperti pada tabel 4.18, sehingga dihasilkan pesan *Very Cold*.

Tabel 4.18 Nilai keanggotaan parameter TEMP

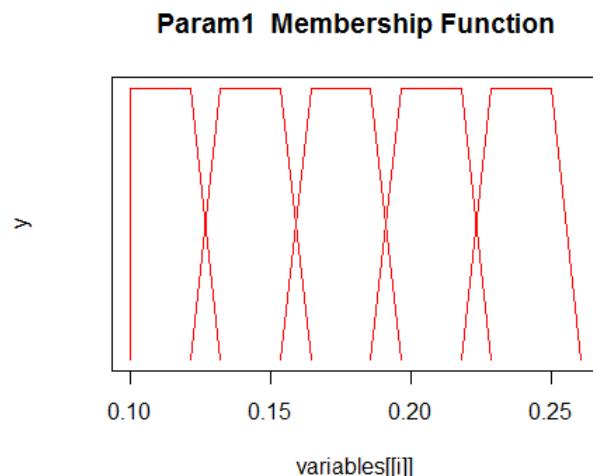
Very cold	Cold	Warm	Hot	Very hot
1	0	0	0	0

2. Data Unspecific

Berbeda dengan penginterpretasian pesan *specific*, proses penginterpretasian pesan *unspecific* ini menggunakan logika *fuzzy* dan *rule* yang dihasilkan dari *unspecific rule generator* yang sangat bergantung pada nilai maksimum dan nilai minimum data. Dimana suatu parameter akan dikategorikan atau dianggap *unspecific* jika parameter tersebut tidak memiliki cara penginterpretasian atau atribut *rule* pada *data description*. Dimana himpunan *fuzzy* dan *rule*

yang digunakan mengacu pada penelitian penentuan trend oleh Castillo-Ortega et al., (2014). Pada penelitian tersebut, Castillo membagi dan menentukan himpunan *fuzzy* untuk *trend* menjadi lima wilayah yang sama besar. Corpus yang digunakan untuk penginterpretasian data *unspecific* adalah *very low, low, medium, high, very high* (Fallah-Ghalhary et al., 2009b).

Sebagai contoh, peneliti akan menginterpretasikan *resume data, last -1 data* dan *current data* dari Param1 yang terdapat pada data tabel 4.9. Langkah pertama dalam melakukan interpretasi pesan *unspecific* hampir sama dengan proses interpretasi untuk pesan *specific*, dimana sistem akan melihat terlebih dahulu cara penginterpretasian parameter tersebut pada *data description*. Dikarenakan Param1 pada tabel 4.9 tidak memiliki cara penginterpretasian pada *data description*, maka proses *unspecific rule generator* akan menghasilkan himpunan *fuzzy* yang nantinya akan digunakan sebagai *rule* dalam menginterpretasikan parameter tersebut. Himpunan *fuzzy* yang dihasilkan dapat dilihat pada gambar 4.11.



Gambar 4.11 Himpunan *fuzzy* Param1 yang dihasilkan oleh *unspecific rule generator*

Seperti yang kita lihat pada gambar 4.11, himpunan *fuzzy* dihasilkan berdasarkan nilai maksimum dan minimum dari

parameter Param1, dimana parameter Param1 memiliki nilai minimum sebesar 0.10, dan nilai maksimum sebesar 0.25. Setelah didapatkan nilai minimum dan nilai maksimum dari parameter Param1 tersebut, proses selanjutnya adalah membagi rentang tersebut menjadi lima wilayah yang sama besar seperti yang dapat kita lihat pada gambar 4.11.

Proses selanjutnya adalah mengkonversi himpunan *fuzzy* tersebut ke dalam bentuk *rule* yang nantinya akan digunakan sebagai acuan untuk menginterpretasikan data. Nilai keanggotaan dari *resume data*, *current data* dan *predict data* untuk parameter Param1 pada tabel 4.9 dapat dilihat pada tabel 4.19.

Tabel 4.19 Nilai keanggotaan dari ringkasan data untuk parameter Param1 pada data *dummy*

	Very low	Low	Medium	High	Very high
Resume Data	0	1	0	0	0
Last (n-1) Data	0	0	1	0	0
Current Data	0	0	0.6	0.4	0

Proses penginterpretasian ringkasan data untuk parameter Param1 pada tabel 4.9 yang digabungkan dengan prediksi data pada tabel 4.13 dilakukan menggunakan *unspecific rule*. Hasil dari proses penginterpretasian tersebut dapat dilihat pada tabel 4.20, sehingga pesan yang dihasilkan sesuai dengan *corpus* untuk data *unspecific*, yang terdiri dari *very high*, *high*, *medium*, *low*, dan *very low*.

Tabel 4.20 Hasil *Data Interpretation* contoh kasus parameter Param1 pada data *dummy*

	Param1
Resume	low
Last (n-1)	Medium
Current	Medium
Predict	Medium

c. Proses Interpretasi *Extreme Event*

Pada bagian ini, akan dilakukan proses interpretasi data untuk hasil dari pendekripsi sinyal *Extreme Event*. Setelah didapatkan hasil

pendeksiian seperti pada tabel 4.10, langkah selanjutnya adalah menginterpretasikan sinyal-sinyal tersebut kedalam bentuk bahasa manusia. Sinyal yang akan diinterpretasikan menjadi pesan adalah sinyal-sinyal yang memiliki kenaikan atau penurunan yang ekstrem. Misalnya jika pada suatu parameter hanya nilai kenaikannya saja yang tergolong pada kategori ekstrem, maka sinyal tersebut akan diinterpretasikan menjadi “*increased*”. Lalu jika hanya penurunannya saja yang tergolong pada kategori ekstrem maka sinyal tersebut akan diinterpretasikan menjadi pesan “*decreased*”. Sedangkan jika sinyal kenaikan dan penurunannya tergolong pada kategori ekstrem, maka sinyal tersebut akan diinterpretasikan menjadi pesan “*fluctuated*”, seperti kasus yang terjadi pada Param2 pada data *dummy* yang dapat dilihat pada gambar 4.4, pada gambar tersebut terdapat kenaikan dan penurunan ekstrem yang tunjukkan oleh garis berwarna hijau dan merah sehingga akan ditampilkan pesan *extreme event* dengan jenis kenaikan *fluctuated*.

d. Proses Interpretasi *Repeated Event*

Pada proses penginterpretasian sinyal *Repeated Event* ini tidak jauh berbeda dengan proses penginterpretasian sinyal untuk *Extreme Event*. Dimana sinyal-sinyal hasil dari pendeksiian pada bagian *Signal Analysis* untuk *Repeated Event* ini akan diinterpretasikan menjadi pesan yang mudah dipahami oleh manusia. Namun yang berbeda pada proses ini adalah, adanya pemilihan *how importans message are*, dimana sinyal yang akan ditampilkan adalah sinyal yang mempunyai nilai perulangan terbanyak.

Jika kita menggunakan data pada tabel 4.12, terlihat pada tabel tersebut siyal *repeated event* terjadi pada dua parameter yakni parameter Param1 dan Param2, namun pada proses interpretasi ini diambil sinyal dengan jumlah perulangan terbesar sehingga pesan yang akan ditampilkan adalah sinyal perulangan parameter Param1 dengan jumlah perulangan sebanyak tiga kali. Sehingga hasil interpretasi *repeated event* yang akan diproses lebih lanjut menjadi sebuah kalimat adalah sinyal *repeated event* untuk Param1.

e. Proses Interpretasi *String Matching*

Dalam proses penginterpretasian sinyal *string matching*, hanya proses *create message* atau pembuatan pesan. Dikarenakan proses ini hanya dilakukan untuk parameter dengan tipe *categorical* saja, sehingga pesan yang ditampilkan tidak perlu dipilih dan disaring terlebih dahulu. Semua sinyal yang didapatkan dari proses *signal analysis* untuk *string matching* akan ditampilkan. Jika kita menggunakan hasil pendekripsi sinyal pada tabel 4.14, dimana pada tabel tersebut terdeteksi bahwa ada pola yang sama untuk parameter Param3 dengan tipe *cataegorical*, maka sinyal tersebut akan diinterpretasikan sebagai pesan *string matching* yang akan dibangun menjadi pesan pada bagian *Document Planning*. Pada pesan tersebut disampaikan bahwa terdapat kesamaan pola antara pola data indeks ke-2 hingga ke-8 dengan pola data 7 hari terakhir.

f. Interpretasi Korelasi Parameter

Untuk menginterpretasikan nilai dari proses pendekripsi sinyal untuk korelasi parameter *Pearson* ini digunakan ketentuan yang ditetapkan oleh De Vaus (2002) dimana nilai yang dihasilkan saat proses pendekripsi sinyal untuk korelasi parameter dibagi menjadi 7 kategori seperti pada tabel 4.21 berikut.

Tabel 4.21 Parameter Correlation Crisp Membership Function (de Vaus, 2002)

No	Coefficient	Class
1	0.00	no linear association
2	0.01 - 0.09	insubstantial relationship
3	0.10 - 0.29	low relationship
4	0.30 - 0.49	moderate relationship
5	0.50 - 0.69	strong relationship
6	0.70 - 0.89	very strong relationship
7	0.9+	almost perfect relationship

Proses penginterpretasian untuk korelasi parameter ini terbagi ke dalam dua bagian, bagian pertama adalah proses penginterpretasian nilai korelasi untuk pembangkitan *routine message* nantinya, proses ini

dilakukan dengan melakukan perhitungan rata-rata dari nilai mutlak setiap koefisien pearson yang sudah dihasilkan pada tabel 4.16, setelah itu dipilih parameter yang memiliki nilai rata-rata terbesar, yang kemudian nilai tersebut akan diinterpretasikan menggunakan himpunan *crisp* pada tabel 4.21. Sehingga dihasilkan pesan interpretasi bahwa parameter Param1 dipilih karena memiliki nilai rata-rata korelasi parameter terbesar, yakni 0.48 yang kemudian diinterpretasikan sehingga didapatkan hasil *moderate relationship*.

Untuk proses penginterpretasian untuk pembangkitan pesan *significant* nantinya pada proses korelasi parameter ini, dilakukan penyaringan sinyal, dimana sinyal yang akan ditampilkan adalah sinyal atau koefisien korelasi yang lebih besar dari 0.7 baik untuk nilai positif maupun negatif. Pesan yang ditampilkan hanyalah pesan yang memiliki nilai keterkaitan *very strong* dan *almost perfect* (lihat tabel 4.21) sedangkan semua nilai diagonal atas dan nilai lainnya akan diabaikan dengan mengubahnya menjadi 0. Jika digunakan hasil pendekripsi sinyal korelasi parameter data *dummy* pada tabel 4.15 maka akan didapatkan hasil dari pemilihan sinyal seperti pada tabel 4.22, seperti yang bisa kita lihat pada tabel tersebut, terdapat nilai korelasi 0.744 yang berarti antara Param1 dan Param2 memiliki keterkaitan yang cukup kuat (*very strong*), sehingga hasil interpretasi nilai ini akan dibentuk menjadi sebuah kalimat pada proses selanjutnya. Dimana untuk nilai korelasi yang bernilai positif, akan menghasilkan pesan yang mempunyai kontras sama pada proses *Document Planning*, misalnya pada pesan tersebut kenaikan pada parameter Param1 akan berdampak pada naiknya nilai parameter Param2, begitu juga untuk penurunan pada parameter tersebut. Namun jika nilai korelasi bernilai negatif maka akan dihasilkan pesan dengan kontras yang berbeda pada proses *Document Planning*, misalnya kenaikan pada parameter *a* akan berdampak pada penurunan parameter *b*, atau sebaliknya.

Tabel 4.22 Hasil proses interpretasi data untuk sinyal korelasi parameter.

	Param1	Param2
Param1	0	0
Param2	0.7449012	0

4.2.4. Model Komputasi untuk *Document Planning*

Pada proses ini dilakukan pemilihan konten (*Content Determination*), dan pembentukan struktur teks (*Document Structure*) yang akan ditampilkan pada teks keluaran (Reiter, 1996). Mengingat banyaknya pesan atau pola yang dihasilkan dari proses-proses sebelumnya, pemilihan konten akan dipilih berdasarkan *Message Inventory*, dan struktur teks yang akan dibangun berdasarkan *schema-based* yang diperkenalkan oleh Turner (Turner *et al.*, 2008). Sehingga *input*, proses, dan *output* pada tahap ini adalah sebagai berikut:

- *Input: Message Inventory*
- Proses: *Document Planning*
- *Output: Content and Structure (Summary Text, Current Text, and Predict Text)*

Pemilihan konten dan pembentukan struktur untuk setiap paragraf akan dijelaskan pada sub-bab berikutnya.

4.2.4.1 Perencanaan Dokumen untuk Ringkasan Data

1. *Content Determination*

Proses pemilihan konten untuk bagian ringkasan data ini mengacu pada penelitian DWP (Putra *et al.*, 2017). Pada penelitian tersebut, konten-konten yang akan ditampilkan dibagi menjadi dua bagian, yaitu *Routine Message* dan *Significant Event Message*. *Routine Message* merupakan pesan-pesan yang akan selalu muncul pada teks yang akan ditampilkan, sedangkan *Significant Event Message* adalah pesan yang akan ditampilkan untuk kondisi tertentu saja. Umumnya *Significant Event Message* memerlukan syarat-syarat tertentu agar pesan-pesannya bisa ditampilkan. Contohnya, dalam kasus pembangkitan berita mengenai kasus gempa bumi, pesan statu hanya akan ditampilkan jika guncangan mencapai status

siaga. Pada penelitian ini, yang termasuk pada *Significant Event Message* adalah *Repeated Event*, *Extreme Event*, dan *String Matching*. Untuk gambaran *Routine Message*, terlihat pada gambar 4.12 terlihat bahwa terdapat variabel parameter[i:n], yang berarti dari parameter ke-i (awal) hingga parameter ke-n (terakhir).

```

Summary – Routine Message ->
Construct Message {
    Date Time, Parameter[i:n]
    Parameter[i:n] Trend
    Parameter[i:n] Comparison
    Parameter[i:n] Correlation
}

```

Gambar 4.12 *Routine Message* untuk ringkasan Data

Significant Event Message pada gambar 4.13, terdapat variabel parameter yang diikuti dengan nama event, seperti parameter [*extreme event*] atau sejenisnya, pesan tersebut menandakan pada parameter mana saja event itu terjadi.

```

Summary – Significant Event Message ->
IF Repeated Event = "True" |
THEN Construct Message {
    Parameter[Repeated Event]
    Repeated Event
}
IF Extreme Event = "True" |
THEN Construct Message {
    Parameter[Extreme Event]
    Extreme Event
}
IF Correleation Event = "True" |
THEN Construct Message {
    Parameter[Correleation Event]
    Correleation Event
}
IF S.Matching Event = "True" |
THEN Construct Message {
    Parameter[S.Matching Event]
    S.Matching Event
}

```

Gambar 4.13 *Significant Event Message* untuk ringkasan data

Sebagai contoh jika kita menggunakan data *dummy* pada tabel 4.6 dan hasil interpretasi data pada proses sebelumnya maka akan didapatkan hasil seperti pada tabel 4.23

Tabel 4.23 Hasil *Content Determination Summary* untuk data klimatologi

Routine Message	According to the daily dummy data between 01/01/2019 00:00 to 01/20/2019 00:00, with parameters: Param1, Param2, and Param3, it illustrated that trend of all variable is increased. All parameters are higher than last week's data. Param1 appears to have a highest impact to all variable with very strong relationship in average.
Significance Event Message	There were some repeating value more than 2 days: Param1 stayed constant at very low during 3-4 Jan 2019. Param1 stayed constant at medium during 10-11 Jan 2019. Param2 fluctuated dramatically (increased 21 points and decreased 21 points). There is a same Param3 data pattern in the last 7 days (13-20 Jan 2019) with data pattern from 2-9 Jan 2019. An increase in Param1 resulted an increase in Param2."

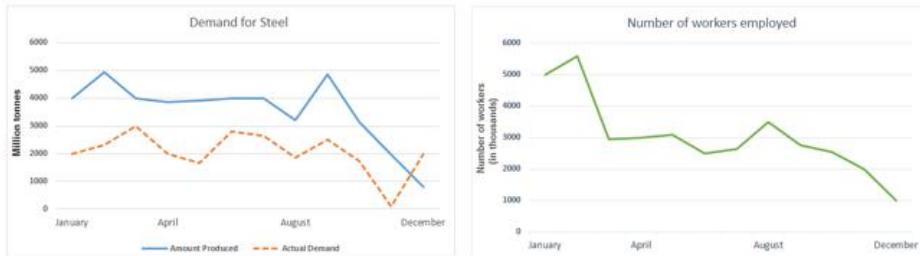
2. *Content Struturing*

Dalam proses penentuan struktur dokumen, Reiter (1996) memaparkan bahwa penentuan struktur dokumen dapat dilakukan dengan cara membangun skema dokumen atau penalaran secara eksplisit. Teks yang dihasilkan diharapkan sesuai dengan pola umum atau konvensional, maka dari itu dalam menentukan struktur dokumen yang akan dibuat, diperlukan sebuah *Initial Corpus* sebagai target dari keluaran teks nantinya, *initial corpus* untuk teks ringkasan data dapat dilihat pada gambar 4.15. Pembentukan *Initial Corpus* ini bertujuan untuk mendikte konten dan memastikan struktur yang koheren, sehingga teks keluaran yang akan dihasilkan lebih padat dan tidak melebar pembahasannya.

Dalam menentukan *Initial Corpus* atau *Target Text* ini, setidaknya ada 4 langkah yang harus dilakukan (Reiter, 1996), diantaranya :

- a. Ambil contoh sejumlah teks dengan bidang yang sama untuk dijadikan sebagai *Target Text*.
- b. Melakukan identifikasi terhadap pesan-pesan yang ada, lalu menentukan bagaimana setiap pesan dapat dibangun berdasarkan data.
- c. Mengusulkan aturan atau struktur yang menjelaskan mengapa pesan "x" ada dalam teks A tetapi tidak ada didalam teks B. Penentuan ini lebih mudah jika disusun dalam bentuk seperti taksonomi.
- d. Diskusikan hasil analisis bersama pakar.

Dalam melakukan pembentukan *initial corpus* untuk teks ringkasan data ini, penulis mengambil contoh dari poin-poin yang disampaikan dalam merepresentasikan sebuah grafik atau data pada *academic writing task International English Language Testing System* (IELTS). Selain itu penulis menggunakan beberapa hasil keluaran dari penelitian sebelumnya sebagai *initial corpus* seperti hasil dari penelitian DWP (Putra *et al.*, 2017) dan GNG (Abidin *et al.*, 2018). Salah satu contoh penulisan grafik pada *academic writing task* pada tes IELTS ini dapat dilihat pada gambar 4.14.



Sample Answer 2:

The line graph outlines the production and demand for steel in million tonnes in the UK in 2010 and the number of workers employed in this sector. Generally speaking, the production of steel in the UK was higher than the demand and the number of workers in this sector directly affected the production capability.

At the beginning of 2010, the demand for steel was 4000 million tonnes which was double than the actual demand. The steel production and demand in April remained almost the same in April. In August, at the end of the third quarter, the demand unchanged but the production dropped to just over 3000 million tonnes. The demand dramatically fell in November and at the end of the year, the demand went higher than the production. During this time the demand for steel was 2000 million tonnes against less than 1000 million tonnes production.

The production of steel is correlated with the employed workers in this sector. In January 2010, the steel industry in the UK employed 5 million workers and it went as high as roughly 5.8 million in February. From March to July, the workers' number stood at an average 2 million and kept on declining from September till December. At the end of the year, the UK steel sector employed 1 million people.

Gambar 4.14 Contoh *academic writing task* pada IELTS

Poin-poin yang disampaikan dalam merepresentasikan sebuah grafik atau data pada tes IELTS diantaranya adalah informasi tentang apa saja yang disampaikan pada data atau grafik tersebut, dan bagaimana kondisi grafik atau data tersebut secara umum. Selain itu disampaikan juga beberapa *event* seperti kondisi kenaikan atau penurunan yang ekstrem atau *extreme event*, lalu disampaikan juga kondisi dimana data tidak berubah-ubah atau stabil, dan *event* lainnya seperti *repeated event* serta korelasi antar parameter.

According to the dataset, from 1/7/2013 00:00:00 to 22/5/2018 00:00:00, with parameters: CloudCoverage, Temperature, WindSpeed, AtmosphericPressure. It is illustrate that, CloudCoverage, and Temperature trend is constant, AtmosphericPressure trend is increase, but WindSpeed trend is decrease. CloudCoverage parameter is more higher than last month. There was some repeated values: CloudCoverage stayed constant at 10 point during 1st to 31st August 2015, and Temperature stayed constant at 20 point during 5th to 31th August 2015. WindSpeed increased extreamly to 100 point from 3rd to 4th December, but AtmosphericPressure decreased significantly to 200 point from 6th to 7th August.

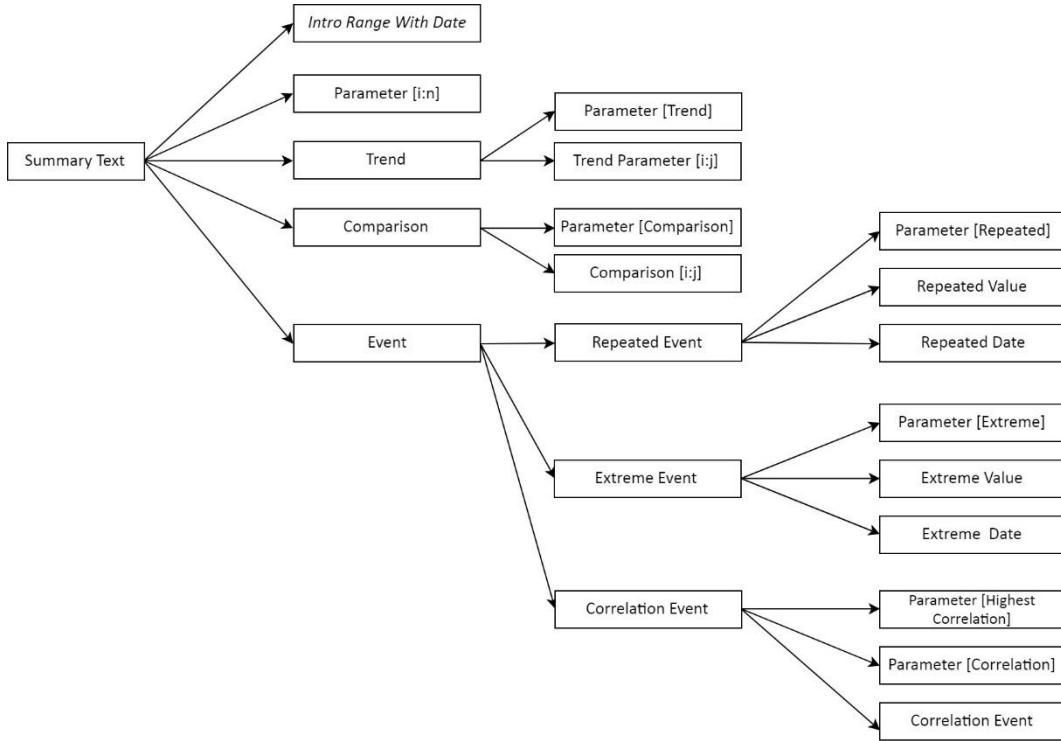
Gambar 4.15 *Initial Corpus* untuk ringkasan data

Dengan mengikuti langkah-langkah dalam pembentukan *Initial Corpus* yang sudah dijelaskan sebelumnya, peneliti membuat *Initial Corpus* yang digambarkan pada gambar 4.15. Sehingga didapatkan skema *initial corpus* seperti pada gambar 4.16.

<i>Summary -></i>	<i>Repeated Event -></i>
<i>Intro with range Date,</i>	<i>Parameter [Repeated],</i>
<i>Parameter [i:n]</i>	<i>Repeated Value,</i>
<i>Trend,</i>	<i>Repeated Date,</i>
<i>Comparison,</i>	<i>Extreme Event -></i>
<i>Event,</i>	<i>Parameter [Extreme],</i>
<i>Trend -></i>	<i>Extreme Value,</i>
<i>Parameter [Trend]</i>	<i>Extreme Date,</i>
<i>Trend Parameter [i:j]</i>	<i>Correlation Event -></i>
<i>Comparison -></i>	<i>Highest Correlation,</i>
<i>Parameter [Comparison],</i>	<i>Parameter [Correlation],</i>
<i>Comparison [i:j]</i>	<i>Correlation Event,</i>
<i>Event -></i>	
<i>Repeated Event,</i>	
<i>Extreme Event,</i>	
<i>Correlation Event,</i>	

Gambar 4.16 Skema teks untuk ringkasan data

Setelah menentukan skema teks untuk ringkasan data, maka didapatkan pohon struktur dokumen seperti pada gambar 4.17. Pada struktur dokumen terdapat variabel i, j dan n, variabel i menunjukkan iterasi dimulai dari i, sedangkan j dan n merupakan batas iterasi, n menunjukkan jumlah keseluruhan parameter, sedangkan j jumlah keseluruhan jenis trend atau pesan. Pada gambar tersebut, struktur dokumen pada ringkasan data terdiri dari *Intro with Range Date*, Parameter ke-i hingga ke-n, *Trend*, dan *Event*. Dimana *Trend* merupakan *Routine Message*, sedangkan *Event* merupakan *Significance Event Message*.

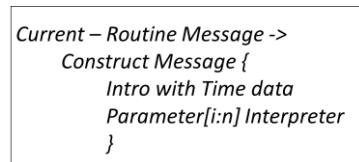


Gambar 4.17 Struktur Pohon untuk teks ringkasan data

4.2.4.2 Perencanaan Dokumen untuk Data Terkini

1. Content Determination

Pemilihan konten untuk data terkini atau data terakhir hampir sama dengan proses pemilihan konten untuk ringkasan data, dimana konen-konten yang akan ditampilkan terbagi menjadi dua kategori, yakni *Routine Message* dan *Significance Event Message*. Untuk konten *Routine Message* dapat dilihat pada gambar 4.18.



Gambar 4.18 *Routine Message* untuk data terkini

Sedangkan penentuan *Significance Event Message* untuk data terkini akan ditampilkan jika data terakhir melebihi atau sama dengan nilai statistik maksimum atau minimum data *statistical summary*, untuk lengkapnya dapat dilihat pada gambar 4.19

```

Current – Significant Event Message ->
IF Max Index == Current Index or Min Index == Current Index |
THEN Construct Message {
    Parameter[Min or Max] Message
}

```

Gambar 4.19 *Significance Event Message* untuk data terkini

Sebagai contoh jika kita menggunakan data partikel udara pada tabel 4.4 dan data *dummy* pada tabel 4.6, maka akan didapatkan hasil *Content Determination* seperti pada tabel 4.24, seperti yang dapat kita lihat pada tabel tersebut, pada bagian paragraf data terkini untuk data *dummy* ini tidak memiliki pesan *significant* berbeda dengan data partikel udara.

Tabel 4.24 Hasil *Content Determination* untuk data terkini

Data partikel udara pada tabel 4.4	
Routine Message	This hour data represent that: pm2.5, and Dew Point in low condition. Temperature in very cold condition. Pressure in high condition.
Significance Event Message	Is, and Ir reached their lowest value on this hour.
Data <i>dummy</i> pada tabel 4.6	
Routine Message	Today data show that: Param1 in medium condition. Param2 in high condition.
Significance Event Message	-

2. Document Structuring

Initial Corpus yang digunakan sebagai acuan dalam menampilkan pesan-pesan untuk data terkini dapat dilihat gambar 4.20.

```

Today data describe that, CloudCoverage still foggy, Temperature stay stable at
warm, WindSpeed is ligh breeze, AtmosphericPressure is very low which is the lowest
value of the month.

```

Gambar 4.20 *Initial Corpus* untuk pesan data terkini

Setelah menentukan *Initial Corpus*, maka dilakukan pembuatan skema teks seperti pada ringkasan data, sehingga didapatkan hasil seperti pada gambar 4.21.

```

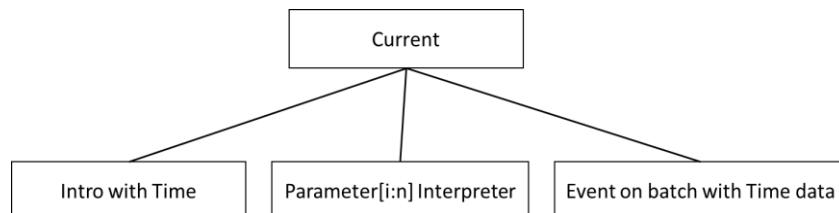
Current ->
  Intro with Time,
  Parameter[i:n] Interpreter,
  Event on batch with Time data

```

Gambar 4.21 Struktur teks data terkini

Sedangkan untuk struktur pohonnya dapat dilihat pada gambar 4.22.

Pada pohon tersebut, kita bisa menemukan variabel i dan n, variabel i merupakan indeks data yang dimulai dari 1, sedangkan variabel n merupakan banyaknya parameter yang ada, sehingga i:n berarti melakukan perulangan sebanyak parameter yang ada. Struktur dokumen untuk data terkini, terdiri dari *Intro With Time* yang kemudian diikuti dengan interpretasi untuk setiap parameter yang ada, jika kondisi dari *Significant Event Message* terpenuhi, maka ditampilkan juga *Event* bersama waktu datanya.



Gambar 4.22 Pohon struktur untuk teks data terkini

4.2.4.3 Perencanaan Dokumen untuk Prediksi

1. Content Determination

Pada pemilihan konten untuk prediksi, terdapat juga dua bagian yang akan ditampilkan seperti pada bagian sebelumnya, yaitu *Routine Message* dan *Significance Event Message*. Untuk *Routine Message* pada bagian prediksi, dapat dilihat pada gambar 4.23.

```

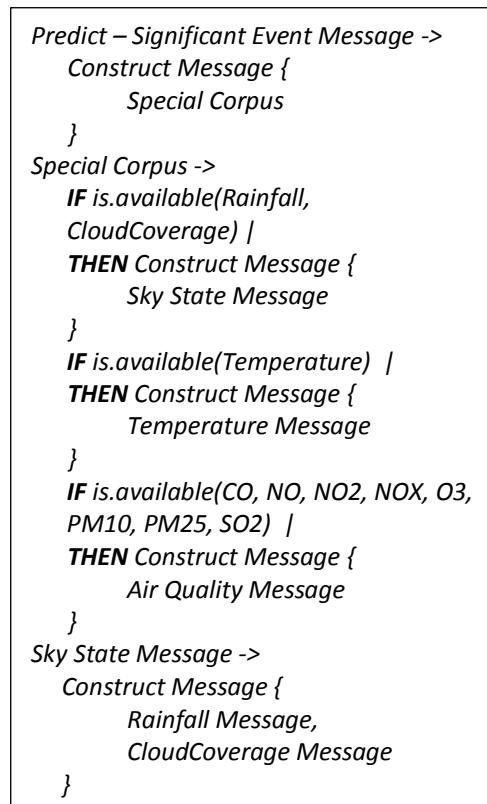
Predict – Routine Message ->
  Construct Message {
    Intro,
    Parameter[i:n] Interpreter Predict,
    Comparasion with last data
  }

```

Gambar 4.23 *Routine Message* untuk Prediksi data

Sedangkan *Significance Event Message* pada bagian prediksi ini berupa *Special Corpus*. *Special Corpus* ini merupakan teks keluaran yang

diadaptasi dari penelitian DWP yang dikembangkan oleh Putra (2017) untuk bagian prediksi. *Special Corpus* terdiri dari tiga bagian, yaitu *Sky State Message*, *Temperature Message*, dan *Air Quality Message*. Pesan *Sky State Message* akan ditampilkan jika terdapat parameter *Rainfall* dan *CloudCoverage* pada data masukan, begitu juga dengan *Temperature Message* hanya akan ditampilkan jika terdapat parameter *Temperature*, untuk *Air Quality Message* akan ditampilkan jika pada data masukan terdapat delapan parameter kualitas udara yang menjadi masukan sistem DWP, yakni CO, NO, NO2, NOX, O3, PM10, PM25, dan SO2. Untuk lebih lengkapnya, *Significant Event Message* ini digambarkan pada gambar 4.24.



Gambar 4.24 *Significant Event Message* untuk Prediksi data

Sebagai contoh jika kita menggunakan data *dummy* pada tabel 4.6 maka akan didapatkan hasil *Content Determination* seperti pada tabel 4.25

Tabel 4.25 Hasil *Content Determination* untuk Prediksi data

Routine Message	From the prediction result, it's estimated that Param1 will still stable at medium. Param2 will normally move to high.
Significant Event Message	-

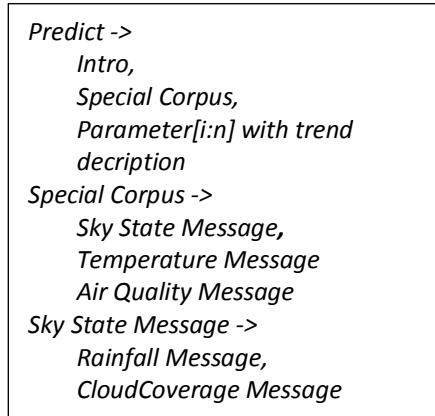
2. Document Structuring

Initial Corpus yang digunakan sebagai acuan dalam menampilkan pesan-pesan untuk prediksi datadapat dilihat gambar 4.25.

Regarding to the prediction result, it's forecasted that tomorrow sky will be light rain although it's covered by partly cloudy sky. Followed by temperature which decreased to warm. Cloud Coverage will averagely turn to partly cloudy. Temperature will stay stable at warm. Wind Speed will still stable at light Breeze. Wind Direction will move to West. Rainfall will constant at light rain. "

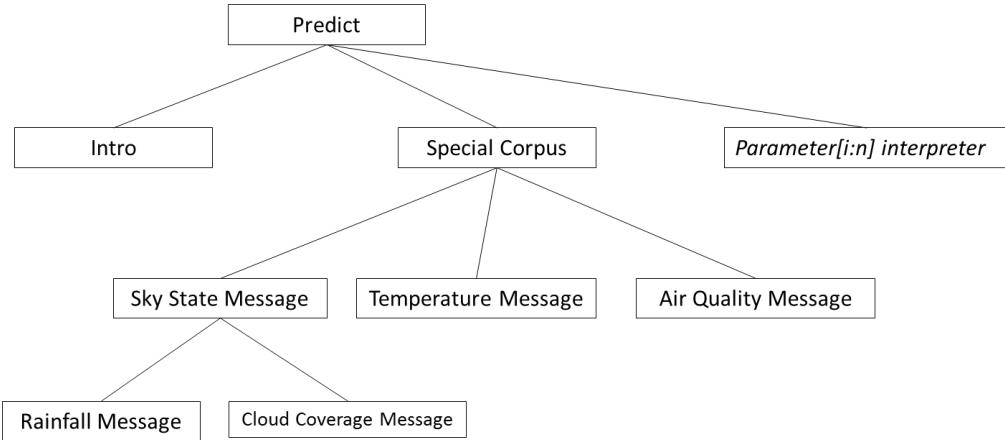
Gambar 4.25 *Initial Corpus* untuk Prediksi Data

Setelah menentukan *Initial Corpus*, maka dilakukan pembuatan skema teks seperti pada ringkasan data, sehingga didapatkan hasil seperti pada gambar 4.26.



