TESIS – IF 185401

ADOPSI METODE PENELUSURAN KEBUTUHAN PADA LINGKUNGAN AGILE (EXTREME PROGRAMMING) UNTUK KLASIFIKASI *DEFECT*

NURAISA NOVIA HIDAYATI
NRP. 05111950010019

Dosen Pembimbing
Ir. Siti Rochimah, M.T., Ph.D.
Dr. Agus Budi Raharjo, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
2021

(Halaman ini sengaja dikosongkan)

TESIS – IF 185401

ADOPSI METODE PENELUSURAN KEBUTUHAN PADA LINGKUNGAN AGILE (EXTREME PROGRAMMING) UNTUK KLASIFIKASI *DEFECT*

NURAISA NOVIA HIDAYATI
NRP. 05111950010019

Dosen Pembimbing
Ir. Siti Rochimah, M.T., Ph.D.
Dr. Agus Budi Raharjo, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
2021

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom)
di
Institut Teknologi Sepuluh Nopember
Oleh:
Nuraisa Novia Hidayati
NRP: 05111950010019

Tanggal Ujian: 15 Juli 2021
Periode Wisuda: September 2021

Disetujui oleh:
Pembimbing:

1. Ir. Siti Rochimah, M.T., Ph.D. .....
   NIP. 196810021994032001

2 Dr. Agus Budi Raharjo, S.Kom., M.Kom .....
   NIP. 1990202011022

Penguji:

1. Daniel O. Siahaan, S.Kom. M,Sc, PD.Eng .....
   NIP. 197411232006041001

2 Dr. Eng. Chastine Fatichah, S.Kom., M.Kom .....
   NIP. 197512202001122002

3 Dr. Radityo Anggoro , S.Kom, M.Sc .....
   NIP. 19841016 2008121002

Kepala Departemen Teknik Infomatika
Fakultas Teknologi Elektro dan Informatika Cerdas

Dr. Eng. Chastine Fatichah, S.Kom., M.Kom
NIP. 197512202001122002

(Halaman ini sengaja dikosongkan)

ADOPSI METODE PENELUSURAN KEBUTUHAN PADA LINGKUNGAN *AGILE* (*EXTREME PROGRAMMING*) UNTUK KLASIFIKASI *DEFECT*

Nama	:	Nuraisa Novia Hidayati
NRP	:	05111950010019
Pembimbing 1	:	Ir. Siti Rochimah, M.T., Ph.D.
Pembimbing 2	:	Dr. Agus Budi Raharjo, S.Kom., M.Kom.

ABSTRAK

Proses pengembangan aplikasi dengan metode *agile* terutama *Extreme Programming* (XP) sangat popular saat ini dikarenakan proses iterasi yang singkat. Namun, metode ini memiliki kelemahan pada sisi dokumentasi sehingga penelusuran penerapan kebutuhan menjadi sulit dilakukan. Kontribusi dari penelitian ini adalah memperbaiki penjabaran kebutuhan dan proses penelusuran penerapan kebutuhan sepanjang pengembangan menggunakan metode XP untuk mengenali bagaimana jenis *defect major* muncul. Pada penelitian ini setiap tahapan penting XP di dokumentasikan serta dilihat keterkaitannya menggunakan Requirement Traceability Matrix (RTM). Setelah RTM terbentuk, kemunculan *defect* ditelusuri melalui dokumen kebutuhan, pengujian, dan sumber kode pada setiap tahapan sebagai *ground truth*. Variabel pada ketiga dokumen berkorelasi dengan kemunculan *defect*. Berdasarkan hasil analisis korelasi Pearson, *defect* dan *bug* berkorelasi erat dengan perubahan maupun penambahan kebutuhan, dan tahapan pengujian terutama *integration testin*g. Transformasi vektor diterapkan pada kata di setiap dokumen menggunakan TF-IDF. Hasil vektorisasi menjadi masukan pada tiga metode pemodelan topik, yaitu Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), dan Non-Negative Matrix Factorization (NMF). Pada dua data set aplikasi metode NMF bekerja dengan baik, meskipun pada aplikasi kedua LSA lebih unggul sedikit dalam nilai akurasi dan *recall*. Kemunculan term pada setiap topik menjadi atribut klasifikasi *defect* yang dilakukan paling terakhir dengan menggunakan *decision tree* dengan dua kelas yaitu *defect major* dan *minor*. Hasil akurasi menggunakan *decision tree* Cart mencapai 94% dengan *precision* 91% , dan *recall* 88% dengan empat cabang penting yaitu jumlah pengujian diulang hingga *defect* teratasi, berapa banyak perubahan kebutuhan yang terjadi pada proses tersebut, konsistensi istilah dari kebutuhan ke pengujian, dan juga dari pengujian ke sumber kode.

Kata kunci: *requirement traceability, RTM, XP, Agile, Information Retrieval, Topic Modelling, TF-Idf, LSA, LDA, NMF, Decision tree*

(Halaman ini sengaja dikosongkan)

ADOPTION OF REQUIREMENT TRACEABILITY METHOD IN AGILE ENVIRONMENT (EXTREME PROGRAMMING) FOR DEFECT CLASSIFICATION

Nama	:	Nuraisa Novia Hidayati
NRP	:	05111950010019
Pembimbing 1	:	Ir. Siti Rochimah, M.T., Ph.D.
Pembimbing 2	:	Dr. Agus Budi Raharjo, S.Kom., M.Kom.

ABSTRACT

The application development process using agile methods, especially extreme programming (XP), is prevalent today due to the short iteration process. However, this method has weaknesses on the documentation side, so tracing the application of requirements becomes difficult. The contribution of this research is to improve the description of requirements and the process of tracing the application of requirements throughout development using the XP method to identify how major defects arise. In this study, each important stage of XP is documented, and its relationship is seen using the Requirement Traceability Matrix (RTM). After the RTM is formed, defects are traced through requirements documents, testing, and source code at each stage as *ground truth*. The variables in the three documents are correlated with the occurrence of defects. Based on the Pearson Correlation Analysis results, defects and bugs are closely correlated with changes or additions to requirements and the testing stages, especially integration testing. Vector transformations are applied to words in each document using TF-IDF. The vectorization results are input for three topic modelling methods, namely Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), and Non-Negative Matrix Factorization (NMF). In the two application data sets, the NMF method worked well, although, in the second application, LSA was slightly superior in accuracy and *recall* values. The term's appearance on each topic becomes an attribute of the defect classification that is carried out most recently by using a Decision tree with two classes, namely major and minor defects. The accuracy results using the Decision tree Cart reached 94% with 91% precision, and 88% recall with four important branches, namely the number of tests repeated until the defect was resolved, how many changes in requirements occurred in the process, consistency of terms from requirements to testing, and also from testing to source code.

Keywords: *requirement traceability, RTM, XP, Agile, Information Retrieval, Topic Modelling, TF-Idf, LSA, LDA, NMF, Decision tree*

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan tesis dengan judul "Adopsi Metode Penelusuruan Kebutuhan Pada Lingkungan *Agile* (*Extreme Programming*) Untuk Klasifikasi *Defect* " sebagai syarat mencapai gelar master di Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika.

Penulis menyadari bahwa penulisan ini tidak dapat terselesaikan tanpa dukungan dari berbagai pihak baik moril maupun materil. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu penyusunan skripsi ini terutama kepada :

1. Kedua orang tua dan adik penulis yang selalu memberikan dukungan, doa, dan motivasi, serta pengertian yang tiada henti selama masa kuliah hingga penyelesaian tesis.
2. Ibu Ir. Siti Rochimah, M.T., Ph.D., selaku dosen pembimbing utama atas bimbingan dan dukungannya hingga terselesaikannya tesis ini.
3. Bapak Dr. Agus Budi Raharjo, S.Kom., M.Kom., selaku dosen pembimbing kedua atas bimbingan dan dukungannya hingga terselesaikannya tesis ini.
4. Bapak Daniel O. Siahaan, S.Kom. M. Sc, PD.Eng. , Ibu Dr. Eng. Chastine Fatichah, S.Kom., M.Kom., dan Bapak Dr. Radityo Anggoro, S.Kom, M.Sc., selaku dosen penguji tesis yang telah memberikan saran demi kemajuan tesis ini.
5. Bapak dan Ibu dosen pascasarjana Teknik Informatika ITS Surabaya yang telah memberikan ilmu dan bimbingan selama masa kuliah.
6. Bapak dan Ibu panitia penyelenggara beasiswa saintek yang telah memberikan beasiswa kepada penulis selama menempuh pendidikan, serta memberikan monitoring dan evaluasi agar penulis dapat menyelesaikan pendidikan master dengan baik.

7. Teman-teman sejawat dan atasan penulis di BPPT yang senantiasa memberikan suntikan semangat, terutama Anne Parlina sebagai teman diskusi selama menempuh pendidikan dan pengerjaan tesis ini.
8. Teman-teman seperjuangan program pascasarjana Angkatan 2019 yang telah menjadi teman sekaligus mentor selama masa perkuliahan.
9. Sahabat-sahabat penulis Arina Pramudita L, Gemala Yanuarita, dan Umi Salama yang terus menanyakan progres tesis dan memberikan dukungan.
10. Serta pihak-pihak lain yang tidak dapat disebutkan di sini yang telah membantu penulis dalam penyusunan tesis ini.

Penulis menyadari masih terdapat banyak kekurangan dan kelemahan. Untuk itu saran dan kritik yang membangun akan sangat membantu agar tesis ini dapat menjadi lebih baik. Akhir kata penulis ucapkan terima kasih.

Surabaya, Juli 2021

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN TESIS i

ABSTRAK iii

ABSTRACT v

KATA PENGANTAR vii

DAFTAR ISI ix

DAFTAR GAMBAR xiii

DAFTAR TABEL xvi

BAB 1 PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 4

1.3 Tujuan 4

1.4 Manfaat 4

1.5 Batasan Penelitian 4

1.6 Kontribusi Penelitian 5

BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI 7

2. 1. *Extreme Programming* (XP) 7

2. 2. Penelusuran Kebutuhan 8

2. 3. Requirement Traceability Matrix (RTM) 9

2. 4. *Preprocessing* 10

2.4.1. *Stopword* 10

2.4.1. *Tokenization* Dan *Stemming* 10

2. 5. TF-IDF 11

2. 6. Pemodelan Topik 12

2.6.1 Latent Semantic Analysis (LSA) 12
2.6.2 Latent Dirichlet Allocation (LDA)..... 13
2.6.3 *Non-negative Matrix Factorization* (NMF)..... 14
2. 7. Korelasi Pearson..... 15
2. 8. Tahap Pengujian untuk Manajemen *Defect* 16
2. 9. *Decision tree* 17
2. 10. *Evaluation Metrics* 18
BAB 3 METODOLOGI PENELITIAN 21
3. 1. Studi Literatur 21
3. 2. Membuat *Ground truth* Penelusuran 21
3.2.1. Identifikasi Tahapan XP 22
3.2.2. Identifikasi Kebutuhan 23
3.2.3. Identifikasi *Defect* 23
3.2.4. Membuat RTM..... 26
3.2.5. Membuat Data Set..... 26
3.2.6. Analisis Korelasi 26
3. 3. Pemodelan Topik..... 26
3.3.1. Preprocessing..... 26
3.3.2. Pembobotan kata dan kemiripan 29
3. 4. Klasifikasi *Defect* 33
3. 5. Pengujian 34
BAB 4 HASIL DAN PEMBAHASAN 37
4. 1. Karakteristik Data..... 37
4. 2. Membuat RTM (*ground truth*)..... 37
4. 2. 1. Data Kebutuhan :..... 39
4. 2. 2. Data Pengujian 45