Gambar 4.26 Struktur teks untuk prediksi data

Sedangkan untuk struktur pohnnya dapat dilihat pada gambar 4.27, dimana struktur pohn tersebut terdiri dari intro, *special corpus*, dan prediksi untuk setiap parameter.



Gambar 4.27 Pohon struktur prediksi data

4.2.5. Model Komputasi untuk *Microplanning and Realisation*

Tugas utama dari tahap *Microplanning* ini adalah mengemas pesan-pesan yang sudah dibangkitkan sebelumnya sehingga dapat tersusun sebaik mungkin mengikuti struktur dokumen yang sudah dibuat pada tahap sebelumnya. Proses *Microplanning* ini terbagi menjadi tiga proses yaitu, *Lexicalisation*, *Aggregation*, dan *Referring Expression Generation*. Proses *lexicalisation* dalam penelitian ini menggunakan pendekatan *rule-based* seperti pada penelitian DWP, dimana setiap pilihan leksikal dibangkitkan berdasarkan rule yang diperoleh dari berbagai sumber. Misalnya, untuk merepresentasikan perubahan setiap parameter pada teks prediksi, pemilihan frase yang tepat akan melalui beberapa tahapan sehingga keluarannya adalah frase seperti “*change progressively*”, “*shifted*”, dan lain-lain. Sedangkan *aggregation* merupakan proses bagaimana pesan yang berkaitan dihubungkan, baik baik pesan yang memiliki kesamaan ataupun pesan yang bertolak belakang. Selanjutnya, *referring expression generation* merupakan bagaimana mendefinisikan subjek informasi, contohnya: “*this month, this day...*” (Putra *et al.*, 2017). Dimana untuk *input*, proses, dan *output* pada tahap ini adalah sebagai berikut:

- *Input: Content and Structure*
- Proses: *Microplanning and Realisation*
- *Output: Summary Text, Current Text, dan Predict Text*

Berikut adalah penjelasan mengenai sub-proses dari tahapan *Microplanning and Realisation*.

4.2.5.1 *Lexicalisation*

Proses *Lexicalisation* atau leksikalisasi adalah proses untuk menentukan bagaimana konsep, informasi, atau elemen pengetahuannya diungkapkan dalam bentuk kata-kata, sehingga teks yang dihasilkan dapat lebih variatif tanpa mengurangi *knowledge* dari konten yang akan disampaikan. Misalnya pada teks untuk prediksi data, banyak frasa yang dapat merepresentasikan kondisi jika perubahan ekstrim terjadi, seperti “*change progressively to*”, “*turn progressively to*”, “*move progressively to*”, “*shifted progressively to*”, dan lainnya. Proses ini merujuk pada representasi *Trend Description* untuk kualitas udara yang diterapkan dalam DWP (Putra *et al.*, 2017).

Proses leksikalisasi pada sistem DWP mengacu pada proses leksikalisasi yang dilakukan oleh (Ramos-Soto *et al.*, 2016), dimana setidaknya ada dua pesan yang direpresentasikan proses *Trend Description* ini. Yang pertama yaitu *Change Type*, dan yang kedua adalah *Change Label*. Untuk contoh hasil dari proses ini dapat dilihat pada gambar 4.22.

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

Gambar 4.22 Contoh dari hasil proses *Trend Description*

Frasa “*change progressively*” merupakan pesan *Change Type* atau pesan merepresentasikan perubahan apa yang terjadi pada suatu parameter, yang kemudian diikuti oleh frasa “*hazzardous*” yang merupakan *Change Label* atau frasa yang merepresentasikan hasil terakhir dari perubahan tersebut. Dalam menentukan *Trend Description*, langkah pertama yang harus dilakukan adalah membuat himpunan berisi hasil representasi data kemarin, data terkini, dan data prediksi, contohnya jika kita menggunakan ringkasan data pada tabel 4.20 untuk parameter Param1 sehingga didapatkan himpunan seperti pada tabel 4.26.

Tabel 4.26 Hasil interpretasi data untuk contoh kasus kualitas udara

Yesterday (TD1)	Today (TD2)	Tomorrow (TD3)
Medium	Medium	Medium

Kemudian setiap anggota himpunan diinterpretasikan sesuai dengan nilai indeksnya. *Indeks* tersebut didapatkan berdasarkan urutan dalam partisi keanggotaan yang sudah dijelaskan pada bagian *Data Interpretation*, urutan tersebut dimulai dengan indeks 0. Urutan tersebut dapat dilihat pada tabel 4.27.

Tabel 4.27 Indeks interpretasi kualitas udara

Partisi	Index
Very low	0
Low	1
Medium	2
High	3
Very High	4

Setelah dipetakan berdasarkan indeks penginterpretasiannya, kini himpunan TD yang memuat hasil dari interpretasi beberapa data, menjadi himpunan TDI yang hanya memuat indeks-indeks penginterpretasiannya saja, maka $TDI = \{0,1,1\}$. Lalu proses selanjutnya adalah mencari *Index Variation* (IV) dengan persamaan berikut (Ramos-Soto *et al.*, 2015):

$$IV = \{ iv1 = td2 - td1, iv2 = td3 - td2, iv3 = val3 \}$$

Maka didapatkan nilai *Index Variation* dengan perhitungan sebagai berikut:

$$IV = \{ iv1 = 2 - 2, iv2 = 2 - 2, iv3 = "Medium" \}$$

$$IV = \{ 0, 0, "Medium" \}$$

Setelah didapatkan nilai IV, perubahan parameter mulai terlihat, dimana pada proses selanjutnya himpunan IV akan direpresentasikan menjadi *Index Variation Lexicalisation* IVL yang berisikan simbol-simbol dengan menggunakan kaidah pada gambar 4.28. Dimana jika IV akan

direpresentasikan menjadi simbol “+” jika bernilai positif, “-“ jika negatif, dan “0” jika IV bernilai 0.

$ IV $	{	'+' jika $ IV > 0$ '-' jika $ IV < 0$ '0' jika $ IV = 0$	—————>	Lebih buruk / menurun Lebih Baik / meningkat Tidak berubah / stabil
--------	---	---	--------	---

Gambar 4.28 Rule penginterpretasian IVL

Dari serangkaian proses diatas, maka dihasilkan nilai dari *Index Variation* untuk *Linguistic Description* (LD) kualitas udara adalah : $LD_{AirQuality} = \{ "0", "0", Medium \}$. Inilah nilai terakhir yang kemudian nilai ini akan direpresentasikan menggunakan skema pada gambar 4.29 yang diperkenalkan oleh (Ramos-Soto et al., 2016).

{TD}	→ {Change Type}{Change Label}
{Change Type}	→ {stable} {MediumChange} {StartChange} {EndChange} {ProgressiveChange}
{stable}	→ MaintainMaintain
{MediumChange}	→ WorsenImprove ImproveWorsen
{StartChange}	→ WorsenMaintain ImproveMaintain
{EndChange}	→ MaintainWorsen MaintainImprove
{ProgressiveChange}	→ WorsenWorsen ImproveImprove
{ChangeLabel}	→ very low low normal high very high interpretation result

Gambar 4.29 Skema untuk mendeskripsikan pesan *Trend Description*

Himpunan LD yang berisi simbol-simbol akan direpresentasikan menjadi sebuah pesan *Trend Description*, dimana simbol “” berarti “*Improve*” atau lebih baik, lalu simbol “-“ berarti “*Worsen*” atau lebih buruk, sedangkan “0” diartikan sebagai “*Maintain*” atau tidak berubah. Dikarenakan himpunan LD yang dihasilkan pada proses sebelumnya berisi $LD_{AirQuality} = \{ "0", "0", Medium \}$, maka *Change Type* dari pesan tersebut adalah “*MaintainMaintain*” yang tergolong pada kategori *Stable*. Setelah didapatkan, *Change Type* dari pesan tersebut, langkah selanjutnya adalah menentukan frasa yang akan disampaikan pada teks keluaran. Dimana penentuan ini dilakukan dengan cara *random* sesuai dengan frasa yang terdapat pada *corpus* seperti yang terdapat pada gambar 4.30. Sehingga

dihadarkan pesan prediksi untuk parameter Param 1 pada data *dummy* sebagai berikut, “*Param1 will still stable at medium*”.

{Stable} -> {"stay stable at", "keep stable at", "stay constant at", "steady at", "still stable at"}
--

Gambar 4.30 *ProgressiveChange Corpus*

Pada penelitian ini, penerapan proses leksikalisis khususnya *Trend Description* dilakukan untuk setiap parameter yang terdapat pada teks prediksi data. Dimana data-data yang sudah diinterpretasikan pada proses *Data Interpretation* akan diproses sehingga dihasilkan teks yang lebih variatif dan *unspecific*.

4.2.5.2 Aggregation

Proses agregasi menentukan bagaimana suatu pesan digabungkan untuk menghasilkan spesifikasi frasa yang sesuai dengan kalimat kompleks (Ramos-Soto et al., 2016). Dalam proses agregasi ini digunakan teknik *Simple Conjunction*, dimana sebuah kalimat dapat digabungkan dengan frasa sederhana seperti “*and*”, “*but*”, “*although*” , dan lainnya sesuai dengan kebutuhan sistem (Reiter & Dale, 1997).

Salah satu contoh penerapan agregasi adalah saat pembentukan pesan *Sky State* pada *Special Corpus* yang terdiri dari pesan-pesan yang mengandung informasi mengenai hujan dan cakupan awan (Putra et al., 2017). Pada gambar 4.31 Dapat dilihat bahwa kalimat yang dibangun dihubungkan dengan kata penghubung atau *Conjunction* berupa ”*covered with*” yang menghubungkan dua

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

Gambar 4.31 Contoh Phrase Aggregation (Putra et al., 2017)

Dalam menentukan kata penghubung yang tepat, sistem perlu mengetahui bagaimana hubungan antara pesan-pesan yang akan dihubungkan nantinya. Contohnya untuk kasus pesan curah hujan dan cakupan awan pada sistem DWP, jika didapatkan pesan curah hujan berupa “*heavy rain*” atau hujan deras dan pesan cakupan awan berupa “*overcast*” yang berarti mendung, maka pesan ini berada dalam keadaan yang sebanding sehingga digunakan kata penghubung “*covered with..*” yang berarti “ditutupi oleh...”. Contoh lainnya, jika pesan curah hujan yang didapatkan adalah “*no rain*” atau tidak ada hujan, namun cakupan awan adalah “*moslty cloudy*” atau umumnya berawan, maka pesan yang bertolak belakang, sehingga digunakan kata penghubung “*although it's covered by*” yang berarti “meskipun ditutupi oleh...” .

Proses selanjutnya adalah penentuan nilai kontras sehingga sistem mampu mengenali mana pesan yang sebanding atau bertolak belakang (Putra *et al.*, 2017). Hal ini hampir sama dengan proses *Trend Description*, dimana penentuan nilai kontras menggunakan bantuan nilai indeks, jika nilai indeks dari kedua pesan yang akan disampaikan bernilai sama, maka pesan tersebut sebanding, jika nilai indeksnya berbeda maka pesan tersebut bertolak belakang. Pemetaan indeks untuk menentukan kontras pada pesan dapat dilihat pada tabel 4.28.

Tabel 4.28 Nilai kontras dalam proses agregasi dengan Simple Conjunction (Putra *et al.*, 2017)

Curah Hujan		Cakupan Awan	
Partisi	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

4.2.5.3 Referring Expression Generation

Pada tahap ini , dilakukan pemilihan kata atau ungkapan untuk merepresentasikan sebuah entitas, sehingga frasa yang digunakan dapat lebih variatif. Tahap ini relatif sederhana, dimana implementasi untuk tahap ini dapat diterapkan secara *hard code* (Reiter & Dale, 1997). Namun, peneliti menerapkan pembangkitan frasa secara *random* seperti yang dilakukan pada sistem DWP (Putra *et al.*, 2017). Contohnya, dalam penentuan *adverb* seperti “*sharply*”, “*extremly*”, “*dramatically*”, “*significantly*”, dan lainnya akan dipilih secara *random* untuk merepresentasikan ketika ada kenaikan atau penurunan ekstrim terjadi.

4.2.6. Hasil Keluaran Sistem

Teks direalisasikan kedalam bentuk aktual berdasarkan struktur yang telah dibuat pada saat proses *document planning* (Reiter, 2011). Realisasi teks dapat ditampilkan menggunakan bahasa pemrograman seperti HTML, LaTeX, RTF, atau lainnya.

Sehingga jika digunakan data *dummy* pada tabel 4.6 akan didapatkan hasil seperti pada tabel 4.29

Tabel 4.29 Hasil akhir untuk contoh kasus data partikel udara

According to the daily Beijing Air Particle data between 01/01/2019 00:00 to 01/20/2019 00:00, with parameters: Param1, Param2, and Param3, it illustrated that trend of all variable is increased. All parameters are higher than last week's data. There were some repeating value more than 2 days: Param1 stayed constant at very low during 3-4 Jan 2019. Param1 stayed constant at medium during 10-11 Jan 2019. Param2 fluctuated dramatically (increased 21 points and decreased 21 points). There is a same Param3 data pattern in the last 7 days (13-20 Jan 2019) with data pattern from 2-9 Jan 2019. Param1 appears to have a highest impact to all variable with very strong relationship in average. An increase in Param1 resulted an increase in Param2.

Today data show that: Param1 in medium condition. Param2 in high condition.

From the prediction result, it's estimated that Param1 will still stable at medium. Param2 will normally move to high.

4.3. Pengembangan Sistem *Data-to-text Unspecific News Generator*

Pada sub-bab ini akan dipaparkan secara menyeluruh mengenai pengembangan sistem *D2T Unspecific News Generator* secara teknis dengan menggunakan proses *Linear Sequential Model* pada sub-bab metode penelitian. Proses pengembangan perangkat lunak dengan menggunakan *Linear Sequential*

Model terdiri dari empat tahap, yaitu analisis, desain, implementasi, dan testing (Pressman, 2001a). Pengembangan model dengan menggunakan *Linear Sequential Model* ini dipilih dikarenakan sistem yang akan dibangun sudah didefinisikan kebutuhan-kebutuhannya secara jelas dan tidak akan mengalami perubahan yang kebutuhan yang signifikan. Selain itu proses model ini dipilih dikarenakan pengembangan tahapnya yang cukup sederhana, sehingga waktu pengembangan sistem tidak memakan waktu yang terlalu lama. Penjelasan mengenai sub-proses pengembangan sistem akan dijelaskan pada sub-bab selanjutnya.

4.3.1 Analisis Sistem D2T UNG

Tahap analisis merupakan tahap pertama dalam pengembangan sistem D2T *Unspecific News Generator (UNG)*. Pemodelan ini diawali dengan mendefinisikan kebutuhan-kebutuhan sistem yang akan diaplikasikan ke dalam bentuk perangkat lunak nantinya. Hal ini sangat penting, dikarenakan pada tahap ini akan ditentukan batasan-batasan dari sistem yang akan dibangun, sehingga wilayah atau *scope* dari sistem yang dibangun tetap jelas dan tidak melebar. Pada tahap ini, didefinisikan terlebih dahulu bagaimana sifat dari perangkat lunak yang akan dibuat, kebutuhan apa saja yang perlu dibuat, dan pemilihan bahasa pemrograman serta bagaimana sistem akan ditampilkan nantinya.

Sehingga pada proses ini didapatkan beberapa kebutuhan utama dari sistem D2T yang akan dikembangkan, diantaranya seperti berikut:

- Sistem D2T yang dikembangkan dapat membangkitkan berita berdasarkan data apapun yang diberikan (data apa saja yang bersifat *time series* dan eksak) baik mempunyai *header* maupun tidak.
- Sistem yang dikembangkan mampu melakukan interpretasi data dengan *corpus* yang didefinisikan oleh pengguna.
- Sistem yang dikembangkan mampu menerima masukan data baik dengan tipe *numerical* maupun *categorical*.
- Sistem yang dikembangkan mampu menganalisa dan mencari kesamaan pola untuk data dengan tipe *categorical*.
- Sistem dapat menghasilkan berita yang berisikan hubungan antara parameter dari data yang dimasukan.

- Sistem yang dikembangkan harus mengacu pada arsitektur D2T yang diperkenalkan oleh (Reiter, 2011).

Setelah didefinisikan beberapa kebutuhan dari sistem yang akan dibangun, proses selanjutnya adalah tahap pembangunan desain yang akan dijelaskan pada sub-bab selanjutnya.

4.3.2 Desain Sistem D2T UNG

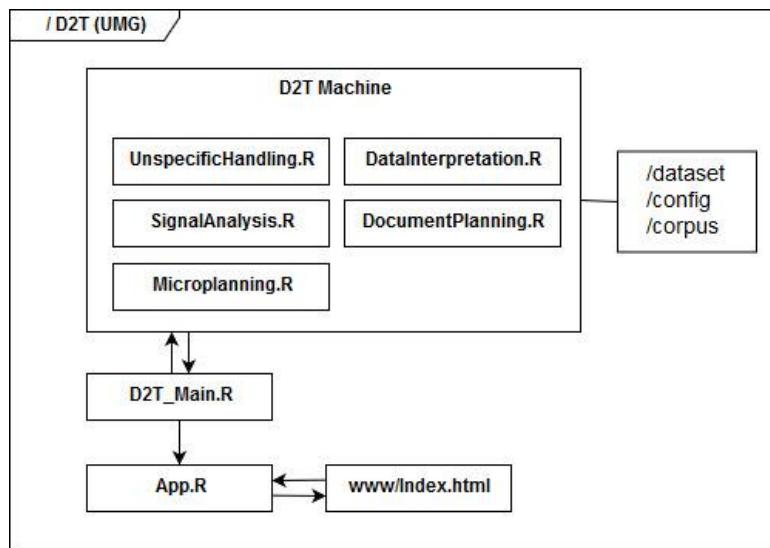
Setelah menentukan kebutuhan-kebutuhan yang diperlukan dalam pembangunan sistem UNG, proses selanjutnya adalah menentukan desain dari sistem UNG yang akan dibangun. Pada tahap ini, kebutuhan-kebutuhan yang sudah didefinisikan sebelumnya akan direpresentasikan ke dalam bentuk *blueprint* perangkat lunak sebelum dilakukan proses *coding*. Setidaknya ada tiga atribut yang akan ditentukan pada tahap ini diantaranya, struktur data, arsitektur perangkat lunak, dan prosedural (algoritmik) secara rinci. Penjelasan sub-proses dari tahap desain adalah sebagai berikut.

a. Struktur Data

Pada pemilihan desain struktur data untuk pembangunan sistem UNG diperlukan bahasa pemrograman yang memiliki struktur data yang fleksibel. Dimana pada pembangunan sistem UNG ini dipilih bahasa pemrograman R, dikarenakan bahasa pemrograman R memiliki kemampuan untuk mengolah berbagai tipe data, kemudahan *casting* variable, dan struktur data yang beragam. Selain itu juga, bahasa pemrograman R dipilih karena sifatnya yang *weakly typed* dimana variabel yang didefinisikan tidak perlu ditentukan tipe variabelnya. Diantara jenis-jenis data yang akan digunakan dalam pembangunan sistem UNG diantaranya, *integer*, *double*, *matrix*, *vector*, *list*, *data frame*, *character*, dan *text*. Konsep pemrograman dalam pengembangan sistem UNG adalah prosedural. Dimana konsep prosedural ini dipilih karena sistem UNG memiliki proses yang cukup kompleks, sehingga tugas-tugas kompleks tersebut akan dipecah menjadi beberapa sub-sub yang lebih kecil sehingga lebih mudah diimplementasikan dibandingkan dengan menggunakan konsep pemrograman berorientasi objek.

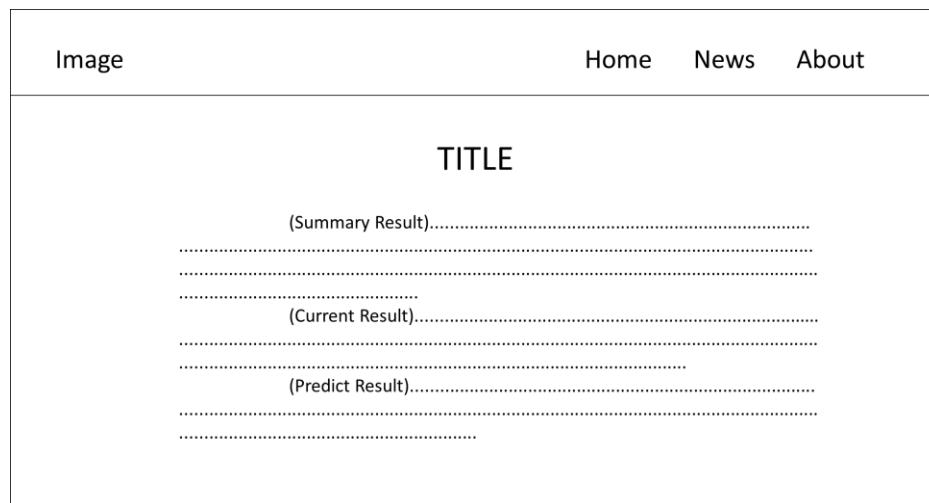
b. Arsitektur perangkat lunak

Sistem UNG dibangun dengan berbasis web menggunakan bahasa pemrograman R, Javascript, dan HTML. Penulis menggunakan *package* yang bernama *ShinyR*, sehingga hasil pemrosesan dari bahasa R akan direpresentasikan ke dalam bentuk web menggunakan *package* tersebut.



Gambar 4.32 Struktur file sistem UNG

Pada gambar 4.32 terlihat bahwa sistem D2T UNG yang dikembangkan menggunakan bahasa R dan ditampilkan dalam bentuk web dengan *package* ShinyR. Semua proses implementasi dan fungsi-fungsi utama disimpan pada D2T Machine yang terdiri dari, *UnspecificHandling.R*, *SignalAnalysis.R*, *DataInterpretation.R*, *DocumentPlanning.R*, dan *Microplanning.R*. Kelima file tersebut merepresentasikan setiap proses dari setiap sub-model dalam pembangunan sistem D2T. Sedangkan file *D2T_Main.R* berisikan kode program untuk memanggil fungsi-fungsi pada file D2T Machine, selain itu juga file *D2T_Main.R* akan berkomunikasi langsung dengan *App.R* yang berisikan kode program untuk mengeksekusi *package* *ShinyR* yang digunakan sebagai perantara dalam menampilkan antarmuka sistem. Gambar 4.33 menunjukan desain antarmuka sistem yang dibangun, dengan menggunakan template *css*, dan *framework bootstrap*.



Gambar 4.33 Antarmuka sistem *Unspecific News Generator* (UNG)

c. Prosedural sistem

Dalam proses pembangunan sistem UNG ini digunakan konsep prosedural, dimana setiap proses akan dipecah menjadi bentuk fungsi-fungsi. Setiap fungsi yang ada pada sistem UNG akan dikelompokan dan disimpan dalam file yang berbeda-beda seperti yang dapat kita lihat pada gambar 4.33. Setiap sub-model dari proses D2T yang sudah dipaparkan pada sub-bab pengembangan model, direpresentasikan dengan fungsi yang terdapat pada file-file yang terdiri dari, *UnspecificHandling.R*, *SignalAnalysis.R*, *DataInterpretation.R*, *DocumentPlanning.R*, dan *Microplanning.R*. Selain itu juga, setiap kebutuhan-kebutuhan dari hasil analisis sebelumnya diimplementasikan ke dalam file-file tersebut dalam bentuk fungsi.

4.3.3 Implementasi Sistem D2T UNG

Pada tahap ini dilakukan proses *coding*, dimana kebutuhan-kebutuhan dan desain yang sudah dihasilkan sebelumnya akan direpresentasikan ke dalam bentuk kode program menggunakan bahasa R mengikuti model proses yang sudah dijelaskan sebelumnya. Untuk lebih lengkapnya, sub-proses dari tahap ini akan dijelaskan pada sub-bab berikut.

4.3.3.1. Implementasi Proses *Unspecific Data Handling*

Untuk mengimplementasikan proses ini, penulis menggunakan *package data.table* untuk mengecek apakah data masukan memiliki header atau tidak, proses pembacaan data masukan menggunakan fungsi fread() yang terdapat pada *package data.table*. Pada gambar 4.34 terlihat proses pembacaan data dilakukan dengan fungsi fread() yang kemudian dikonversikan kedalam bentuk *data frame*. Penggunaan fungsi fread() ini bertujuan agar parameter yang tidak memiliki *header* akan tetap diproses dan diberi *default header* dengan nama *v2*, *v3*, *v4* dan seterusnya, lalu mengubah *header* parameter pertama menjadi *DateTime*, sehingga sistem akan tetap berjalan walaupun data masukan tidak memiliki header sekalipun.

```
#-----
# UNSPECIFIC DATA HANDLER |
#-----
# Force read, with default parameter v2,v3,v4,etc if there's no
# header available
dataset <- as.data.frame(fread(file="DatasetsExperiment/
Climatology#1.csv"))

# Rename first parameter header to DateTime
colnames(dataset)[1] <- "DateTime"

# Dataset with datetime Column dropped
datasetWithoutDate <- dataset[ , colnames(dataset) != 
"DateTime"]

# Parameter Header
columnName <- colnames(datasetWithoutDate)

# Parameter Config
mainConfig <- ReadConfig()
```

Gambar 4.34 Proses *Unspecific Data Handler*

Setelah proses pembacaan selesai, langkah selanjutnya adalah menentukan *data description*. Penentuan *data description* ini dilakukan dengan memanggil fungsi ReadConfig(). Pada gambar 4.36 diperlihatkan bagaimana proses penentuan *data description* parameter ini dilakukan.

```

ReadConfig <- function (){
  # Initializing
  nullSequence <- rep(NA, length(columnName))
  dfResult <- data.frame(ColName = columnName, Type =
  nullSequence, Rule = nullSequence, Alternate=nullSequence,
  stringsAsFactors=FALSE)

  # Read config from file
  mainConfig <- read.table("Config/datadescription.csv",
  header=TRUE, sep=", ")

  # Load setting from default file
  i<-1
  for(i in i:nrow(mainConfig)){
    tempColumn <- as.character(unlist(mainConfig$ColName[i]))
    dfResult[dfResult$ColName == tempColumn,] <-
    as.vector(unlist(mainConfig[mainConfig$ColName == tempColumn,]))
  }

  # Checking variable type with typeof()
  headerClass <- ClassHeaderChecker(dataset)
  headerClass <- headerClass[names(headerClass) != "DateTime"]

  # Merging R config with Default config
  i<-1
  for(i in i:length(headerClass)){
    tempColumn <- names(headerClass)[i]

    if(is.na(dfResult[dfResult$ColName == tempColumn,"Type"])){
      dfResult[dfResult$ColName == tempColumn,"Type"] <-
      headerClass[names(headerClass) == tempColumn]
    }
  }

  return(dfResult)
}

```

Gambar 4.35 Fungsi *data description* dalam proses *Unspecific Data Handler*

D2T_Apps > Config			
Name	Date modified	Type	Size
datadescription	12/01/2018 03:10 ...	Microsoft Excel C...	1 KB

Gambar 4.36 File *datadescription.csv* pada folder *Config*

Dalam proses pendata descriptionan parameter, yang dilakukan pertama kali adalah membaca file *datadescription.csv* yang terdapat folder *Config* seperti yang dapat dilihat pada gambar 4.36. File ini berisikan *data description* berupa tipe dari suatu parameter, cara penginterpretasiannya beserta bagaimana sebuah parameter akan ditampilkan pada teks keluaran nantinya, untuk lebih lengkapnya dapat dilihat pada lampiran. Setelah proses pembacaan selesai, proses selanjutnya adalah menyimpan *data description* tersebut ke dalam bentuk *data frame* yang

sudah diinisialisasi sebelumnya yaitu variable *mainConfig*. *Data description* yang akan disimpan hanyalah *data description* untuk parameter-parameter yang menjadi masukan.

Setelah disimpan kedalam bentuk variabel *data frame*, sistem akan melakukan pengecekan tipe parameter dengan menggunakan fungsi *typeof()* pada R. Jika pengguna tidak mendefinisikan tipe parameter pada file *datadescription.csv*, maka sistem otomatis menggunakan hasil dari fungsi *typeof()* tersebut sebagai acuan. Namun, jika pengguna sudah mendefinisikan tipe parameter pada file *datadescription.csv*, maka *data description* tersebut yang akan menjadi acuan. Misalnya jika kita menggunakan data klimatologi pada tabel 4.2, lalu ingin mendefinisikan data tersebut dengan *corpus* pada penelitian DWP (Putra *et al.*, 2017), maka akan dihasilkan *data description* seperti pada tabel 4.30.

Tabel 4.30 *Data description* untuk contoh kasus data klimatologi.

ColName	Type	Rule	Alternate
WindSpeed	numeric	crisp	Wind Speed
WindDirection	numeric	crisp	Wind Direction
CloudCoverage	numeric	crisp	Cloud Coverage
Temperature	numeric	fuzzy	NA
Rainfall	categorical	fuzzy	NA

Untuk contoh lainnya jika kita menggunakan data nilai tukar pada tabel 4.1, dikarenakan *data description* untuk parameter-parameter tersebut belum didefinisikan sebelumnya, maka akan dihasilkan *data description* seperti pada tabel 4.31, pada tabel tersebut terlihat *data description* yang bernilai NA karena tidak didefnisikan sebelumnya kini berubah menjadi tipe *numeric* yang merupakan hasil dari pendekripsi variabel secara *default*.

Tabel 4.31 *Data description* untuk contoh kasus data nilai tukar

ColName	Type	Rule	Alternate
USD	numeric	NA	U.S. Dollar
JPY	numeric	NA	Japan Yen
GBP	numeric	NA	Great British Pounds
CHF	numeric	NA	Confoederatio Helvetica Franc
SGD	numeric	NA	Singapore Dollar

ColName	Type	Rule	Alternate
MYR	numeric	NA	Malaysian Ringgit
HKD	numeric	NA	Hong Kong dollar
AUD	numeric	NA	Australian Dollar
CAD	numeric	NA	Canadian Dollar

4.3.3.2. Implementasi Proses *Signal Analysis*

a. Ringkasan data

Implementasi dalam proses peringkasan data, penulis menggunakan fungsi *min*, *max* dan *mean* yang terdapat pada R seperti pada gambar 4.37 berikut.

```
StatisticalAnalysis <- function(dataset2){
  dataset2WithoutDate <- dataset2[, colnames(dataset2) != "DateTime"]
  ColName <- Average <- MaxValue <- MaxIndex <- MaxDate <- MinValue <-
  MinIndex <- MinDate <- Trend <- c("")

  i=1
  n=length(dataset2WithoutDate)
  for(i in i:n){
    ColName[i] <- colnames(dataset2WithoutDate[i])
    #MAX
    MaxValue[i] <- max(dataset2WithoutDate[i])
    max_index2<- as.integer(which(dataset2WithoutDate[i]
                                   == max(dataset2WithoutDate[i])))
    MaxIndex[i] <- max_index2[1]
    max_index0 <- max_index2[1]
    MaxDate[i] <- as.character(dataset2$DateTime[max_index0])

    #MIN
    MinValue[i] <- min(dataset2WithoutDate[i])
    min_index2 <- as.integer(which(dataset2WithoutDate[i]
                                   == min(dataset2WithoutDate[i])))
    MinIndex[i] <- min_index2[1]
    min_index0 <- min_index2[1]
    MinDate[i] <- as.character(dataset2$DateTime[min_index0])

    #AVERAGE
    Average[i] <- mean(dataset2WithoutDate[,i])
    Trend[i] <- TrendAnalysis(ColName[i],dataset2WithoutDate[[i]],
                               as.double(MaxValue[i]),
                               as.double(MinValue[i]), nrowDataset)
  }

  dataset2Statistical <- data.frame(ColName, MaxDate, MaxValue,
  MaxIndex, MinDate, MinValue, MinIndex, Average, Trend);
  return(dataset2Statistical)
}
```

Gambar 4.37 Statistical Summary Function

Dalam menentukan *trend*, penulis menggunakan pendekatan *linear model* dengan fungsi lm() yang terdapat pada R. Hasil dari *linear model* tersebut digunakan untuk menentukan kecenderungan apakah *trend* data menaik atau menurun seperti pada gambar 4.38.

```
TrendAnalysis <- function(start,dataset, min, max){
  # Dataset is vector
  plot(as.numeric(unlist(dataset)), type="o", col="blue")
  dataset <- dataset[start:length(dataset)]

  if(length(unique(dataset)) == 1) {
    result <- "0"
  }else{
    x = c(1:length(dataset))
    reg = lm(dataset~x)

    #linear model range
    df <- reg$coefficients[2] + reg$coefficients[1]
    dl <- reg$coefficients[2]*length(dataset) +
    reg$coefficients[1]
    range <- dl-df

    #stats
    stat <- 0
    if(range < 0){
      range <- range * (-1)
      stat <- 1
    }
    #dataset range
    rangeReal <- max-min
    #5% minimum treshold
    if(range > 0.05*rangeReal){
      if(stat == 0){
        result <- "+"
      }else{
        result <- "-"
      }
    }else{
      result <- "0"
    }
    abline(reg,col="red")
  }
  return(result)
}
```

Gambar 4.38 *Trend Analysis function*

b. *Extreme Event*

Dalam pendekripsi *Extreme Event*, penulis menggunakan fungsi ResumeEventExtreme(), dengan *input* parameter yang bertipe *numerical*, dan *statistical summary* dari hasil peringkasan data. Kenaikan atau

penurunan yang digolongkan ke dalam *extreme event* jika kenaikan atau penurunan lebih besar dari 65% dari interval data (data maksimum – data minimum). Setelah didapatkan nilai kenaikan berserta indeks dan interpretasinya, maka informasi tersebut akan disimpan dalam variabel dengan tipe *data frame*. Untuk lebih lengkapnya implementasi *Extreme Event* dapat dilihat pada gambar 4.39.

```
ResumeEventExtreme <- function(datasetWithoutCatDate,
statisticalResume, type=NULL) {
  # initializing
  i <- 1
  vectorGrowth <- vectorStartIndex <- vectorEndIndex <-
  vectorInterpreter <- c()
  for(i in 1:length(datasetNumericalWithoutDate)){
    listColumn <- datasetNumericalWithoutDate [[i]]

    # Get highest growth/decay with their index
    if(type == "Growth"){
      listExtremeAnalysisResult <-
      ResumeHighestGrowthAnalysis(diff(listColumn), "Growth")
    }else if (type == "Decay"){
      listExtremeAnalysisResult <-
      ResumeHighestGrowthAnalysis(diff(listColumn), "Decay")
    }

    #store growth/decay value, start/end index into vector
    vectorGrowth[i] <-listExtremeAnalysisResult$valueResult
    vectorStartIndex[i] <-
    listExtremeAnalysisResult$startIndexResult
    vectorEndIndex[i] <-listExtremeAnalysisResult$endIndexResult

    #checking if range > 65% data range
    vectorInterpreter[i] <-
    InterpreterExtremeEvent(vectorGrowth[i], statisticalResume[i,])
  }
  #exception
  vectorEndIndex <- vectorEndIndex + 1

  #Combine all process into df
  .....
}
return(dfExtremeEvent)
}
```

Gambar 4.39 *ResumeEventExtreme function*

c. *Repeated Event*

Pada pendektsian sinyal *Repeated Event* dilakukan dengan menggunakan fungsi *ResumeRepeatedAnalysis()*, yang didalamnya terdapat fungsi *rle()* atau *run length encoding* dimana hasil dari fungsi *rle()*

adalah sebuah vektor boolean yang menunjukkan terjadinya perulangan data, jika ditemukan nilai yang berulang, maka boolean akan bernilai *true*. Perulangan yang termasuk kedalam kategori *Repeated Event* adalah jika perulangan tersebut melebihi 10% dari jumlah data. Gambar 4.40 menunjukkan bagaimana proses ini dimplementasikan.

```
#Repeated Value analysis
ResumeRepeatedAnalysis <- function(dataset){

  lengthEncoding <- rle(dataset)

  #limit
  n <- length(dataset) * 0.1

  repeatedSequence <- rep(lengthEncoding$lengths >= n,
  times=lengthEncoding$lengths)

  #Example repeatedSequence
  #[1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE

  RepValue <- as.numeric(table(repeatedSequence)[["TRUE"]])
  if(!is.na(RepValue)){
    # Get
    dt <- data.frame(number = rle(repeatedSequence)$values,
lengths = rle(repeatedSequence)$lengths)
    # Get the end
    dt$end <- cumsum(dt$lengths)
    # Get the start
    dt$start <- dt$end - dt$lengths + 1
    # Selecting column
    dt <- dt[dt$number == TRUE, c("number", "start", "end")]
    result <- list(RepValue = nrow(dt), Start = dt$start, End =
dt$end)
  }else{
    result <- list(RepValue = 0, Start = 0, End = 0)
  }

  return(result)
}
```

Gambar 4.40 *ResumeRepeatedAnalysis function*

d. Prediksi Data

Dalam proses prediksi data, penulis menggunakan *package xts* dan *smooth*. Package *xts* digunakan untuk mengkonversi data masukan kedalam bentuk *time-series* yang kemudian diurutkan berdasarkan *DateTime* secara *ascending*. Sedangkan package *smooth* digunakan untuk melakukan prediksi data dengan *exponential smoothing*. Pada gambar 4.41 proses

prediksi data dilakukan didalam fungsi `ResumeEventRepeat()`, setelah mengkonversi data menjadi bentuk *time-series* dengan menggunakan fungsi `xts()`, langkah selanjutnya adalah prediksi data dengan fungsi `forecast()` yang terdapat pada *package smooth*.

```
PredictDataset<-function(dataset, format="%m/%d/%Y %H:%M") {
  result <- c()
  lengthWithoutDate <- length(dataset[,-which(colnames(dataset) ==
  == "DateTime")])

  # Convert to TS data
  dataSeries <- xts(dataset[,-which(colnames(dataset) ==
  == "DateTime")], order.by=as.Date(dataset[,"DateTime"], format))

  # Forecasting with Ex.SMOOTH
  i<-1
  for(i in i:lengthWithoutDate){
    result[i] <- forecast(dataSeries[,i], h=1)$mean
  }

  names(result) <- colnames(dataset[ , colnames(dataset) != "DateTime"])

  # print(result)
  return(result)
}
```

Gambar 4.41 *PredictDataset function*

e. *String Matching*

Pendeteksian sinyal *string matching* ini hanya diterapkan untuk parameter dengan tipe *categorical*. Sehingga meskipun data masukan memiliki parameter dengan tipe *categorical*, sistem akan tetap bekerja seperti biasa. Proses pendeteksian sinyal *string matching* ini menggunakan algoritma *Knuth Morris Prat* (KMP). Dimana pola yang akan cari adalah sebanyak n data terakhir, nilai n ditentukan berdasarkan dengan interval data, jika data dengan interval per jam, maka diambil data atau pola acuan yang diambil sebanyak 6 baris atau data 6 jam terakhir. Untuk data dengan interval per hari maka data yang diambil sebanyak 7 baris atau data seminggu terakhir. Selebihnya, untuk data dengan interval bulanan atau tahunan, maka data yang diambil sebanyak 4 baris data terakhir. Implementasi proses *string matching* ini terdapat pada fungsi `MotifDiscoveryAnalysis()` yang dapat dilihat pada gambar 4.42.

```

MotifDiscoveryAnalysis <- function(colName, dataset,
datasetIntervalValue){
  #Splitting n last data
  n <- DataInterpreterInterval(datasetIntervalValue, type =
"limit")
  index <- length(dataset)+1 - n

  #pattern
  pattern <- dataset[index:length(dataset)]

  #dataest
  dataset <- dataset[1:index]
  print(dataset)
  print(pattern)

  #KMP Process
  result <- list()
  if(!is.null(KMP(dataset, pattern))){
    result$total <- length(KMP(dataset,pattern))
    result$pattern <- KMP(dataset,pattern)
  }else{
    result$total <- 0
    result$pattern <- NA
  }

  return(result)
}

```

Gambar 4.42 *MotifDiscoveryAnalysis function*

Fungsi KMP() yang terdapat pada gambar 4.43 mengacu pada implementasi algoritma KMP yang pada bahasa R yang dibuat oleh (Rahman, 2017) pada penelitiannya. Untuk lebih lengkapnya, implementasi algoritma KMP ini dapat dilihat pada gambar 4.44.

```
#Function to seacrh the pattern in the string
KMP <- function(string, pattern){

  #inisiasi variabel
  prefix <- KMP_Prefix(pattern)
  n_string <- length(string)
  n_pattern <- length(pattern)
  index <- c()
  total <- 0
  i <- 0

  #Perulangan sesuai dengan jumlah string
  for(j in 1:n_string){
    while(i > 0 && pattern[i+1] != string[j]){
      i <- prefix[i]
    }
    if(pattern[i+1] == string[j]){
      i <- i+1
    }
    if(i == n_pattern){
      index <- c(index, j-n_pattern+1)
      total <- total+1
      i <- prefix[i]
    }
  }
  return(index)
}
```

Gambar 4.43 Implementasi algoritma KMP (Rahman, 2017)

Sebelum proses pencocokan pola menggunakan algoritma KMP, diperlukan penentuan *prefix* sebagai acuan dalam proses pencocokan pola nantinya, dimana proses penentuan *prefix* dilakukan pada fungsi KMP_Prefix dapat dilihat pada gambar 4.44.

```
#Function for get the prefix from the pattern
KMP_Prefix <- function(pattern){
  n_pattern <- length(pattern)
  prefix <- c(0)
  a <- 0
  #pattern making
  for(b in 2:n_pattern){
    while(a > 0 && pattern[a+1] != pattern[b]){
      a <- prefix[a]
    }
    if(pattern[a+1] == pattern[b]){
      a <- a+1
    }
    prefix[b] <- a
  }
  #return the result
  return(prefix)
}
```

Gambar 4.44 Penentuan prefix pada algoritma KMP (Rahman, 2017)

f. Korelasi antar parameter

Pendeteksian korelasi parameter ini menggunakan *Pearson Correlation Coefficient*, yang dalam implementasinya menggunakan fungsi cor() pada R. Implementasi proses ini dapat dilihat pada gambar 4.45 ini.

```
CorrelationAnalysis <- function(data) {
  #using pearson correlation coefficient
  return(cor(data))
}
```

Gambar 4.45 *CorrelationAnalysis function*.

Dimana hasil dari fungsi tersebut berupa matriks dengan ukuran n,n dimana n merupakan jumlah parameter *numerical* pada data masukan. Nilai yang dihasilkan merepresentasikan seberapa kuatnya hubungan linear antar paramater

Setelah didapatkan nilai korelasi pada tabel 4.16, proses selanjutnya adalah menentukan parameter yang mana yang memiliki dampak paling besar terhadap parameter lain. Cara ini dilakukan dengan mengubah semua data pada tabel 4.16 ke dalam bentuk mutlak terlebih dahulu menggunakan fungsi abs(), lalu dicari nilai rata-ratanya untuk setiap parameter, proses ini terdapat pada fungsi CorrelationRoutineMessage() yang dapat dilihat pada gambar 4.46.

```
CorrelationRoutineMessage <- function(corMatrix) {
  #Correlation Routine Message Analysis
  #Mean with absolute value
  coreMean <- apply(abs(corMatrix), 2, mean)

  #get the highest mean with their index
  highestMean <- max(coreMean)
  highestIndex <- which.max(coreMean)

  message <- correlationRoutineDocPlan(highestMean,
  names(highestIndex))
  return(message)
}
```

Gambar 4.46 *CorrelationRoutineMessage function*

Setelah dilakukan proses pada gambar 4.46, maka didapatkan hasil seperti pada tabel 4.15 dimana nilai tertinggi ada pada parameter Param1 yang nantinya akan diproses menjadi pesan *routine* untuk korelasi parameter. Setelah didapatkan pesan *routine*, maka dilakukan proses analisis sinyal untuk pesan *significant*, analisis dilakukan dengan

memproses data pada tabel 4.15 lalu mengambil nilai yang lebih besar dari 0.7 untuk nilai positif, dan lebih kecil dari -0.7 untuk nilai negatif. Proses ini dilakukan pada fungsi CorrelationSignificantMsgContent Determination() yang dapat dilihat pada gambar 4.47

```
CorrelationSignificantMsgContentDetermination <-
function(matrix) {
  # ContentDetermination
  # Only showing var more than 0.7

  matrix[!lower.tri(matrix)] <- 0
  matrix <- as.data.frame(matrix)
  matrix[matrix < 0.7 & matrix >= 0 | matrix > -0.7 &
matrix <= 0] <- 0

  return(matrix)
}
```

Gambar 4.47 CorrelationSignificantMsgContentDetermination function

Setelah dilakukan proses pada gambar 4.47 maka didapatkan hasil seperti pada tabel 4.22. Nilai inilah yang akan diambil untuk ditampilkan sebagai *significant message* untuk korelasi parameter. Jika tidak didapatkan hasil pada proses ini, maka *significant message* untuk proses korelasi parameter tidak akan ditampilkan.

4.3.3.3. Implementasi Proses *Data Interpretation*

a. *Generate rule*

Pada tahap ini implementasi dilakukan dengan cara menginterpretasikan setiap parameter sesuai cara penginterpretasian yang terdapat pada *data description*. Jika tipe penginterpretasiannya menggunakan *fuzzy* maka akan digunakan *Fuzzy Membership Function*, jika pentinterpretasiannya adalah *crisp* maka digunakan *Crisp Membership Function* dalam proses penginterpretasiannya. Namun, jika tidak terdapat cara penginterpretasian atau *data description Rule* pada *data description*, maka digunakan *Unspecific Membership Function*.

Fungsi keanggotaan bagi parameter yang sudah didefinisikan cara penginterpretasiannya disimpan dalam file berbentuk *csv* (separator koma) dengan format *file [parameter]Adjective.csv* lalu disimpan pada folder *Corpus/Fuzzy* atau *Corpus/Crisp* sesuai dengan tipe penginterpretasiannya.

Seperti pada gambar 4.48, sehingga sistem dapat bekerja secara *unspecific* untuk data apapun sesuai kebutuhan pengguna.

D2T_Apps > Corpus > Fuzzy				
Name	Date modified	Type	Size	
GeneralAdjective	08/12/2018 09:08 ...	Microsoft Excel C...	1 KB	
RainfallAdjective	08/12/2018 09:08 ...	Microsoft Excel C...	1 KB	
TemperatureAdjective	08/12/2018 09:08 ...	Microsoft Excel C...	1 KB	
TrendFuzzyAdjective	08/12/2018 09:08 ...	Microsoft Excel C...	1 KB	

Gambar 4.48 *Fuzzy Corpus for Data Interpretation*

Proses interpretasi ini dilakukan dengan memanggil fungsi *DataInterpreterAdjective()*, fungsi ini bekerja dengan cara menyamakan nama kolom pada data dengan daftar parameter yang ada, jika terdapat dalam *data description* maka lakukan data interpretasi sesuai fungsi keanggotannya, jika tidak terdapat cara penginterpretasianya maka proses ini menggunakan cara *unspecific*. Fungsi *DataInterpreterAdjective()* ini dapat dilihat pada gambar 4.49.

```
DataInterpreterAdjective <- function(value,
type="Unspecific", statisticalResume=NULL) {
  if(!is.na(mainConfig[mainConfig$ColName == type,]$Rule)) {
    if(mainConfig[mainConfig$ColName == type,]$Rule == "fuzzy") {
      corpus <- read.table(file=paste0(
        "Corpus/Fuzzy/", type, "Adjective.csv"),
        sep=",", header=TRUE)
      if(type == "Rainfall" && value == 0) {
        result <- list(InterpreterResult =
          as.character("no rain"), InterpreterIndex = 0)
      } else{
        result <- MembershipFuzzy(value, corpus);
      }
    } else{
      # Read Crisp corpus Corpus/Crisp/[Parameter]Adjective.csv
      result <- MembershipClassifier(value, corpus);
    }
  } else{
    corpus <- UnspecificFuzzyGenerator(type, statisticalResume)
    result <- MembershipFuzzy(value, corpus);
  }
  return(result)
}
```

Gambar 4.49 *DataInterpreterAdjective function*

Jika tipe penginterpretasian suatu parameter adalah *fuzzy* maka akan digunakan *Fuzzy Membership Function* yang diimplementasikan pada fungsi *MembershipFuzzy()* seperti pada gambar 4.50.

```
MembershipFuzzy <- function(value, corpus){
  if(is.null(corpus)){
    return (list(InterpreterResult = as.character("Constant"),
    InterpreterIndex = 0))
  }
  i <- 1;
  n <- nrow(corpus);
  m <- length(corpus);

  membershipValue <- c()
  for(i in 1:n){
    v1<-corpus[i, "v1"];
    v2<-corpus[i, "v2"];
    v3<-corpus[i, "v3"];
    v4<-corpus[i, "v4"];

    ##/ \ <- 1st area, 2nd area, 3rd area
    #first area
    if((value>=v1)&&(value<=v2)){
      membershipValue[i] <- ( (value-v1) / (v2-v1) );
    #second area (optimum)
    }else if((value>v2)&&(value<=v3)){
      membershipValue[i] <- 1;
    #third area
    }else if((value>v3)&&(value<=v4)){
      membershipValue[i] <- ( (v4-value) / (v4-v3) );
    #fourth, default condition (outside)
    }else{
      membershipValue[i] <- 0;
    }
    if(is.nan(as.numeric(membershipValue[i]))){
      membershipValue[i] <- 9999
    }
  }

  print(membershipValue)
  #check highest membership result
  interpreterResult <- corpus[which.max(membershipValue),
  "Category"]

  #Get Message with highest membership value
  ....
  return(result)
}
```

Gambar 4.50 *Fuzzy Membership Function*

Namun jika penginterpretasian suatu data adalah *crisp* maka akan digunakan *Crisp Membership Function* yang diimplementasikan pada fungsi *MembershipClassifier()* yang dapat dilihat pada gambar 4.51.

```
MembershipClassifier <- function(value, corpus) {
  interpreterResult <- sapply(value, function(v) corpus[v >=
corpus["Lower"] & v < corpus["Upper"], "Category"])
  interpreterIndex <- which(interpreterResult ==
corpus$Category)
  return (list(InterpreterResult =
as.character(interpreterResult), InterpreterIndex =
interpreterIndex))
}
```

Gambar 4.51 *Crisp Membership Function*

b. *Specific Message*

Untuk parameter yang sudah didefinisikan cara penginterpretasiannya, setelah pembacaan corpus pada fungsi DataInterpreterAdjective(), proses selanjutnya adalah memetakan kedalam *rule* sesuai cara penginterpretasiannya. Jika cara penginterpretasiannya menggunakan logika *fuzzy* maka data tersebut akan diinterpretasikan menggunakan fungsi MembershipFuzzy() pada gambar 4.50. Namun jika tipe penginterpretasiannya menggunakan logika *crisp*, maka data tersebut akan diinterpretasikan menggunakan fungsi MembershipClassifier() pada gambar 4.51.

c. *Unpecific Message*

Untuk parameter yang tidak memiliki cara penginterpretasian maka akan digunakan *Unspecific Memership Function* yang diimplementasikan pada fungsi UnspecificFuzzyGenerator() seperti pada gambar 4.52 yang kemudian diterapkan fungsi keanggotaan *Fuzzy Membership Function* pada gambar 4.50 sehingga didapatkan hasil penginterpretasiannya.

```

UnspecificFuzzyGenerator <-function(type, statisticalResume) {
  corpus <-
  read.table(file=paste0("Corpus/Fuzzy/UnspecificAdjective.csv"),
  sep=", ", header=TRUE)
  maxRange <-
  as.character(statisticalResume[statisticalResume$ColName ==
  type, "MaxValue"])
  minRange <-
  as.character(statisticalResume[statisticalResume$ColName ==
  type, "MinValue"])

  listUnspecificPartition <- list()
  if(minRange == maxRange){
    return(NULL)
  }else{
    n = nrow(corpus)
    node = (2*n)+n-1

    maxRange <- as.double(maxRange)
    minRange <- as.double(minRange)
    rangenode = (maxRange-minRange)/node

    i=1
    j=0
    membershipValue <- c()
    for (i in 1:n) {
      if(i == 1){
        v1<-minRange;
        v2<-minRange;
        v3<-minRange+(2*rangenode);
        v4<-minRange+(3*rangenode);

        j <- i+1
        listUnspecificPartition[[i]] <- c(v1,v2,v3,v4)
      }else{
        v1<-minRange+(j)*rangenode;
        v2<-minRange+(j+1)*rangenode;
        v3<-minRange+(j+3)*rangenode;
        v4<-minRange+(j+4)*rangenode;

        listUnspecificPartition[[i]] <- c(v1,v2,v3,v4)
      }
    }
  }

  v1 <-unlist(lapply(listUnspecificPartition, `[[` , 1)))
  v2 <-unlist(lapply(listUnspecificPartition, `[[` , 2)))
  v3 <-unlist(lapply(listUnspecificPartition, `[[` , 3)))
  v4 <-unlist(lapply(listUnspecificPartition, `[[` , 4)))

  result <- data.frame(Category=corpus$Category, v1, v2, v3, v4)

  return(result)
}

```

Gambar 4.52 *UnspecificFuzzyGenerator Function*

d. Interpretasi Event

Implementasi proses penginterpretasian untuk *event* ini dilakukan berbarengan dengan implementasi *content determination* pada bagian *document planning*. Dikarenakan proses penginterpretasian *event* ini cukup sederhana, sehingga implementasi dilakukan sekaligus dengan pembangkitan pesan pada bagian *document planning* yang akan dibahas pada sub-bab implementasi *document planning*.

e. Kualitas Udara

Dalam melakukan implementasi *rule-based* untuk kualitas udara ini hanya akan dilakukan jika semua parameter kualitas udara ada dalam data masukan. Interpretasi kualitas udara ini dilakukan dengan menghitung nilai PSI untuk kualitas udara yang ada pada fungsi AirQualityCalculation() (Putra *et al.*, 2017) seperti pada gambar 4.53 dengan modifikasi pengecekan parameter yang ada, sehingga interpretasi kualitas udara tidak akan dilakukan jika parameter PM25, PM10, CO, SO2, dan O3 tidak terdapat pada dataset.

```
AirQualityCalculation <- function (dataset){
  #menghitung sub-index value dari variabel PM25
  a2<-a1<-b2<-b1<-PM25_PSI_value<-PM10_PSI_value<-O_PSI_value<-
  NO2_PSI_value<-SO2_PSI_value<-O3_PSI_value <- 0;

  #CHECKING PROCES
  ...
  PM25_PSI_value <- ((a2-a1)/(b2-b1))* (PM25-b1)+a1
  PM10_PSI_value <- ((a2-a1)/(b2-b1))* (PM10-b1)+a1
  SO2_PSI_value <- ((a2-a1)/(b2-b1))* (SO2-b1)+a1
  CO_PSI_value <- (((a2-a1)/(b2-b1))* (CO-b1)+a1)
  O3_PSI_value <- (((a2-a1)/(b2-b1))* (O3-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))
  return(PSI_value);
}
```

Gambar 4.53 Implementasi perhitungan PSI untuk kualitas udara (Putra *et al.*, 2017)

Sehingga setelah implementasi pada gambar 4.53 dilakukan, maka didapatkan implementasi *rule-based* untuk kualitas udara seperti pada tabel

4.32 yang disimpan dalam file *AirQualityAdjective.csv* pada folder *Corpus/Crisp* dengan tipe Crisp.

Tabel 4.32 *Air Quality Crisp Membership Value*

Category	Lower	Upper
Good	0	51
Moderate	51	101
Unhealthy	101	201
Very Unhealthy	201	301
Hazardous	301	501

4.3.3.1. Implementasi Proses *Document Planning*

Berdasarkan penjelasan pada sub-bab model proses *Document Planning* yang telah dipaparkan sebelumnya, implementasi pada proses ini berupa *Content Determination* atau pemilihan konten yang akan ditampilkan pada teks keluaran nantinya. Karena banyaknya sinyal dan pesan yang dihasilkan pada proses sebelumnya, maka sinyal-sinyal dan pesan-pesan tersebut perlu diseleksi terlebih dahulu sehingga pesan yang ditampilkan tetap relevan.

Pada pendekripsi sinyal *Repeated Event*, konten yang akan ditampilkan adalah ketika ada nilai perulangan pada suatu parameter. Dapat dilihat pada gambar 4.54, dimana ada proses pengecekan variabel *maxValue*, variabel ini berisikan jumlah perulangan terbesar pada suatu adata, jika variabel *maxValue* ini bernilai 0, maka pesan *Repeated Event* tidak akan dimunculkan, dan diganti dengan pesan default “*There were no repeating values within [limit] [interval] or more, every value changed from time to time.*” yang menandakan bahwa tidak terdapat nilai yang berulang yang melebihi *limit* sebesar 10% dari range data, dengan *interval* data perjam, perhari, perminggu, atau pun yang lainnya. Namun, jika terdapat perulangan pada suatu parameter, akan ditampilkan pesan seperti ini “*There were some repeating value more than 876 hours: IR stayed constant at very low during 1 Jan 2010 00:00 to 4 Mar 2010 15:00, 24 Oct 2010 14:00 to 31 Dec 2010 23:00.*”.

```

resumeRepeatedLimit <- as.integer(nrow(dataset) * 0.1)
#Content Determination
if(maxValue != 0){
  resumeRepeated <- ResumeRepeated2(columnName[[maxIndex]],
dataset, vectorRepeatedInterpretResult,
listRepeated[[maxIndex]]$Start, listRepeated[[maxIndex]]$End)
  resumeRepeated <- paste("There were some repeating value
more than @limit @interval: ", resumeRepeated)
} else{
  resumeRepeated <- "There were no repeating values within
@limit @interval or more, every value changed from time to
time."
}

```

Gambar 4.54 Implementasi *Content Determination* untuk *Repeated Event*

Implementasi *Content Determination* untuk *Extreme Event* dilakukan dengan memilih kenaikan atau penurunan yang mempunyai nilai interpreter *extreme*. Seperti yang digambarkan pada gambar 4.55 dimana jika suatu kenaikan dan penurunan dua-duanya merupakan sinyal *extreme* maka konten *fluctuate extremely* akan ditampilkan. Namun jika hanya kenaikannya saja yang merupakan sinyal *extreme* maka sinyal tersebut akan menghasilkan pesan *increase extremely*, begitu juga jika penurunan suatu parameter merupakan sinyal *extreme* maka sinyal tersebut akan menghasilkan pesan berupa *decrease extremely*. Contoh pesan *extreme* yang akan ditampilkan adalah “*O3 increased significantly from 29-30 Sep 2016 (increased 95 points)*”.

```

#Fluctuated Extremely
if(dfGrowth$IncInterpreter[i] == "extreme" &&
dfGrowth$DecInterpreter[i] == "extreme"){
  event <- "fluctuated"
  adverb <- "significantly"
  phrase <- paste(event, adverb)

  sentence <- paste0(dfGrowth$columnNameNumerical[i], " ",
phrase, " (increased ", dfGrowth$IncValue[i]," points ","and
decreased ", abs(dfGrowth$DecValue[i])," points)")

  flResult[flIndex] <- as.character(sentence)
  flIndex <- flIndex +1
#Increased Extremely
}else if(dfGrowth$IncInterpreter[i] == "extreme"){

  event <- "increased"
  adverb <- "significantly"
  phrase <- paste(event, adverb)

  dateRange <-
LexicalDateRange(as.character(dateTime[dfGrowth$IncstartIndex[i]]),

as.character(dateTime[dfGrowth$Inc endIndex[i]]))

  sentence <- paste0(dfGrowth$columnNameNumerical[i], " ",
phrase, " from ", dateRange, " ","(increased ",
dfGrowth$IncValue[i]," points) " )

  incResult[incIndex] <- as.character(sentence)
  incIndex <- incIndex +1
#Decreased Extremely
}else if(dfGrowth$DecInterpreter[i] == "extreme"){

  event <- "decreased"
  adverb <- "significantly"
  phrase <- paste(event, adverb)

  dateRange <-
LexicalDateRange(as.character(dateTime[dfGrowth$DecstartIndex[i]]),

as.character(dateTime[dfGrowth$Dec endIndex[i]]))

  sentence <- paste0(dfGrowth$columnNameNumerical[i], " ",
phrase, " from ", dateRange, " ","(decreased ",
abs(dfGrowth$DecValue[i])," points) " )

  decResult[decIndex] <- as.character(sentence)
  decIndex <- decIndex +1
}

```

Gambar 4.55 Implementasi *Content Determination* untuk *Extreme Event*

Dalam pendeksi sinyal *String Matching*, implementasi *Content Determination* dilakukan pada fungsi MotifDiscoveryDocPlan() dengan melakukan pengecekan hanya pada parameter dengan tipe *categorical*. Jika

tidak terdapat pola yang sama pada parameter tersebut, maka variabel *listColumn* pada gambar 4.51 akan bernilai *NA*, dan ditampilkan pesan “*For the past 7 days , no same patterns were found for each categorical parameters*”. Namun jika suatu parameter memiliki pola yang sama, maka variable *listColumn* pada gambar 4.56 akan berisikan nilai sesuai indeks dari parameter tersebut. Sehingga akan ditampilkan pesan seperti berikut “*There are a duplicate CBWD data pattern in the last 6 hours (31 Dec 2010 17:00 to 31 Dec 2010 23:00) with data patterns from 3 May 2010 20:00 to 4 May 2010 02:00, and 1 Oct 2010 10:00 to 1 Oct 2010 16:00.*”

```
#Content Determination
MotifDiscoveryDocPlan <- function(listMD){
  #For categorical parameter only
  listCategorical <- mainConfig[which(mainConfig$type ==
  ("character") | mainConfig$type == ("categorical") |
  mainConfig$type == ("factor"))]

  if(nrow(listCategorical) != 0) {
    listColumn <- rownames(listCategorical)

    i<-1
    for(i in i:length(listColumn)){
      #If there's no pattern in data, value = NA
      if(is.na(listMD[[as.numeric(listColumn[i])]]$pattern[1])){
        listColumn[i] <- NA
      }
    }
  }else{
    listColumn <- NULL
  }

  return(listColumn)
}
```

Gambar 4.56 Implementasi *Content Determination* untuk pendeksiian *String Matching*

Implementasi *Content Determination* pada korelasi parameter dilakukan pada fungsi CorrelationSignificantMsgContentDetermination() yang dapat dilihat pada gambar 4.57. Implementasi *Content Determination* untuk korelasi parameter terbilang sederhana, dimana nilai yang akan diproses sebagai pesan keluaran adalah nilai korelasi yang lebih besar dari 0.7 untuk korelasi positif, dan nilai korelasi yang lebih kecil dari -0.7 untuk korelasi negatif. Sehingga didapatkan pesan keluaran seperti ini “*An*

increase in TEMP resulted a growth in DEWP. As a result of the decay in PRES, it can be seen that DEWP, and TEMP is increasing”.

```
CorrelationSignificantMsgContentDetermination <-
function(matrix) {
  # ContentDetermination
  matrix[!lower.tri(matrix)] <- 0 # Removing top triangle
  matrix <- as.data.frame(matrix)
  # Only showing var more than 0.7
  matrix[matrix < 0.7 & matrix >= 0 | matrix > -0.7 & matrix <=
0] <- 0

  return(matrix)
}
```

Gambar 4.57 Implementasi *Content Determination* untuk korelasi antar parameter

Implementasi *Content Determination* pada *Current Text* dilakukan pada fungsi yang digambarkan pada gambar 4.58 dimana pesan hanya akan ditampilkan jika suatu parameter mendapatkan nilai tertinggi atau nilai terendah pada data terakhir. Sehingga jika data terakhir merupakan nilai tertinggi atau nilai terendah, maka akan ditampilkan pesan seperti berikut “*LWS reached their highest value on this hour*”.

```
CurrentHighest <- function(data, statSummary, interval){
  analysisResult <- c()
  analysisIndex <- c()

  i <- 1
  result <- ""
  for(i in i:length(data)){
    #get highgest/lowest value from statsum
    ....
    sentence <- ""
    analysisResult[i] <- NA
    analysisIndex[i] <- NA
    #if last data is highest data
    if(data[i] >= maxVal){
      analysisResult[i] <- "+"
      analysisIndex[i] <- i
    #if last data is lowest data
    }else if(data[i] <= minVal){
      analysisResult[i] <- "-"
      analysisIndex[i] <- i
    }
    result <- paste0(result, sentence)
  }
  result <- CurrentHighestAggregation(data, analysisResult,
analysisIndex, interval)
  return(result)
}
```

Gambar 4.58 Implementasi *Content Determination* untuk *Current Text*

Implementasi *Document Structuring* dilakukan dalam hal analisa struktur dokumen pada pembahasan sebelumnya, yang selanjutnya struktur dokumen tersebut akan direalisasikan pada sub-bab *Structure Realisation*.

4.3.3.4. Implementasi Proses *Microplanning and Realisation*

Pada tahap ini implementasi ini dilakukan pada setiap perbandingan data, contoh implementasi pada tahap *Lexicalisation* adalah ketika menentukan frasa untuk mendefinisikan rentang tanggal. Misalnya, format rentang “2-3 Dec 2018” digunakan untuk mendefinisikan rentang tanggal yang memiliki nilai bulan dan tahun yang sama, lalu format “2 Oct -3 Dec 2018” digunakan untuk mendefinisikan rentang tanggal dengan tahun yang sama, sedangkan format “20 Jan 2017 - 08 Jan 2018” digunakan untuk mendefinisikan rentang tanggal yang tahunnya berbeda. Proses penentuan frasa ini dilakukan pada fungsi LexicalDateRange() yang dapat dilihat pada gambar 4.59.

```
LexicalDateRange <- function(dateStart, dateEnd) {
  #FORMAT INPUT: mm/dd/yyyy -> "07/01/2018"

  #Exception for hourly data
  if(datasetIntervalValue == 1) {
    #Hourly data default: 1 June 2018 00:00 to 1 June 2018 12:00
  }

  timeRepeated <- ""
  if(startYear == endYear) {
    if(startMonth == endMonth) {
      #Example: 1 - 8 June 2018
      timeRepeated <- paste0(startDate, "-", endDate, " ",
      month.abb[startMonth], " ", startYear)
      #if month different
    } else{
      #Example: 1 Mar - 8 June 2018
      timeRepeated <- paste(startDate, month.abb[startMonth], "-",
      endDate, month.abb[endMonth], startYear)
    }
  } else{
    #Example: 20 Jan 2017 - 08 Jan 2018
    timeRepeated <- paste(startDate, month.abb[startMonth],
    startYear, "-", endDate, month.abb[endMonth], endYear)
  }
  return(timeRepeated)
}
```

Gambar 4.59 Implementasi *Lexicalisation* untuk format range tanggal

Implementasi *Aggregation* pada umumnya diterapkan pada proses penggabungan sinyal-sinyal yang memiliki jenis kontras yang sama. Misalnya, proses penggabungan sinyal untuk proses perbandingan parameter pada *Resume Text*, dimana sinyal-sinyal perbandingan tersebut akan dikelompokan menjadi tiga kelompok menggunakan *simple conjunction* seperti *but* dan *and*. Contoh implementasi proses *Aggregation* ini dapat dilihat pada gambar 4.60.

```
ComparsionAggregation <- function(group) {
  if(is.null(group)){
    return(group)
  }else if(length(group) == 1){
    return(group)
  }else{
    i <- 1
    message <- ""
    for(i in i:length(group)){
      #aggregation process for last data with same contrast
      if(i == length(group)){
        message <- paste0(message, "and ", group[i])
      }else{
        message <- paste0(message, group[i], ", ")
      }
    }
    return(message)
  }
}
```

Gambar 4.60 Implementasi *Aggregation* untuk mengelompokan sinyal pada perbandingan data

Selain itu, implementasi proses *Aggregation* ini dilakukan pada proses-proses yang membutuhkan proses *grouping* didalamnya, seperti proses *grouping* pada pesan *trend*, lalu proses *grouping* untuk perbandingan parameter, dan juga proses *grouping* pada *Current text* dan *Predict Text*. Ciri khas dari proses *Aggregation* dengan *Simple Conjunction* ini adalah parameter-parameter yang ditampilkan sekaligus. Contohnya, pesan pada *Current Text* berikut ini “*PM2.5, DEWP, IS, and IR in very low condition. TEMP in low condition. IS, and IR reached their lowest value on this hour.*”. Dimana pada pesan tersebut, parameter-parameter yang memiliki hasil interpretasi *very low* dan parameter-parameter yang merupakan *lowest value*

dikelompokan dan ditampilkan sekaligus menggunakan *simple conjunction* berupa *and*.

Sedangkan contoh implementasi *Referring Expression Generation* terdapat pada gambar 4.61 dimana dilakukan pemilihan *intro* secara *random* berdasarkan *corpus* yang ada.

```
ReadIntro <- function(source="Data", type="Unspecific") {
  type
  ...
  as.matrix(read.table(file=paste0("Corpus/", type, "Intro.csv"),
  header=FALSE, sep=';'))
  # print(corpus)
  n <- length(corpus)
  random_value <- as.integer(runif(1,1,n+0.5))
  ...
  # return ("Woops no data intro!");
}

ReadResumeIntro <- function(dataset2, ColName, source="dataset2") {

  corpus           <-      as.matrix(read.table(file=paste0("Corpus/",
  "ResumeIntro.csv"), header=FALSE, sep=';'))

  #Randoming corpus
  n <- length(corpus)
  random_value <- as.integer(runif(1,1,n))
  result <- corpus[random_value]
  ....
  return (result)
}
```

Gambar 4.61 Implementasi *Referring Expression Generation* untuk menentukan *intro* secara *random*

Selain itu, penentuan interval waktu data dalam bahasa manusia terdapat pada gambar 4.62 dimana pertama-tama diambil sampel data yang kemudian ditentukan interval datanya dalam jam, yang kemudian direpresentasikan kedalam bentuk manusia menggunakan fungsi DataInterpreterInterval() yang terdapat pada gambar 4.62.

```

DataInterpreterInterval <- function (interval, type =
"default") {
  #Interval data
  if(type == "interval"){
    if(interval == 1){
      result <- "hourly"
    }else if(interval == 24){
      result <- "daily"
    }else if(interval == 168){
      result <- "weekly"
    }else if(interval == 672 || interval == 696 || interval ==
720 || interval == 744){
      result <- "monthly"
    }else if(interval == 8760 || interval == 8736){
      result <- "yearly"
    }else{
      result <- ""
    }
  }
  #Intro with interval data
}else if(type == "intro"){
  if(interval == 1){
    result <- "This hour"
  }else if(interval == 24){
    result <- "Today"
  }else if(interval == 168){
    result <- "This week"
  }else if(interval == 720 || interval == 744){
    result <- "This month"
  }else if(interval == 8760 || interval == 8736){
    result <- "This year"
  }else{
    result <- ""
  }
}
.....
return(result)
}

```

Gambar 4.62 Implementasi *Referring Expression Generation*
untuk *Time Description*

Sedangkan implementasi *Structure Realisation* dilakukan seperti pada gambar 4.63.

```

#Resume Text
resumeResult <- paste(resumeIntro, resumeTrend,
resumeComparsion, resumeRepeated, resumeExtremeEvent,
resumeMotifDiscovery, resumeCorrelation)
#Current Text
currentResult <- paste(currentIntro, currentDesc, currentHighest)
#Predict Text
predictResult <- paste(predictIntro, predictContent)

```

Gambar 4.63 Implementasi *Structure Realisation*

4.3.4 Testing Sistem D2T UNG

Adapun rencana pengujian sistem UNG ini dilakukan dengan menggunakan teknik pengujian *blackbox*. Dimana fitur-fitur sistem akan diuji untuk membuktikan bahwa pembangunan sistem UNG ini menjawab kebutuhan-kebutuhan dari hasil analisis sebelumnya. Untuk rencana pengujian sistem UNG ini ditampilkan pada tabel 4.33.