4. 2. 3. Membuat RTM 48
4. 2. 4. Uji Korelasi 52
4. 3. Hasil Pemodelan Topik 56
4. 3. 1. Aplikasi Pertama 56
4.3.1.1. Kebutuhan 56
4.3.1.2. Pengujian 64
4.3.1.3. *Bug* and *Defect* 67
4.3.1.4. Evaluasi Hasil Pemodelan Topik 69
4. 3. 2. Aplikasi Kedua 78
4.3.2.1. Kebutuhan 78
4.3.2.2. Pengujian 85
4.3.2.3. *Bug* and *Defect* 89
4.3.2.4. Evaluasi Hasil Pemodelan Topik 90
4. 3. 3. Analisa Hasil Pemodelan Topik Kedua Aplikasi 98
4. 4. Klasifikasi *Defect* 99
BAB 5 PENUTUP 108
5.1 Kesimpulan 108
5.2 Saran 110
DAFTAR PUSTAKA 111
BIODATA PENULIS 118

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 3.1 Diagram alir tahapan penelitian..... 21
Gambar 3.2 Diagram Alir pembuatan *ground truth* penelusuran..... 22
Gambar 3.3 Kumpulan *user story* dan *user story point* 23
Gambar 3.4 Contoh Kode Pengujian Unit 25
Gambar 4.1 Korelasi Pearson Heatmap Aplikasi Pertama..... 54
Gambar 4.2 Korelasi Pearson Heatmap Aplikasi Kedua 54
Gambar 4.3 Data *Ground truth* Persebaran Topik Aplikasi Pertama 57
Gambar 4.4 Kata Penyusun Topik Unigram-LSA Aplikasi Pertama 57
Gambar 4.5 Persebaran Kebutuhan Unigram-LSA Aplikasi Pertama..... 58
Gambar 4.6 Kata Penyusun Topik Kebutuhan LSA-Bigram Trigram Aplikasi Pertama..... 59
Gambar 4.7 Persebaran Topik Kebutuhan LSA- Bigram Trigram Aplikasi Pertama 59
Gambar 4.8 Kata Penyusun Topik Kebutuhan Unigram-LDA Aplikasi Pertama 60
Gambar 4.9 Persebaran Topik Kebutuhan Unigram-LDA Aplikasi Pertama..... 60
Gambar 4.10 Kata Penyusun Topik Kebutuhan LDA-Bigram Trigram Aplikasi Pertama..... 61
Gambar 4.11 Persebaran Topik Kebutuhan LDA- Bigram Trigram Aplikasi Pertama..... 61
Gambar 4.12 Kata Penyusun Topik Kebutuhan Unigram -NMF Aplikasi Pertama 62
Gambar 4.13 Persebaran Topik Kebutuhan Unigram -NMF Aplikasi Pertama.... 62
Gambar 4.14 Kata Penyusun Topik Kebutuhan Bigram dan Trigram – NMF Aplikasi Pertama..... 63
Gambar 4.15 Persebaran Topik Kebutuhan Bigram dan Trigram – NMF Aplikasi Pertama..... 63
Gambar 4.16 Kata Penyusun Topik Pengujian Pemodelan LSA-Unigram Aplikasi Pertama..... 64

Gambar 4.17 Kata Penyusun Topik Pengujian Pemodelan LSA-Bigram dan Trigram Aplikasi Pertama 64
Gambar 4.18 Kata Penyusun Topik Pengujian Pemodelan LDA-Unigram Aplikasi Pertama 65
Gambar 4.19 Kata Penyusun Topik Pengujian LDA- Bigram dan Trigram Aplikasi Pertama 65
Gambar 4.20 Kata Penyusun Topik Pengujian Pemodelan NMF-Unigram 66
Gambar 4.21 Kata Penyusun Topik Pengujian Pemodelan NMF-Bigram dan Trigram 66
Gambar 4.22 Kata Penyusun Topik *Bug* dan *Defect* Pemodelan LSA-Unigram Aplikasi Pertama 67
Gambar 4.23 Kata Penyusun Topik *Bug* dan *Defect* Pemodelan LDA-Unigram Aplikasi Pertama 67
Gambar 4.24 Kata Penyusun Topik *Bug* dan *Defect* NMF-Unigram Aplikasi Pertama 68
Gambar 4.25 Persebaran *Ground truth* Topik Kebutuhan Aplikasi Kedua 78
Gambar 4.26 Kata Penyusun Topik Kebutuhan LSA-Unigram Aplikasi Kedua .. 79
Gambar 4.27 Persebaran Topik Kebutuhan LSA-Unigram Aplikasi Kedua 80
Gambar 4.28 Kata Penyusun Topik Kebutuhan LSA-Bigram Trigram Aplikasi Kedua 80
Gambar 4.29 Persebaran Topik Kebutuhan LSA-Bigram Trigram Aplikasi Kedua 81
Gambar 4.30 Kata Penyusun Topik Kebutuhan LDA-Unigram Aplikasi Kedua . 81
Gambar 4.31 Persebaran Topik Kebutuhan LDA-Unigram Aplikasi Kedua..... 82
Gambar 4.32 Kata Penyusun Topik Kebutuhan LDA-Bigram Trigram Aplikasi Kedua 82
Gambar 4.33 Persebaran Topik Kebutuhan LDA-Bigram Trigram Aplikasi Kedua 83
Gambar 4.34 Kata Penyusun Topik Kebutuhan NMF-Unigram Aplikasi Kedua . 83
Gambar 4.35 Persebaran Topik Kebutuhan NMF-Unigram Aplikasi Kedua 84
Gambar 4.36 Kata Penyusun Topik Kebutuhan NMF-Bigram Trigram Aplikasi Kedua 84

Gambar 4.37 Persebaran Topik Kebutuhan NMF Bigram Trigram Aplikasi Kedua 85
Gambar 4.38 Kata Penyusun Topik Pengujian LSA-Unigram Aplikasi Kedua ... 85
Gambar 4.39 Kata Penyusun Topik Pengujian LSA-Bigram Trigram Aplikasi Kedua 86
Gambar 4.40 Kata Penyusun Topik Pengujian LDA-Unigram Aplikasi Kedua... 86
Gambar 4.41 Kata Penyusun Topik Pengujian LDA-Bigram Trigram Aplikasi Kedua 87
Gambar 4.42 Kata Penyusun Topik Pengujian NMF-Unigram Aplikasi Kedua .. 88
Gambar 4.43 Kata Penyusun Topik Pengujian NMF-Bigram Trigram Aplikasi Kedua 88
Gambar 4.44 Kata Penyusun Topik *Bug* dan *Defect* LSA-Unigram Aplikasi Kedua 89
Gambar 4.45 Kata Penyusun Topik *Bug* and *Defect* LDA-Unigram Aplikasi Kedua 89
Gambar 4.46 Kata Penyusun Topik *Bug* and *Defect* NMF-Unigram Aplikasi Kedua 90
Gambar 4.47 Persebaran Data *Defect* 102
Gambar 4.48 Confusion Matrix *Decision tree* Awal..... 103
Gambar 4.49 Pohon Keputusan Klasifikasi *Defect* 104
Gambar 4.50 Confusion Matrix *Decision tree* dengan SMOTE 105
Gambar 4.51 Pohon Keputusan Klasifikasi *Defect* Akhir 105

DAFTAR TABEL

Tabel 2.1 Metriks Kualitas Pengujian 17
Tabel 2.2 Confusion Matrix 19
Tabel 4.1 *User story* Aplikasi 38
Tabel 4.2 Contoh *Story point* Aplikasi Pertama 41
Tabel 4.3 Contoh *Story point* Aplikasi Kedua 44
Tabel 4.4 Contoh Pengujian Aplikasi Pertama 45
Tabel 4.5 Contoh Pengujian Aplikasi Kedua 46
Tabel 4.6 Contoh Matrix Kualitas Pengujian Aplikasi Pertama 47
Tabel 4.7 Contoh Matrix 48
Tabel 4.8 Tabel Penyimpanan Database Aplikasi Pertama 49
Tabel 4.9 Tabel Penyimpanan Database Aplikasi Kedua 49
Tabel 4.10 Contoh RTM Aplikasi Pertama 50
Tabel 4.11 Contoh RTM Aplikasi Kedua 51
Tabel 4.12 Data Set Uji Korelasi Aplikasi Pertama 53
Tabel 4.13 Data Set Uji Korelasi Aplikasi Kedua 53
Tabel 4.14 Nilai Akurasi Kebutuhan dan Pengujian Aplikasi Pertama 70
Tabel 4.15 Nilai Cohen Kappa Kebutuhan dan Pengujian Aplikasi Pertama 70
Tabel 4.16 *Precision* dan *Recall* Kebutuhan dan Pengujian Aplikasi Pertama 70
Tabel 4.17 Nilai Akurasi *Bug* dan *Defect* Aplikasi Pertama 75
Tabel 4.18 Nilai Cohen Kappa *Bug* dan *Defect* Aplikasi Pertama 76
Tabel 4.19 Nilai *Precision* dan *Recall Bug* dan *Defect* Aplikasi Pertama 76
Tabel 4.20 Nilai Akurasi Kebutuhan dan Pengujian Aplikasi Kedua 93
Tabel 4.21 Nilai Cohen Kappa Kebutuhan dan Pengujian Aplikasi Kedua 93
Tabel 4.22 *Precision* dan *Recall* Kebutuhan dan Pengujian Aplikasi Kedua 94
Tabel 4.23 Nilai akurasi Pemodelan Topik *Bug* dan *Defect* Aplikasi Kedua 96
Tabel 4.24 Nilai Cohen Kappa *Bug* dan *Defect* Aplikasi Kedua 96
Tabel 4.25 Nilai *Precision* dan *Recall Bug* dan *Defect* Aplikasi Kedua 97
Tabel 4.26 Contoh Data *Defect* 101

(Halaman ini sengaja dikosongkan)

BAB 1
PENDAHULUAN

Pada bab ini dijelaskan mengenai beberapa hal dasar dalam pembuatan proposal penelitian yang meliputi latar belakang, perumusan masalah, tujuan, manfaat, kontribusi penelitian, dan batasan masalah.

1.1 Latar Belakang

Metodologi *agile*, terutama *extreme programming* (XP) banyak diterapkan pada pengembangan perangkat lunak saat ini. Pada prosesnya komunikasi antara tim pengembang dan pengguna sering terjadi sehingga dapat mengakomodir penambahan maupun perubahan kebutuhan yang cepat. Akan tetapi, prinsip yang ada pada XP memiliki banyak kelemahan seperti minimnya dokumentasi dan proses pengerjaan ulang yang terus menerus pada satu kebutuhan, mengakibatkan penurunan kualitas perangkat lunak. Perangkat lunak dapat memiliki resiko gagal mewujudkan seluruh kebutuhan pengguna dikarenakan tidak adanya kontrol terhadap prioritas kebutuhan dan sejauh mana perubahan dapat dilakukan.Ketika kebutuhan berubah atau bertambah tidak jarang terjadi kesalahan penafsiran, yang kemudian mengakibatkan kegagalan fungsi pada kebutuhan lain sebagai satu kesatuan. Penelusuran kegagalan fungsi terhadap kebutuhan awal pengguna perlu diperhatikan sebagai proses perbaikan yang berkelanjutan. Penelusuran kebutuhan hingga ke sumber kode juga diharapkan dapat mempermudah proses pengenalan resiko yang muncul selama proses pengembangan dan memastikan bahwa semua kebutuhan telah dapat diterapkan pada perangkat lunak.

Membangun sebuah penelusuran mulai dari kebutuhan hingga ke sumber kode memiliki beberapa hambatan, yaitu bertambahnya waktu pengerjaan yang juga berarti menambah biaya dan menghambat laju pengembangan. Selain permasalahan waktu dan biaya, penerapan penelusuran kebutuhan juga memiliki tantangan lain seperti otomatisasi, tingkat kehandalannya, penentuan ruang

lingkup, proses koordinasi individu yang terlibat, pembaharuan secara real time, melakukan identifikasi kebutuhan non fungsional, kemudahan untuk bisa dipahami, dan nilai apa yang diperoleh (Wang *et al.*, 2018). Proses penelusuran pada dasarnya tetap memerlukan proses manual, dikarenakan kompleksitas hubungan diantara kebutuhan dan artifak yang ada seperti desain dan sumber kode. Penelusuran kebutuhan yang paling mudah diterapkan adalah RTM (*Requirement Traceability Matrix*). Metode RTM merupakan identifikasi hubungan kebutuhan dengan desain hingga ke sumber kode.