Tabel 4.33 Tabel Rencana pengujian *blackbox* sistem UNG

No.	Test Case	Kode Uji
1.	Pengujian proses <i>signal analysis: Event Detection (Comparison, Extreme, Repeated value, String Matching, dan, Korelasi parameter)</i>	1-UJI-01
2.	Pengujian proses data interpretation (<i>Crisp membership function, Fuzzy membership function, dan Unspecifc Fuzzy Generator</i>)	1-UJI-02
3.	Pengujian proses <i>Content Determintation</i>	1-UJI-03
4.	Pengujian proses <i>Lexicalisation</i>	1-UJI-04
5.	Pengujian proses <i>Aggregation</i>	1-UJI-05
6.	Pengujian proses <i>Referring Expression Generation</i>	1-UJI-06
7.	Pengujian proses <i>Structure Realization</i>	1-UJI-07
8.	Pengujian untuk dataset <i>numerical</i> dan <i>categorical</i>	1-UJI-08
9.	Pengujian untuk dataset tanpa <i>header</i>	1-UJI-09
10.	Pengujian untuk dataset dengan <i>header</i>	1-UJI-10
11.	Pengujian untuk dataset dengan <i>Specific corpus</i>	1-UJI-11
12.	Pengujian untuk dataset dengan <i>Unspecific corpus</i>	1-UJI-12

Setelah menentukan rencana proses pengujian *blackbox* proses selanjutnya adalah proses pengujian. Adapun hasil pengujian sistem UNG dapat dilihat pada tabel 4.34

Tabel 4.34 hasil pengujian *blackbox* sistem UNG

Kode Uji	Hasil yang diharapkan	Hasil keluaran
1-UJI-01	<i>Event</i> yang terdeteksi sesuai dengan visualisasi data yang dibuat.	OK
1-UJI-02	Interpretasi data sesuai dengan aturan yang diberikan.	OK
1-UJI-03	<i>Event</i> yang tidak memenuhi syarat tidak muncul didalam teks.	OK
1-UJI-04	Representasi kalimat sesuai dengan kondisi data sebenarnya, sesuai dengan visualisasi yang dibuat.	OK
1-UJI-05	Penerapan <i>Simple Conjunction Referring to Contrast Value</i> sesuai dengan kondisi real.	OK
1-UJI-06	Teks keluaran bervariasi.	OK
1-UJI-07	Struktur teks sesuai dengan <i>Target Text</i> .	OK
1-UJI-08	Sistem dapat menghasilkan <i>text</i> dengan masukan data <i>numerical</i> maupun <i>categorical</i>	OK
1-UJI-09	Sistem tetap berkerja dan menghasilkan keluaran meskipun data masukan tidak memiliki header	OK
1-UJI-10	Sistem tetap berkerja dan menghasilkan keluaran dengan data masukan yang memiliki header	OK
1-UJI-11	Sistem menghasilkan pesan <i>specific</i> sesuai dengan <i>corpus</i> yang ditentukan ole pengguna	OK
1-UJI-12	Sistem menghasilkan pesan <i>unspecific</i>	OK

4.3.5 Panduan Penggunaan Aplikasi UNG

Berikut tahapan untuk menggunakan aplikasi D2T dalam penelitian ini:

1. *Extract* semua file D2T.rar kedalam sebuah folder
2. Install *package* yang diperlukan
3. Buka file apps.R
4. Lalu tekan tombol RunApp pada RStudio
5. Jendela browser akan terbuka, lalu masukkan judul berita dan tekan tombol enter, tunggu sebentar hingga pesan keluaran muncul.
Direkomendasikan untuk menggunakan browser eksternal seperti *Firefox* maupu *Goole Chrome* agar aplikasi berjalan optimal dan responsif.

4.4. Rancangan Eksperimen

Dalam melakukan proses evaluasi pada pembangkitan kalimat bahasa alami oleh suatu sistem, Belz (2007) memaparkan bahwa diperlukan serangkaian proses seperti evaluasi menggunakan *NIST* dan *BLEU*, evaluasi oleh *expert* pada bidang data terkait, evaluasi komputasi, dan evaluasi waktu pembangunan sistem. Selain itu Ramos-Soto et al., (2015) dalam penelitiannya, dilakukan cara pengevaluasian yang berbeda. Dimana pada penelitiannya, Ramos-Soto melakukan 20 kali pembangkitan berita dengan *input* yang berbeda-beda yang kemudian setiap hasil pembangkitan tersebut dievaluasi oleh *expert* lalu dilakukan penilaian. Penilaian dilakukan berdasarkan pertanyaan terkait *relevance* dan *truthfullnes*, yang kemudian setelah semua hasil dinilai maka langkah selanjutnya adalah melakukan perhitungan rata-rata sehingga dapat dilakukan penarikan kesimpulan.

Dengan demikian, penulis memutuskan untuk membuat skenario eksperimen dengan mempertimbangkan dua aspek, yakni:

1. *Readability*, yaitu mengevaluasi kualitas teks dengan menggunakan penilaian *Flesch Reading Ease* seperti pada penelitian (Putra et al., 2017). Nilai tersebut merepresentasikan tingkat kemudahan dalam membacaan suatu teks. Dalam penilaian ini, penulis menggunakan situs *Readability Analyzer* (<https://datayze.com/readability-analyzer.php>) dan *Readability Formula* (<http://www.readabilityformulas.com/flesch-reading-ease-readability-formula.php>). Semua hasil eksperimen akan dinilai, kemudian dihitung rata-ratanya, setelah itu hasil perhitungan rata-rata akan diterjemahkan dengan memasukan nilai tersebut pada tabel 4.35.
2. *Computation Time*, mengevaluasi waktu komputasi sistem dengan menggunakan perintah *System.time()* sebagai contoh *system.time(source("D2T_Main.R"))*. Perhitungan dilakukan pada semua hasil eksperimen, yang kemudian dihitung rata-ratanya. Selain itu, fungsi ini dapat digunakan sebagai indikator apakah terdapat error pada *source code* atau tidak, karena jika terjadi sedikit saja kesalahan, maka fungsi tersebut tidak akan mengeluarkan hasil.

Tabel 4.35 Flesch Reading Ease

Score	School Level	Notes
100.0–90.0	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>

Selain aspek-aspek tersebut, peneliti juga melakukan analisis terhadap kemampuan sistem dalam memproses data masukan, dan mengecek kebenaran informasi yang dibangkitkan dengan cara sebagai berikut:

1. Melakukan pengecekan *Unspecific handling*, dimana sistem akan dilihat kemampuannya dalam menangani data yang beragam, apakah sistem tetap berjalan dan memproses data masukan meskipun terdapat kondisi seperti data masukan yang tidak memiliki *header*, lalu terdapat parameter dengan tipe *categorical*, atau data yang memiliki kustomisasi dalam penginterpretasiannya, dan kondisi-kondisi lainnya.
2. Membandingkan *Representative Text* dengan kondisi real, baik itu *trend*, *Repeated Event*, korelasi parameter, atau *event* lainnya.

Terdapat empat jenis data yang akan digunakan, yaitu data kurs nilai mata uang asing terhadap rupiah (KEMENDAG RI), data klimatologi dan kualitas udara (MeteoGalicia), serta data partikel udara Kota Beijing (Badan Meteorologi Kota Beijing, China). Untuk memproses data dilakukan dengan mengubah variabel *fileName* pada file D2T_Main.R sesuai dengan nama dataset yang akan digunakan, dataset disimpan pada folder *DatasetExperiment*. Untuk setiap data masukan, akan dilakukan simulasi sebanyak tiga kali, dimana simulasi pertama data akan diproses tanpa menggunakan *header*, lalu untuk simulasi kedua data akan menggunakan *header* aslinya, dan simulasi ke-tiga data akan menggunakan *header* aslinya dan

dengan kustomisasi *corpus*, sehingga terdapat 12 *testcase* yang akan diberikan pada sistem seperti pada tabel 4.36.

Tabel 4.36 Rancangan Eksperimen Sistem UNG

Kode Dataset	Dataset	Detail dan Sumber
CE_NH	1 Jan 2002 – 1 Okt 2018 (bulanan)	Situs web Bank Indonesia (https://www.bi.go.id/) kurs rupiah terhadap valuta asing dengan rentang bulanan, selama 1 Januari 2002 hingga 1 Oktober 2018 tanpa menggunakan header
CE_WH	1 Jan 2002 – 1 Okt 2018 (bulanan)	Situs web Bank Indonesia (https://www.bi.go.id/) kurs rupiah terhadap valuta asing dengan rentang bulanan, selama 1 Januari 2002 hingga 1 Oktober 2018 dengan menggunakan header
CE_WHM	1 Jan 2002 – 1 Okt 2018 (bulanan)	Situs web Bank Indonesia (https://www.bi.go.id/) kurs rupiah terhadap valuta asing dengan rentang bulanan, selama 1 Januari 2002 hingga 1 Oktober 2018 dengan kustomisasi <i>Corpus</i> , dengan menggunakan header
CL_NH	1 Januari 2016 - 31 Desember 2017 (harian)	Data klimatologi pada situs web www.MeteoGalicia.gal , selama satu tahun pada periode 2016-2017 tanpa menggunakan header
CL_WH	1 Januari 2016 - 31 Desember 2017 (harian)	Data klimatologi pada situs web www.MeteoGalicia.gal , selama satu tahun pada periode 2016-2017 dengan menggunakan header
CL_WHM	1 Januari 2016 - 31 Desember 2017 (harian)	Data klimatologi pada situs web www.MeteoGalicia.gal , selama satu tahun pada periode 2016-2017 dengan kustomisasi <i>Corpus</i> , dengan menggunakan header
AQ_NH	1 Januari 2016 - 31 Desember 2017 (harian)	Data kualitas udara pada situs web www.MeteoGalicia.gal , selama satu tahun pada periode 2016-2017 tanpa menggunakan header
AQ_WH	1 Januari 2016 - 31 Desember 2017 (harian)	Data kualitas udara pada situs web www.MeteoGalicia.gal , selama satu tahun pada periode 2016-2017 dengan menggunakan header
AQ_WHM	1 Januari 2016 - 31 Desember 2017 (harian)	Data kualitas udara pada situs web www.MeteoGalicia.gal , selama satu tahun pada periode 2016-2017 dengan kustomisasi <i>Corpus</i> , dengan menggunakan header
BPM_NH	1 Januari 2016 - 31 Desember 2017 (per jam)	Data partikel udara pada situs web http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data , selama dua tahun pada periode 2010-2011 tanpa menggunakan header
BPM_WH	1 Januari 2016 - 31 Desember 2017 (per jam)	Data partikel udara pada situs web http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data , selama dua tahun pada periode 2010-2011 dengan menggunakan header
BPM_WHM	1 Januari 2016 - 31 Desember 2017 (per jam)	Data partikel udara pada situs web http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data , selama dua tahun pada periode 2010-2011 dengan kustomisasi <i>corpus</i> , dengan menggunakan header

4.5. Hasil dan Pembahasan Hasil Eksperimen

Pada sub-bab ini, akan dijelaskan mengenai hasil eksperimen yang telah dilakukan, serta menganalisis hasil eksperimen tersebut, dimulai dengan data nilai tukar mata uang asing dengan interval bulanan, data klimatologi dengan interval harian, data kualitas udara dengan interval harian, dan data partikel udara dengan interval per jam. Hasil dan pembahasan hasil eksperimen untuk setiap *testcase* akan dipaparkan pada sub-bab selanjutnya.

4.7.1. Hasil dan Pembahasan Hasil Eksperimen Data Nilai Tukar Mata Uang Asing

Eksperimen menggunakan data nilai tukar mata uang asing atau kurs ini didapatkan dari situs KEMENDAG pada periode 1 januari 2002 hingga 1 Oktober 2018 dengan interval bulanan. Data ini dipilih dikarenakan mempunyai keunikan dimana *trend* dari setiap parameter adalah menaik, dan juga semua parameter pada data ini bertipe *numerical*.

4.7.1.1 Hasil Eksperimen Data Tukar Mata Uang Asing Tanpa *Header*

Eksperimen pertama dilakukan dengan menggunakan data kurs mata uang asing dengan *header* yang dihilangkan terlebih dahulu atau *currency exchange no header* (CE_NH). Hal ini bertujuan untuk melihat apakah sistem akan tetap berjalan meskipun data yang dimasukkan tidak dikenali oleh sistem sebelumnya dalam kasus ini data masukan tidak memiliki *header*. Cuplikan dari data tersebut dapat dilihat pada tabel 4.37 sehingga didapatkan hasil seperti pada gambar 4.64.

Tabel 4.37 Cuplikan data nilai tukar tanpa menggunakan *header* (CE_NH)

01/01/2001 00:00	9.45000	8.13149	13.81498	5.74364	...	6.28451
02/01/2001 00:00	9.83500	8.45297	14.17963	5.85644	...	6.43274
03/01/2001 00:00	10.40000	8.37000	14.85227	6.01400	...	6.60907
...
08/01/2018 00:00	14.711	13.25614	19.14198	15.18951	...	11.30
09/01/2018 00:00	14.929	13.14462	19.52714	15.28281	...	11.46622
10/01/2018 00:00	15.227	13.45914	19.35428	15.152	...	11.6103

Currency Exchange NEWS

According to the monthly currency exchange data from 01/01/2001 00:00 to 10/01/2018 00:00, with parameters: V2, V3, V4, V5, V6, V7, V8, V9, and V10, it indicated that trend of all variable is increased. All parameters are higher than last year's data. There were no repeating values within 21 months or more, every value changed from time to time. V3 increased sharply from 1 Aug - 1 Nov 2008 (increased 4.353 points), and V10 increased significantly from 1 Dec 2002 - 1 May 2003 (increased 4.051 points). V8 fluctuated dramatically (increased 1.3619 points and decreased 1.33185 points). V6 appears to have a highest impact to all variable with very strong relationship in average. An increase in V2 resulted an increase in V3, V5, V6, V7, and V8. As a result of the growth in V3, it can be seen that V5, V6, V7, V9, and V10 is increasing. A rise in V5 causes an attendant increase in V6, V7, V8, V9, and V10. An increase in V6 resulted an increase in V7, V8, V9, and V10. An increase in V7 resulted an increase in V9, and V10. An increase in V9 resulted an increase in V10.

This month data indicate that: V2, V3, V5, V6, V7, V9, and V10 in very high condition. V4 in high condition. V8 in medium condition. V2, V3, V6, and V10 reached their highest value on this month.

From the prediction result, it's expected that V2, V3, V5, V6, V7, V9, and V10 will steady at very high. V4 will stay stable at high. V8 will still stable at medium.

Gambar 4.64 Hasil eksperimen pertama menggunakan data nilai tukar tanpa menggunakan *header* (CE_NH)

Hasil keluaran pada gambar 4.64 dapat dilihat bahwa meskipun data tersebut tidak memiliki *header* sehingga tidak dikenali oleh sistem, sistem akan tetap memproses data tersebut lalu memberi nama setiap parameter dengan *header* V2, V3, V4 dan lainnya. Dimana dalam informasi pada gambar 4.64 dijelaskan bahwa *trend* dari setiap parameter adalah menaik, yang mengakibatkan nilai pada data terakhir lebih besar dari data pada tahun sebelumnya. Terdapat beberapa *extreme event* pada periode tersebut dimana parameter ke-tiga atau V3 mengalami kenaikan yang cukup ekstrem dari tanggal 1 Agustus hingga 1 November 2008 dengan jumlah kenaikan sebanyak 4.353 poin, penurunan yang cukup ekstrem terjadi juga pada parameter ke-sepuluh atau V10 pada periode 1 Desember 2002 sampai dengan 1 Mei 2003 dengan jumlah penurunan sebanyak 4.051 poin, lalu terjadi fluktuasi yang ekstrem pada parameter ke-delapan atau V8 dengan jumlah kenaikan sebanyak 1.3619 poin dan penurunan sebanyak 1.33815 poin. Disebutkan juga, bahwa parameter V6 memiliki pengaruh yang sangat kuat terhadap perubahan pada parameter lainnya. Selain itu kenaikan parameter V2, V3, V5, V6, V7, dan V9 memiliki peran dalam kenaikan yang terjadi pada beberapa parameter lainnya. Pada paragraf ke-dua terlihat deskripsi untuk setiap parameter, dan terdapat *event* yang menunjukan bahwa parameter V2, V3, V6, dan V10 mencapai nilai tertinggi pada bulan tersebut. Pada paragraf ke-tiga disimpulkan bahwa semua parameter akan tetap konstan berada pada kondisi seperti pada paragraf ke-dua.

4.7.1.2 Hasil Eksperimen Data Tukar Mata Uang Asing dengan *Header*

Currency Exchange NEWS

Regarding to the monthly currency exchange data between 01/01/2001 00:00 to 10/01/2018 00:00, with parameters: USD, JPY, GBP, CHF, SGD, MYR, HKD, AUD, and CAD, it showed that trend of all variable is increased. All parameters are higher than last year's data. There were no repeating values within 21 months or more, every value changed from time to time. JPY increased rapidly from 1 Aug - 1 Nov 2008 (increased 4.353 points), and CAD increased sharply from 1 Dec 2002 - 1 May 2003 (increased 4.051 points). HKD fluctuated rapidly (increased 1.3619 points and decreased 1.33185 points). SGD appears to have a highest impact to all variable with very strong relationship in average. As a result of the growth in USD, it can be seen that JPY, CHF, SGD, MYR, and HKD is increasing. As a result of the growth in JPY, it can be seen that CHF, SGD, MYR, AUD, and CAD is increasing. As a result of the growth in CHF, it can be seen that SGD, MYR, HKD, AUD, and CAD is increasing. An increase in SGD resulted an increase in MYR, HKD, AUD, and CAD. A rise in MYR causes an attendant increase in AUD, and CAD. A rise in AUD causes an attendant increase in CAD.

This month data describe that: USD, JPY, CHF, SGD, MYR, AUD, and CAD in very high condition. GBP in high condition. HKD in medium condition. USD, JPY, SGD, and CAD reached their highest value on this month.

Based on prediction result, it's projected that USD, JPY, CHF, SGD, MYR, AUD, and CAD will stay constant at very high. GBP will stay stable at high. HKD will keep stable at medium.

Gambar 4.65 Eksperimen kedua menggunakan data nilai tukar dengan menggunakan *header* (CE_WH)

Untuk eksperimen ke-dua pada gambar 4.65 digunakan data nilai tukar dimana data tersebut diproses dengan menggunakan *header* aslinya namun untuk penginterpretasiannya masih digunakan cara *unspecific*. Dimana tidak ada perbedaan seperti pada eksperimen kedua, perbedaan yang mencolok hanya terletak pada penyebutan parameter, dimana pada eksperimen pertama penyebutan parameter menggunakan nama *header* secara default, sedangkan pada eksperimen kedua penyebutan parameter menggunakan nama *header* aslinya seperti pada tabel 4.1 dimana V2 adalah parameter kedua yakni USD, V3 adalah JPY, V4 adalah GBP, V5 adalah CHF, V6 adalah SGD, V7 adalah MYR, V8 adalah HKD, V9 adalah AUD, dan V6 adalah CHF. Selain itu perbedaan terletak pada frasa hasil dari proses *Reffering Expression Generation* dalam mengungkapkan kondisi kenaikan dan penurunan seperti *increase*, *growth*, *rise*, *decline*, *decrease*, *decay* dan frasa untuk mengungkapkan *extreme event* seperti *extremely*, *dramatically*, *significantly* dan *sharply*.

4.7.1.3 Hasil Eksperimen Data Tukar Mata Uang Asing dengan Kustomisasi *Corpus*

Untuk eksperimen ke-tiga digunakan data nilai tukar yang sama seperti pada eksperimen kedua, namun pada eksperimen ke-tiga ini digunakan kustomisasi *corpus*. Dimana kustomisasi ini didefinisikan pada *data description* yang dapat dilihat pada tabel 4.38.

Tabel 4.38 Kustomisasi *data description* pada eksperimen ke-tiga

ColName	Type	Rule	Alternate
USD	numeric	NA	U.S. Dollar
JPY	numeric	NA	Japan Yen
GBP	numeric	NA	Great British Pounds
CHF	numeric	NA	Confoederatio Helvetica Franc
SGD	numeric	NA	Singapore Dollar
MYR	numeric	NA	Malaysian Ringgit
HKD	numeric	NA	Hong Kong dollar
AUD	numeric	NA	Australian Dollar
CAD	numeric	NA	Canadian Dollar

Seusai dengan *data description* tersebut, setiap parameter akan diganti namanya menggunakan *Custom Alternate* sehingga didapatkan hasil seperti pada gambar 4.66 dimana pada gambar tersebut terlihat, nama parameter pada teks keluaran berubah mengikuti *data description alternate* pada *data description*. Dimana parameter seperti USD ditampilkan sebagai U.S Dollar, parameter JPY ditampilkan sebagai Japan Yen, begitu juga dengan parameter-parameter lainnya. Untuk pendekripsi *event* pada eksperimen ke-tiga ini tidak ada perbedaan dengan eksperimen sebelumnya, dikarenakan pendekripsi *event* tidak dipengaruhi oleh perubahan *header*.

Currency Exchange NEWS

From the monthly currency exchange data between 01/01/2001 00:00 to 10/01/2018 00:00, with parameters: U.S. Dollar, Japan Yen, Great British Pounds, Confoederatio Helvetica Franc, Singapore Dollar, Malaysian Ringgit, Hong Kong dollar, Australian Dollar, and Canadian Dollar, it showed that trend of all variable is increased. All parameters are higher than last year's data. There were no repeating values within 21 months or more, every value changed from time to time. Japan Yen increased significantly from 1 Aug - 1 Nov 2008 (increased 4.353 points), and Canadian Dollar increased dramatically from 1 Dec 2002 - 1 May 2003 (increased 4.051 points). Hong Kong dollar fluctuated dramatically (increased 1.3619 points and decreased 1.33185 points). Singapore Dollar appears to have a highest impact to all variable with very strong relationship in average. An increase in U.S. Dollar resulted a growth in Japan Yen, Confoederatio Helvetica Franc, Singapore Dollar, Malaysian Ringgit, and Hong Kong dollar. An increase in Japan Yen resulted a growth in Confoederatio Helvetica Franc, Singapore Dollar, Malaysian Ringgit, Australian Dollar, and Canadian Dollar. A rise in Confoederatio Helvetica Franc causes an attendant increase in Singapore Dollar, Malaysian Ringgit, Hong Kong dollar, Australian Dollar, and Canadian Dollar. An increase in Singapore Dollar resulted a growth in Malaysian Ringgit, Hong Kong dollar, Australian Dollar, and Canadian Dollar. A rise in Malaysian Ringgit causes an attendant increase in Australian Dollar, and Canadian Dollar. An increase in Australian Dollar resulted a growth in Canadian Dollar.

This month data describe that: U.S. Dollar, Japan Yen, Confoederatio Helvetica Franc, Singapore Dollar, Malaysian Ringgit, Australian Dollar, and Canadian Dollar in very high condition. Great British Pounds in high condition. Hong Kong dollar in medium condition. U.S. Dollar, Japan Yen, Singapore Dollar, and Canadian Dollar reached their highest value on this month.

According to the prediction result, it's estimated that U.S. Dollar, Japan Yen, Confoederatio Helvetica Franc, Singapore Dollar, Malaysian Ringgit, Australian Dollar, and Canadian Dollar will steady at very high. Great British Pounds will stay stable at high. Hong Kong dollar will stay stable at medium.

Gambar 4.66 Eksperimen ke-tiga menggunakan data nilai tukar dengan kustomisasi *corpus* (CE_WHM)

4.7.1.4 Analisis Hasil Eksperimen Data Tukar Mata Uang Asing

Analisis hasil eksperimen dengan menggunakan data nilai tukar mata uang asing atau kurs pada periode bulan Januari 2002 hingga Oktober 2018 adalah sebagai berikut:

1. Analisis aspek *Readability*

Evaluasi dilakukan dengan menggunakan metode *Flesch Reading Ease* menggunakan *Readability Analyzer* pada situs www.datayze.com dan aplikasi *Automatic Readability Checker* pada situs www.readabilityformulas.com. Hasil pengujian aspek *Readability* ini dapat dilihat pada tabel 4.39. Dimana rata-rata yang didapatkan dari proses analisis ini adalah 81.22 yang kemudian berdasarkan *Flesch Reading Ease Score* pada tabel 4.35 didapatkan bahwa informasi dari teks keluaran sangat mudah dipahami bahkan oleh kalangan anak sekolah dasar sekalipun. Untuk hasil lebih lengkap terdapat didalam lampiran.

Tabel 4.39 Hasil evaluasi *Readability* dengan data kurs

No	Kode Dataset	<i>Flesch Reading Ease Score</i> (Datayze)	<i>Flesch Reading Ease Score</i> (Readability Formula)
1	CE_NH	82.01	81
2	CE_WH	82.02	79.8
3	CE_WHM	81.71	80.8
Rata-rata		81.92	80.53
Rata-rata Keseluruhan		81.22	

2. Analisis aspek *Computation Time*

Hasil perhitungan *Computation Time* pada data nilai tukar mata uang asing terdapat pada tabel 4.40 terlihat bahwa rata-rata *running time* sistem ini adalah 3.97 detik. Untuk hasil lebih lengkapnya terdapat pada lampiran.

Tabel 4.40 Hasil evaluasi *Computation Time* dengan data kurs

No	Kode Dataset	<i>Running Time (s)</i>
1	CE_NH	4.62
2	CE_WH	3.72
3	CE_WHM	3.58
Rata-rata		3.97

3. Analisis *Unspecific handling*

Pada analisis *General handling* ini akan dilihat bagaimana kemampuan sistem dalam menangani data masukan yang beragam, apakah sistem akan tetap bekerja meskipun data yang menjadi masukan berbeda-beda jenisnya. Seperti pada tabel 4.41 dapat disimpulkan bahwa sistem tetap bejalan dan memproses data masukan, meskipun data kurs yang dimasukan tidak memiliki header dengan tipe *numerical*, tidak memiliki cara penginterpretasian, tidak memiliki *data description*, dan memiliki *data description* ketika eksperimen ke-tiga.

Tabel 4.41 Hasil evaluasi *Responsiveness*

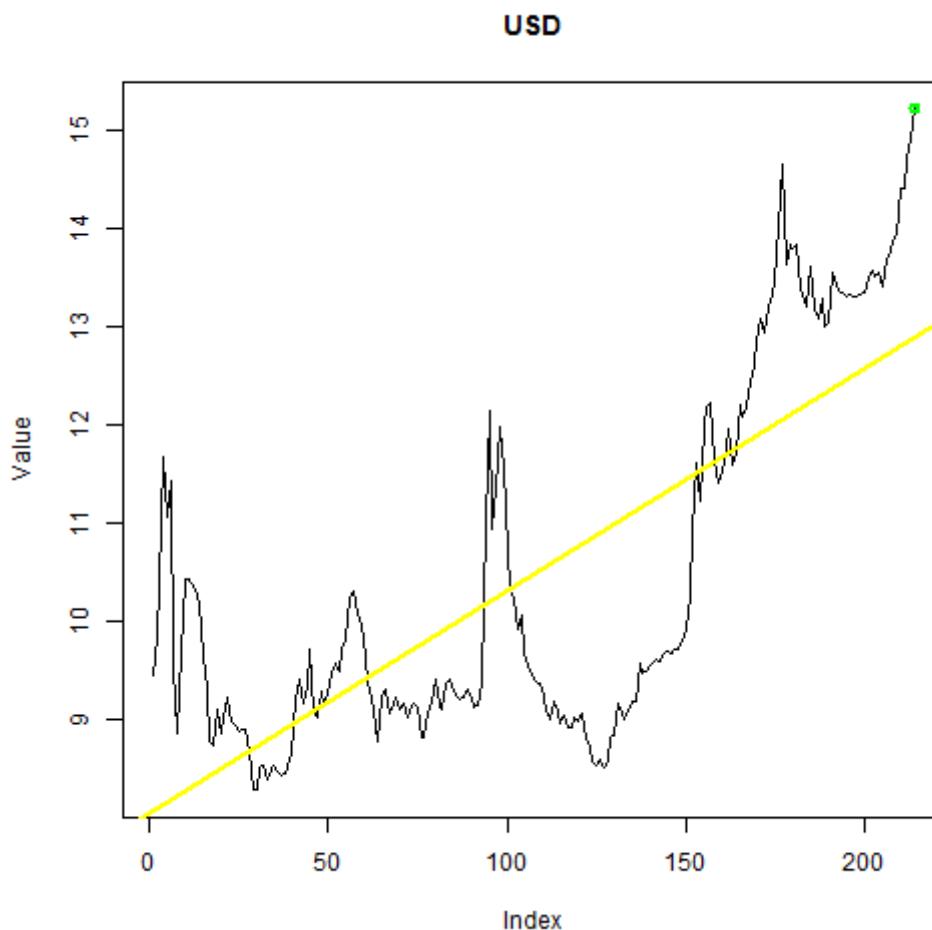
No	Kode Dataset	Header	Config Rule	Rule	Numerical	Categorical	Config Alternate	Output
1	CE_NH	No	No	Unspecific	Yes	No	No	Yes
2	CE_WH	Yes	No	Unspecific	Yes	No	No	Yes
3	CE_WHM	Yes	Yes	Unspecific	Yes	No	Yes	Yes

4. Analisis *Representative Text*

Untuk Analisis *Representative Text* ini dilakukan dengan cara pembuatan grafik garis dengan menggunakan indikator warna untuk merepresentasikan *event* yang terjadi pada parameter tersebut. Warna hitam pada grafik merepresentasikan data suatu parameter dimana sumbu *x* merepresentasikan indeks *Date Time* dan sumbu *y* adalah nilai dari parameter yang bersangkutan, warna kuning menandakan *trend* dari parameter tersebut, warna hijau menandakan kenaikan ekstrem, warna merah menandakan penurunan ekstrem, warna biru menandakan *repeated event*, sedangkan lingkaran hijau menandakan bahwa data terakhir merupakan data tertinggi, dan lingkaran merah menandakan bahwa data terakhir merupakan data terendah pada parameter tersebut. Untuk hasil keseluruhan plot dapat dilihat pada lampiran.

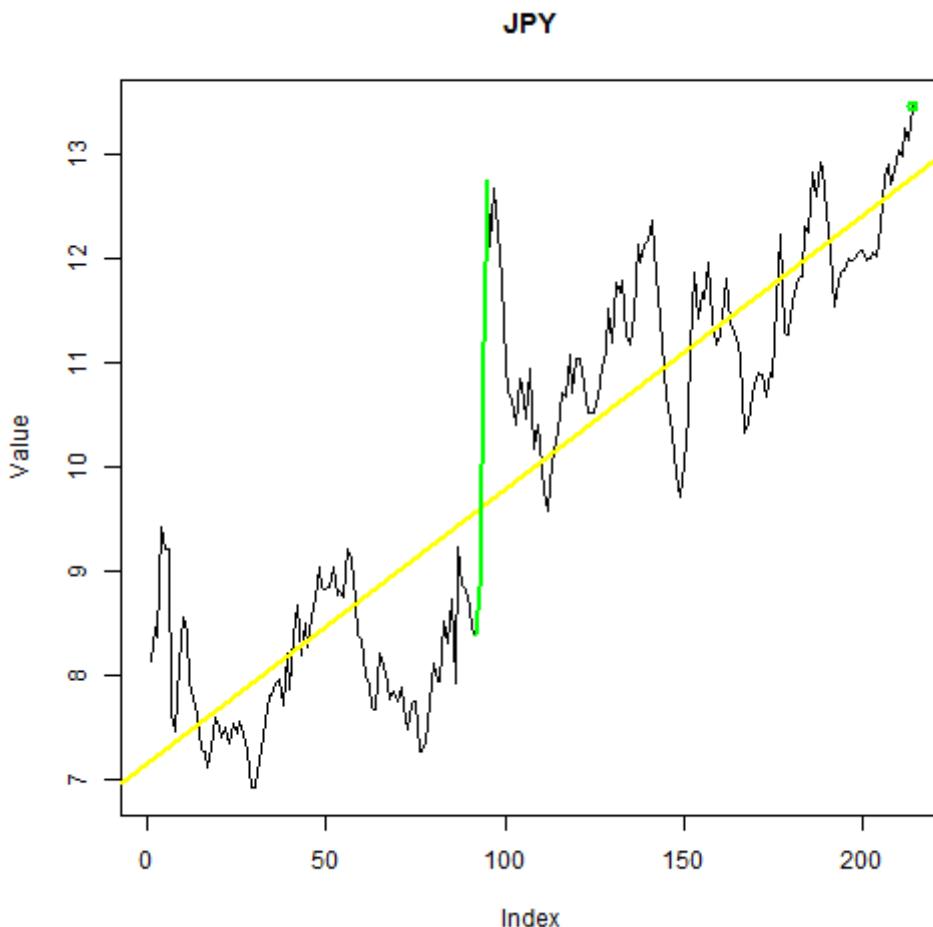
Analisis *Representative Text* ini hanya dilakukan pada beberapa parameter saja, mengingat tidak semua *event* terjadi pada setiap parameter. *Event* yang terjadi pada parameter tidak dipengaruhi oleh *header*, *data description*, maupun tipe penginterpretasian, sehingga analisis *Representative Text* ini hanya dilakukan sekali saja, karena eksperimen

pertama hingga eksperimen ke-tiga menghasilkan *event* yang sama, selama nilai dari data masukan tidak mengalami perubahan. Parameter yang akan dibahas pada analisis *Representative Text* untuk kasus data tukar ini adalah parameter USD, JPY, SGD, HKD, dan CAD.



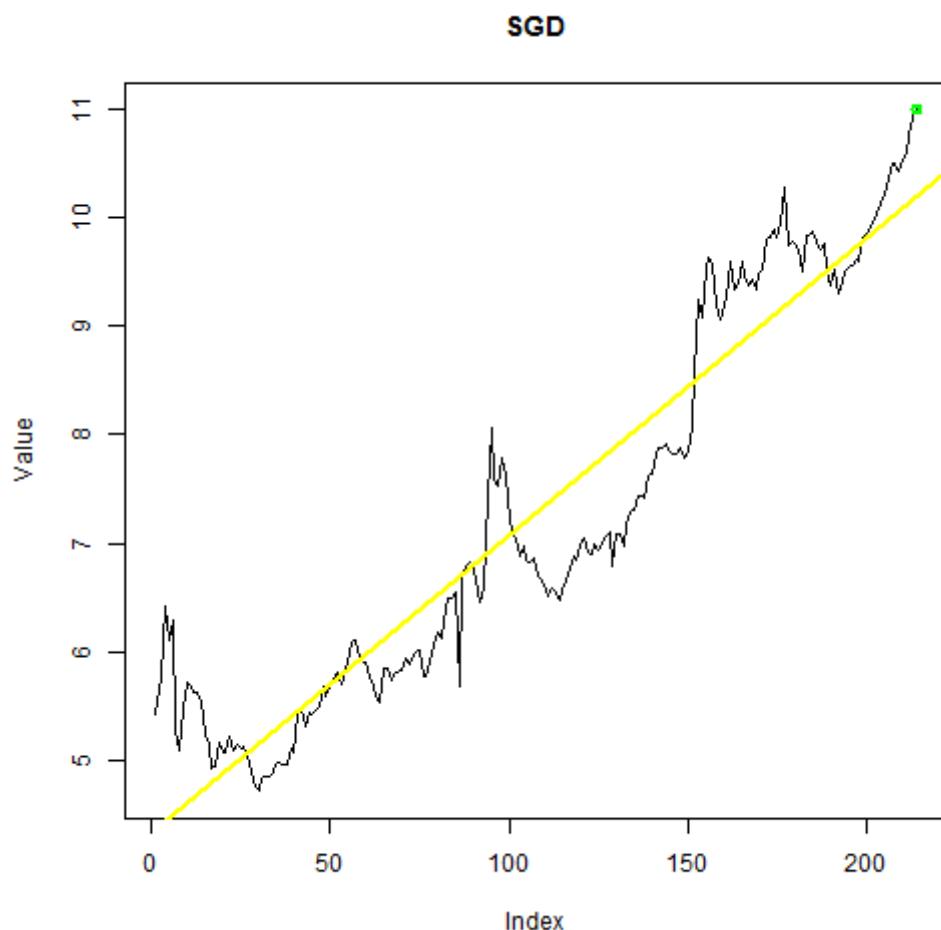
Gambar 4.67 Plot *Representative Text* untuk parameter US.Dollar

Dalam teks keluaran pada gambar 4.67 disebutkan bahwa *trend* parameter USD adalah menaik hal tersebut dibuktikan dengan *linear model* atau garis berwarna kuning menaik. Selain itu dikatakan bahwa parameter data terakhir pada parameter USD merupakan data dengan nilai tertinggi sejauh ini (*highest value*) yang dibuktikan dengan titik berwarna hijau pada data terakhir.



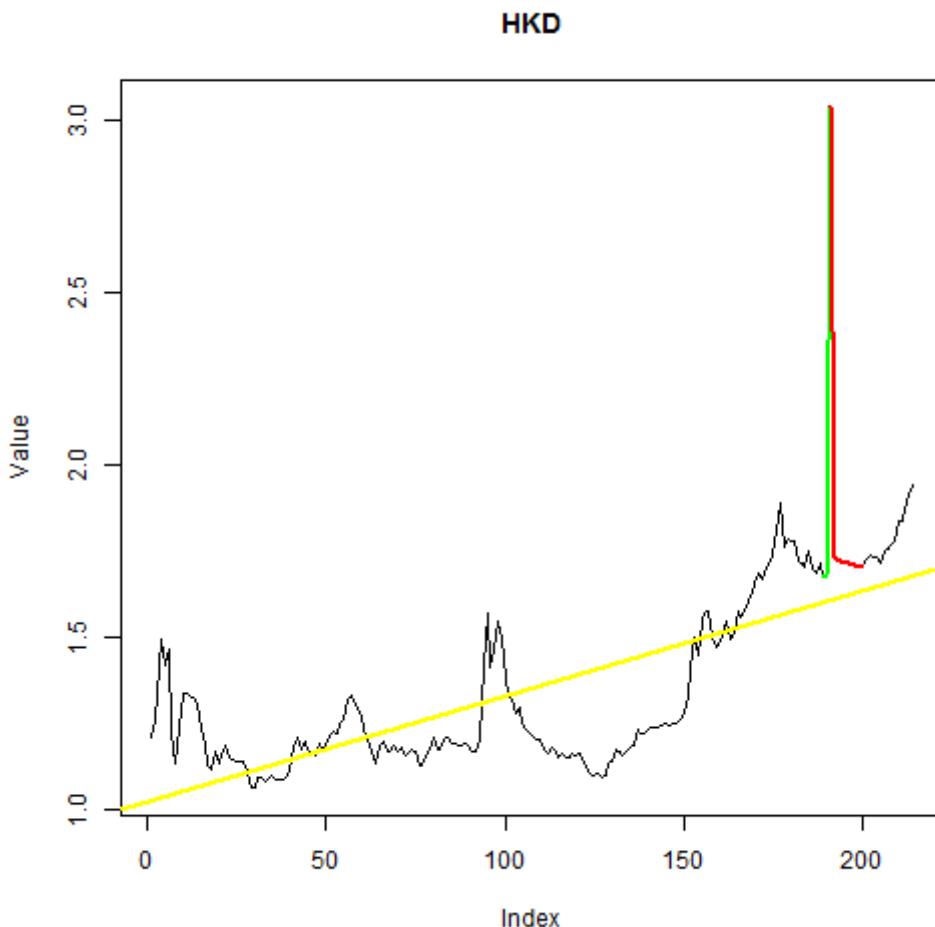
Gambar 4.68 Plot *Representative Text* untuk parameter Japan yen (JPY)

Pada teks keluaran, dikatakan bahwa *trend* parameter JPY menaik seperti yang terlihat pada gambar 4.68, selain itu dikatakan bahwa parameter data terakhir pada parameter JPY merupakan data nilai tertinggi sejauh ini (*highest value*) hal ini dapat dibuktikan dengan plot berupa titik hijau pada data terakhir yang posisinya berada pada puncak grafik. Selain itu teks keluaran mengatakan bahwa terjadi kenaikan yang ekstrem pada parameter JPY sebesar 4.353 poin, hal ini dibuktikan dengan garis hijau yang mengalami kenaikan signifikan dari titik 8 koma sekian hingga titik 12 koma sekian.



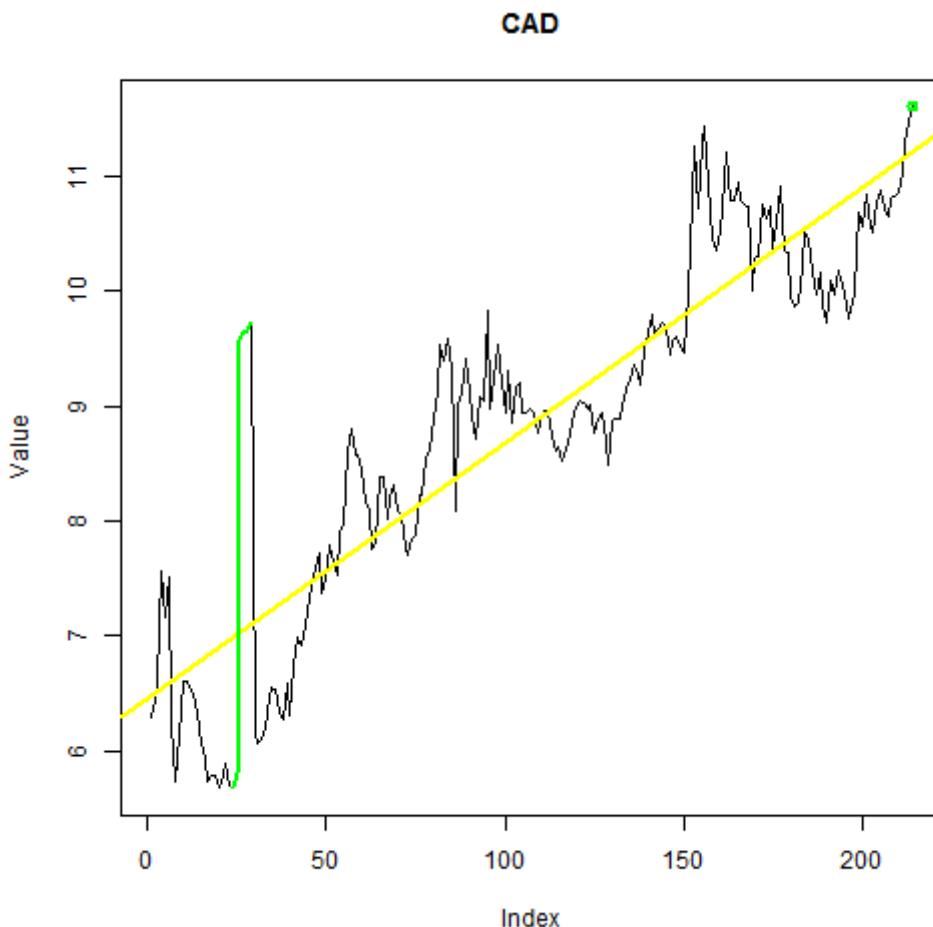
Gambar 4.69 Hasil plot *Representative Text* untuk parameter Singapore Dollar (SGD)

Trend yen Jepang pada teks keluaran dikatakan menaik sesuai dengan *Linear Model* pada gambar 4.69 yang berwarna kuning. Selain itu dikatakan bahwa data terakhir pada parameter SGD merupakan nilai tertinggi (*highest value*) hal ini dibuktikan dengan lingkaran berwarna yang merupakan data terakhir, mengingat posisinya yang berada pada puncak grafik, sehingga dimunculkan pesan untuk menandakan bahwa parameter SGD mencapai puncaknya pada data terakhir.



Gambar 4.70 Hasil plot *Representative Text* untuk parameter Hong Kong Dollar (HKD)

Pada teks keluaran dikatakan bahwa *trend* HKD adalah menaik, hal ini dibuktikan dengan *linear model* pada gambar 4.70 yang juga menaik. Selain itu pada teks keluaran disebutkan bahwa pada parameter HKD terjadi fluktuasi yang ekstrem dimana terjadi kenaikan sebesar 1.3619 poin dan penurunan sebesar 1.33185 poin, fluktuasi ini dapat kita lihat pada gambar 4.70 dimana terdapat warna hijau dan kuning yang merepresentasikan kenaikan dan penurunan yang ekstrem. Jumlah kenaikan dan penurunan terjadi pada kisaran nilai 1.7 hingga 3.0, sehingga pesan fluktuasi pada teks dapat dibuktikan karena nilai kenaikan dan penurunannya sekitar 1.3.



Gambar 4.71 Hasil plot *Representative Text* untuk parameter Canadian Dollar (CAD)

Trend pada parameter CAD adalah *increase* atau menaik, hal ini serupa dengan *linear model* yang direpresentasikan pada gambar 4.71 dengan garis kuning. Selain itu pada teks keluaran dikatakan bahwa parameter CAD mengalami kenaikan yang cukup ekstrem, dimana kenaikan tersebut direpresentasikan dengan garis hijau yang mengalami kenaikan dari sekitar angka 5.8 hingga angka 9.8 hal ini sejalan dengan pesan yang terdapat pada teks keluaran yang menyatakan bahwa terjadi kenaikan ekstrem pada parameter CAD sebesar 4.051 poin. Lingkara hijau menandakan bahwa CAD menyentuh nilai tertinggi pada pada terakhir, seperti yang dapat kita lihat dimana posisi lingkaran hijau pada gambar 4.71 menempati posisi puncak pada grafik.



Gambar 4.72 Hasil plot *Representative Text* untuk korelasi parameter

Pada teks keluaran dikatakan bahwa SGD memiliki dampak paling besar pada parameter lain, hal ini dibuktikan dengan menghitung nilai rata-rata hasil korelasi parameter SGD pada gambar 4.72 sehingga didapatkan hasil sebesar 0.8675 ($(0.9 + 0.88 + 0.63 + 0.99 + 0.89 + 0.84 + 0.9 + 0.88) / 8 = 0.8675$) yang merupakan nilai rata-rata tertinggi dibandingkan dengan hasil korelasi pada parameter lainnya. Selain itu, pada teks keluaran dikatakan bahwa kenaikan pada parameter USD berpengaruh pada kenaikan parameter JPY, CHF, SGD, MYR dan HKD hal ini dibuktikan dengan nilai korelasi USD-JPY sebesar 0.72, USD-CHF sebesar 0.86, USD-SGD sebesar 0.9, USD-MYR sebesar 0.75, dan USD-HKD sebesar 0.94. Kenaikan pada parameter JPY pun berdampak pada kenaikan parameter CHF, SGD, MYR, AUD, dan CAD. Kenaikan pada parameter CHF berpengaruh pada SGD, MYR, HKD, AUD, dan CAD. Kenaikan pada parameter SGD berpengaruh pada kenaikan parameter MYR, HKD, AUD, dan CAD. Serta kenaikan pada parameter MYR berpengaruh pada kenaikan parameter AUD, dan CAD. Lalu kenaikan pada parameter AUD berdampak

pada kenaikan parameter CAD. Keterkaitan parameter akan ditampilkan pada pesan jika nilai korelasi kedua parameter tersebut melebihi nilai 0.7 yang berarti hubungan setiap parameter tersebut tergolong pada *very strong relationship*.

4.7.2. Hasil dan Pembahasan Hasil Eksperimen Data Klimatologi

Eksperimen menggunakan data klimatologi pada situs web www.MeteoGalicia.gal, selama satu tahun pada periode 2016-2017 dengan interval data harian. Data ini dipilih dikarenakan setiap parameternya memiliki *trend* yang berbeda-beda, memiliki parameter dengan fluktuasi yang ekstrem, dan parameter dengan nilai konstan, selain itu data ini juga dipakai sebagai masukan pada penelitian DWP (Putra *et al.*, 2017).

4.7.4.4. Hasil Eksperimen Data Klimatologi Tanpa Header

Eksperimen keempat dilakukan dengan menggunakan data klimatologi pada tabel 4.2 dengan *header* yang dihilangkan terlebih dahulu atau *climate no header* (CL_NH). Cuplikan dari data tersebut dapat dilihat pada tabel 4.42 sehingga didapatkan hasil seperti pada gambar 4.73.

Tabel 4.42 Cuplikan data klimatologi tanpa menggunakan *header* (CE_NH)

07/06/2016 00:00	40.8	21.3	5.47	315	0
07/07/2016 00:00	20.9	20.1	6.41	315	0
07/08/2016 00:00	27.2	19.5	7.02	315	0
07/09/2016 00:00	23.2	19.1	5.94	315	0
07/10/2016 00:00	77.5	18.7	5.44	180	0
...
07/02/2017 00:00	12.6	18.9	7.34	315	0
07/03/2017 00:00	13.3	21.8	4.86	315	0
07/04/2017 00:00	18.7	24	6.59	225	0
07/05/2017 00:00	81.1	19.2	8.35	225	0
07/06/2017 00:00	58.3	17.9	6.48	315	0

Climatology NEWS

Regarding to the daily MeteoGalicia Climatology data between 07/06/2016 00:00 to 07/06/2017 00:00, with parameters: V2, V3, V4, V5, and V6, it represented that V3 trend is decreased and V6 trend is constant but the rest is increased. V3, and V5 parameters are higher than last week's data, but the rest parameters are lower than last week's data. There were some repeating value more than 36 days: V6 stayed constant at very low during 6 Jul - 18 Aug 2016. V2 fluctuated significantly (increased 90.9 points and decreased 87.6 points), V4 fluctuated extremely (increased 20.05 points and decreased 27.03 points), V5 fluctuated rapidly (increased 270 points and decreased 270 points), and V6 fluctuated significantly (increased 67.2 points and decreased 64.1 points). V6 appears to have a highest impact to all variable with moderate relationship in average.

Today data illustrate that: V2 in medium condition. V3 in high condition. V4 in low condition. V5 in very high condition. V6 in very low condition. V6 reached their lowest value on this day. V5 reached their highest value on this day.

Regarding to the prediction result, it's projected that V2 will start to change to medium. V3 will keep stable at high. V4 will stay stable at low. V5 will begin to turn to very high. V6 will still stable at very low.

Gambar 4.73 Hasil eksperimen ke-empat menggunakan data klimatologi tanpa menggunakan *header* (CL_NH)

Hasil keluaran eksperimen ke-empat pada gambar 4.73 dapat dilihat bahwa ketika data masukan tidak dikenali oleh sistem dikarenakan tidak adanya *header* pada data tersebut, sistem akan tetap memproses data tersebut lalu memberi nama setiap parameter dengan *header* V2, V3, V4 dan seterusnya. Dimana dalam informasi pada gambar 4.73 dikatakan bahwa *trend* dari parameter ke-tiga atau V3 adalah menurun, lalu *trend* untuk parameter V6 adalah konstan dan sisanya menurun. Parameter V3 dan V5 memiliki nilai yang lebih besar dibandingkan dengan nilai pada minggu lalu. Selain itu, dikatakan bahwa terdapat nilai yang berulang pada parameter ke-enam atau V6, dimana parameter tersebut tidak mengalami perubahan selama lebih dari 36 hari dari mulai tanggal 6 Juli hingga tanggal 18 Agustus 2016. Terdapat beberapa *extreme event*, dimana parameter V2, V4, V5, dan V6 mengalami fluktuasi yang cukup ekstrem. Dikatakan bahwa V6 memiliki nilai rata-rata korelasi tetinggi dengan interpretasi korelasi moderat. Pada paragraf ke-dua terlihat deskripsi kondisi setiap parameter, dan terdapat *event* yang menunjukkan bahwa parameter V6 mencapai nilai tertinggi, dan V5 mencapai nilai terendah pada bulan tersebut. Pada paragraf ke-tiga disimpulkan bahwa parameter V2, dan V5 akan mulai beralih ke kondisi *medium*, dan *very high*.

4.7.4.5. Hasil Eksperimen Data Klimatologi Tanpa *Header*

Climatology NEWS

According to the daily MeteoGalicia Climatology data between 07/06/2016 00:00 to 07/06/2017 00:00, with parameters: CloudCoverage, Temperature, WindSpeed, WindDirection, and Rainfall, it is clear that Temperature trend is decreased and Rainfall trend is constant but the rest is increased. Temperature, and WindDirection parameters are higher than last week's data, but the rest parameters are lower than last week's data. There were some repeating value more than 36 days: Rainfall stayed constant at very low during 6 Jul - 18 Aug 2016. CloudCoverage fluctuated sharply (increased 90.9 points and decreased 87.6 points), WindSpeed fluctuated sharply (increased 20.05 points and decreased 27.03 points), WindDirection fluctuated sharply (increased 270 points and decreased 270 points), and Rainfall fluctuated extremely (increased 67.2 points and decreased 64.1 points). Rainfall appears to have a highest impact to all variable with moderate relationship in average.

Today data indicate that: CloudCoverage in medium condition. Temperature in high condition. WindSpeed in low condition. WindDirection in very high condition. Rainfall in very low condition. Rainfall reached their lowest value on this day. WindDirection reached their highest value on this day.

According to the prediction result, it's projected that CloudCoverage will start to change to medium. Temperature will stay stable at high. WindSpeed will keep stable at low. WindDirection will start to change to very high. Rainfall will steady at very low.

Gambar 4.74 Hasil eksperimen ke-lima menggunakan data klimatologi dengan menggunakan *header* (CL_WH)

Untuk eksperimen ke-lima pada gambar 4.74 digunakan klimatologi dimana data tersebut diproses dengan menggunakan *header* aslinya namun untuk penginterpretasiannya masih digunakan cara *unspecific*. Dimana tidak ada perbedaan seperti pada eksperimen ke-lima, perbedaan yang mencolok hanya terletak pada penyebutan parameter, dimana pada eksperimen ke-empat penyebutan parameter menggunakan nama *header* secara default, sedangkan pada eksperimen ke-lima penyebutan parameter menggunakan nama *header* aslinya seperti pada tabel 4.2 dimana V2 adalah parameter kedua yakni CloudCoverage, V3 adalah Temperature, V4 adalah WindSpeed, dan V5 adalah WindDirection Selain itu perbedaan terletak pada frasa hasil dari proses *Reffering Expression Generation* dalam mengungkapkan kondisi kenaikan dan penurunan seperti *increase*, *growth*, *rise*, *decline*, *decrease*, *decay* dan frasa untuk mengungkapkan *extreme event* seperti *extremely*, *dramatically*, *significantly* dan *sharply*.

4.7.4.6. Hasil Eksperimen Data Klimatologi Dengan Kustomisasi *Corpus*

Untuk eksperimen ke-enam digunakan data nilai tukar yang sama seperti pada eksperimen ke-lima, namun pada eksperimen ke-enam ini digunakan kustomisasi *corpus*. Dimana kustomisasi ini didefinisikan pada *data description* yang dapat dilihat pada tabel 4.43.

Tabel 4.43 Kustomisasi *data description* pada eksperimen ke-enam

ColName	Type	Rule	Alternate
AirQuality	numeric	range	Air Quality
WindSpeed	numeric	range	Wind Speed
WindDirection	categorical	NA	Wind Direction
CloudCoverage	numeric	range	Cloud Coverage
Temperature	numeric	fuzzy	NA
Rainfall	numeric	fuzzy	NA

Sehingga didapatkan hasil seperti pada gambar 4.75 dimana pada gambar tersebut terlihat, nama parameter pada teks keluaran berubah mengikuti atribut *alternate* pada *data description*. Dimana parameter WindSpeed ditampilkan sebagai Wind Speed, parameter Wind Direction ditampilkan sebagai Wind Direction, dan CloudCoverage ditampilkan menjadi Cloud Coverage. Untuk pendekripsi *event* pada eksperimen ke-enam ini tidak ada perbedaan dengan eksperimen sebelumnya, dikarenakan pendekripsi *event* tidak dipengaruhi oleh perubahan *header*. Namun proses kustomisasi ini sangat berpengaruh pada hasil interpretasi data, dimana jika pada sebelumnya semua parameter diinterpretasikan menggunakan cara *unspecific*, sedangkan pada eksperimen ke-enam ini semua parameter memiliki cara penginterpretasiannya masing-masing, cara penginterpretasian pada eksperimen ke-enam ini digunakan kaidah penginterpretasian seperti pada penelitian (Putra *et al.*, 2017).

Tidak dilakukan analisis pendekripsi *event* seperti *trend*, *extreme event*, *predict* pada parameter WindDirection dikarenakan kini parameter WindDirection didefinisikan dengan tipe *categorical* tidak seperti pada eksperimen ke-lima dimana tipe dari parameter WindDirection adalah *numerical*, sehingga hanya dilakukan proses pendekripsi sinyal berupa *string matching* yang ditandakan dengan munculnya pesan hasil dari pendekripsi sinyal *String Matching* berupa “*For the past 7 days , no equivalent patterns were found for each categorical parameters.*”. Jika pada eksperimen ke-lima parameter Cloud Coverage berada pada kondisi *medium* yang merupakan hasil penginterpretasian dengan cara *unspecific*, kini pada eksperimen ke-tiga ini setelah dilakukan kustomisasi cara pendefinisian, parameter Cloud Coverage berada pada kondisi *mostly cloudy* yang merupakan hasil interpretasi data seperti yang sudah dijelaskan pada sub-bab *Data Interpretation*.

sebelumnya. Selain itu, parameter Temperature berada pada kondisi *warm*, lalu parameter Wind Speed berada pada kondisi *light breeze*

Pada paragraf ke-tiga terdapat perbedaan yang cukup mencolok, yakni dibangkitkannya beberapa kalimat prediksi pada penelitian (Putra *et al.*, 2017) yang berisi “*Based on prediction result, it's projected that tomorrow sky will be light rain although it's covered by partly cloudy sky. Favored by temperature which decreased to warm.*”. Hal ini dikarenakan, jika parameter-parameter yang digunakan pada penelitian oleh Putra (2017) terdapat pada data masukan beserta lengkap dengan *data description* cara penginterpretasiannya, maka *special corpus* yang diadaptasi pada penelitian oleh Putra (2017) akan ikut dibangkitkan.

Climatology NEWS

Regarding to the daily MeteoGalicia Climatology data between 07/06/2016 00:00 to 07/06/2017 00:00, with parameters: Cloud Coverage, Temperature, Wind Speed, Wind Direction, and Rainfall, it represented that Temperature trend is decreased and Rainfall trend is constant but the rest is increased. Temperature, and Wind Direction parameters are higher than last week's data, but the rest parameters are lower than last week's data. There were some repeating value more than 36 days: Rainfall stayed constant at no rain during 6 Jul - 18 Aug 2016. Cloud Coverage fluctuated sharply (increased 90.9 points and decreased 87.6 points), Wind Speed fluctuated dramatically (increased 20.05 points and decreased 27.03 points), Wind Direction fluctuated sharply (increased 270 points and decreased 270 points), and Rainfall fluctuated significantly (increased 67.2 points and decreased 64.1 points). Rainfall appears to have a highest impact to all variable with moderate relationship in average.

Today data illustrate that: Cloud Coverage in mostly cloudy condition. Temperature in warm condition. Wind Speed in light Breeze condition. Wind Direction in North West condition. Rainfall in no rain condition. Rainfall reached their lowest value on this day. Wind Direction reached their highest value on this day.

Regarding to the prediction result, it's projected that tomorrow sky will be light rain although it's covered by partly cloudy sky. Favored by temperature which decreased to warm. Cloud Coverage will change progressively to partly cloudy. Temperature will still stable at warm. Wind Speed will still stable at light Breeze. Wind Direction will averagely change to West. Rainfall will shifted to light rain.

Gambar 4.75 Hasil eksperimen ke-enam menggunakan data klimatologi dengan kustomisasi *corpus* (CL_WHM)

4.7.4.7. Analisis Hasil Eksperimen Data Klimatologi

Analisis hasil eksperimen dengan menggunakan data klimatologi pada periode 2016-2017 adalah sebagai berikut:

1. Analisis aspek *Readability*

Evaluasi dilakukan dengan menggunakan metode *Flesch Reading Ease* menggunakan *Readability Analyzer* pada situs www.datayze.com dan aplikasi *Automatic Readability Checker* pada situs www.readabilityformulas.com. Hasil pengujian aspek *Readability* ini dapat dilihat pada tabel 4.44. Dimana rata-rata yang didapatkan dari proses analisis ini adalah 60.41 yang kemudian berdasarkan *Flesch Reading Ease*

Score pada tabel 4.35 didapatkan bahwa informasi dari teks keluaran tergolong pada kategori *plain english* sehingga mudah dipahami bahkan oleh anak usia remaja sekalipun. Untuk hasil lebih lengkap terdapat didalam lampiran.

Tabel 4.44 Hasil evaluasi *Readability* dengan data klimatologi

No	Kode Dataset	Flesch Reading Ease Score (Datayze)	Flesch Reading Ease Score (Readability Formula)
1	CL_NH	68.93	70.2
2	CL_WH	47.46	52.6
3	CL_WHM	59.98	63.3
Rata-rata		58.79	62.03
Rata-rata Keseluruhan		60.41	

2. Analisis aspek *Computation Time*

Hasil perhitungan *Computation Time* pada data klimatologi terdapat pada tabel 4.45. Terlihat bahwa rata-rata *running time* sistem ini adalah 3.99 detik. Untuk hasil lebih lengkapnya terdapat pada lampiran.

Tabel 4.45 Hasil evaluasi *Computation Time* dengan data klimatologi

No	Kode Dataset	Running Time (s)
1	CL_NH	4.36
2	CL_WH	4.24
3	CL_WHM	3.37
Rata-rata		3.99

3. Analisis *Unspecific Handling*

Pada analisis *Unspecific Handling* ini akan dilihat bagaimana respon sistem dalam menangani berbagai jenis data masukan, apakah sistem akan tetap bekerja meskipun data yang menjadi masukan berbeda-beda jenisnya. Seperti pada tabel 4.46 terlihat bahwa sistem tetap bejalan dan memproses data masukan, meskipun data klimatologi yang dimasukan tidak memiliki header, tidak memiliki cara penginterpretasian, tidak memiliki *data description*, memiliki kostumisasi *corpus* dan memiliki parameter dengan tipe *categorical* pada eksperimen ke-tiga.

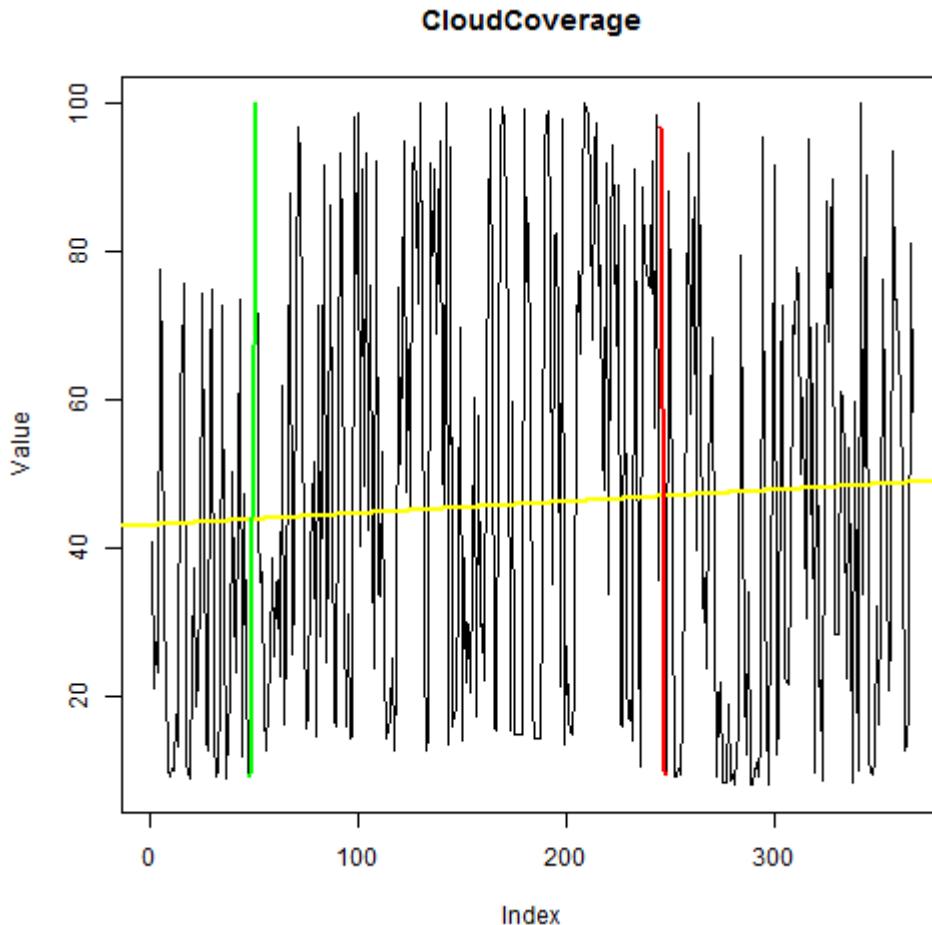
Tabel 4.46 Hasil evaluasi *Unspecific Handling*

No	Kode Dataset	Header	Config Rule	Rule	Numerical	Categorical	Config Alternate	Output
1	CL_NH	No	No	Unspecific	Yes	No	No	Yes
2	CL_WH	Yes	No	Unspecific	Yes	No	No	Yes
3	CL_WHM	Yes	Yes	Crisp, Fuzzy	Yes	Yes, 1	Yes	Yes

4. Analisis *Representative Text*

Untuk Analisis *Representative Text* ini dilakukan dengan cara pembuatan grafik garis dengan menggunakan indikator warna untuk merepresentasikan *event* yang terjadi pada parameter tersebut. Warna hitam pada grafik merepresentasikan data suatu parameter, warna kuning menandakan *trend* dari parameter tersebut, warna hijau menandakan kenaikan ekstrem, warna merah menandakan penurunan ekstrem, warna biru menandakan *repeated event*, sedangkan lingkaran hijau menandakan bahwa data terakhir merupakan data tertinggi, dan lingkaran merah menandakan bahwa data terakhir merupakan data terendah pada parameter tersebut. Untuk hasil keseluruhan plot dapat dilihat pada lampiran.

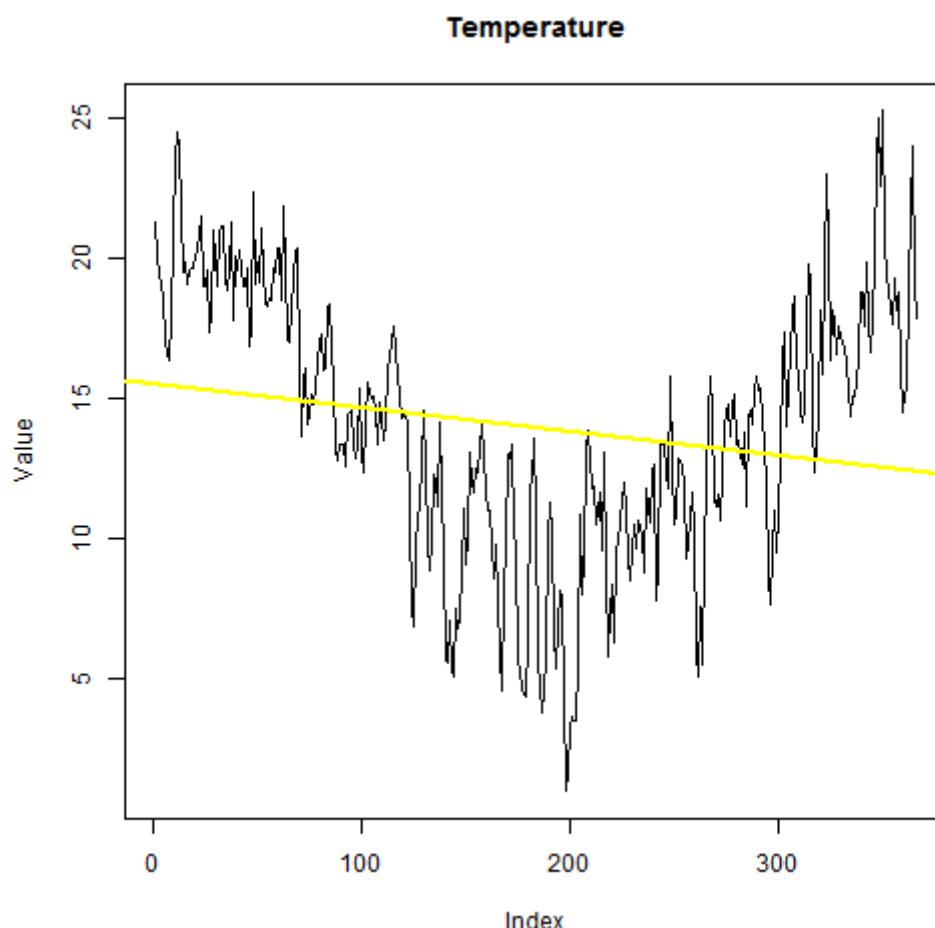
Analisis *Representative Text* dilakukan hampir sama seperti pada analisis *Representative Text* eksperimen data kurs sebelumnya. Dimana analisis ini dilakukan pada semua parameter yang ada pada data. Parameter yang akan dibahas pada analisis *Representative Text* ini yakni CloudCoverage, Temperature, WindSpeed, WindDirection, dan Rainfall.