RTM disajikan dalam bentuk tabel dengan kolom berisi kebutuhan yang dipetakan dengan kolom selanjutnya berupa desain seperti use case, activity diagram, class diagram, sequence, UI, dan artifak desain lainnya. Pemetaan kemudian berlanjut pada kolom artifak pendukung, yaitu tabel, *User Interface*, *test-case*, dan status verifikasinya (Putro and Wibowo, 2018). RTM semacam ini diterapkan pada proses pengembangan tradisional dimana fase desain memiliki waktu yang cukup panjang dan detail. Pada lingkungan pengembangan Agile kolom disederhanakan hanya menjadi kebutuhan, User Interface , dan sumber kodenya. Pembuatan RTM pada lingkungan agile pada penelitian sebelumnya menerapkan algoritma sederhana pada dokumen *spreadsheet* untuk mengidentifikasi artifak yang berhubungan. Sementara itu, kebutuhan dikelompokkan sesuai proses yang terjadi, yaitu pemecahan, penghapusan, pergantian, dan penghentian (Jeong, Cho and Lee, 2018).

Algoritma tersebut akan membantu melakukan generate RTM pada sheet selanjutnya. Akan tetapi, proses memasukkan artifak terkait tetaplah manual. Kebutuhan selalu di dokumentasikan berupa teks, dan pada lingkungan agile berupa *user story* pendek, maka penelusuran mengarah pada *Information Retrieval* (IR). Terdapat penelitian yang melakukan ekstraksi kata pada artifak kebutuhan, desain, dan sumber kode. Kata-kata tersebut kemudian disimpan ke dalam masing-masing tabel dan diproses dengan mencari tingkat similarity menggunakan metode K-gram (Suhartoyo and Rochimah, 2016). Latent Semantic Analysis (LSA) dan LDA Latent Dirichlet Allocation (LDA) juga pernah dipergunakan sebelumnya untuk melakukan penelusuran sumber kode ke beberapa kebutuhan (Nanang, Penelusuran

and Penelusuran, 2018). Proses IR secara umum mencari kemiripan kata atau istilah untuk membuat sebuah keterhubungan. Kemiripan kata akan sulit ditemukan apabila penggunaan Bahasa antara kebutuhan dan framework pemrograman sumber kode berbeda. Terlebih lagi penggunaan istilah yang berbeda-beda pada setiap dokumen.

Penelusuran berdasarkan kemiripan kata atau istilah bisa digunakan untuk mengetahui ketergantungan antar kebutuhan yang di peroleh dari *user story* saja. Penerapan penelusuran pada kebutuhan selama proses pengembangan terbukti mampu menekan jumlah kemunculan *defect* pada sebuah produk perangkat lunak yang dihasilkan (Rempel and Mader, 2017). Terdapat banyak variabel penelusuran kebutuhan yang kemudian dilihat pengaruhnya pada kemunculan *defect*, dan yang memiliki pengaruh kuat adalah penelusuran pada tingkat kebutuhan, yaitu ketergantungan antar kebutuhan. Kebutuhan juga dipecah menjadi dua, yaitu mengalami proses perubahan atau merupakan penambahan.

Penelitian ini mengusulkan untuk memperbaiki proses pengembangan XP dengan mengadopsi penelusuran kebutuhan. Penelusuran Kebutuhan bertujuan untuk membuat proses pengembangan XP terarah dan tepat sasaran, sehingga sesuai dengan kebutuhan pengguna. Hal pertama yang akan dilakukan adalah menganalisis proses penelusuran pada lingkup pengembangan XP menggunakan RTM yang dibuat secara manual. Variabel yang muncul pada saat pembuatan RTM inilah yang kemudian akan dilihat pengaruhnya terhadap *defect* yang dihasilkan. Proses selanjutnya adalah menjadikan RTM yang telah ada sebagai *ground truth* dan membuat penelusuran menggunakan metode IR seperti TF-IDF yang kemudian digabungkan dengan metode pemodelan topik seperti LSA, dan LDA, dan NMF. Hubungan antar artifak akan dibuat berdasarkan kemiripan kemunculan kata dan topiknya untuk membangun keterkaitan antar artifak, termasuk dokumentasi kemunculan *defect*. Diharapkan akan terlihat variabel apa saja yang berpengaruh dengan kemunculan *defect*. *Defect* sendiri akan dibagi menjadi dua tipe, yaitu major dan minor. Klasifikasi sederhana dengan *decision tree* akan diberlakukan untuk melihat kecenderungan variabel apa saja yang menimbulkan *defect* tipe tertentu.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana melakukan identifikasi kebutuhan pada lingkungan pengembangan XP ?
2. Bagaimana melakukan identifikasi *defect* pada lingkungan tersebut ?
3. Apa sajakah variabel yang memiliki korelasi dengan kemunculan *defect* ?
4. Bagaimana cara melakukan klasifikasi *defect* dengan penelusuran kebutuhan?

1.3 Tujuan

Tujuan yang dicapai dalam penelitian ini adalah menerapkan metode penelusuran yang sesuai dengan lingkungan pengembangan perangkat lunak XP. Setiap artifak yang dihasilkan pada tahapan XP akan dilihat keterkaitannya dengan membuat RTM, variabel yang berpengaruh pada setiap tahapan dihitung korelasinya dengan kemunculan *defect*. Penelusuran dokumen yang ada pada RTM dapat dilihat keterkaitannya dengan pemodelan topik seperti Tf-Idf, LSA, LDA, dan NMF yang kemudian dibandingkan manakah yang lebih sesuai untuk melakukan penelusuran. Kemudian dapat melakukan klasifikasi kemunculan tipe *defect* minor dan major menggunakan algoritma klasifikasi sederhana, yaitu *decision tree.*

1.4 Manfaat

Manfaat dari penelitian ini diantaranya adalah untuk membantu tim pengembang menemukan metode penelusuran kebutuhan yang sesuai pada lingkungan pengembangan perangkat lunak XP. Penelusuran tersebut dapat dipergunakan untuk penanganan kesalahan penerapan kebutuhan pada perangkat lunak, sehingga mampu melakukan klasifikasi pada *defect* yang muncul.

1.5 Batasan Penelitian

Terdapat beberapa batasan masalah terkait topik yang dikerjakan dalam penelitian ini.

1. Perangkat lunak yang diteliti memiliki minimal tiga actor pengguna, empat modul.
2. Pengembangan perangkat lunak menggunakan Bahasa pemrograman PHP dan menggunakan generator PHPMaker atau Yii2.
3. Perangkat lunak telah digunakan dan memiliki versi pengembangan, dalam penelitian ini hanya diambil tiga versi terakhir
4. Pemroresan data menggunakan Bahasa Python 3 dan library yang bekerja dalamnya.
5. Klasifikasi *defect* pada tingkatan *severity* dan hanya pada tipe *major* dan *minor*.

1.6 Kontribusi Penelitian

Kontribusi yang diharapkan dari penelitian ini adalah terciptanya model penelusuran yang dapat membuat proses pengembangan XP menjadi lebih baik. Model penelusuran yang diajukan juga diharapkan meningkatkan penanganan perubahan kebutuhan pada lingkungan pengembangan perangkat lunak XP, sehingga tim pengembang mampu mengidentifikasi variabel terlibat pada setiap tahapan yang memiliki pengaruh kuat akan terjadinya *defect* dengan jenis tertentu.

(Halaman ini sengaja dikosongkan)

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini dijelaskan mengenai pustaka yang digunakan sebagai landasan ilmiah metode yang akan digunakan dalam penelitian dan pustaka penelitian yang telah dilakukan sebelumnya terkait penelitian yang akan dilakukan. Kajian pustaka yang dilakukan mencangkup RTM, Pemodelan Topik, dan teknik klasifikasi menggunakan *decision tree*, dan evaluasi yang akan diterapkan.

2. 1. *Extreme Programming* (XP)

XP adalah salah satu teknik pengembangan aplikasi perangkat lunak secara *agile*. Pada teknik ini seluruh tim terlibat aktif dalam siklus pengembangan yang singkat dan pada setiap perulangan tahapan harus menghasilkan suatu bagian dari produk (Sadath, Karim and Gill, 2018). Kebutuhan di definisikan melalui cerita pengguna yang tidak terlalu detail, pengguna atau klien akan bekerja berdampingan dengan pengembang. Komunikasi diantara keduanya harus terjalin terus menerus baik dalam mendefinisikan penerapan kebutuhan maupun memecahkan permasalahan saat proses pengembangan berjalan . Pengembang bekerja secara berpasangan pada satu modul, secara bergantian mengoreksi kekurangan kode. Ketika memulai menulis beberapa kode produksi, pengembang harus terlebih dahulu menulis sebuah kode untuk tes pengujian sederhana dan kemudian mereka harus menulis kode minimum yang diperlukan untuk lulus tes itu.

Proses-proses ini menghasilkan banyak pengujian unit kecil yang berkembang selama implementasi kode produksi dan setiap kegagalan pada pengujian tersebut dapat menjadi indikator fitur yang rusak. Selain itu, tes unit ini membantu untuk memperbaiki dan meningkatkan kualitas kode dan melindungi basis kode untuk perubahan di masa mendatang. Tahapan pada XP dijelaskan lebih lanjut pada standar RUP (Rational Unified Process) yang telah digunakan secara luas untuk mendefinisikan eksekusi pada setiap tahapan dan tanggung jawab apa

yang harus diselesaikan pada tiap tahapan tersebut (Putra, Yuliawati and Mursanto, 2012). Kelemahan pada metode ini adalah minimnya dokumentasi secara formal pada setiap tahapan sehingga sulit untuk melakukan penelusuran bila bukan berasal dari tim pengembang. Tahapan yang paling penting ketika melakukan deteksi *defect* tentunya adalah penjelasan kebutuhan oleh pengguna, pengujian, dan implementasi. Oleh karena itu pada penelitian kali ini dilakukan dokumentasi ulang pada ketiga tahapan tersebut.

2. 2. Penelusuran Kebutuhan

Penelusuran kebutuhan adalah kemampuan untuk mengikuti alur dari perjalanan kebutuhan baik secara maju maupun mundur mulai dari tahapan spesifikasi, pengembangan hingga penyempurnaan. Dalam praktiknya, proses RT generik biasanya mencakup perencanaan dan pengelolaan strategi keterlacakan, menghasilkan jejak hubungan, menyempurnakan jejak hubungan, memelihara jejak hubungan, dan menggunakan jejak hubungan. B. Wang, R. Peng, Y. Li, H. Lai, dan Z. Wang menggolongkan penelitian terhadap RT menjadi enam kelas, yaitu teknologi penghasil, teknologi penyempurnaan , teknologi pemeliharaan , teknologi penelusuran kebutuhan non-fungsional, teknologi representatif, dan aplikasi RT (Wang *et al.*, 2018).

Salah satu hal yang perlu diperhatikan pada lingkungan pengembangan XP adalah adanya perubahan yang terus menerus. Setiap perubahan akan mempengaruhi kebutuhan awal dan konfigurasi perangkat lunak secara keseluruhan. Untuk itulah M. S. Murtazina dan T. V. Avdeenko menyatakan bahwa penelusuran kebutuhan diperlukan agar dapat menetapkan hubungan antara kebutuhan dengan artifak perangkat lunak lainnya yang didasarkan pada Ontology (Murtazina and Avdeenko, 2019). Terdapat tujuh jenis artefak yang sering digunakan sebagai sumber informasi saat meneliti ketertelusuran perangkat lunak, yaitu fitur, persyaratan, kode sumber, arsitektur, komponen, pengujian, dan model variabilitas (Vale *et al.*, 2017). Pada proses pengembangan perangkat lunak artifak XP adalah *user story*, kriteria penerimaan, desain antar muka, test scenario, acceptance test, dan lain sebagainya. Menurut F. Furtado dan A. Zisman Pemetaan

terhadap transisi pengembangan secara tradisional dengan pengembangan agile memiliki empat permasalahan, yaitu pengerjaan ulang (rework), pengertian kebutuhan secara menyeluruh (high level scope), kurangnya dokumentasi kebutuhan non-fungsional, tidak terkendalinya manajemen pengembangan (Furtado and Zisman, 2016).