Gambar 4.76 Hasil plot *Representative Text* untuk parameter CloudCoverage

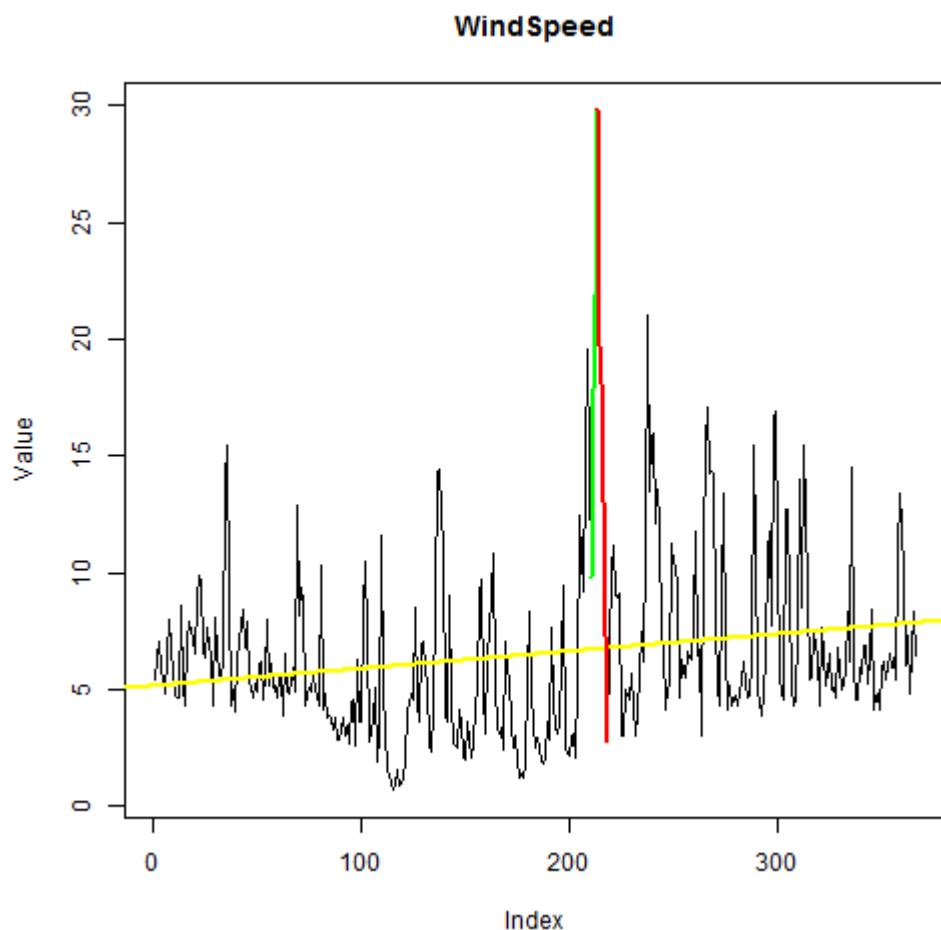
Dalam teks keluaran disebutkan bahwa *trend* parameter CloudCoverage adalah menaik hal tersebut dibuktikan dengan *linear model* atau garis berwarna kuning pada gambar 4.76 yang menaik. Selain itu pada teks keluaran disebutkan bahwa pada parameter CloudCoverage terjadi fluktuasi yang ekstrem dimana terjadi kenaikan sebesar 90.9 poin dan penurunan sebesar 87.6 poin, fluktuasi ini dapat kita lihat pada gambar 4.76 dimana terdapat warna hijau dan kuning yang merepresentasikan kenaikan dan penurunan yang ekstrem. Jumlah kenaikan dan penurunan terjadi pada kisaran nilai 5 hingga 100, sehingga pesan fluktuasi pada teks dapat dibuktikan karena nilai kenaikan dan penurunannya sekitar 90-95. Pada paragraf ke-dua dari teks keluaran dikatakan bahwa parameter CloudCoverage ini berada pada kondisi *mostly cloudy* jika diterjemahkan menggunakan *rule* yang sudah didefinisikan, dan *medium* jika diinterpretasikan menggunakan cara *unspecific*, hal ini dapat dilihat bahwa

data terakhir berada pada kisaran 50 yang merupakan nilai tengah dari keseluruhan data.



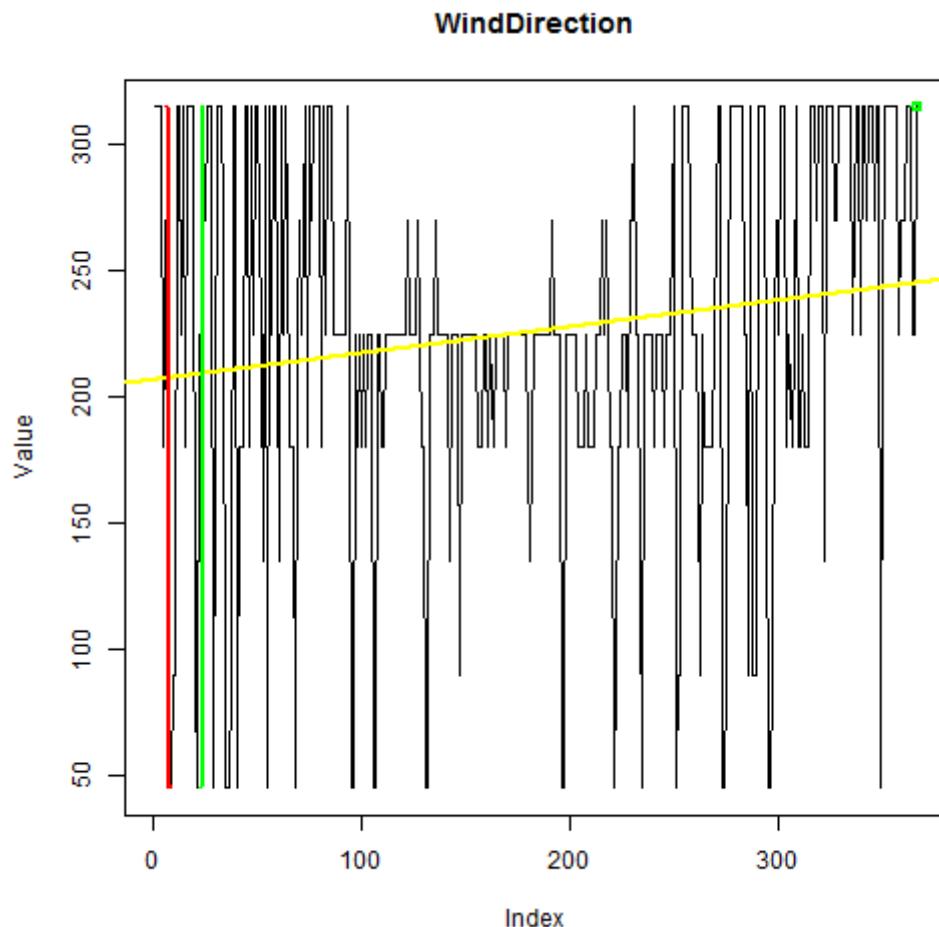
Gambar 4.77 Plot *Representative Text* untuk parameter Temperature

Pada teks keluaran, dikatakan bahwa *trend* parameter JPY adalah menurun seperti yang terlihat pada *linear model* yang terdapat pada gambar 4.77, tidak ada *event* yang terjadi pada parameter Temperature ini. Namun pada teks keluaran dikatakan bahwa data terakhir berada pada kondisi *warm*, dan *high* jika diinterpretasikan menggunakan cara *unspecific*, hal ini sesuai dengan hasil pada gambar 4.77 dimana pada grafik tersebut data terakhir berada pada rentang nilai 15 dan 20 yang berada pada 3/5 wilayah dari data keseluruhan dihitung dari 0.



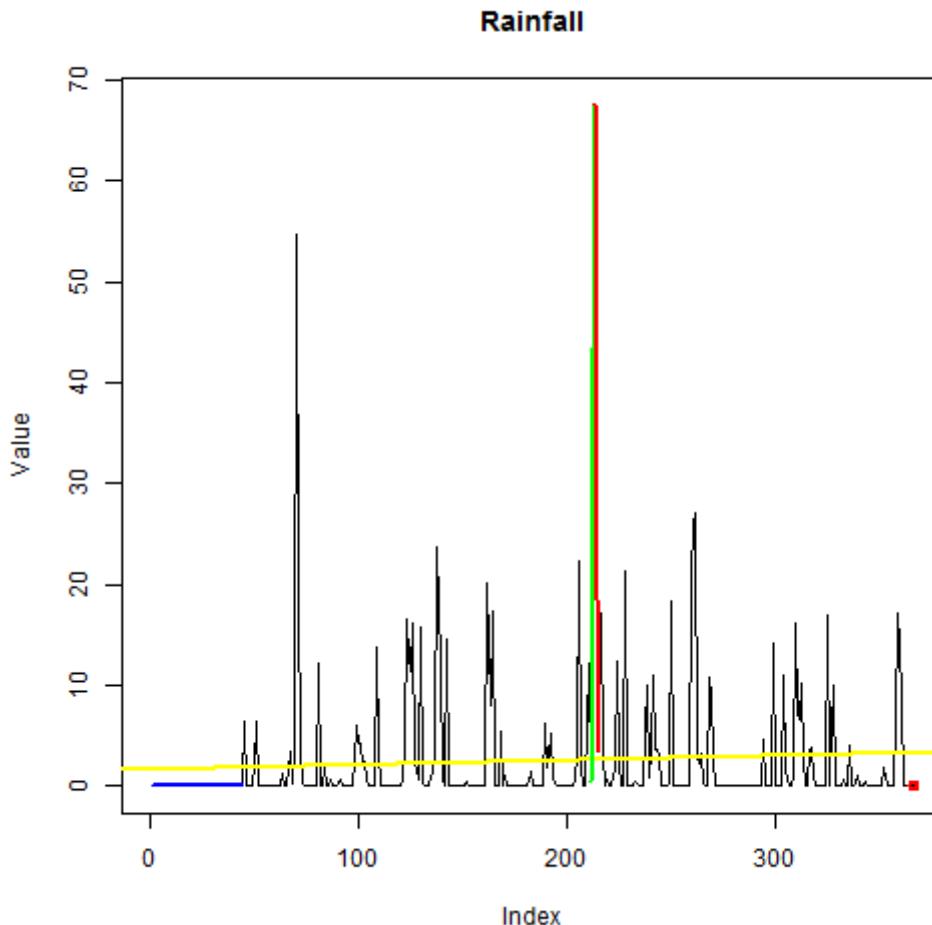
Gambar 4.78 Hasil plot *Representative Text* untuk parameter WindSpeed

Trend WindSpeed pada teks keluaran dikatakan menaik sesuai dengan *Linear Model* pada gambar 4.78 yang berwarna kuning. Selain itu pada teks keluaran disebutkan bahwa pada parameter WinSpeed terjadi fluktuasi yang ekstrem dimana terjadi kenaikan sebesar 20.05 poin dan penurunan sebesar 27.03 poin, fluktuasi ini dapat kita lihat pada gambar 4.78 dimana terdapat warna hijau merepresentasikan kenaikan yang ekstrem dan kuning yang merepresentasikan penurunan yang ekstrem.



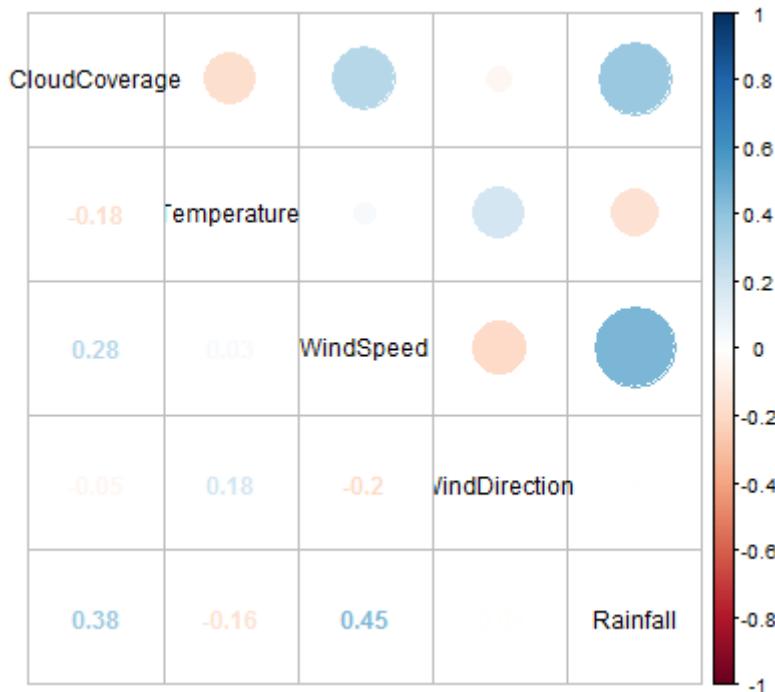
Gambar 4.79 Hasil plot *Representative Text* untuk parameter WindDirection

Pada teks keluaran dikatakan bahwa *trend* WindDirection adalah menaik, hal ini dibuktikan dengan *linear model* pada gambar 4.79 yang juga menaik. Selain itu juga, hasil dari eksperimen ke-empat dan ke-lima dikatakan bahwa data terakhir pada parameter WindDirection merupakan data dengan nilai tertinggi pada tahun ini (*highest value*). Namun pada eksperimen ke-enam parameter WindDirection ini didefinisikan sebagai *categorical* parameter sehingga tidak ada analisis *highest value*.



Gambar 4.80 Hasil plot *Representative Text* untuk parameter Rainfall

Pada teks keluaran dikatakan bahwa *trend* pada parameter Rainfall adalah *constant*, walau *linear model* yang direpresentasikan pada gambar 4.80 dengan garis kuning yang menaik, tetapi sistem tetap mengkategorikan *trend* kedalam konstan dikarenakan kenaikan pada parameter Rainfall tidak melebihi minimum tershold sebanyak 10%. Selain itu pada teks keluaran dikatakan bahwa parameter Rainfall mengalami flutuasi yang cukup ekstrem, dimana flutuasi yang ekstrem direpresentasikan dengan garis hijau yang merepresentasikan kenaikan ekstrem dan garis merah yang merepresentasikan penurunan ekstrem. Lingkara merah menandakan bahwa parameter Rainfall menyentuh nilai terendah pada pada terakhir, seperti yang dapat kita lihat dimana posisi lingkaran merah pada gambar 4.80 menempati posisi terendah pada grafik.



Gambar 4.81 Hasil plot *Representative Text* untuk korelasi parameter data klimatologi

Pada teks keluaran dikatakan bahwa Rainfall memiliki dampak paling besar pada parameter lain, hal ini dibuktikan dengan menghitung nilai rata-rata hasil korelasi parameter Rainfall pada gambar 4.81 dimana secara perhitungan kasar parameter Rainfall memiliki nilai rata-rata korelasi paramter paling tinggi dibanding parameter lainnya. Dikarenakan tidak ada korelasi parametr yang melebihi angka 0.7 maka tidak ditampilkan *significant message* untuk korelasi antar paramter.

4.7.3. Hasil dan Pembahasan Hasil Eksperimen Data Kualitas Udara

Eksperimen menggunakan data kualitas udara pada situs web www.MeteoGalicia.gal, selama satu tahun pada periode 2016-2017 dengan interval data harian. Data ini dipilih karena pada setiap datanya terdapat fluktuasi yang signifikan, selain itu data ini bertipe *integer*, yang selanjutnya akan digunakan

modifikasi *corpus*. Data ini juga digunakan pada penelitian DWP (Putra *et al.*, 2017).

4.7.3.1. Hasil Eksperimen Data Kualitas Udara Tanpa Header

Eksperimen ke-tujuh dilakukan dengan menggunakan data klimatologi pada tabel 4.3 dengan *header* yang dihilangkan terlebih dahulu atau *air quality no header* (AQ_NH). Cuplikan dari data tersebut dapat dilihat pada tabel 4.47 sehingga didapatkan hasil seperti pada gambar 4.82.

Tabel 4.47 Cuplikan data klimatologi tanpa menggunakan *header* (AQ_NH)

07/06/2016 00:00	0.13	3	15	19	51	18	10	1
07/07/2016 00:00	0.11	1	10	10	56	14	7	1
07/08/2016 00:00	0.1	1	8	8	59	12	8	1
07/09/2016 00:00	0.1	2	10	11	57	12	7	1
07/10/2016 00:00	0.11	2	10	12	53	12	11	1
...
07/02/2017 00:00	0.17	2	35	37	10	14	20	7
07/03/2017 00:00	0.17	23	43	78	1	11	14	9
07/04/2017 00:00	0.17	31	42	90	1	0	14	9
07/05/2017 00:00	0.18	32	41	90	1	0	12	7
07/06/2017 00:00	0.18	32	61	90	1	0	12	7

Air Quality NEWS

Regarding to the daily MeteoGalicia Air Quality data from 07/06/2016 00:00 to 07/06/2017 00:00, with parameters: V2, V3, V4, V5, V6, V7, V8, and V9, it illustrated that V7 trend is increased and the rest is constant. V9 parameter is equal with last week's data. V3, V4, and V5 parameters are higher than last week's data, but the rest parameters are lower than last week's data. There were no repeating values within 36 days or more, every value changed from time to time. V6 increased extremely from 29-30 Sep 2016 (increased 95 points), V7 increased extremely from 19-20 Oct 2016 (increased 38 points), and V8 increased significantly from 30 Sep - 1 Oct 2016 (increased 32 points). V2 decreased sharply from 24-29 Sep 2016 (decreased 0.37 points), and V4 decreased significantly from 25-29 Sep 2016 (decreased 61 points). V3 fluctuated rapidly (increased 56 points and decreased 64 points), V5 fluctuated dramatically (increased 121 points and decreased 157 points), and V9 fluctuated significantly (increased 9 points and decreased 11 points). V5 appears to have a highest impact to all variable with strong relationship in average. As a result of the decay in V4, it can be seen that V2, and V3 is decreasing. A decrease in V5 resulted a decrease in V2, V3, and V4.

Today data illustrate that: V2, and V8 in low condition. V3, V5, and V9 in medium condition. V4 in very high condition. V6, and V7 in very low condition. V7 reached their lowest value on this day.

Regarding to the prediction result, it's predicted that V2, and V8 will still stable at low. V3, V5, and V9 will stay stable at medium. V4 will move to very high. V6, and V7 will keep stable at very low.

Gambar 4.82 Hasil eksperimen ke-tujuh menggunakan data kualitas udara tanpa menggunakan *header* (AQ_NH)

Hasil keluaran eksperimen ke-tujuh pada gambar 4.82 dapat dilihat bahwa ketika data masukan tidak dikenali oleh sistem dikarenakan tidak adanya *header* pada data tersebut, sistem akan tetap memproses data tersebut lalu memberi nama setiap parameter dengan *header* V2, V3, V4 dan seterusnya. Dimana V2 merupakan

parameter ke-dua mewakili CO, V3 mewakili NO, V4 mewakili NO₂, V5, mewakili NOX, V6 mewakili O₃, V7 mewakili PM10, V8 mewakili PM25, dan V9 mewakili SO₂.

Dimana dalam informasi pada gambar 4.82 dikatakan bahwa *trend* dari parameter ke-tiga atau V3 adalah menurun, lalu *trend* untuk parameter V7 adalah menaik dan sisanya adalah konstan. Parameter V9 memiliki nilai yang sama dengan nilai minggu lalu. Parameter V3, V4 dan V5 memiliki nilai yang lebih besar dibandingkan dengan nilai pada minggu lalu, dan sisanya lebih kecil. Terdapat beberapa *extreme event*, dimana parameter V6, V7, dan V8 mengalami kenaikan yang cukup ekstrem. Dikatakan juga bahwa parameter V2, dan V4 mengalami kenaikan yang cukup ekstrem, lalu parameter V3, V5 dan V9 mengalami fluktuasi yang cukup ekstrem. Dikatakan bahwa V5 memiliki nilai rata-rata korelasi tetinggi dengan interpretasi korelasi *strong*. Lalu terdapat beberapa parameter yang memiliki dampak yang kuat pada parameter lainnya, misalnya pada parameter V4 terjadi penurunan maka hal tersebut berdampak pada menurunya nilai parameter V2 dan V3, lalu jika penurunan terjadi pada parameter V5, hal ini akan berdampak juga pada menurunya nilai parameter V5.

Pada paragraf ke-dua terlihat deskripsi kondisi setiap parameter, dan terdapat *event* yang menunjukkan bahwa parameter V7 mencapai nilai terendah pada bulan tersebut. Pada paragraf ke-tiga disimpulkan bahwa parameter V4 akan mulai beralih ke kondisi *very high*, dan sisanya akan tetap berada pada hasil interpretasi pada paragraf ke-dua.

4.7.3.2. Hasil Eksperimen Data Kualitas Udara dengan *Header*

Air Quality NEWS

According to the daily MeteoGalicia Air Quality data from 07/06/2016 00:00 to 07/06/2017 00:00, with parameters: CO, NO, NO2, NOX, O3, PM10, PM25, and SO2, it can be seen that PM10 trend is increased and the rest is constant. SO2 parameter is equal with last week's data. NO, NO2, and NOX parameters are higher than last week's data, but the rest parameters are lower than last week's data. There were no repeating values within 36 days or more, every value changed from time to time. O3 increased rapidly from 29-30 Sep 2016 (increased 95 points), PM10 increased sharply from 19-20 Oct 2016 (increased 38 points), and PM25 increased extremely from 30 Sep - 1 Oct 2016 (increased 32 points). CO decreased extremely from 24-29 Sep 2016 (decreased 0.37 points), and NO2 decreased extremely from 25-29 Sep 2016 (decreased 61 points). NO fluctuated significantly (increased 56 points and decreased 64 points), NOX fluctuated significantly (increased 121 points and decreased 157 points), and SO2 fluctuated sharply (increased 9 points and decreased 11 points). NOX appears to have a highest impact to all variable with strong relationship in average. As a result of the decay in NO2, it can be seen that CO, and NO is decreasing. A fall in NOX causes an attendant decrease in CO, NO, and NO2.

Today data represent that: CO, and PM25 in low condition. NO, NOX, and SO2 in medium condition. NO2 in very high condition. O3, and PM10 in very low condition. PM10 reached their lowest value on this day.

According to the prediction result, it's predicted that air quality will stay stable at Good . CO, and PM25 will stay constant at low. NO, NOX, and SO2 will keep stable at medium. NO2 will shifted to very high. O3, and PM10 will steady at very low.

Gambar 4.83 Hasil eksperimen ke-delapan menggunakan data kualitas udara dengan menggunakan *header* (AQ_WH)

Untuk eksperimen ke-delapan pada gambar 4.83 digunakan data kualitas udara dimana data tersebut diproses dengan menggunakan *header* aslinya namun untuk penginterpretasiannya masih digunakan cara *unspecific*. Pada eksperimen ini terdapat beberapa perbedaan yang mencolok yakni terletak pada penyebutan parameter, dimana pada eksperimen ke-tujuh penyebutan parameter menggunakan nama *header* secara default, sedangkan pada eksperimen ke-delapan penyebutan parameter menggunakan nama *header* data aslinya seperti pada tabel 4.3 dimana V2 diinterpretasikan menjadi CO, V3 diinterpretasikan menjadi NO, V4 diinterpretasikan menjadi NO2, V5 diinterpretasikan menjadi NOX, dan seterusnya. Perbedaan lainnya adalah ditampilkannya pesan *Special Corpus* yang mengadaptasi hasil keluaran teks prediksi kualitas udara pada penelitian DWP (Putra *et al.*, 2017). Selain itu perbedaan lain terletak pada frasa hasil dari proses *Reffering Expression Generation* dalam mengungkapkan kondisi kenaikan dan penurunan seperti *increase*, *growth*, *rise*, *decline*, *decrease*, *decay* dan frasa untuk mengungkapkan *extreme event* seperti *extremely*, *dramatically*, *significantly* dan *sharply*.

4.7.3.3. Hasil Eksperimen Data Kualitas Udara Dengan Kustomisasi *Corpus*

Untuk eksperimen ke-sembilan digunakan data nilai tukar yang sama seperti pada eksperimen ke-delapan, namun pada eksperimen ke-sembilan ini digunakan kustomisasi *corpus*. Dimana kustomisasi ini didefinisikan pada *data description* yang dapat dilihat pada tabel 4.48.

Tabel 4.48 Kustomisasi *data description* pada eksperimen ke-sembilan

ColName	Type	Rule	Alternate
PM25	categorical	NA	NA
SO2	categorical	NA	NA

Sehingga didapatkan hasil seperti pada gambar 4.84 dimana pada gambar tersebut terlihat, bahwa pendektsian *event* pada eksperimen ke-sembilan ini tidak ada perbedaan dengan eksperimen sebelumnya, dikarenakan pendektsian *event* tidak dipengaruhi oleh perubahan *header*. Namun proses kustomisasi ini sangat berpengaruh pada hasil interpretasi data, dimana jika pada sebelumnya semua parameter diinterpretasikan menggunakan cara *unspecific*, parameter PM25 dan SO2 didefinisikan dengan tipe *categorical*, cara penginterpretasian pada eksperimen ke-enam ini digunakan kaidah penginterpretasian seperti pada penelitian (Putra *et al.*, 2017) sehingga didapatkan pesan *good* pada paragraf ketiga.

Dikarenakan parameter PM25 dan SO2 kini bertipe *categorical*, maka *tidak* dilakukan analisis pendektsian *event* seperti *trend*, *extreme event*, *predict* seperti pada eksperimen ke-lima, dimana tipe dari parameter PM25 dan SO2 adalah *numerical*, sehingga hanya dilakukan proses pendektsian sinyal berupa *string matching* yang ditandakan dengan munculnya pesan hasil dari pendektsian sinyal *String Matching* berupa “*For the past 7 days , no equivalent patterns were found for each categorical parameters.*”.

Pada paragraf ke-tiga terdapat perbedaan yang cukup mencolok, yakni dibangkitkannya beberapa kalimat prediksi pada penelitian (Putra *et al.*, 2017) yang berisi “*According to the prediction result, it's projected that air quality will stay stable at Good*”. Hal ini dikarenakan, jika parameter-parameter yang digunakan pada penelitian oleh Putra (2017) terdapat pada data masukan maka

special corpus yang diadaptasi pada penelitian oleh Putra (2017) akan ikut dibangkitkan.

Air Quality NEWS

According to the daily MeteoGalicia Air Quality data from 07/06/2016 00:00 to 07/06/2017 00:00, with parameters: CO, NO, NO2, NOX, O3, PM10, PM25, and SO2, it is clear that PM10 trend is increased and the rest is constant. NO, NO2, and NOX parameters are higher than last week's data, but the rest parameters are lower than last week's data. There were no repeating values within 36 days or more, every value changed from time to time. O3 increased dramatically from 29-30 Sep 2016 (increased 95 points), and PM10 increased dramatically from 19-20 Oct 2016 (increased 38 points). CO decreased sharply from 24-29 Sep 2016 (decreased 0.37 points), and NO2 decreased extremely from 25-29 Sep 2016 (decreased 61 points). NO fluctuated rapidly (increased 56 points and decreased 64 points), and NOX fluctuated dramatically (increased 121 points and decreased 157 points). For the past 7 days, no duplicate patterns were found for each categorical parameters. NOX appears to have a highest impact to all variable with strong relationship in average. A fall in NO2 causes an attendant decrease in CO, and NO. As a result of the decay in NOX, it can be seen that CO, NO, and NO2 is decreasing.

Today data describe that: CO in low condition. NO, and NOX in medium condition. NO2 in very high condition. O3, and PM10 in very low condition. PM10 reached their lowest value on this day.

Regarding to the prediction result, it's expected that air quality will still stable at Good. CO will keep stable at low. NO, and NOX will stay stable at medium. NO2 will normally move to very high. O3, and PM10 will stay stable at very low.

Gambar 4.84 Hasil eksperimen ke-sembilan menggunakan data kualitas udara dengan kustomisasi *corpus* (AQ_WHM)

4.7.3.4. Analisis Hasil Eksperimen Data Kualitas Udara

Analisis hasil eksperimen dengan menggunakan data kualitas udara pada periode 2016-2017 adalah sebagai berikut:

1. Analisis aspek *Readability*

Evaluasi dilakukan dengan menggunakan metode *Flesch Reading Ease* menggunakan *Readability Analyzer* pada sistem www.datayze.com dan aplikasi *Automatic Readability Checker* pada situs www.readabilityformulas.com. Hasil pengujian aspek *Readability* ini dapat dilihat pada tabel 4.49. Dimana rata-rata yang didapatkan dari proses analisis ini adalah 71.35 yang kemudian berdasarkan *Flesch Reading Ease Score* pada tabel 4.35 didapatkan bahwa informasi dari teks keluaran tergolong pada kategori *very easy to read* yang berarti teks keluaran mudah dipahami bahkan oleh kelas satu SMP sekalipun. Untuk hasil lebih lengkap terdapat didalam lampiran.

Tabel 4.49 Hasil evaluasi *Readability* dengan data kualitas udara

No	Kode Dataset	Flesch Reading Ease Score (Datayze)	Flesch Reading Ease Score (Readability Formula)
1	AQ_NH	70.76	69.4
2	AQ_WH	74.96	72.4
3	AQ_WHM	70.78	69.8

No	Kode Dataset	Flesch Reading Ease Score (Datayze)	Flesch Reading Ease Score (Readability Formula)
	Rata-rata	72.16	70.53
	Rata-rata Keseluruhan		71.35

2. Analisis aspek *Computation Time*

Hasil perhitungan *Computation Time* pada data klimatologi terdapat pada tabel 4.50. Terlihat bahwa rata-rata *running time* sistem ini adalah 5.40 detik. Untuk hasil lebih lengkapnya terdapat pada lampiran.

Tabel 4.50 Hasil evaluasi *Computation Time* dengan data kualitas udara

No	Kode Dataset	Running Time (s)
1	CE_NH	5.19
2	CE_WH	5.91
3	CE_WHM	5.11
	Rata-rata	5.40

3. Analisis *Unspecific Handling*

Pada analisis *Unspecific Handling* ini akan dilihat bagaimana respon sistem dalam menangani berbagai jenis data masukan, apakah sistem akan tetap bekerja meskipun data yang menjadi masukan berbeda-beda jenisnya. Seperti pada tabel 4.51 terlihat bahwa sistem tetap bejalan dan memproses data masukan, meskipun data klimatologi yang dimasukan tidak memiliki header, tidak memiliki cara penginterpretasian, tidak memiliki *data description*, dan data memiliki parameter dengan tipe *categorical* berjumlah dua.

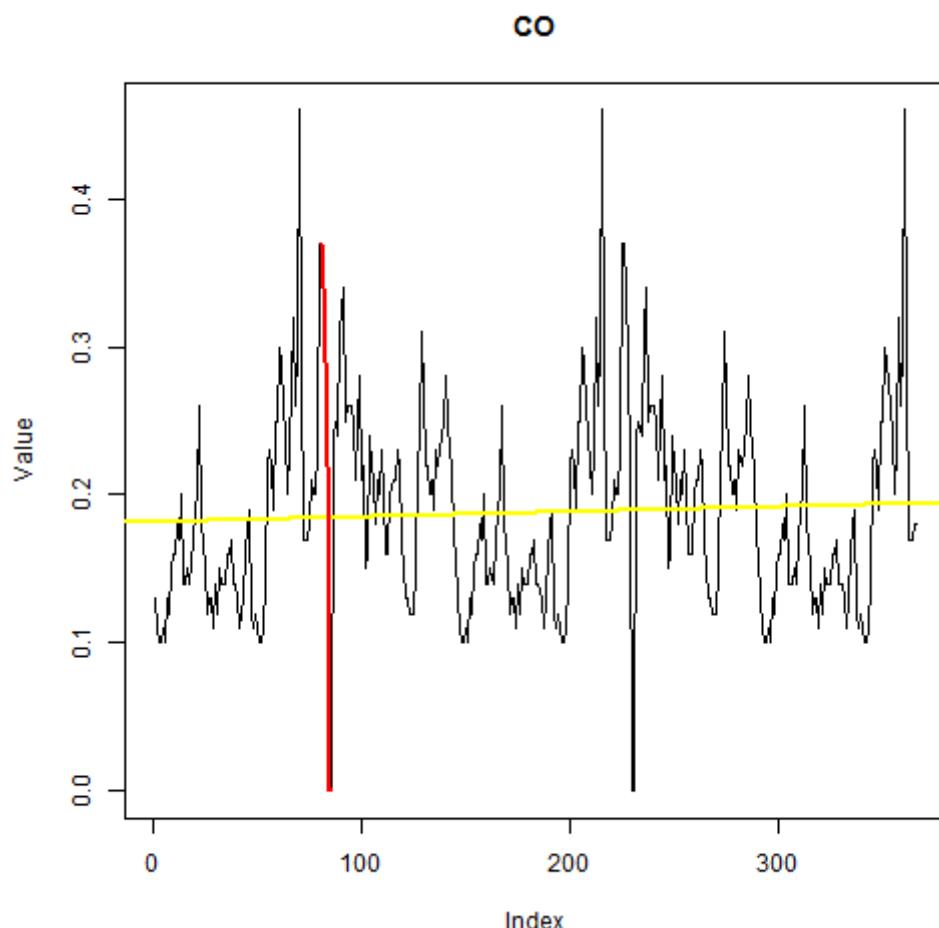
Tabel 4.51 Hasil evaluasi *Unspecific Handling*

No	Kode Dataset	Header	Config Rule	Rule	Numerical	Categorical	Config Alternate	Output
1	CL_NH	No	No	Unspecific	Yes	No	No	Yes
2	CL_WH	Yes	No	Unspecific	Yes	No	No	Yes
3	CL_WHM	Yes	Yes	Unspecific, Crisp	Yes	Yes, 2	No	Yes

4. Analisis *Representative Text*

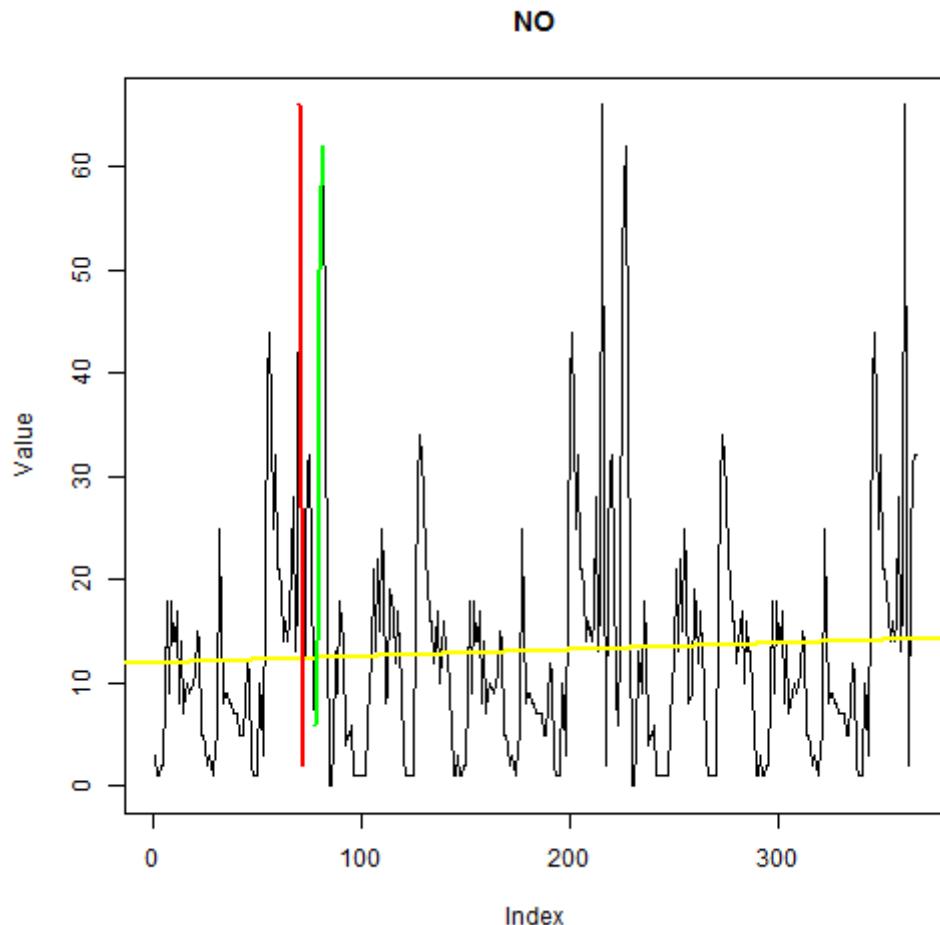
Analisis *Representative Text* yang dilakukan hampir sama seperti pada analisis hasil eksperimen data kurs sebelumnya. Dimana tidak semua parameter pada data kualitas udara akan dibahas pada analisis *Representative Text* ini. Parameter yang akan dianalisis, diantaranya CO, NO dan PM10, dikarenakan beberapa parameter memiliki *event* yang sama.