2. 3. Requirement Traceability Matrix (RTM)

Teknik paling populer untuk melakukan penelusuran kebutuhan adalah RTM, pada penelitian sebelumnya H.P Putro dan A.F Wibowo membuat RTM yang disajikan dalam bentuk tabel dengan kolom berisi persyaratan yang dipetakan dengan kolom berikutnya yaitu desain terkait seperti *use case*, *activity diagram, class diagram*, *sequence*, UI, dan artifak desain lainnya. Sementara itu, pemetaan hubungan dilanjutkan dengan kolom berikutnya yaitu tabel, *User Interface*, *test-case*, dan status verifikasi (Putro and Wibowo, 2018).

RTM semacam ini diterapkan pada proses pengembangan tradisional di mana fase desain memiliki jumlah waktu yang panjang dan terperinci. Dalam lingkungan pengembangan *agile* S. Jeong, H. Cho, dan S. Lee menyederhanakan kolom menjadi hanya terdiri dari persyaratan, antarmuka pengguna dan aktivitas server, serta *source code* (Jeong, Cho and Lee, 2018). Penerapan algoritma sederhana yang diterapkan pada spreadsheet, perlu diidentifikasi secara unik sebagai penyelesaian, penghapusan, perubahan, dan penghentian, algoritma membantu menghasilkan RTM pada lembar berikutnya, tapi proses penyisipan artifak terkait tetaplah dilakukan manual [3].

R.Elamin dan R. Osman melakukan identifikasi seluruh kebutuhan yang ada dengan pemberian kode dan konten kebutuhan, berlanjut dengan pemberian kode dan konten desain, pemberian kode dan konten pengujian. Prinsip RTM juga diterapkan kedalam pembuatan metamodel dalam pola XML yang berisi identifikasi artifak, alur hubungan (trace link), dan peraturan hubungan (trace rule) dikombinasikan dengan graph database (Elamin and Osman, 2018).

Berdasarkan trace link yang dibangun pada RTM, beberapa atribut dapat dilihat dan ditemukan yang berkorelasi kuat dengan munculnya *defect*. Pada

penelitian sebelumnya M. Rath, D. Lo, dan P. Mäder, kejelasan persyaratan dan proses perubahan memiliki korelasi yang sangat kuat dengan munculnya cacat (Rath, Lo and Mäder, 2018). Persyaratan yang dapat dilacak hingga kode implementasi meskipun telah berubah membuat pengembang menyadari dampak yang dihasilkan dari perubahan tersebut. Sehingga ketika perubahan diterapkan, persyaratan lain yang terpengaruh dapat diprediksi untuk menyesuaikan dan dapat meminimalkan terjadinya cacat produk.

2. 4. *Preprocessing*

2.4.1. *Stopword*

Merupakan daftar kata dalam pengindeksan otomatis untuk menyaring kata-kata yang tidak memiliki tujuan nyata dalam menggambarkan konten dokumen terutama dalam pencarian. Menghilangkan kata-kata seperti itu menjadi pertimbangan sejak awal dalam pemrosesan kecepatan pengindeksan otomatis, menghemat banyak ruang dalam indeks, dan tidak merusak keefektifan pengambilan informasi. Penghapusan daftar kata ini harus mengikuti kaedah setiap Bahasa yang berbeda-beda dengan memenuhi dua persyaratan, yaitu keakuratan penggalian informasi pada teks seharusnya tidak berkurang dan dimensi ruang fitur teks seharusnya berkurang (Yao and Ze-Wen, 2011).

2.4.1. *Tokenization* Dan *Stemming*

Tokenisasi adalah proses pemecahan aliran teks ke dalam kata, frasa, simbol, atau elemen bermakna lainnya yang disebut token. Tujuan dari tokenization adalah eksplorasi kata-kata dalam sebuah kalimat. Daftar token menjadi input untuk selanjutnya pemrosesan seperti penguraian atau penambangan teks (Paper, 2016). Semua proses dalam pencarian informasi memerlukan set data berupa kata-kata. Hasil dari tokenisasi masih dapat berupa kata berimbuhan dan tanda baca. Untuk itu tanda baca juga harus dihapuskan untuk lebih fokus pada kata yang nantinya akan digali informasinya. *Stemming* adalah proses menggabungkan bentuk varian kata ke dalam representasi umum, akar kata (Paper, 2016).

Proses ini adalah prosedur yang banyak digunakan dalam pemrosesan teks untuk pengambilan informasi (IR) berdasarkan asumsi bahwa mengajukan

pertanyaan dengan istilah memindai menyiratkan minat pada dokumen yang mengandung kata-kata pindai. Menghilangkan imbuhan dan awalan dari kata dalam istilah-istilah yang ada pada dokumen. Asumsi yang berlaku adalah kata-kata yang sama memiliki makna yang serupa. Dalam hal keefektifan *stemming* dapat meningkatkan *recall* karena kata berubah ke bentuk kata dasarnya. Dalam pengolahan dokumen Bahasa Indonesia telah terdapat algoritma untuk mengubah kata-kata berimbuhan menjadi bentuk kata dasar, yaitu stemmer sastrawi.

2. 5. TF-IDF

Sebuah dokumen sebagian besar terdiri dari kata-kata yang berhubungan satu sama lain, sehingga untuk mengelompokkan dokumen perlu dipilih kata-kata representatif dari setiap dokumen yang ada. Modifikasi terhadap distribusi kata pada setiap kelas telah dilakukan oleh A.Guo dan T. Yang untuk membuat bobot representasi kata dalam menentukan tema dokumen menjadi lebih baik (Guo and Yang, 2016). Pada TFIDF, TF adalah jumlah item fitur yang muncul dalam dokumen, IDF adalah frekuensi anti-dokumen seperti yang ditunjukkan pada persamaan (2.2).

$$w_{td} = TF_{td} \times log\left[\frac{N}{DF_t}\right] \qquad (2.1)$$

- TF_{td} adalah jumlah kemunculan t dalam dokumen d.
- DF_t adalah jumlah dokumen yang mengandung istilah t.
- N adalah jumlah total dokumen dalam korpus.

IDF adalah kebalikan dari frekuensi dokumen yang mengukur keinformatifan istilah t. Ketika IDF dihitung, maka nilainya akan sangat rendah untuk kata-kata yang paling sering terjadi seperti *stopword* (karena kata *stopword* seperti "yang" ada di hampir semua dokumen, dan *N / df* akan memberikan nilai yang sangat rendah untuk kata itu) . Sehingga pada akhirnya akan memberikan apa yang diinginkan, yaitu bobot relatif. Pencarian kemiripan dokumen berdasarkan tampilan kata yang dipotong menjadi Unigram, bigram, atau trigram telah

dilakukan untuk menelusuri penerapan kebutuhan namun masih menghasilkan nilai *precision* dan *recall* yang rendah dikarenakan istilah yang berbeda pada setiap artifak yang terlibat (Suhartoyo and Rochimah, 2016).

2. 6. Pemodelan Topik

LSA, LDA, dan NMF adalah metode pemodelan topik yang paling sering digunakan saat ini. S. Rani dan M. Kumar membandingkan LSA dengan PLSA dan LDA. Ketiga metode tersebut dibandingkan kekuatan dan keterbatasannya masing-masing dengan kesimpulan dimana LSA memiliki keterbatasan dalam menangani polisemi, PLSA cenderung *overfitting*, dan LDA mengalami kesulitan untuk menemukan korelasi kata atau topik (Rani and Kumar, 2021).

J. Zhao, Q. P. Feng, P. Wu, J. L. Warner, J. C. Denny, dan W. Q. Wei juga menemukan bahwa NMF dapat memberikan hasil topik yang signifikan terkait dengan identifikasi penyakit, korelasi antara topik dan nilai untuk variabel tertentu menunjukkan korelasi yang baik (Zhao *et al.*, 2019). R. Albalawi, T. H. Yeap, dan M. Benyoucef menyimpulkan pada penelitiannya bahwa LDA dan NMF dianggap sebagai metode pemodelan topik yang menghasilkan output paling berharga pada teks pendek dibandingkan dengan metode lain seperti LSA, Principal Component Analysis (PCA), dan proyeksi acak (RP) (Albalawi, Yeap and Benyoucef, 2020).

2.6.1 Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) adalah suatu metode model statistik matematika untuk menganalisis struktur semantik suatu teks (Nanang, Penelusuran and Penelusuran, 2018). LSA mengembalikan konsep daripada topik untuk mewakili dokumen yang diberikan. Konsepnya adalah daftar kata-kata yang mewakili dokumen dengan cara terbaik. Dalam prosesnya mencoba untuk menemukan kumpulan kata-kata terbaik yang dikenal sebagai konsep untuk mewakili dokumen menggunakan istilah matriks kemunculan bersama. Matriks istilah dokumen sangat jarang dan berukuran besar. Perhitungan pada matriks sebesar itu memberikan hasil yang tidak terlalu signifikan dan banyak nilai dalam matriks itu nol. Untuk mengurangi kompleksitas komputasi dan untuk mendapatkan hasil yang lebih relevan dan berguna SVD digunakan. SVD menguraikan matriks

menjadi tiga matriks yang berbeda: matriks kolom ortogonal, matriks baris ortogonal dan satu matriks tunggal seperti pada persamaan (2.3).

$$M = U\sum V^* \quad (2.2)$$

- M adalah matriks m × m
- U adalah matriks tunggal kiri mxn
- Σ adalah matriks diagonal n × n dengan bilangan real non-negatif.
- V adalah matriks tunggal kanan m × n
- V* adalah matriks n × m, yang merupakan transpose dari V.

✓ Matriks Identitas: Ini adalah matriks kuadrat di mana semua elemen diagonal utama adalah satu, dan semua elemen lainnya adalah nol.

✓ Matriks Diagonal: Ini adalah matriks di mana entri selain diagonal utama semuanya nol.

✓ Singular Matrix: Sebuah matriks adalah singular jika determinannya adalah 0 atau matriks kuadrat yang tidak memiliki invers matriks.

2.6.2 Latent Dirichlet Allocation (LDA)

LDA mengekstrak fitur dokumen dari level kata, lalu menuju ke level dokumen, dan akhirnya mencapai level korpus. Ekstraksi fitur dengan LDA dilakukan melalui tahap pengetahuan akan alasan keterhubungan dan hasilnya dapat diimplementasikan untuk penggalian informasi. Alasan keterhubungan menjadi penting dalam proses LDA karena menentukan distribusi topik dalam dokumen. Karena itu, proses pemilihan metode alasan keterhubungan perlu mendapat perhatian khusus. Penelitian terhadap pemrosesan teks Bahasa Indoenesia sebelumnya mencoba membandingkan metode *mean variational inference* dan Gibbs sampling (Prihatini *et al.*, 2017) .

Hasil percobaan menunjukkan bahwa Gibbs Sampling memiliki kinerja yang lebih baik daripada Mean Variational Inference untuk LDA. Algoritma Gibbs Sampling LDA lebih relevan dalam mengekstraksi fitur unik dalam dokumen untuk menemukan topik tersembunyi dalam dokumen pada teks berbahasa Indonesia. Apa yang dilakukan adalah berusaha menemukan distribusi probabilitas bersyarat dari

penugasan topik satu kata yang dikondisikan pada penugasan topik lainnya. Mengabaikan semua perhitungan matematis, yang akan didapatkan adalah persamaan probabilitas bersyarat yang terlihat seperti ini untuk satu kata *w* dalam dokumen d yang termasuk dalam topik *k*, seperti yang ditunjukkan oleh persamaan (2.4).

$$p(z_{d,n}) = \frac{n_{d,k}+\alpha_k}{\sum_i^K n_{d,i}+\alpha_i} \frac{V_{k,w_{d,n}}+\lambda_{w_{d,n}}}{\sum_i V_{k,i}+\lambda_i} \quad (2.3)$$

dimana:

n (d, k): Jumlah kali dokumen d menggunakan topik k

v (k, w): Frekuensi topik k menggunakan kata yang diberikan

αk: Parameter Dirichlet untuk distribusi dokumen ke topik

λw: Parameter Dirichlet untuk distribusi topik ke kata.