Analisis *Representative Text* ini dilakukan dengan cara pembuatan grafik garis dengan menggunakan indikator warna untuk merepresentasikan *event* yang terjadi pada parameter tersebut. Warna hitam pada grafik merepresentasikan data suatu parameter, warna kuning menandakan *trend* dari parameter tersebut, warna hijau menandakan kenaikan ekstrem, warna merah menandakan penurunan ekstrem, warna biru menandakan *repeated event*, sedangkan lingkaran hijau menandakan bahwa data terakhir merupakan data tertinggi, dan lingkaran merah menandakan bahwa data terakhir merupakan data terendah pada parameter tersebut. Untuk hasil keseluruhan plot dapat dilihat pada lampiran.



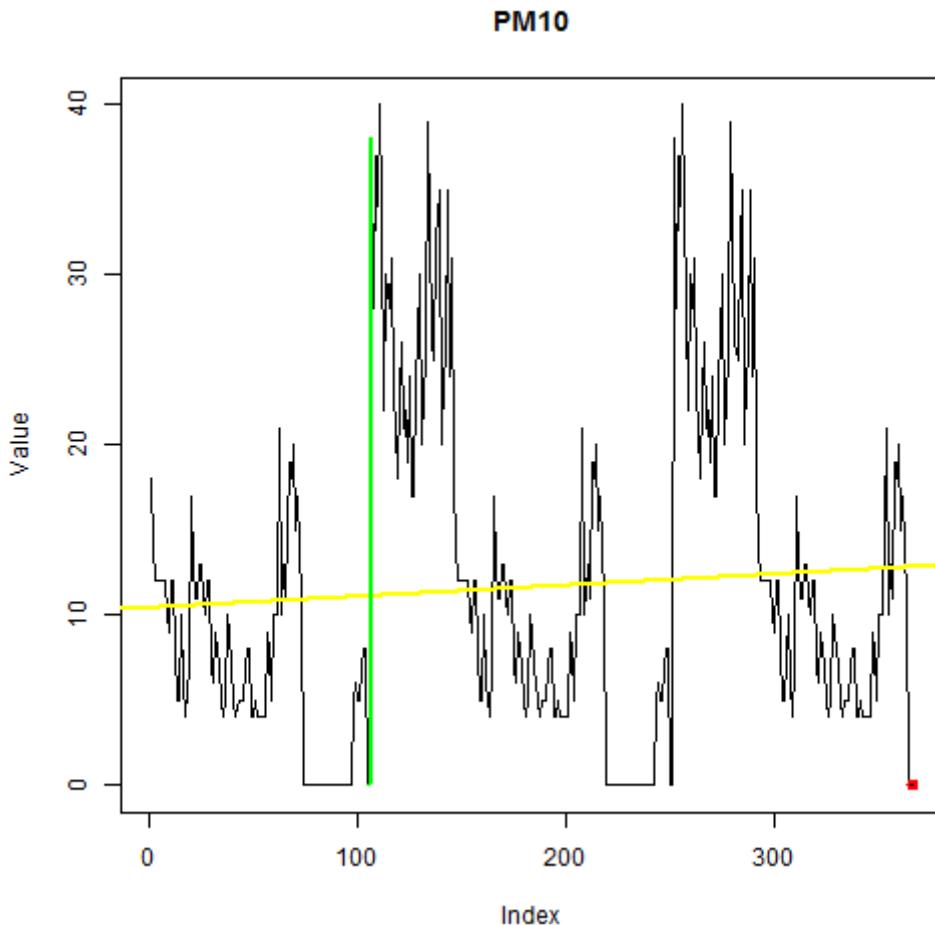
Gambar 4.85 Hasil plot *Representative Text* untuk parameter CO

Dalam teks keluaran disebutkan bahwa *trend* parameter CO adalah konstan hal tersebut dibuktikan dengan *linear model* atau garis berwarna kuning pada gambar 4.85 yang menaik namun tidak melebihi *minimum threshold* sebesar 10%. Selain itu pada teks keluaran disebutkan bahwa pada parameter CO terjadi penurunan yang ekstrem sebesar 0.37 poin, fluktuasi ini dapat kita lihat pada gambar 4.85 dimana terdapat kuning yang merepresentasikan penurunan yang ekstrem. Jumlah kenaikan dan penurunan terjadi pada kisaran nilai 0 hingga 0.4, sehingga pesan penurunan ekstrem pada teks dapat dibuktikan karena nilai kenaikan dan penurunannya sebesar 0.37.



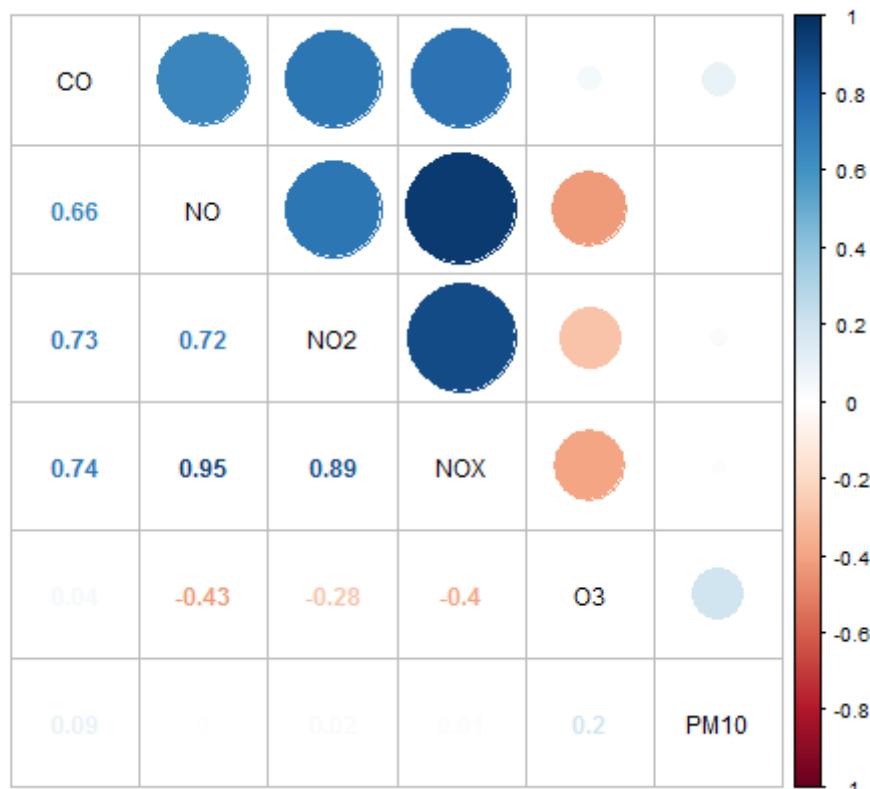
Gambar 4.86 Plot *Representative Text* untuk parameter NO

Pada teks keluaran, dikatakan bahwa *trend* parameter JPY adalah konstan hal tersebut dibuktikan dengan *linear model* atau garis berwarna kuning pada gambar 4.86 yang menaik namun tidak melebihi *minimum threshold* sebesar 10%, pada teks keluaran disebutkan bahwa pada parameter NO terjadi fluktuasi yang ekstrem dimana terjadi kenaikan sebesar 56 poin dan penurunan sebesar 64 poin, fluktuasi ini dapat kita lihat pada gambar 4.86 dimana terdapat warna hijau merepresentasikan kenaikan yang ekstrem dan kuning yang merepresentasikan penurunan yang ekstrem. Selain itu pada teks keluaran dikatakan bahwa data terakhir parameter CO berada pada kondisi *medium*, yang diinterpretasikan menggunakan cara *unspecific*, hal ini sesuai dengan hasil pada gambar 4.86 dimana pada grafik tersebut data terakhir berada pada rentang nilai 30 dan 40 yang berada pada wilayah tengah grafik,



Gambar 4.87 Hasil plot *Representative Text* untuk parameter PM10

Trend PM10 pada teks keluaran dikatakan sebagai satu-satunya parameter dengan *trend* yang menaik sesuai dengan *Linear Model* pada gambar 4.87 yang berwarna kuning. Selain itu pada teks keluaran disebutkan bahwa pada parameter PM10 terjadi kenaikan yang ekstrem dimana terjadi kenaikan sebesar 20.05 poin kenaikan ini dapat kita lihat pada gambar 4.87 dimana terdapat warna hijau merepresentasikan kenaikan yang ekstrem. Selain itu dikatakan bahwa PM10 mencapai nilai terendah pada pulan ini, hal ini dibuktikan dengan lingkaran merah pada data terakhir yang berada pada dasar grafik .



Gambar 4.88 Hasil plot *Representative Text* untuk korelasi parameter data kualitas udara

Pada teks keluaran dikatakan bahwa NOX memiliki dampak paling besar pada parameter lain, hal ini dibuktikan dengan menghitung nilai rata-rata hasil korelasi parameter NOX pada gambar 4.88 dimana secara perhitungan kasar parameter NOX memiliki nilai rata-rata korelasi parameter paling tinggi dibanding parameter lainnya. Selain itu pada teks keluaran dikatakan bahwa penurunan pada nilai NO₂ berdampak pada penurunan CO dan NO begitu juga untuk kenaikan, hal ini sejalan dengan hasil korelasi parameter NO₂-CO sebesar 0.73 dan korelasi NO₂-NO sebesar 0.72 yang tergolong pada kategori *strong relationship*. Dikatakan juga pada teks keluaran bahwa penurunan yang terjadi pada NOX berdampak pada menurunnya nilai parameter CO, NO, dan NO₂, hal ini dibuktikan dengan nilai korelasi NOX-CO sebesar 0.7, lalu nilai korelasi

NOX-NO sebesar 0.95, dan nilai korelasi antara NOX dan NO2 sebesar 0.89 yang jika diinterpretasikan hasilnya tergolong pada kategori *strong, almost perfect, dan very strong relationship.*

4.7.4. Hasil dan Pembahasan Hasil Eksperimen Data Partikel Udara

Eksperimen menggunakan data partikel udara udara pada situs web <http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data>, selama dua tahun pada periode 2010-2011 dengan interval data adalah per jam. Dimana data ini memiliki keunikan tersendiri yaitu adanya parameter *categorical* pada data masukan. Data ini memiliki keunikan tersendiri dibanding dataset lainnya, dimana data ini memiliki parameter *categorical* dan beberapa parameter yang konstan, selain itu parameter lain memiliki fluktuasi yang cukup ekstrem. Jumlah data pada dataset ini adalah sebanyak 17 ribu lebih data sehingga dataset ini menarik.

4.7.4.1. Hasil Eksperimen Data Partikel Udara Tanpa Header

Eksperimen ke-sepuluh dilakukan dengan menggunakan data klimatologi pada tabel 4.4 dengan *header* yang dihilangkan terlebih dahulu atau *Beijing PM25 no header* (BPM_NH). Cuplikan dari data tersebut dapat dilihat pada tabel 4.52. sehingga didapatkan hasil seperti pada gambar 4.89.

Tabel 4.52 Cuplikan data kualitas udara tanpa menggunakan *header* (BPM_NH)

01/01/2010 00:00	138	-21	-11	1021	NW	1.79	0	0
01/01/2010 01:00	125	-21	-12	1020	NW	4.92	0	0
01/01/2010 02:00	249	-21	-11	1019	NW	6.71	0	0
01/01/2010 03:00	210	-21	-14	1019	NW	9.84	0	0
01/01/2010 04:00	98	-20	-12	1018	NW	12.97	0	0
...
01/31/2010 19:00	54	-15	-1	1025	NE	2.68	0	0
01/31/2010 20:00	71	-13	-1	1025	cv	0.89	0	0
01/31/2010 21:00	129	-8	-1	1026	SE	5.81	0	0
01/31/2010 22:00	145	-7	-2	1027	SE	10.73	0	0
01/31/2010 23:00	101	-7	-2	1027	SE	13.86	0	0

Beijing PM25 NEWS

According to the hourly Beijing air particle data between 01/01/2010 00:00 to 12/31/2011 23:00, with parameters: V2, V3, V4, V5, V6, V7, V8, and V9, it can be seen that V3, V4 trend is increased and the rest is constant. V8, and V9 parameters are equal with yesterday's data. V4, and V7 parameters are lower than yesterday's data, but the rest parameters are higher than yesterday's data. There were some repeating value more than 1752 hours: V8 stayed constant at very low during 18 Mar 2010 11:00 to 9 Feb 2011 22:00, 27 Feb 2011 13:00 to 2 Dec 2011 06:00. V2 fluctuated dramatically (increased 929 points and decreased 969 points), V7 fluctuated dramatically (increased 584.71 points and decreased 583.81 points), V8 fluctuated dramatically (increased 27 points and decreased 27 points), and V9 fluctuated sharply (increased 36 points and decreased 36 points). There are a duplicate V6 data pattern in the last 6 hours (31 Dec 2011 17:00 to 31 Dec 2011 23:00) with data patterns from 2 Feb 2010 01:00 to 2 Feb 2010 07:00, 23 Nov 2010 04:00 to 23 Nov 2010 10:00, and 25 Sep 2011 04:00 to 25 Sep 2011 10:00. V3 appears to have a highest impact to all variable with moderate relationship in average. An increase in V4 resulted an increase in V3. A decrease in V5 resulted a growth in V3, and V4.

This hour data illustrate that: V2, V3, and V4 in low condition. V5 in high condition. V7, V8, and V9 in very low condition. V8, and V9 reached their lowest value on this hour.

Based on prediction result, it's projected that V2, V3, and V4 will steady at low. V5 will stay stable at high. V7, V8, and V9 will steady at very low.

Gambar 4.89 Hasil eksperimen ke-sepuluh menggunakan data partikel udara tanpa menggunakan *header* (BPM_NH)

Hasil keluaran eksperimen ke-sepuluh pada gambar 4.89 dapat dilihat bahwa ketika data masukan tidak dikenali oleh sistem dikarenakan tidak adanya *header* pada data tersebut, sistem akan tetap memproses data tersebut lalu memberi nama setiap parameter dengan *header* V2, V3, V4 dan seterusnya. Dimana V2 merupakan parameter ke-dua mewakili . PM2.5, V3 mewakili DEWP, V4 mewakili TEMP, V5, mewakili PRES, V6 mewakili CBWD, V7 mewakili LWS, V8 mewakili IS, dan V9 mewakili IR.

Dimana dalam informasi pada gambar 4.89 dikatakan bahwa *trend* dari parameter ke-tiga atau V3 dan V4 adalah menurun, dan sisanya adalah konstan. Parameter V3, V4 dan V5 memiliki nilai yang lebih kecil dibandingkan dengan nilai pada minggu lalu, dan sisanya lebih besar. Terjadi *repeated event* pada parameter V8 dimana parameter tersebut berada pada keadaan *very low* selama lebih dari 1752 jam, dengan dua kali perulangan terjadi yaitu pada periode 18 Mar 2010 11:00 sampai 9 Feb 2011 22:00, dan pada periode 27 Feb 2011 13:00 sampai 2 Dec 2011 06:00. Terdapat beberapa *extreme event*, dimana parameter V2, V7, V8, dan V9 mengalami fluktuasi yang cukup ekstrem. Dikatakan bahwa V3 memiliki nilai rata-rata korelasi tetinggi dengan interpretasi korelasi *moderate*. Lalu terdapat beberapa parameter yang memiliki dampak yang kuat pada parameter lainnya, misalnya pada parameter V4 terjadi kenaikan maka hal tersebut berdampak pada meningkatnya nilai parameter V3, lalu jika penurunan terjadi pada parameter V5, hal ini akan berdampak juga pada menaiknya nilai parameter V3 dan V4.

Pada paragraf ke-dua terlihat deskripsi kondisi setiap parameter, dan terdapat *event* yang menunjukan bahwa parameter V8 dan V9 mencapai nilai terendah pada jam tersebut. Pada paragraf ke-tiga disimpulkan bahwa semua parameter tidak mengalami perubahan yang berarti dan akan tetap berada pada kondisi seperti pada paragraf ke-dua.

4.7.4.2. Hasil Eksperimen Data Partikel Udara Dengan *Header*

Beijing PM25 NEWS

According to the hourly Beijing air particle data between 01/01/2010 00:00 to 12/31/2011 23:00, with parameters: pm2.5, DEWP, TEMP, PRES, cbwd, lws, ls, and lr, it showed that DEWP, TEMP trend is increased and the rest is constant. ls, and lr parameters are equal with yesterday's data. TEMP, and lws parameters are lower than yesterday's data, but the rest parameters are higher than yesterday's data. There were some repeating value more than 1752 hours: ls stayed constant at very low during 18 Mar 2010 11:00 to 9 Feb 2011 22:00, 27 Feb 2011 13:00 to 2 Dec 2011 06:00. pm2.5 fluctuated extremely (increased 929 points and decreased 969 points), lws fluctuated sharply (increased 584.71 points and decreased 583.81 points), ls fluctuated extremely (increased 27 points and decreased 27 points), and lr fluctuated extremely (increased 36 points and decreased 36 points). There are an identical cbwd data pattern in the last 6 hours (31 Dec 2011 17:00 to 31 Dec 2011 23:00) with data patterns from 2 Feb 2010 01:00 to 2 Feb 2010 07:00, 23 Nov 2010 04:00 to 23 Nov 2010 10:00, and 25 Sep 2011 04:00 to 25 Sep 2011 10:00. DEWP appears to have a highest impact to all variable with moderate relationship in average. An increase in TEMP resulted an increase in DEWP. A decrease in PRES resulted an increase in DEWP, and TEMP.

This hour data reveal that: pm2.5, DEWP, and TEMP in low condition. PRES in high condition. lws, ls, and lr in very low condition. ls, and lr reached their lowest value on this hour.

Regarding to the prediction result, it's predicted that pm2.5, DEWP, and TEMP will keep stable at low. PRES will keep stable at high. lws, ls, and lr will keep stable at very low.

Gambar 4.90 Hasil eksperimen ke-sebelas menggunakan data partikel udara dengan menggunakan *header* (BPM_WH)

Untuk eksperimen ke-sebelas pada gambar 4.90 digunakan data partikel udara dimana data tersebut diproses dengan menggunakan *header* aslinya namun untuk penginterpretasiannya masih digunakan cara *unspecific*. Pada eksperimen ini terdapat beberapa perbedaan yang mencolok yakni terletak pada penyebutan parameter, dimana pada eksperimen ke-enam penyebutan parameter menggunakan nama *header* secara default, sedangkan pada eksperimen ke-tujuh penyebutan parameter menggunakan nama *header* data aslinya seperti pada tabel 4.3 dimana V2 diinterpretasikan menjadi pm2.5, V3 diinterpretasikan menjadi DEWP, V4 diinterpretasikan menjadi TEMP, V5 diinterpretasikan menjadi PRES, dan seterusnya. Selain itu perbedaan lain terletak pada frasa hasil dari proses *Reffering Expression Generation* dalam mengungkapkan kondisi kenaikan dan penurunan seperti *increase*, *growth*, *rise*, *decline*, *decrease*, *decay* dan frasa untuk mengungkapkan *extreme event* seperti *extremely*, *dramatically*, *significantly* dan *sharply*.

4.7.4.3. Hasil Eksperimen Data Partikel Udara Dengan Kustomisasi *Corpus*

Untuk eksperimen ke-dua belas ini digunakan data partikel udara yang sama seperti pada eksperimen ke-sebelas, namun pada eksperimen ke-dua belas ini digunakan kustomisasi *corpus*. Dimana kustomisasi ini didefinisikan pada *data description* yang dapat dilihat pada tabel 4.53.

Tabel 4.53 Kustomisasi *data description* pada eksperimen ke-enam

ColName	Type	Rule	Alternate
TEMP	Numerical	Fuzzy	Temperature

Sehingga didapatkan hasil seperti pada gambar 4.91 dimana pada gambar tersebut terlihat, nama parameter TEMP pada teks keluaran berubah menjadi Temperature, hal ini disebabkan karena adanya kustomisasi atribut *alternate* pada *data description*. Selain itu juga perbedaan yang mencolok terdapat pada cara penginterpretasian parameter TEMP, pada eksperimen ini digunakan penginterpretasian temperatur menggunakan *corpus* pada penelitian DWP (Putra *et al.*, 2017). Akibatnya, hasil interpretasi parameter TEMP yang asalnya *low* kini berubah menjadi *very cold condition*.

Beijing PM25 NEWS

Regarding to the hourly Beijing air particle data (01/01/2010 00:00 - 12/31/2011 23:00), with parameters: pm2.5, DEWP, Temperature, PRES, cbwd, lws, ls, and lr, it can be seen that DEWP, Temperature trend is increased and the rest is constant. ls, and lr parameters are equal with yesterday's data. Temperature, and lws parameters are lower than yesterday's data, but the rest parameters are higher than yesterday's data. There were some repeating value more than 1752 hours: ls stayed constant at very low during 18 Mar 2010 11:00 to 9 Feb 2011 22:00, 27 Feb 2011 13:00 to 2 Dec 2011 06:00. pm2.5 fluctuated dramatically (increased 929 points and decreased 969 points), lws fluctuated significantly (increased 584.71 points and decreased 583.81 points), ls fluctuated dramatically (increased 27 points and decreased 27 points), and lr fluctuated significantly (increased 36 points and decreased 36 points). There are a equivalent cbwd data pattern in the last 6 hours (31 Dec 2011 17:00 to 31 Dec 2011 23:00) with data patterns from 2 Feb 2010 01:00 to 2 Feb 2010 07:00, 23 Nov 2010 04:00 to 23 Nov 2010 10:00, and 25 Sep 2011 04:00 to 25 Sep 2011 10:00. DEWP appears to have a highest impact to all variable with moderate relationship in average. A rise in Temperature causes an attendant increase in DEWP. As a result of the decay in PRES, it can be seen that DEWP, and Temperature is increasing.

This hour data illustrate that: pm2.5, and DEWP in low condition. Temperature in very cold condition. PRES in high condition. lws, ls, and lr in very low condition. ls, and lr reached their lowest value on this hour.

According to the prediction result, it's forecasted that pm2.5, and DEWP will keep stable at low. Temperature will keep stable at very cold. PRES will steady at high. lws, ls, and lr will steady at very low.

Gambar 4.91 Hasil eksperimen ke-dua belas menggunakan data partikel udara dengan kustomisasi *corpus* (BPM_WHM)

4.7.4.4. Analisis Hasil Eksperimen Data Partikel Udara

Analisis hasil eksperimen dengan menggunakan Eksperimen menggunakan data partikel udara selama dua tahun pada periode 2010-2011 dengan interval data per jam adalah sebagai berikut:

1. Analisis aspek *Readability*

Evaluasi dilakukan dengan menggunakan metode *Flesch Reading Ease* menggunakan *Readability Analyzer* pada situs www.datayze.com dan aplikasi *Automatic Readability Checker* pada situs www.readabilityformulas.com. Hasil pengujian aspek *Readability* ini dapat dilihat pada tabel 4.54. Dimana rata-rata yang didapatkan dari proses analisis ini adalah 76.27 yang kemudian berdasarkan *Flesch Reading Ease Score* pada tabel 4.35 didapatkan bahwa informasi dari teks keluaran tergolong pada kategori *plain english* yang berarti teks keluaran mudah dipahami bahkan oleh siswa usia remaja sekalipun. Untuk hasil lebih lengkap terdapat didalam lampiran.

Tabel 4.54 Hasil evaluasi *Readability* dengan data partikel udara

No	Kode Dataset	<i>Flesch Reading Ease Score (Datayze)</i>	<i>Flesch Reading Ease Score (Readability Formula)</i>
1	BPM_NH	77.95	75.6
2	BPM_WH	81.92	79.6
3	BPM_WHM	70.78	71.8
Rata-rata		76.88	75.66
Rata-rata Keseluruhan		76.27	

2. Analisis aspek *Computation Time*

Hasil perhitungan *Computation Time* pada data klimatologi terdapat pada tabel 4.55. Terlihat bahwa rata-rata *running time* sistem ini adalah 44.79 detik. Untuk hasil lebih lengkapnya terdapat pada lampiran.

Tabel 4.55 Hasil evaluasi *Computation Time* dengan data kualitas udara

No	Kode Dataset	Running Time (s)
1	BPM_NH	45.02
2	BPM_WH	44.64
3	BPM_WHM	45.25
Rata-rata		44.79

3. Analisis *Unspecific Handling*

Pada analisis *Unspecific Handling* ini akan dilihat bagaimana respon sistem dalam menangani berbagai jenis data masukan, apakah sistem akan tetap bekerja meskipun data yang menjadi masukan berbeda-beda jenisnya. Seperti pada tabel 4.56 terlihat bahwa sistem tetap bejalan dan memproses data masukan, meskipun data klimatologi yang dimasukan tidak memiliki header, tidak memiliki cara penginterpretasian, tidak memiliki *data description*, dan data memiliki parameter dengan tipe *categorical* berjumlah 1. Hal ini menandakan bahwa dengan inputan yang memiliki tipe *categorical* sekalipun, sistem akan tetap bekerja tanpa menghasilkan *error*.

Tabel 4.56 Hasil evaluasi *Unspecific Handling*

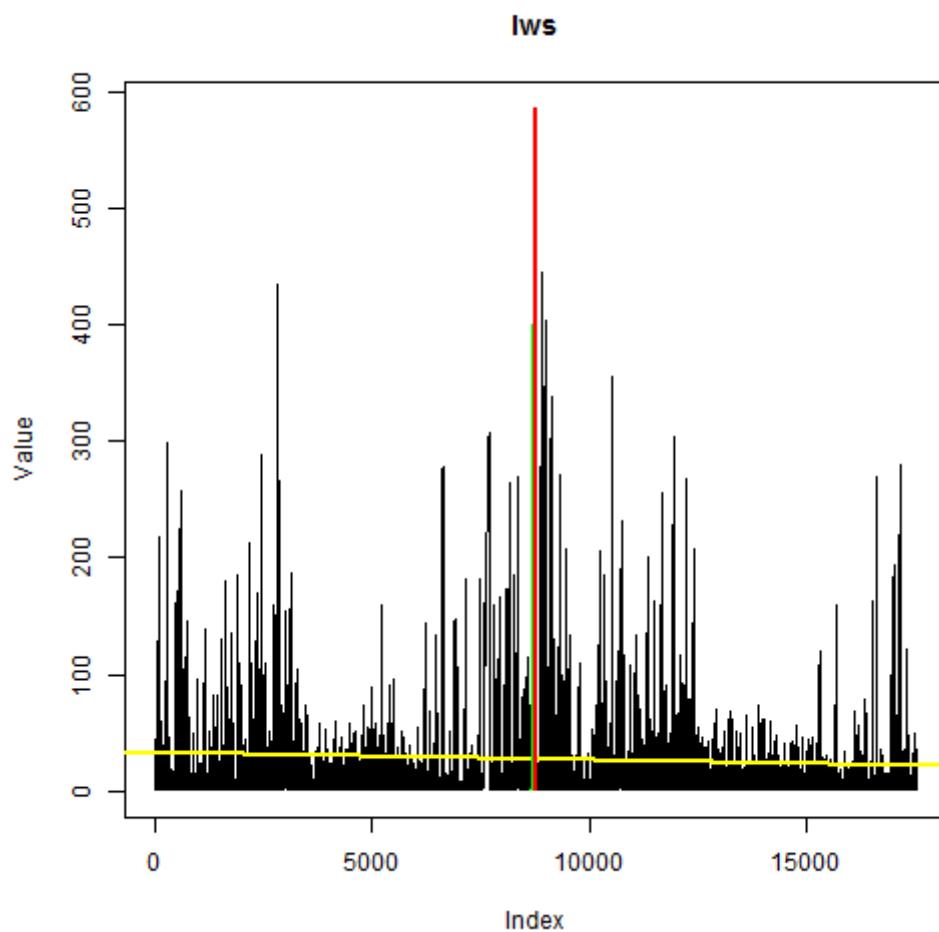
No	Kode Dataset	Header	Config Rule	Rule	Numerical	Categorical	Config Alternate	Output
1	BPM_NH	No	No	Unspecific	Yes	Yes, 1	No	Yes
2	BPM_WH	Yes	No	Unspecific	Yes	Yes, 1	No	Yes
3	BPM_WHM	Yes	Yes	Unspecific, Fuzzy	Yes	Yes, 1	Yes	Yes

4. Analisis *Representative Text*

Analisis *Representative Text* yang dilakukan berbeda seperti pada analisis hasil eksperimen data kurs sebelumnya. Dimana analisis ini dilakukan pada semua parameter yang ada pada data. Tidak semua parameter pada data kualitas udara akan dibahas pada analisis *Representative Text* ini. Parameter yang akan dianalisis, diantaranya LWS, IS dan IR, dikarenakan beberapa parameter memiliki *event* yang sama.

Analisis *Representative Text* ini dilakukan dengan cara pembuatan grafik garis dengan menggunakan indikator warna untuk merepresentasikan

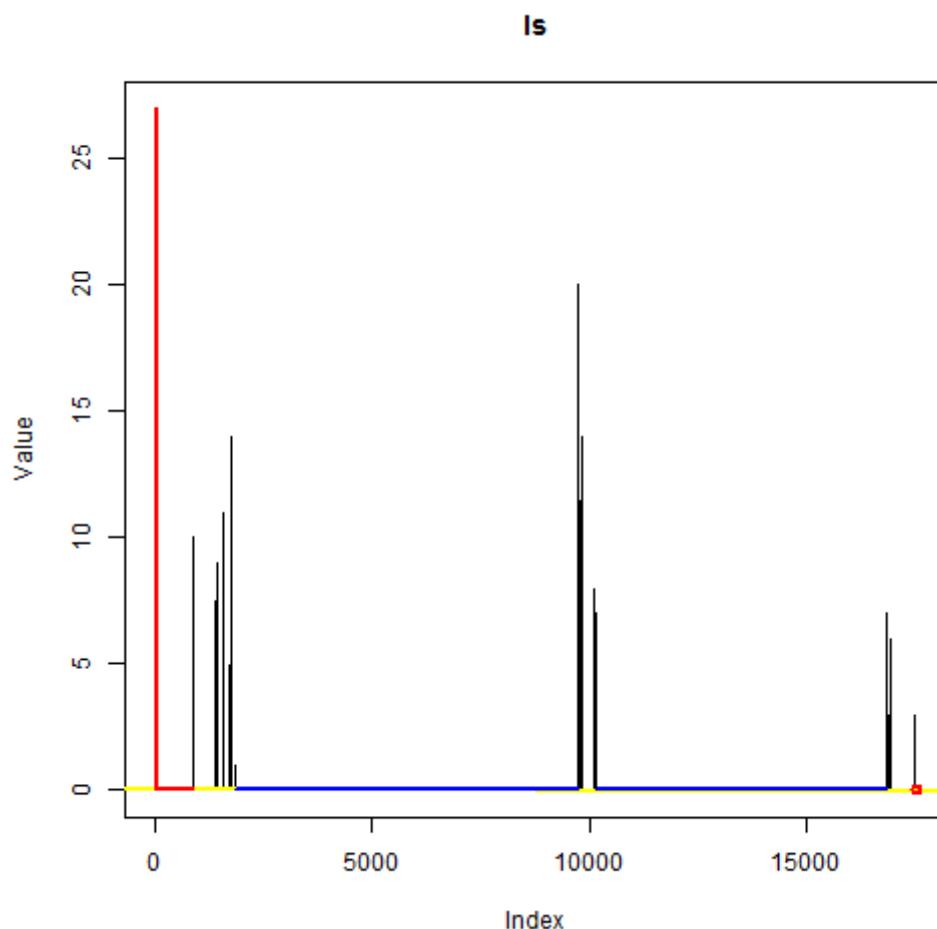
event yang terjadi pada parameter tersebut. Warna hitam pada grafik merepresentasikan data suatu parameter, warna kuning menandakan *trend* dari parameter tersebut, warna hijau menandakan kenaikan ekstrem, warna merah menandakan penurunan ekstrem, warna biru menandakan *repeated event*, sedangkan lingkaran hijau menandakan bahwa data terakhir merupakan data tertinggi, dan lingkaran merah menandakan bahwa data terakhir merupakan data terendah pada parameter tersebut. Untuk hasil keseluruhan plot dapat dilihat pada lampiran.



Gambar 4.92 Hasil plot *Representative Text* untuk parameter LWS

Dalam teks keluaran, disebutkan bahwa *trend* parameter LWS atau *Cumulated wind speed* adalah konstan hal tersebut dibuktikan dengan *linear model* atau garis berwarna kuning pada gambar 4.92 yang sedikit menaik namun tidak melebihi *minimum treshold* sebesar 10%. Selain itu pada teks

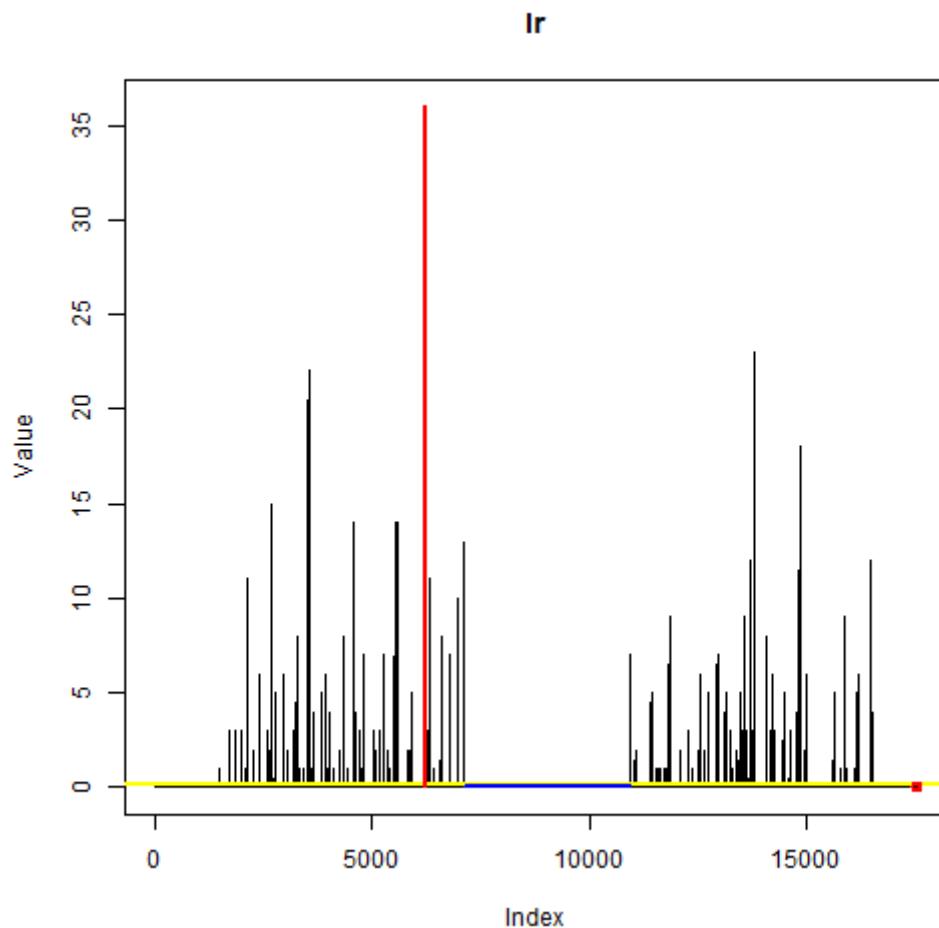
keluaran disebutkan bahwa pada parameter PM25 terjadi fluktuasi yang ekstrem dimana terjadi kenaikan sebesar 584.71 poin dan penurunan sebesar 583.81 poin, fluktuasi ini dapat kita lihat pada gambar 4.92 dimana terdapat garis hijau yang merepresentasikan kenaikan dan garis merah yang menunjukkan penurunan yang ekstrem.