2.6.3 *Non-negative Matrix Factorization* (NMF)

NMF dapat diterapkan pada analisis statistik data multivariat dengan cara diberikan satu set vektor data multivariat n-dimensi, vektor ditempatkan di kolom dari *n x m* matriks *V* dimana m adalah jumlah contoh dalam kumpulan data. Matriks tersebut kemudian difaktorkan menjadi *n x r* matriks *W* dan *r x m* matriks *H*, dimana *r* adalah jumlah topik yang akan dihasilkan (Suri and Roy, 2017). Biasanya *r* dipilih lebih kecil dari atau m, sehingga *W* dan *H* lebih kecil dari *V* sebagai matriks aslinya . Hal Ini menghasilkan versi yang terkompresi dari data matriks aslinya (Hien *et al.*, 2015). Beberapa ukuran kesalahan rekonstruksi antara *V* dan perkiraan *WH* dihitung dengan menggunakan persamaan (2.5)

$$\frac{1}{2}\|V - WH\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(A_{ij} - (WH)_{ij}\right)^2 \quad (2.4)$$

Optimasi maksimalisasi untuk memperhalus *W* dan *H* guna meminimalkan kesalahan rekonstruksi. Pendekatan yang umum adalah melakukan iterasi antara dua aturan pembaruan perkalian sampai konvergensi dengan persamaan (2.6)

$$H_{cj} \leftarrow H_{cj} \frac{(W\boldsymbol{V})_{cj}}{(W\boldsymbol{WH})_{cj}} \quad W_{ic} \leftarrow \frac{(\boldsymbol{A}H)_{ic}}{(\boldsymbol{WH}H)_{ic}} \qquad (2.5)$$

2. 7. Korelasi Pearson

Sebuah makalah telah menunjukkan bahwa uji signifikansi koefisien korelasi sangat penting dalam penelitian karena derajat hubungan saja tidak cukup untuk menyimpulkan bahwa koefisien korelasi yang dihitung sudah memadai (Obilor and Amadi, 2018). Korelasi antara set data adalah ukuran seberapa baik mereka terkait. Ukuran korelasi yang paling umum dalam statistik adalah korelasi Pearson. Nama lengkapnya adalah Pearson Product Moment Correlation (PPMC).

PPMC adalah ukuran kekuatan dan arah hubungan yang ada antara dua variabel yang diukur setidaknya pada skala interval. Korelasi Pearson mencoba menarik garis yang paling cocok melalui data dua variabel, dan koefisien korelasi Pearson, *r*, menunjukkan seberapa jauh semua titik data ini dari garis yang paling cocok, *r* menunjukkan hubungan linear antara dua set data, yaitu *x* dan *y* seperti pada persamaan (2.1).

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \qquad (2.6)$$

Di mana:

- *r* : koefisien korelasi r pearson
- *n*: jumlah sampel/observasi
- *x*: variabel bebas/variabel pertama
- *y*: variabel terikat/variabel kedua.

Dari perhitungan dengen rumus (2.6), maka koefisien yang dihasilkan dapat diartikan sebagai berikut :

a. Koefisien korelasi 1 berarti bahwa untuk setiap peningkatan positif dalam satu variabel, ada peningkatan positif dari proporsi tetap di variabel lain.

b. Koefisien korelasi -1 berarti bahwa untuk setiap peningkatan positif dalam satu variabel, ada penurunan negatif dari proporsi tetap di variabel lainnya.
c. Nol berarti bahwa untuk setiap kenaikan, tidak ada kenaikan positif atau negatif. Keduanya tidak terkait.

Penelitian K. Satyander dan S. Kamna menggunakan korelasi Pearson dan korelasi Spearman memberikan hasil adanya korelasi antara jumlah *defect* sebelum produk rilis dengan jumlah *defect* setelah rilis dan digunakan oleh pengguna (Kumar and Solanki, 2015). Pada penelitian tersebut kemampuan pengembang dalam mengenali kecacatan fungsi menggunakan proses pengujian yang tepat akan mengurangi munculnya kecacatan fungsi yang timbul saat produk telah rilis dan digunakan secara luas.

2. 8. Tahap Pengujian untuk Manajemen *Defect*

Tahapan dalam pengujian adalah pengujian unit, pengujian integrasi, dan pengujian regresi. Ketika perubahan demi perubahan pada kebutuhan, sumber kode juga akan terpengaruh dan menjadi tidak stabil. Oleh karena itu, pada setiap satu *iterasi* terselesaikan K. Jinzenji, T. Hoshino, L. Williams, dan K. Takahashi menegaskan bahwa pengujian regresi memainkan peran penting dalam mengukur kualitas perangkat lunak karena ketidakstabilan ini (Jinzenji *et al.*, 2013). K. Jinzenji, T. Hoshino, L. Williams, dan K. Takahashi membuat metriks kualitas seperti pada Tabel 2.1 dalam tahap pengujian.

Pada setiap tahap pengujian, setiap *bug* harus memiliki hubungan dengan kode kelas tempat pengujian dilakukan. Hubungan ini diperlukan untuk penanganan lebih lanjut agar aplikasi berjalan dengan baik. R. Noor dan M. Fahad Khan menyatakan bahwa *test case* yang dilakukan pada masing-masing *code class* juga perlu diperhatikan, yang memudahkan ketertelusuran dan melihat korelasi antara jumlah test case dan kemungkinan adanya *bug,* memungkinkan adanya masukan dari iterasi sebelumnya untuk membuat keputusan pada iterasi selanjutnya (Noor and Fahad Khan, 2014). Semakin banyak *bug* yang terdeteksi, semakin kecil *defect* yang akan terjadi saat produk dirilis.

Tabel 2.1 Metriks Kualitas Pengujian

No	Proses pengujian	Jumlah kelas yang diuji	Jumlah kasus pengujian (*test case*)	Jumlah *bug*s
1	Uji Unit	✓	✓	✓
2	Uji Integrasi	-	✓	✓
3	Uji Regresi	-	✓	✓

Sumber kode yang tidak stabil selama iterasi perlu dilakukan pengujian regresi untuk menangani adanya kemungkinan dampak pada kode lain. Menurut A. Ngah, M. Munro, dan M. Abdallah disaat melakukan uji regresi, hal yang perlu diperhatikan adalah frekuensi integrasi, frekuensi regresi, dan jumlah uji regresi (Ngah, Munro and Abdallah, 2017). Terdapat banyak teknik pengujian regresi yang ada hingga saat ini. Karena pengujian ini bertujuan untuk melihat sejauh mana dampak dari perubahan persyaratan yang diterapkan pada perbaikan kode, tentu lebih baik untuk melakukan pengujian secara keseluruhan. Namun, seluruh proses pengujian memakan waktu dan biaya. Untuk alasan ini, perlu untuk memilih beberapa yang memiliki kemungkinan terkena dampak dan harus segera disesuaikan dengan teknik pemilihan uji regresi .

2. 9. *Decision tree*

Decision tree adalah salah satu metode klasifikasi populer karena ini lebih mudah dipahami oleh manusia. Model pohon keputusan membuat prediksi berdasarkan struktur pohon atau struktur hierarkis. Model ini dibagi oleh root, branch dan node yang setiap node mewakili prediksi kelas dan apakah mungkin untuk node menjadi *parent* dari node lain di bawahnya dan branch mewakili konjungsi ke node (Sharma and Kumar, 2016). Ramadika Jankovic melakukan penelitian untuk mengklasifikasikan citra warisan clutural menggunakan pohon keputusan. Dalam penelitian ini, metode yang diusulkan adalah pohon keputusan C4.5 untuk mengklasifikasikan gambar. Metode ini menggunakan pencarian

kompleks untuk membuat model pohon keputusan dan paramater dapat diubah untuk mendapatkan akurasi yang lebih baik (Janković, 2019).

Nilai akurasi yang tinggi sering kali terjadi pada klasifikasi menggunakan *decision tree*. Sebuah penelitian yang melakukan klasifikasi potensi pengidap diabetes pada data riwayat kesehatan pasien menggunakan *decision tree* mencapi akurasi 82,41%, bahkan metode ini dianggap paling membantu dibandingkan dengan strategi pengenalan potensi pengidap diabetes lainnya pada saat itu (Kanmani and Murugan, 2020). Pada penilitian lain, klasifikasi pada ribuan kondisi operasional stabilitas *power system voltage* menggunakan *decision tree* juga mencapai nilai akurasi yang sangat tinggi, yaitu hingga 99,06% (Vanfretti and Arava, 2020). Model *decision tree* serupa yang digunakan pada penelitian ini, yaitu CART menunjukkan performa lebih baik dibandigkan dengan ANN, Naïve Bayes, C4.5, SVM, dan J48 dalam melakukan klasifikasi pasien berpotensi menderita coronary artery disease (CAD) dengan nilai akurasi sebesar 92,41% (Ghiasi, Zendehboudi and Mohsenipour, 2020).

2. 10. *Evaluation Metrics*

Untuk mengetahui seberapa baik metode sebuah pemodelan topik dijalankan, diperlukan pengukuran menggunakan akurasi, *precision,* dan *Recall*. Ketika sebuah model pembelajaran sebuah system dijalankan pada data set, maka akan terdapat confusion matrix seperti pada tabel Tabel 2.2. *Precision* (juga disebut nilai prediktif positif) adalah fraksi dari *instance* yang relevan di antara *instance* yang diambil seperti pada formula (2.7) (Dalianis and Dalianis, 2018). *Recall* (juga dikenal sebagai sensitivitas) adalah fraksi dari jumlah total instance yang relevan yang sebenarnya diambil seperti pada formula (2.8) (Dalianis and Dalianis, 2018). Dengan kata lain Precision untuk mengukur ketepatan informasi yang sebenarnya diinginkan dengan jawaban yang diberikan oleh fungsi model. Kemudian, *Recall* mengukur keberhasilan model untuk menemukan adanya informasi yang diinginkan.

Akurasi didefinisikan sebagai proporsi nyata baik positive maupun negative pada dari data keseluruhan, mengukur seberapa dekat hasil yang prediksi yang

benar dibandingkan dengan nilai sebenarnya seperti pada formula (2.9) (Dalianis and Dalianis, 2018). Bila data yang digunakan tidak seimbang jumlahnya pada setiap kelas maka diperlukan koefisien Cohen Kappa dimana $E(Z_k)$ adalah ekpektasi dari Z_k dalam populasi dari sebuah item dan $E_{ind}(Z_k)$ adalah ekspektasi yang mengasumsikan independensi statistic dari rating yang dibuat oleh dua pengamat (Vanbelle, 2017). Ditunjukkan pada formula (2.10).