Gambar 4.93 Plot *Representative Text* untuk parameter IS

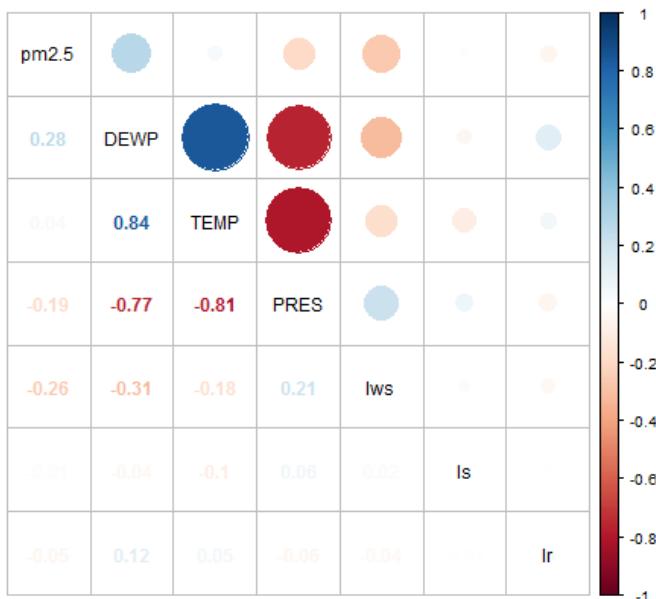
Pada teks keluaran, dikatakan bahwa *trend* parameter IS atau *Cumulated hours of snow* adalah konstan hal tersebut dibuktikan dengan *linear model* atau garis berwarna kuning pada gambar 4.93. Pada teks keluaran disebutkan bahwa pada parameter IS terjadi fluktuasi yang ekstrem dimana terjadi kenaikan sebesar 27 poin dan penurunan sebesar 27 poin, fluktuasi ini dapat kita lihat pada gambar 4.93 dimana terdapat warna hijau merepresentasikan kenaikan yang ekstrem dan kuning yang

merepresentasikan penurunan yang ekstrem, dalam kasus ini warna hijau tidak muncul karena tertimpa oleh warna merah disebabkan padatnya data, untuk membuktikan kenaikan tersebut dapat dilihat pada data secara langsung. Terjadi *repeated event* sebanyak dua kali yang direpresentasikan dengan garis berwarna biru dimana pada teks keluaran disebutkan bahwa parameter IS berada pada kondisi *very low* selama periode tersebut. Selain itu pada teks keluaran dikatakan bahwa data terakhir parameter IS berada pada kondisi *very low*, yang diinterpretasikan menggunakan cara *unspecific*, hal ini sesuai dengan hasil pada gambar 4.93 dimana pada grafik tersebut data terakhir berada terletak pada dasar grafik, yang membuat parameter IS mencapai *lowest value* pada data jam terakhir.



Gambar 4.94 Hasil plot *Representative Text* untuk parameter IR

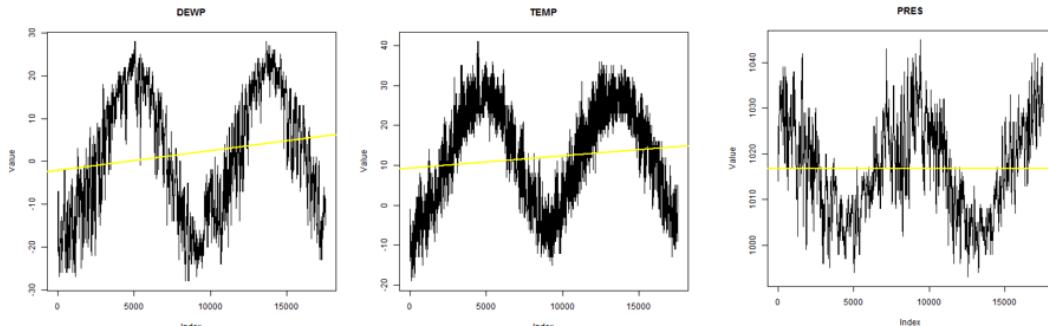
Trend IR atau *Cumulated hours of snow* pada teks keluaran dikatakan konstan sesuai dengan *Linear Model* pada gambar 4.94 yang berwarna kuning. Pada teks keluaran disebutkan bahwa pada parameter IR terjadi fluktuasi yang ekstrem dimana terjadi kenaikan sebesar 36 poin dan penurunan sebesar 36 poin, fluktuasi ini dapat kita lihat pada gambar 4.94 dimana terdapat warna hijau merepresentasikan kenaikan yang ekstrem dan kuning yang merepresentasikan penurunan yang ekstrem, dalam kasus ini warna hijau tidak muncul karena tertimpa oleh warna merah disebabkan padatnya data, untuk membuktikan kenaikan tersebut dapat dilihat pada data secara langsung. Terjadi *repeated event* sebanyak satu kali yang direpresentasikan dengan garis berwarna biru dimana pada teks keluaran disebutkan bahwa parameter IR berada pada kondisi *very low* selama periode tersebut. Namun dikarenakan sinyal *repeated event* pada parameter IR tidak lebih banyak dibandingkan parameter IS, maka pesan yang ditampilkan hanya untuk parameter IS. Selain itu pada teks keluaran dikatakan bahwa data terakhir parameter IR berada pada kondisi *very low*, yang diinterpretasikan menggunakan cara *unspecific*, hal ini sesuai dengan hasil pada gambar 4.94 dimana pada grafik tersebut data terakhir berada terletak pada dasar grafik, yang membuat parameter IR mencapai *lowest value* pada data jam terakhir.



Gambar 4.95 Hasil plot *Representative Text* untuk korelasi parameter data kualitas udara

Pada teks keluaran dikatakan bahwa DEWP atau *dew point* memiliki dampak paling besar pada parameter lain, hal ini dibuktikan dengan menghitung nilai rata-rata hasil korelasi parameter DEWP pada gambar 4.95 dimana secara perhitungan kasar parameter DEWP memiliki nilai rata-rata korelasi paramter paling tinggi dibanding parameter lainnya. Selain itu pada teks keluaran dikatakan bahwa kenaikan pada nilai TEMP atau *temperature* berdampak pada kenaikan DEWP begitu juga untuk penurunan, hal ini sejalan dengan hasil korelasi parameter TEMP-DEWP sebesar 0 yang tergolong pada kategori *very strong relationship*. Dikatakan juga pada teks keluaran bahwa penurunan yang terjadi pada PRES berdampak pada menaiknya nilai parameter DEWP, dan TEMP, hal ini dibuktikan dengan nilai korelasi PRES- DEWP sebesar -0.77, lalu nilai korealasi PRES -TEMP sebesar 0.81, dan nilai korelasi antara NOX dan NO2 sebesar 0.89 yang jika diinterpretasikan hasilnya tergolong pada

kategori *strong relationship*. Untuk lebih lengkapnya korelasi parameter ini dapat dilihat dengan hasil plot pada gambar 4.96.



Gambar 4.96 Hasil plot Representative Text untuk parameter DEWP, TEMP dan PRES

4.6. Perbandingan dengan Penelitian Sebelumnya

Setelah didapatkan hasil eksperimen dan dilakukan analisis pada hasil eksperimen, langkah selanjutnya adalah membandingkan hasil dari keluaran sistem dengan hasil dari penelitian-penelitian sebelumnya. Penjelasan perbandingan penelitian ini dengan penelitian sebelumnya dapat dilihat pada tabel 4.57.

Tabel 4.57 Hasil perbandingan dengan penelitian sebelumnya

Penelitian	Output
UNG Output #Without Header	<p>According to the daily data (07/06/2016 00:00 - 07/06/2017 00:00), with parameters: V2, V3, V4, V5, and V6. It is clear that, V3 trend is decreased and V6 trend is constant but the rest is increased. V3, and V5 parameters are higher than last week's data, but the rest parameters are lower than last week's data. There were some repeating value more than 36 days: V6 stayed constant at very low during 6 Jul - 18 Aug 2016. V2 fluctuated significantly (increased 90.9 points and decreased 87.6 points), V4 fluctuated significantly (increased 20.05 points and decreased 27.03 points), V5 fluctuated significantly (increased 270 points and decreased 270 points), and V6 fluctuated significantly (increased 67.2 points and decreased 64.1 points). For the past 7 days , no equivalent patterns were found for each categorical parameters. V6 appears to have a highest impact to all variable with moderate relationship in average.</p> <p>Today data indicate that: V2 in medium condition. V3 in high condition. V4 in low condition. V5 in very high condition. V6 in very low condition. V6 reached their lowest value on this day. V5 reached their highest value on this day.</p> <p>According to the prediction result, it's predicted that V2 will move to medium. V3 will still stable at high. V4 will still stable at low. V5 will begin to turn to very high. V6 will stay constant at very low.</p>
UNG Output #With Header	Regarding to the daily data, from 07/06/2016 00:00 to 07/06/2017 00:00, with parameters: Cloud Coverage, Temperature, Wind Speed, Wind Direction, and Rainfall. It represented that, Temperature trend is decreased and Rainfall trend is constant but the rest is increased.

Penelitian	Output
	<p>Temperature, and Wind Direction parameters are higher than last week's data, but the rest parameters are lower than last week's data. There were some repeating value more than 36 days: Rainfall stayed constant at no rain during 6 Jul - 18 Aug 2016. Cloud Coverage fluctuated significantly (increased 90.9 points and decreased 87.6 points), Wind Speed fluctuated significantly (increased 20.05 points and decreased 27.03 points), Wind Direction fluctuated significantly (increased 270 points and decreased 270 points), and Rainfall fluctuated significantly (increased 67.2 points and decreased 64.1 points). For the past 7 days , no equivalent patterns were found for each categorical parameters. Rainfall appears to have a highest impact to all variable with moderate relationship in average.</p> <p>Today data represent that: Cloud Coverage in mostly cloudy condition. Temperature in warm condition. Wind Speed in light Breeze condition. Wind Direction in North West condition. Rainfall in no rain condition. Rainfall reached their lowest value on this day. Wind Direction reached their highest value on this day.</p> <p>From the prediction result, it's expected that tomorrow sky will be light rain although it's covered by partly cloudy sky. Favored by temperature which decreased to warm. Cloud Coverage will change progressively to partly cloudy. Temperature will still stable at warm. Wind Speed will stay constant at light Breeze. Wind Direction will moderately turn to West. Rainfall will shifted to light rain.</p>
DWP (Putra, 2017) <i>Output</i>	<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>
(Gkatzia <i>et al.</i> , 2016) <i>Output</i>	<ul style="list-style-type: none"> -Light rian showers are likely -Sunny intervals with rain being possible – less likely than not. -Sunny with rain being unlikely
(Ramos- Soto <i>et</i> <i>al.</i> , 2015) <i>Output</i>	<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>
(Kittredge & Driedger, 1994) <i>Output</i>	<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>

Berdasarkan hasil perbandingan pada tabel 4.57 dapat disimpulkan bahwa *output* dan konten-konten yang dihasilkan dari penelitian ini lebih banyak dibanding dengan yang lainnya, namun jika dilihat dari penyampaian konten pada teks keluaran, sistem DWP lebih unggul, dikarenakan sistem DWP dikembangkan untuk membangkitkan berita dengan domain cuaca (Putra *et al.*, 2017). Pada penelitian DWP, hasil keluaran dibagi menjadi dua paragraf, dimana paragraf pertama berisikan prediksi cuaca, dan paragraf kedua berisi tentang rangkuman data selama satu bulan. Pada sistem DWP terdapat dua data masukan yaitu data klimatologi dan kualitas udara, sehingga konten yang muncul terbagi ke dalam dua bagian. Sedangkan pada penelitian Ramos (2016), hanya ada dua konten yang ditampilkan yaitu konten kualitas udara dan kecepatan angin. Jika pada penelitian Kittredge dan Driedger (1994) dihasilkan konten mengenai kondisi angin dan salju secara terpisah-pisah, sedangkan pada penelitian Gkatzia (2016) konten yang disampaikan hanya terkait keadaan langit saja. Hal ini menunjukkan bahwa sistem D2T yang dibangun sebelumnya terikat pada satu domain spesifik saja. Berbeda dengan sistem D2T yang dikembangkan pada penelitian ini, dimana sistem akan tetap menghasilkan meskipun yang menjadi masukan adalah data *unspecific* ataupun data yang tidak memiliki header. Dimana data-data tersebut akan diberi header *default* yang secara otomatis dihasilkan oleh sistem dan lalu diinterpretasikan nilainya menggunakan fungsi keanggotaan secara *unspecific*.

Keluaran dari sistem ini terdiri dari tiga paragraf, dimana paragraf pertama menceritakan rangkuman data dalam satu *batch*, paragraf ke-dua menceritakan deskripsi data terkini, dan paragraf ke-tiga memaparkan informasi prediksi. Sehingga dapat dikatakan bahwa dalam penelitian ini menjadi pelengkap dari semua konten penelitian sebelumnya dimana pada DWP (Putra *et al.*, 2017) hanya terdapat 2 paragraf berisikan ringkasan data dan prediksi untuk besok. Sedangkan pada penelitian Ramos-soto (2015) hanya menghasilkan satu kalimat prediksi saja. Pada penelitian Gkatzia (2016), keluaran sistem berupa gambar dan 1 kalimat untuk pilihan kuis dalam game dan 1 kalimat untuk deskripsi sesuai gambar. Pada aplikasi FOG (Kittredge & Driedger, 1994) keluaran sistem berupa beberapa baris kalimat,

tergantung pada konten yang ditampilkan. Keseluruhan keluaran sistem menggunakan bahasa Inggris.

Kesimpulan dari hasil perbandingan pada tabel 4.55 adalah sebagai berikut:

1. *Unspecific*

Pada penelitian ini dikembangkan sistem D2T untuk data *unspecific* yang menjadi hal baru dibandingkan dengan penelitian lainnya. Dimana sistem dapat menghasilkan dan merepresentasikan data, meskipun dari tersebut memiliki cara penginterpretasian (*specific*), ataupun tidak (*unspecific*).

2. Responsif

Pada penelitian ini dikembangkan sistem D2T yang mampu menerima masukan data apapun, baik itu *numerical* maupun *categorical*. Selain itu, sistem yang dibangun mampu memproses data tanpa *header* sekalipun.

3. Waktu data

Pada penelitian ini dilakukan pengembangan untuk membaca data secara dinamis, dalam periode waktu yang berbeda beda, dapat berupa data setiap detik, menit, jam, hari bahkan tahunan. Namun format waktu data harus mengikuti aturan pada sistem UNG.

4. Evaluasi

Evaluasi pada penelitian DWP (Putra et al, 2017) dan Ramos-soto (2016) melakukan evaluasi *Relevance* dan *Truthfulness* dengan memberikan kuisioner kepada *expert* yang berisi pertanyaan mengenai kedua aspek tersebut. Pada penelitian ini dilakukan validasi data dengan membandingkan teks keluaran dengan visualisasi data dan menilai aspek *Readability* dan *Computation Time* seperti pada penelitian DWP (Putra et al, 2017). Sedangkan pada penelitian Gkatzia (2016) evaluasi dilakukan dengan mengevaluasi nilai dari hasil permainan.

Keseluruhan detil perbandingan dapat dilihat pada tabel 4.58.

Tabel 4.58 Perbandingan dengan penelitian sebelumnya

Penelitian	Output	Korelasi antar parameter	Categorikal	Ringkasan data	Machine Learning	Prediksi	Deskripsi data terkini	Waktu data	Evaluasi	Bahasa	Pengaplikasian Sistem	Readability
(Kittredge & Driedger, 1994)	1 – 11 baris	Tidak	Tidak	Tidak	<i>Forecast Production Assistant (FPA)</i>	Ya	Tidak	48 jam	-	Inggris dan Prancis	Ya	59.61
(Ramos-Soto et al., 2015)	N kalimat (tergantung banyak data)	Tidak	Tidak	Tidak	<i>Short-term Prediction</i>	Ya	Ya	Unspecified	Kuisisioner expert (<i>Relevance & Truthfulness</i>)	Inggris, Spanyol dan Galician	Ya	71.14
(Gkatzia et al., 2016)	Gambar dan 2 baris teks	Tidak	Tidak	Tidak	-	Ya	Ya	Harian	Kuis dalam Game	Inggris	Ya	47.43
DWP (Putra, 2017)	2 paragraf	Tidak	Tidak	Ya	<i>Exponential Smoothing dan Gradient Descent</i>	Ya	Tidak	Tahunan	<i>Readability, Computation Time, kuisisioner expert (Relevance & Truthfulness)</i>	Inggris	Tidak	63.53
GNG (Abidin et al., 2018)	3 paragraf	Tidak	Tidak	Ya	<i>Piecewise Linear Approximation dengan Least Square Method</i>	Ya	Ya	Unspecified	<i>Readability, Computation Time, dan Validasi Grafis</i>	Inggris	Tidak	35.425
UNG	3 paragraf	Ya	Ya	Ya	<i>Exponential Smoothing dan Gradient Descent</i>	Ya	Ya	Unspecified	<i>Readability, Computation Time, dan Validasi Grafis</i>	Inggris	Tidak	72,31

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

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

1. Penelitian ini berhasil membuat sebuah model sistem *Data-to-Text* (D2T) untuk data *unspecific* dengan menggunakan model penelitian sebelumnya sebagai acuan, dan dilakukan penambahan model proses *Unspecific Data Handling* sehingga dataset dipastikan memiliki *header* dan *data description* sebelum diproses. Selain itu, diterapkan beberapa analisis seperti *Pearson Correlation Coefficient* untuk parameter dengan tipe *numerical* sehingga penelitian ini menjawab saran pada penelitian GNG oleh Abidin et al., (2018) dan penerapan *String Matching* untuk parameter dengan tipe *categorical*.
2. Penelitian ini berhasil mengembangkan *software* dengan model yang telah dirancang pada tujuan pertama, dengan menggunakan bahasa pemrograman R sebagai inti *Data-to-Text* dan beberapa *package* yang mendukung pembuatan sistem D2T untuk data *unspecific*.
3. Kesimpulan dari keseluruhan hasil eksperimen yang dilakukan, keluaran dari sistem terbukti merepresentasikan data yang diberikan. Penelitian ini memeroleh nilai rata-rata keseluruhan 72.31 pada aspek *Readability* yang artinya keluaran dari sistem ini tergolong dalam kategori *plain english* yang berarti dapat dipahami oleh anak usia remaja sekalipun. Sehingga hal ini menjawab masalah pada latar belakang dimana sistem ini mampu menghasilkan teks keluaran yang mudah dipahami untuk berbagai input data. Sedangkan pada aspek *Computation Time* diperoleh rata-rata waktu komputasi 2-5 detik untuk data berukuran kurang dari seribu baris. Namun untuk data berukuran lebih dari 18 ribu data proses terjadi lebih lama dengan durasi sekitar 45 detik.

5.2. Saran

Dalam pelaksanaan penelitian, penulis menyadari bahwa masih banyak kekurangan yang dilakukan oleh penulis dalam penelitian ini. Oleh karena itu, penulis menyampaikan beberapa saran yang dapat dilakukan di kemudian hari agar penelitian selanjutnya dapat menghasilkan analisis yang jauh lebih baik. Berikut beberapa saran yang penulis anjurkan:

1. Pengembangan *corpus* untuk kasus umum, atau menambahkan *corpus-corpus* untuk kasus khusus pada proses *Data Interpretation* sehingga dapat meningkatkan nilai *Readability*.
2. Menggunakan algoritma *Maching Learning* dalam proses *Content Determination* sehingga hasil keluaran yang dihasilkan lebih variatif seperti *Reinforcement Learning* yang dikembangkan oleh (Gkatzia *et al.*, 2013).
3. Menggunakan penerapan *Machine Learning* seperti *Classification*, ataupun *Clustering* untuk analisis data sehingga konten pada teks keluaran bisa lebih variatif.

DAFTAR PUSTAKA

- Abidin, A. Z., Riza, L. S., & Nurdin, E. A. (2018). Pengembangan Sistem Data-To-Text (D2T) untuk Membangkitkan Berita pada Data Streaming.
- Athoillah, M., Irawan, M., I., & Imah, Elly, M. (2015). Study Comparison of SVM-, K-NN- and Backpropagation-Based Classifier for Image Retrieval. *Jurnal Ilmu Komputer Dan Informasi (Journal of Computer Science and Information)*, 8(1), 11–18.
- Banaee, H., Ahmed, M. U., & Loutfi, A. (2013). Towards NLG for Physiological Data Monitoring with Body Area Networks. *Proceedings of the 14th European Workshop on Natural Language Generation (ENLG'13)*, 193–197. Retrieved from <http://www.aclweb.org/anthology/W13-2127>
- Bateman, J., & Zock, M. (2012). Natural Language Generation. *The Oxford Handbook of Computational Linguistics*, 9780199276(April 2018), 1–21. <https://doi.org/10.1093/oxfordhb/9780199276349.013.0015>
- Belz, A. (2007). Probabilistic Generation of Weather Forecast Texts. *Naacl-Hlt*, 164–171. Retrieved from <http://www.aclweb.org/anthology/N07-1021>
- Bowden, R., & Bullington, S. F. (1996). Development of manufacturing control strategies using unsupervised machine learning. *IIE Transactions (Institute of Industrial Engineers)*, 28(4), 319–331. <https://doi.org/10.1080/07408179608966279>
- Boyd, S. (1998). TREND: A System for Generating Intelligent Descriptions of Time-Series Data. *Icips*, 1–5. <https://doi.org/10.1.1.57.3705>
- Budiharto, W. (2013). *Pengantar Praktis Pemrograman R untuk Ilmu Komputer*.
- Castillo-Ortega, R., Marín, N., Martínez-Cruz, C., & Sánchez, D. (2014). A proposal for the hierarchical segmentation of time series. Application to trend-based linguistic description. *IEEE International Conference on Fuzzy Systems*, 489–496. <https://doi.org/10.1109/FUZZ-IEEE.2014.6891840>
- Chen, Y., & Wu, Y. (2016). On the Massive String Matching Problem, 350–355.
- Chowdhury, G. G. (2005). Natural language Processing, 51–89.
- de Vaus, D. A. (2002). *Surveys in Social Research* (5th editio).
- Demir, S., Carberry, S., & McCoy, K. F. (2012). Summarizing Information

- Graphics Textually. *Computational Linguistics*, 38(3), 527–574.
<https://doi.org/10.1162/COLI>
- Fallah-Ghalhary, G. A., Mousavi-baygi, M., & Nokhandan, M. H. (2009). Environmental Sciences. *Journal of Chromatography B*, 6(3), 271–275.
<https://doi.org/10.1111/ijfs.12122>
- Fallah-Ghalhary, G. A., Mousavi-Baygi, M., & Nokhandan, M. H. (2009). Annual Rainfall Forecasting by Using Mamdani Fuzzy Interface System. *Research Journal of Environmental Sciences*, 3.
- Gatt, A., Portet, F., Reiter, E., Hunter, J., Mahamood, S., & Moncur, W. (2009). From data to text in the Neonatal Intensive Care Unit : Using NLG technology for decision support and information management, 22, 153–186.
<https://doi.org/10.3233/AIC-2009-0453>
- Gkatzia, D., Hastie, H., Janarthanam, S., & Lemon, O. (2013). Generating student feedback from time-series data using Reinforcement Learning, 115–124.
- Gkatzia, D., Lemon, O., & Rieser, V. (2016). Natural Language Generation enhances human decision-making with uncertain information. *The 54th Annual Meeting of the Association for Computational Linguistics*, 264.
- Gkatzia, D., Lemon, O., & Rieser, V. (2017). Data-to-Text Generation Improves Decision-Making Under Uncertainty. *IEEE Computational Intelligence Magazine*, 12(3), 10–17. <https://doi.org/10.1109/MCI.2017.2708998>
- Hallett, C., Power, R., & Scott, D. (2006). Summarisation and visualisation of e-Health data repositories Conference Item Repositories. *UK E-Science All-Hands Meeting*, 18–21.
- Härdle, W., & Simar, L. (2007). *Applied Multivariate Statistical Analysis. Applied Statistics* (Vol. 22007). Berlin: Springer. <https://doi.org/10.2307/2347962>
- Hospital, Z. (2003). Predicting hepatitis B virus – positive metastatic hepatocellular carcinomas using gene expression profiling and supervised machine learning. *Nature Medicine*, 9(4), 416. <https://doi.org/10.1038/nm843>
- Hunter, J., Freer, Y., Gatt, A., Reiter, E., Sripada, S., Sykes, C., & Westwater, D. (2011). Bt-Nurse: Computer generation of natural language shift summaries from complex heterogeneous medical data. *Journal of the American Medical Informatics Association*, 18(5), 621–624. <https://doi.org/10.1136/amiajnl->

2011-000193

- Ihaka, R., & Gentleman, R. (2012). R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3), 299–314. <https://doi.org/10.1080/10618600.1996.10474713>
- Käll, L., Canterbury, J. D., Weston, J., Noble, W. S., & MacCoss, M. J. (2007). Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods*, 4(11), 923–925. <https://doi.org/10.1038/NMETH1113>
- Karl Pearson, F. R. S. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302), 157–175. <https://doi.org/10.1080/14786440009463897>
- Kittredge, R. I., & Driedger, N. (1994). Using Natural-Language Processing to Produce Weather Forecasts. *IEEE Expert-Intelligent Systems and Their Applications*, 9(2), 45–53. <https://doi.org/10.1109/64.294135>
- Kukich, K. (1983). Design of a knowledge-based report generator. *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics* -, 145. <https://doi.org/10.3115/981311.981340>
- Liddy, E. D. (2001). Natural Language Processing. In *Encyclopedia of Library and Information Science*.
- McKeown, K., Kukich, K., & Shaw, J. (1994). Practical issues in automatic documentation generation. *Proceedings of the Fourth Conference on Applied Natural Language Processing* -, 7. <https://doi.org/10.3115/974358.974361>
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of Machine Learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 17). MIT Press. https://doi.org/10.1007/978-3-642-34106-9_15
- Ng, A. Y., Coates, A., Diel, M., Ganapathi, V., Schulte, J., Tse, B., ... Liang, E. (2006). Autonomous inverted helicopter flight via reinforcement learning. *Springer Tracts in Advanced Robotics*, 21, 363–372. https://doi.org/10.1007/11552246_35

- Paliouras, G., Papatheodorou, C., Karkaletsis, V., & Spyropoulos, C. D. (2002). Discovering user communities on the Internet using unsupervised machine learning techniques. *Interacting With Computers*, 14(6), 761–791. [https://doi.org/10.1016/S0953-5438\(02\)00015-2](https://doi.org/10.1016/S0953-5438(02)00015-2)
- Palpanas, T., Vlachos, M., Keogh, E., Gunopoulos, D., & Truppel, W. (2004). Online amnesic approximation of streaming time series. *Data Engineering, 2004. Proceedings. 20th International Conference On*, 339–349. <https://doi.org/10.1109/ICDE.2004.1320009>
- Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., & Sykes, C. (2009). Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7–8), 789–816. <https://doi.org/10.1016/j.artint.2008.12.002>
- Portet, F., Reiter, E., Hunter, J., & Sripada, S. (2007). Automatic Generation of Textual Summaries from Neonatal Intensive Care Data. *Lecture Notes in Artificial Intelligence*, 227–236.
- Pressman, R. S. (2001a). *Software engineering: a practitioner's approach* (5th ed.). New York: McGraw-Hill Publishing Company, Inc.
- Pressman, R. S. (2001b). *Software Engineering A Practitioner's Approach*. (B. Jones & E. Gray, Eds.) (5th ed.). Palgrave Macmillan.
- Putra, B., Riza, L. S., & Wihardi, Y. (2017). Pengembangan Sistem Data-to-Text untuk Membangkitkan Berita Cuaca dengan Pendekatan Time-Series dalam R.
- Rahman, A. B. (2017). *Deteksi Genomic Repeats Menggunakan Algoritma Knuth-Morris-Pratt pada R High-Performance Computing Package*. Bandung.
- Ramos-Soto, A., Bugarín, A., & Barro, S. (2016a). Fuzzy Sets Across the Natural Language Generation Pipeline. *Progress in Artificial Intelligence*. <https://doi.org/10.1007/s13748-016-0097-x>
- Ramos-Soto, A., Bugarín, A., & Barro, S. (2016b). On the role of linguistic descriptions of data in the building of natural language generation systems. *Fuzzy Sets and Systems*, 285, 31–51. <https://doi.org/10.1016/j.fss.2015.06.019>
- Ramos-Soto, A., Bugarin, A., Barro, S., Gallego, N., Rodriguez, C., Fraga, I., & A.Saunders. (2015). Automatic Generation of Air Quality Index Textual

- Forecasts Using a Data-To-Text Approach. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9422(May), 164–174.
<https://doi.org/10.1007/978-3-319-24598-0>
- Reddington, J., & Tintarev, N. (2011). Automatically Generating Stories from Sensor Data. *Proceedings of the 16th International Conference on Intelligent User Interfaces*, (November 2010), 407–410.
<https://doi.org/10.1145/1943403.1943477>
- Reiter, E. (1996). Building Natural-Language Generation Systems, 91–93.
- Reiter, E. (2010). 20 Natural Language Generation. ... of *Computational Linguistics and Natural Language* Retrieved from http://gendocs.ru/docs/20/19207/conv_1/file1.pdf#page=600
- Reiter, E. (2011). An Architecture for Data-to-Text Systems. *Computational Intelligence*, 27(1), 23–40. <https://doi.org/10.1111/j.1467-8640.2010.00370.x>
- Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1), 57–87.
<https://doi.org/10.1017/S1351324997001502>
- Riza, L. S. (2015). *Data Science and Big Data Processing in R: Representations and Software*.
- Riza, L. S., Nasrulloh, I. F., Junaeti, E., Zain, R., & Nandiyanto, A. B. D. (2016). GradDescentR: An R package implementing gradient descent and its variants for regression tasks. *Proceedings - 2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2016*, 125–129. <https://doi.org/10.1109/ICITISEE.2016.7803060>
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210–229.
<https://doi.org/10.1147/rd.33.0210>
- Schneider, A. H., Mort, A., Mellish, C., Reiter, E., & Wilson, P. (2013). MIME - NLG in Pre-Hospital Care. *Fourteenth European Workshop on Natural Language Generation*, 152–156.
- Shannon, C. E. (1950). A Chess-Playing Machine. *Scientific American*, 182(2), 48–51.

- Soehn, J., Zinsmeister, H., & Rehm, G. (2007). Requirements of a User-Friendly , General-Purpose Corpus Query Interface.
- Spector, P. (2004). An Introduction to R. *Statistical Computing Facility*, (x), 1–10.
- Sripada, S. G., & Gao, F. (2007). Summarizing dive computer data: A case study in integrating textual and graphical presentations of numerical data. *Proceedings of the Workshop on Multimodal Output Generation (MOG-2007)*, 149–157.
- Sripada, S. G., & Reiter, E. (2003). S U M T I M E -M OUSAM : Configurable Marine Weather Forecast Generator. *Expert Update*, 6(1), 4–1.
- Sripada, S. G., Reiter, E., Hunter, J., & Yu, J. (2001). A two-stage model for content determination. *Proceedings of the 8th European Workshop on Natural Language Generation-Volume 8*, 8, 1–8. <https://doi.org/10.3115/1117840.1117842>
- Sripada, S. G., Reiter, E., Hunter, J., & Yu, J. (2003). Generating English summaries of time series data using the Gricean maxims. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '03*, 187. <https://doi.org/10.1145/956755.956774>
- Stone, P., Sutton, R. S., & Kuhlmann, G. (2005). Reinforcement Learning for RoboCup-Soccer Keepaway. *Adaptive Behavior*, 13(3), 165–188. <https://doi.org/10.1177/105971230501300301>
- Tang, F., Brennan, S., Zhao, Q., & Tao, H. (2007). Co-Tracking Using Semi-Supervised Support Vector Machines. *Computer Vision*, 1–8.
- Thomas, K. E., Sripada, S., & Noordzij, M. L. (2012). Atlas.txt: Exploring linguistic grounding techniques for communicating spatial information to blind users. *Universal Access in the Information Society*, 11(1), 85–98. <https://doi.org/10.1007/s10209-010-0217-5>
- Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations : A simple and general method for semi-supervised learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 384–394.
- Turing, A. (1950). Introducci ón a la Inteligencia Artificial. *Intelligence*, 59, 433–460.
- Turner, R., Sripada, S., Reiter, E., & Davy, I. P. (2008). Using spatial reference

- frames to generate grounded textual summaries of georeferenced data. *Proceedings of the Fifth International Natural Language Generation Conference*, 16–24. <https://doi.org/10.3115/1708322.1708328>
- Vijayarani, S., & Janani, R. (2016). Information From Desktop – Comparative Analysis. *2016 International Conference on Inventive Computation Technologies (ICICT)*. <https://doi.org/https://doi.org/10.1109/INVENTIVE.2016.7830233>
- Williamson, R., & Andrews, B. J. (2000). Gait Event Detection for FES Using Accelerometers and Supervised Machine Learning. *IEEE Transactions on Rehabilitation Engineering*, 8(3), 312–319.
- Ye, Q., Zhang, Z., & Law, R. (2009). Expert Systems with Applications Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems With Applications*, 36(3), 6527–6535. <https://doi.org/10.1016/j.eswa.2008.07.035>
- Yu, J., Reiter, E., Hunter, J., & Mellish, C. (2007). Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering*, 13(1), 25–49. <https://doi.org/10.1017/S1351324905004031>
- Zanero, S., & Savaresi, S. M. (2004). Unsupervised learning techniques for an intrusion detection system. *Proceedings of the 2004 ACM Symposium on Applied Computing - SAC '04*, 412. <https://doi.org/10.1145/967900.967988>
- Zhang, L., Peng, Y., Liang, J., Liu, X., Yi, J., & Wen, Z. (2015). An Improved String Matching Algorithm for HTTP Data Reduction, 345–348. <https://doi.org/10.1109/IIH-MSP.2015.18>