Tabel 2.2 Confusion Matrix

		Nilai Sebenarnya	
		True	*False*
Nilai Prediksi	*True*	TP (*True Positive*) Hasil yang benar di prediksi	FP (*False Positive*) Hasil yang tidak diharapkan
	False	FN (*False Negative*) Hasil yang meleset dari nilai sebenarnya	TN (*True Negative*) Hasil yang di prediksi benar tidak sesuai

$$P = \frac{TP}{TP + FP} \tag{2.7}$$

$$R = \frac{TP}{TP + FN} \tag{2.8}$$

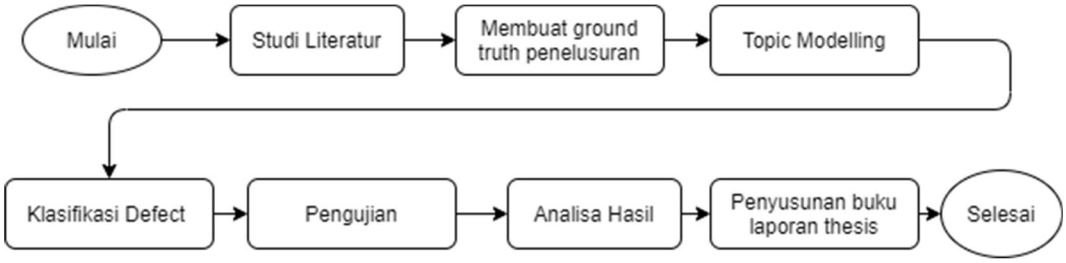
$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

$$\kappa = 1 - \frac{E(Z_k)}{E_{ind}(Z_k)} \tag{2.10}$$

(Halaman ini sengaja dikosongkan)

BAB 3
METODOLOGI PENELITIAN

Bab ini akan memaparkan tentang metodologi penelitian yang digunakan pada penelitian ini, yang terdiri dari (1) studi literatur, (2) penelitian, dan (3) dokumentasi. Ilustrasi alur metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram alir tahapan penelitian

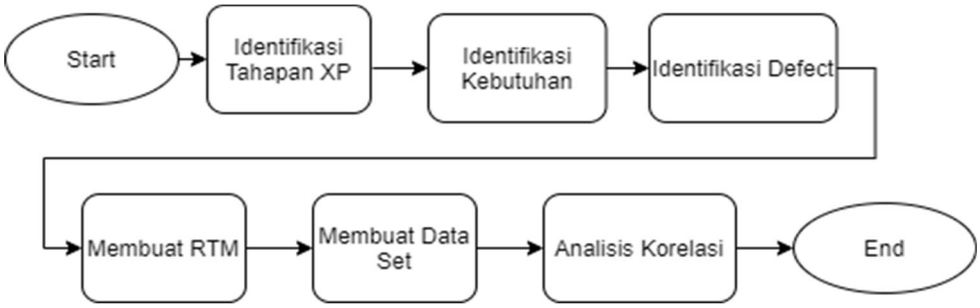
Penjelasan tahapan metode penelitian akan diterangkan secara terperinci pada sub bab berikut.

3. 1. Studi Literatur

Penelitian diawali dengan studi literature, yaitu mempelajari sistem penelusuruan kebutuhan yang telah dilakukan pada penelitian-penelitian sebelumnya. Kemudian bagaimana ketika penelusuran diterapkan pada proses pengembangan agile, tahapan apa saja yang dilewati ketika menerapkan kebutuhan hingga ke sumber kode dan bagaimana identifikasi *defect* yang ada. Beberapa metode pemodelan topik dan IR perlu dipelajari juga untuk membuat penelusuran antar dokumen kebutuhan, dokumen pengujian, dan sumber kode.

3. 2. Membuat *Ground truth* Penelusuran

Metode penelusuran kebutuhan yang paling mudah dan sering digunakan adalah RTM. Pada penelitian kali ini, RTM akan dibuat secara manual berdasarkan artifak pengembangan aplikasi yang ada. Tahapan pembuatan RTM terdapat pada Gambar 3.2 .



Gambar 3.2 Diagram Alir pembuatan *ground truth* penelusuran

3.2.1. Identifikasi Tahapan XP

Metode pengembangan XP memiliki tahapan yang sedikit dan iterasi terus menerus untuk setiap tahapan secara cepat. Tahapan XP yang akan diteliti adalah sebagai berikut :

- proses penjabaran kebutuhan oleh pengguna melalui *user story* di awal proses pengembangan perangkat lunak,
- proses pengujian baik pengujian unit, integrasi, dan regresi,
- pengembangan kode berbasis pengujian beserta proses iterasinya yang cepat.

User story akan berkembang menjadi *User story point* berupa penjelasan kebutuhan secara teknis oleh tim pengembang dengan tentunya melibatkan pengguna untuk memberikan timbal balik. Proses komunikasi tim pengembang dan pengguna berjalan secara intens dalam pertemuan singkat setiap dua minggu. Pada pertemuan ini dihasilkan catatan perubahan dan perbaikan proses pengembangan. Proses pengujian unit akan berlangsung secara otomatis menggunakan kode yang dibuat tim pengembang. Sementara itu, pengujian integrasi dan regresi akan melibatkan pengguna. Proses pengembangan kode akan berlangsung secara *pair-programming* dimana dua programmer akan bekerja pada modul yang sama dan saling memberikan masukan. Perbaikan dan pembaharuan kode akan tersimpan ke dalam subversion beserta komentar setiap programmer.

3.2.2. Identifikasi Kebutuhan

Penelitian menggunakan data pengembangan perangkat lunak aplikasi kepegawaian oleh lembaga pemerintah. Catatan penting ketika pertemuan dan umpan balik dari pengguna selama proses pengembangan diproses untuk menemukan kebutuhan yang harus dipenuhi oleh perangkat lunak tersebut. Kebutuhan dibagi menjadi dua jenis, fungsional dan non-fungsional. Kemudian berdasarkan waktu kemunculannya dalam proses iterasi diklasifikasikan ke dalam penambahan atau perubahan. Penting untuk dicatat semua dokumen ditulis dalam Bahasa Indonesia. Pada Gambar 3.3 semua kebutuhan dicatat sebagai *user story* dan didetailkan kembali menjadi poin-poin teknis, yang kemudian disebut sebagai *user story point*.

Kode	Nama Kebutuhan Awal		Kode	Rincian Kebutuhan Level 1	Parent	Status Kebutuhan		Kode	Rincian Kebutuhan Level 2	Parent	Status Kebutuhan
K1	Melakukan scan dokumen yang diperlukan untuk setiap pegawai lalu diunggah ke dalam sistem		K1.1	Menghubungkan scanner dengan aplikasi E-Doc	K1	High		K1.3.1	Memilih atribut pegawai pemilik dokumen	K1.3	High
K2	Melakukan validasi dokumen yang di unggah secara pribadi oleh pegawai		K1.2	Melakukan penyimpanan fisik dokumen ke file repository aplikasi e-doc	K1	High		K1.3.2	Memilih klasifikasi dokumen	K1.3	High
K3	Melakukan upload dokumen dari file komputer		K1.3	Melakukan penyimpanan atribut file ke database	K1	High		K1.2.1	Membuat formulir pemilihan dokumen dari hasil scan	K1.2	High
K4	Melakukan Hapus File baik yang diunggah sendiri Maupun yang diunggah oleh pegawai		K2.1	Melakukan Filtering atau penanda pada masing-masing status dokumen	K2	High		K1.2.2	Melakukan pemilihan dokumen hasil scan	K1.2	High
			K2.2	Melakukan update status dokumen diterima menjadi valid	K2	High					
NFR			K2.3	Melakukan update status dokumen yang ditolak	K2	High					
1	Menggunakan plugin TWAIN berbasis Web		K3.1	Membuat formulir upload unggah dokumen untuk setiap pegawai	K3	Medium					
2	Menggunakan template adminLTE yang memudahkan pengguna		K3.2	Melakukan pembatasan jenis dokumen hanya PDF saja	K3	Medium					
3	Menggunakan server windows		K4.1	Membuat formulir penghapusan file	K4	Medium					
4	Menggunakan dynamic security pada PHP maker		K4.2	Menghapus atribut file pada database dan repository aplikasi	K4	Medium					
			K4.3	Memberikan konfirmasi penghapusan file	K4	Medium					

Gambar 3.3 Kumpulan *user story* dan *user story point*

3.2.3. Identifikasi *Defect*

Dikarenakan tahapan pengujian pada XP terdapat pengujian otomatis. maka pengujian otomatis ini akan di dokumentasikan ke dalam dokumen test case seperti pada Gambar 3.5. Dokumen test case ini nantinya akan mencakup tiga jenis pengujian sebagai berikut :

- Uji unit, pada pengujian unit dilakukan pada setiap fungsi yang dijalankan. Pengujian ini dilakukan ketika di awal tahapan pada sumber kode yang nantinya akan dikembangkan menjadi fungsi utuh.
- Uji Integrasi, pada pengujian integrasi keseluruhan fungsi kode yang saling berkaitan akan diuji sejauh mana telah sesuai dengan kebutuhan.
- Uji Regresi, pengujian regresi adalah pengujian yang kembali dilakukan ketika terdapat perubahan atau penambahan pada kebutuhan (Tarhini, Ismail and Mansour, 2008).

Pengujian otomatis dilakukan dengan PHP-Unit dan subversi menggunakan svn tortoise. Kode tes akan dijalankan pada beberapa unit kode. Salah satu contohnya adalah seperti pada Gambar 3.4 adalah kode tes ambilniptest.php yang dijalankan untuk menguji apakah kode ambilnip.php sudah berjalan dengan benar. Fungsi yang akan diuji adalah mencari nip18 berdasarkan nip yang dimasukkan. Kemudian pertama kumpulan data palsu terdiri dari nip18, nip, dan nama variabel array $ eventRow. Kemudian fungsi pencarian nip carinip () dijalankan, ketika nip dimasukkan sesuai dengan data palsu menggunakan fungsi phpunit default assertEquals harus menghasilkan nip yang cocok pada kumpulan data palsu. Jika pengujian ini berjalan dengan baik tanpa *error*, maka kode unit ambilnip.php dapat berjalan sesuai fungsi yang diharapkan.

Pada saat melakukan dokumentasi pengujian akan tertera juga kegagalan yang ada pada kolom *Actual Result* dengan status *Fail.* Kegagalan ketika hasil dari pengujian tidak sesuai dengan yang diharapkan dalam memenuhi kebutuhan akan dikumpulkan menjadi *defect.* Pengujian yang dilakukan ada tiga tahapan, yaitu pengujian unit, pengujian integrasi, dan pengujian regresi.

```php
require 'ambilnip.php';

class AmbilnipTests extends PHPUnit_Framework_TestCase
{
    private $ambilnip;

    protected function setUp()
    {
        $this->nip = new nip();
    }

    protected function tearDown()
    {
        $this->nip = NULL;
    }

    public function carinip() {
  //Set fake data
  $eventRow = array(
    'nip' => 3451,
    'nip18' =>3456789,
    'nama'=>'nuraisa'
  );

  //$dao = $this->getMock('EventDao');
  $ambilnip = $this->getMockBuilder('nip')
                  ->getMock();

  //Set up the mock to return the fake row when findById is called
  $ambilnip->expects($this->once())
      ->method(carinip)
      ->with(nip18)
      ->will($this->returnValue($eventRow));

  //$repo = new EventRepository($dao);

  //$event = $repo->nip();

  //Confirm ID of the returned Event is correct
  $this->assertEquals(3451, $event->getId());
}
}
```

Gambar 3.4 Contoh Kode Pengujian Unit

Test Case ID	TC-P1	Test Case Description	Menguji plugin Twain untuk scan dokumen		
Created By		Reviewed By		Version	1

QA Tester's Log

Tester's Name		Date Tested	March 18, 2015	Test Case (Pass/Fail/Not Executed)	Fail

S #	Prerequisites:
1	Browser Chrome
2	Bahasa pemrograman PHP
3	
4	

S #	Test Data Requirement
1	Dokumen lebih dari satu halaman
2	
3	
4	

Test Conditions

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Membuka menu upload operator	driver scanner muncul pada field	as expected	Pass
2	Klik tombol pindai	muncul pop up halaman untuk melakukan scan	as expected	Pass
3	Klik tombol pindai pada pop up	mesin scan melakukan pindai dan hasil dokumen muncul pada kotak	Tampil sebagian	Fail

Gambar 3.5 Contoh Format Dokumen Test Case

3.2.4. Membuat RTM

Penulis akan mengkompilasi RTM dari studi sebelumnya (Putro and Wibowo, 2018) (Jeong, Cho and Lee, 2018) dan menambahkan kolom *defect* yang dihasilkan. Jadi artefak yang perlu dipertimbangkan adalah kebutuhan, ketergantungan antar kebutuhan, dokumen *test case*, kode apa pun yang terkait dengan *test case* , tabel terkait dalam database, dan *defect* nya. Pada RTM id kebutuhan pada kolom kebutuhan, id dokumen kasus pengujian pada kolom pengujian, id tabel yang terkait pada kolom tabel, sumber kode terkait pada kolom *code*, dan kegagalan fungsi akan dimasukkan kedalam kolom *defect*.

3.2.5. Membuat Data Set

Pelacakan telah dibangun di atas RTM, serta ketergantungan dokumen di antara persyaratan. Dari dua dokumen, akan terlihat dalam satu persyaratan berapa banyak *story point* akan muncul, berapa banyak dari mereka telah berubah atau meningkat, jumlah kasus uji yang dilakukan dalam 3 tahap pengujian (pengujian unit, pengujian integrasi, sistem pengujian), pengujian regresi, waktu penyelesaian, dan jumlah *defect*.

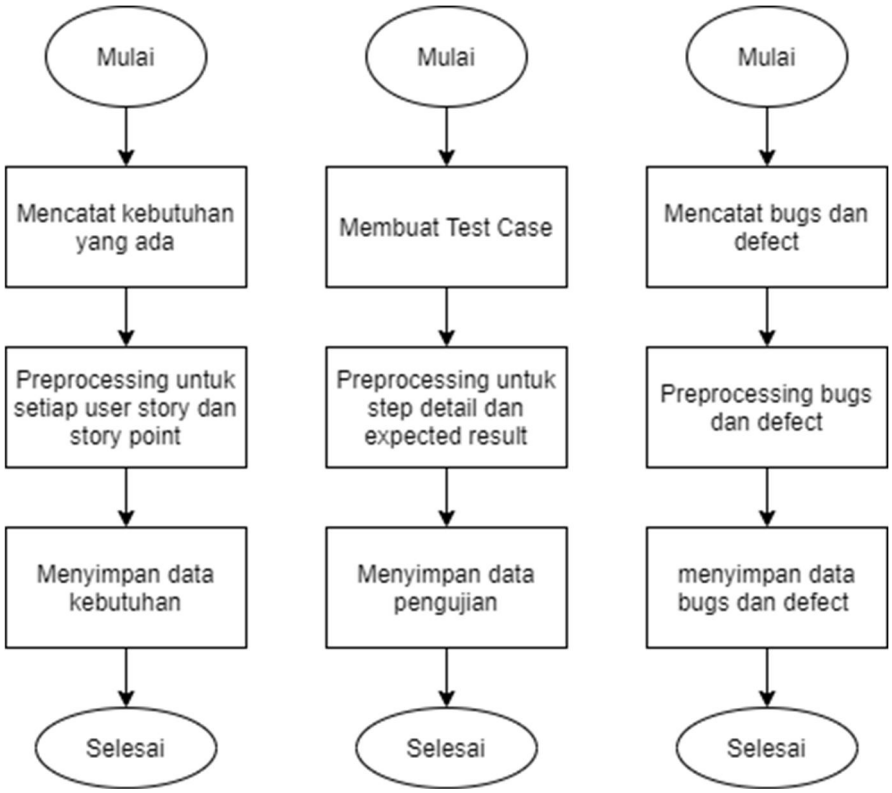
3.2.6. Analisis Korelasi

Variabel yang berhubungan dengan terjadinya *defect* kemudian di analisa, baik yang berasal dari tahapan pengembangan, faktor eksternal yang mempengaruhinya, dan artifak yang dihasilkan. Hal ini untuk memperkuat apakah RTM yang telah dibuat memang memiliki korelasi dengan terjadinya *defect*. Korelasi nantinya akan dihitung secara kuantitatif menggunakan korelasi Pearson (Agarwal, Patyal and Nindrajog, 2015). Data set yang ada akan dimasukkan kedalam python dan dilihat seberapa besar korelasi yang ditimbulkan antar variabel. *Defect* yang terkumpul sendiri juga akan di klasifikasikan berdasarkan dampaknya. Klasifikasi yang dilakukan adalah klasifikasi biner, yaitu major dan minor.

3. 3. Pemodelan Topik

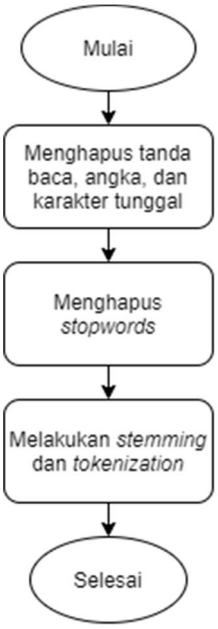
3.3.1. Preprocessing

Dokumen kebutuhan, test case, hingga sumber kode menggunakan Bahasa tulis. Sehingga dibutuhkan ekstraksi teks untuk melihat kemiripan kata dan keterkaitannya. Keterkaitan kata-kata inilah yang nantinya akan mempermudah dan memungkinkan otomatisasi penelusuran kebutuhan. Terdapat tiga artifak yang akan di ekstrak kontennya, yaitu dokumen *user story* dan test case. Proses ektraksi ketiganya tergambar pada Gambar 3.6.



Gambar 3.6 Diagram alir preprocessing artifak

Ketiganya akan melalui tahapan preprocessing terlebih dahulu, untuk menghilangkan karakter hingga kata yang tidak diperlukan. Proses preprocessing akan mengikuti tahapan pada Gambar 3.7.



Gambar 3.7 Diagram alir tahapan Preprocessing

Tanda baca tidak diperlukan untuk melihat kemiripan kata dan tidak menggambarkan tujuan apapun dari dokumen sehingga harus dihilangkan agar tidak dihitung. Kata penghubung tidak hanya menghubungkan antar kalimat dan bukan inti dari objek kata yang akan dicari kemiripannya, maka harus dihilangkan juga. *Stemming* diperlukan untuk mencari akar kata, hal ini dikarenakan kemiripan tidak akan terdeteksi ketika mendapatkan imbuhan. *Tokenization* diperlukan untuk memecah kalimat menjadi per kata.

Hasil preprocessing kata pada *user story* akan masuk dijadikan satu table dengan kolom Requirement, setiap *user story* berasal dari dokumen catatan pertemuan dan diskusi yang ada. Dokumen tersebut tentunya akan memiliki *Code* untuk memudahkan pengenalan. Status kebutuhan pada *user story* juga akan disimpan, yaitu prioritas tinggi, sedang, atau rendah. Kebutuhan juga berada pada dua kategori yaitu perubahan atau penambahan. Pada kolom terakhir, adalah kelompok modul yang manakah kebutuhan tersebut.

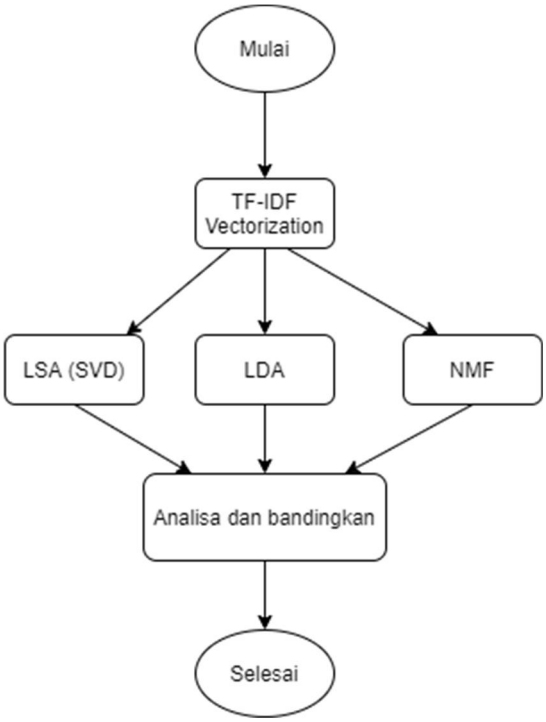
Hasil preprocessing kata pada pengujian akan masuk pada satu tabel dengan kolom untuk detail langkah (*step detail)* dan hasil yang diharapkan (*expected result)* akan disimpan secara terpisah karena akan diproses secara

terpisah pula. Hasil *preprocessing step detail* Setiap pengujian akan diberikan identitas *Code* untuk memudahkan berasal dari dokumen pengujian yang mana . Pada kolom terakhir, adalah masuk kedalam kelompok modul yang manakah kebutuhan tersebut.

Hasil preprocessing kata pada *bug*, *defect*, dan expected result pada pengujian akan masuk pada satu tabel dengan kolom *content*. Kemudian pembeda apakah *content* merupakan *bug*, *defect*, ataukah *expected result* dimasukkan kedalam *type.* Setiap kata pada konten akan diberikan identitas *Code* untuk memudahkan berasal dari dokumen pengujian yang mana. Pada kolom terakhir, adalah asal kelompok modul. Sumber kode juga akan dikumpulkan, tetapi tidak melalui *preprocessing* pada tabel tersendiri dengan kode fungsi penting akan masuk pada kolom *source code*. Sementara itu nama dari filenya akan masuk pada kolom name. Pada kolom terakhir, adalah asal kelompok modulnya.

3.3.2. Pembobotan kata dan kemiripan

Tahapan selanjutnya tertera pada Gambar 3.8. Kemunculan kata pada setiap dokumen dan keunikan katanya akan menjadi langkah awal melihat kemiripan konten antara artifak yang satu dengan artifak lainnya.



Gambar 3.8 Tahapan Pemodelan Topik Dokumen

Setiap konten akan dikonversi, direpresentasikan sebagai daftar *token* (*lemmas*) di atas, menjadi vektor yang dapat dimengerti oleh model *machine learning*. Hal ini akan dilakukan dalam tiga langkah menggunakan model *bag-of-words*:

1. Hitung berapa kali sebuah kata muncul di setiap pesan (Dikenal sebagai frekuensi istilah)
2. Membobot jumlah, sehingga token yang sering mendapatkan bobot yang lebih rendah (frekuensi dokumen terbalik)
3. Normalisasi vektor ke panjang satuan, untuk abstrak dari panjang teks asli (norma L2)

Setiap vektor akan memiliki dimensi sebanyak adanya kata-kata unik dalam tabel ekstraksi. Konversi menggunakan *CountVectorizer SciKit Learn*, model ini akan mengonversi kumpulan dokumen teks menjadi matriks jumlah token. Dapat di analogikan sebagai matriks 2-Dimensi. Di mana 1-dimensi adalah seluruh kosakata (1 baris per kata) dan dimensi lainnya adalah dokumen yang sebenarnya, dalam hal ini kolom per kalimat teks, seperti pada tabel dibawah ini

Table 3.1 Sparse Matrix

	Kalimat 1	Kalimat 2	Kalimat 3
Perhitungan kata 1	0	0	2
Perhitungan kata 2	1	1	0
Perhitungan kata N	0	1	1

Biasanya, bobot TF-idf terdiri dari dua istilah: yang pertama menghitung *Term Frequency* (TF) yang dinormalisasi, dengan kata lain berapa kali sebuah kata muncul dalam dokumen, dibagi dengan jumlah total kata dalam dokumen tersebut. Istilah kedua adalah Inverse *Document Frequency* (IDF), dihitung sebagai logaritma dari jumlah dokumen dalam korpus dibagi dengan jumlah dokumen tempat istilah tertentu muncul. Misalkan dokumen yang berisi 100 kata dimana kata pindai muncul 3 kali. Frekuensi istilah (TF) untuk pindai adalah (3/100) = 0,03. Anggaplah terdapat 10 juta dokumen dan kata pindai muncul dalam seribu dokumen ini. Kemudian, frekuensi dokumen terbalik (Idf) dihitung sebagai log (10,000,000 / 1.000) = 4. Dengan demikian, bobot Tf-idf adalah : 0,03 * 4 = 0,12.

Cara lain untuk melihat kemiripan dan keterkaitan konten adalah dengan metode LSA. LSA mempelajari topik laten dengan melakukan dekomposisi matriks pada matriks term-dokumen menggunakan dekomposisi nilai singular (Nanang, Penelusuran and Penelusuran, 2018). LSA biasanya digunakan sebagai teknik pengurangan dimensi atau pengurangan noise. Metode pada python yang akan dipakai adalah Singular Value Decomposition (SVD). SVD menguraikan matriks menjadi tiga matriks yang berbeda: matriks kolom ortogonal, matriks baris ortogonal dan satu matriks tunggal. Terdapat beberapa opsi untuk menentukan jumlah topik optimal

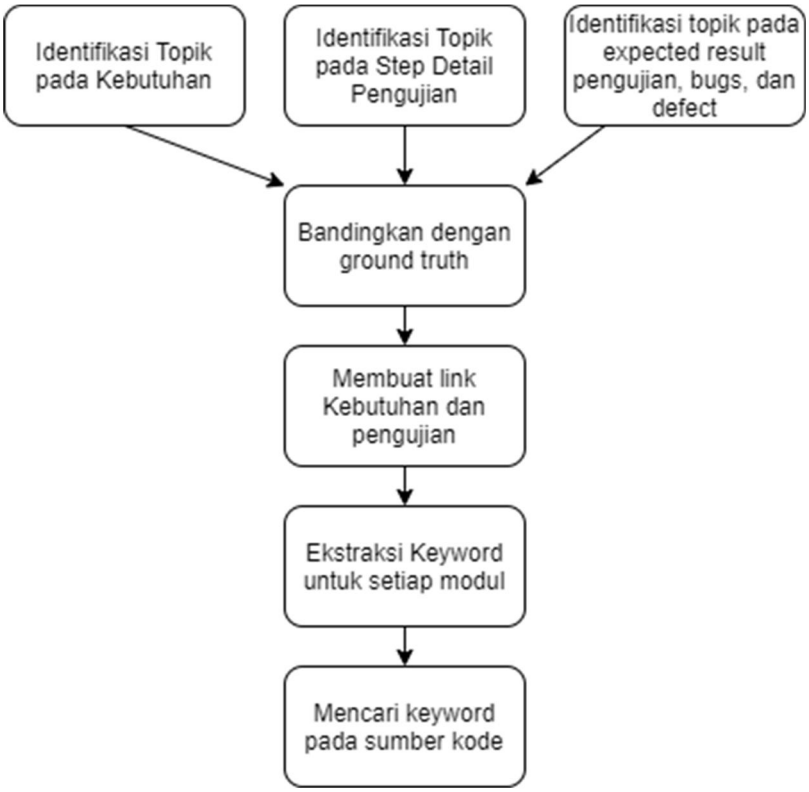
- mempertimbangkan setiap topik sebagai sebuah kluster dan mencari tahu keefektifan suatu kluster dengan menggunakan koefisien Silhouette.
- Ukuran koherensi topik adalah ukuran realistis untuk mengidentifikasi jumlah topik.

Melakukan analisa topik juga dapat dilakukan dengan LDA. Pendekatan LDA untuk pemodelan topik adalah mempertimbangkan setiap dokumen sebagai kumpulan topik dalam proporsi tertentu. Dan setiap topik sebagai kumpulan kata kunci, sekali lagi, dalam proporsi tertentu. Setelah algoritma dengan jumlah topik dijalankan, maka perlu mengatur kembali distribusi topik dalam dokumen dan distribusi kata kunci dalam topik untuk mendapatkan komposisi distribusi topik dan kata kunci yang baik. Input utama pada LDA adalah data corpus yang berasal dari TF-Idf. Proses generatif LDA dapat digambarkan sebagai mengingat jumlah dokumen *M*, jumlah kata *N*, dan jumlah topik *K* sebelumnya, model melatih untuk output:

- *psi*, distribusi kata untuk setiap topik *K*
- *phi*, distribusi topik untuk setiap dokumen *i*

Cara yang terakhir adalah bentuk optimalisasi dari metode sebelumnya, yaitu NMF. Pada metode ini nilai negatif dihilangkan pada matriks yang secara simultan mengurangi dimensi dan kemudian melakukan clustering. Pada penelitian sebelumnya NMF merupakan algoritma yang lebih cepat mengeluarkan hasil dibandingkan LDA, meskipun hasil yang diberikan oleh LDA dinilai lebih baik dalam melakukan clustering (Suri and Roy, 2017). Hasil LSA, LDA, dan NMF berupa kumpulan kata dalam beberapa topik yang jumlahnya sudah ditentukan sebelumnya. Topik disusun berdasarkan modul yang tertera pada *user story*.

Untuk itu jumlah topik disesuaikan dengan jumlah modul. Setiap dokumen di kenali topiknya. Kata dan istilah yang digunakan pada dokumen kebutuhan, pengujian, dan sumber kode tidaklah sama. Terlihat sekilas kemiripan yang ada sangatlah sedikit. Maka proses pengerjaannya adalah seperti pada gambar Gambar 3.9. Identifikasi topik untuk kebutuhan dilakukan terpisah dengan identifikasi topik *step detail* pengujian dengan k topik dan kategori topik yang sama. Setelah itu, metode *preprocessing* yang paling sesuai diterapkan untuk pemodelan topik untuk *expected result, bugs,* dan *defect.*



Gambar 3.9 Tahapan Pemodelan Topik Artifak Dokumen

Kemudian dilihat apakah dokumen berada pada topik yang sesuai dengan *ground truth* RTM sebelumnya. Ketiga metode di uji coba pada dataset yang sama sehingga tracing link terbentuk, proses penelusuran mulai dari *defect* hingga ke level kebutuhan hingga letaknya pada sumber kode. Pemodelan topik ini membantu melakukan pemetaan secara cepat dan juga untuk mengetahui bagaimanakah pernyataan kebutuhan yang benar.

3. 4. Klasifikasi *Defect*

Defect dibagi menjadi dua kelas, yaitu minor dan major. *Defect minor* adalah kegagalan fungsi yang dapat dilewati dan tidak terlalu berpengeruh terhadap aliran data dan menggagalkan fungsional tertentu pada suatu modul. Contoh *defect minor* adalah membuka halaman yang tidak seharusnya ketika melakukan aktifitas penyimpanan, notifikasi yang tidak muncul ketika menghapus atau menyimpan data sehingga pengguna tidak mengetahui status data yang dikerjakan, dan tampilan formulir yang terlalu kecil dan menyulitkan pengisian data. *Defect major* adalah

kegagalan fungsi yang mengakibatkan terhentinya aliran data sehingga pengguna tidak dapat menggunakan sistem sebagaimana mestinya.

Contoh *defect* major adalah nilai suatu atribut yang gagal tersimpan sehingga mengakibatkan keseluruhan data tidak dapat di proses lebih lanjut, data yang telah disimpan ternyata tidak sesuai atau bahkan rusak, dan plugin tidak berfungsi untuk memproses data. Setelah seluruh *defect* yang ada di klasifikasikan menjadi major dan minor oleh pakar. Berdasarkan klasifikasi inilah dilihat pola terjadinya *defect* berdasarkan penelusuran yang telah dibuat sebelumnya. Faktor apa sajakah yang menyertai timbulnya *defect* major dan minor dari tahapan pengujian hingga kebutuhan. Algoritma klasifikasi sederhana *Decision tree* diterapkan. Atribut yang muncul berdasarkan analisis dokumen tertulis adalah :

- Banyaknya *user story*
- Panjang *user story*
- Berapa banyak kata-kata yang ada pada *user story* muncul pada pengujian dan *defect* untuk melihat konsistensi penerapan
- *User story point* yang berisi hal teknis apakah tercantum pada komentar sumber kode atau bahkan variabel di kode

Sementara itu atribut yang muncul berdasarkan analisis proses pada setiap tahapan adalah :

- Berapa banyak perubahan atau penambahan kebutuhan terjadi
- Berapa banyak pengujian yang dilakukan dan apakah semua fungsi telah di uji.
- Berapa banyak waktu yang diperlukan untuk perbaikan fungsi hingga dinyatakan lulus pengujian

3. 5. Pengujian

Data training dan testing berasal dari dokumen yang terlibat pada penelusuran berdasarkan *ground truth*. Topik modelling dilakukan menggunakan 3 metode LSA, LDA, dan NMF. Hasil dari pemodelan topik dibandingkan dengan *ground truth* yang ada. Sementara itu nilai pada setiap metode topik modelling

dilihat akurasi, *Precision*, dan *Recal l*nya. Baik *precision* dan *recall* didasarkan pada pemahaman dan ukuran relevansi. Sementara itu untuk megukur performa pemodelan topik dengan dataset dengan jumlah data yang tidak seimbang pada setiap kelasnya menggunakan koefisien Cohen Kappa. Nilai akurasi, *precision*, dan *recall* juga menjadi ukuran performa metode klasifikasi *defect* menggunakan *decision tree*.

(Halaman ini sengaja dikosongkan)

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan mengenai lingkungan pengembangan perangkat lunak dengan metode XP yang digunakan pada penelitian beserta paparan hasil penerapan metode seperti yang dijelaskan pada BAB 3. Dalam bab ini juga dijelaskan analisis terhadap hasil tersebut untuk kemudian menjadi masukan untuk perbaikan pengembangan aplikasi dengan metode XP selanjutnya.

4. 1. Karakteristik Data

Terdapat dua aplikasi perangkat lunak yang diambil data dan proses pengembangannya, yaitu sebagai berikut :

a) E-Document, merupakan aplikasi kepegawaian yang dikembangkan dengan Bahasa PHP 5 diatas platform PHP MAKER 12. Aplikasi ini dikembangkan oleh 2 orang Programmer
b) E-DAKIN, merupakan aplikasi pelaporan sederhana yang dikembangkan dengan Bahasa pemrograman PHP 7 diatas platform Yii 2. Aplikasi ini juga dikembangkan oleh 2 orang Programmer.

Jumlah tim pengembang adalah 5 orang dengan rincian sebagai berikut :

- 1 orang project manager
- 1 orang business analyst (dari pihak klien yang dilibatkan)
- 1 orang system analyst
- 2 orang Programmer
- 1 orang QA.

4. 2. Membuat RTM (*ground truth*)

Hal paling pertama yang dilakukan adalah melakukan analisis terhadap kebutuhan. Kebutuhan umum berasal dari pengguna dan business analyst berupa identifikasi tahapan proses bisnis yang diterapkan, penjelasan kebutuhan belum

detail dan berupa gambaran secara garis besar alur bagaimana perangkat lunak dapat membantu kinerja pegawai. Sementara itu, kebutuhan secara teknis muncul dari System Analyst dan masukan dari programmer. Kebutuhan teknis memberikan gambaran lebih detil mengenai atribut apa saja yang terlibat, bagaimana atribut tersebut di proses, dan penerapannya kedalam sumber kode. Seperti penjelasan mengenai jenis data pegawai, status dokumen, dan proses penyimpanan ke dalam database.

Pertama-tama memiliki kumpulan kebutuhan yang disepakati untuk tidak terlalu detail antara pengguna dan tim pengembang. Kebutuhan berdasarkan gambaran awal ini disimpan dan diberi kode dan terus berubah selama proses pengembangan, yang kemudian disebut sebagai *user story* seperti yang tertera pada Tabel 4.1. Tim pengembang menjelaskan lebih lanjut tahapan pelaksanaan teknis pada setiap poin penting yang ada. Setiap *user story* dipecah menjadi beberapa poin penting yang kemudian disebut sebagai *story point*. Setiap *story point* memiliki ketergantungan satu sama lain, serta memiliki urgensi penerapan yang berbeda. Hal ini untuk memudahkan tim pengembang dalam memprioritaskan implementasi, perbaikan implementasi, dan tentunya mengantisipasi jika ada perubahan. Tingkat prioritas dapat berubah ketika perubahan dan penambahan kebutuhan dengan prioritas tinggi sebelumnya telah terpenuhi. Hal ini untuk memastikan bahwa implementasi fungsi utama harus dipenuhi sebelum fungsi lain dikerjakan.

Tabel 4.1 *User story* Aplikasi

User story Aplikasi 1 : Aplikasi ini dikembangkan untuk mengelola file-file penting dan memudahkan biro SDMO dalam pemrosesan hal-hal yang berhubungan dengan kepegawaian. File- file tersebut perlu diubah menjadi bentuk digital dan diatur dengan aplikasi ini dikarenakan permasalahan sebagai berikut : 1. File dalam bentuk kertas susah dikelola dan terkadang terselip atau hilang. 2. Mengumpulkan file dan memproses file kepegawaian yang begitu banyak terkadang akan terlewat dan memakan banyak waktu

File ini akan menyimpan file penting terkait kepegawaian dalam bentuk digital. File ini digunakan oleh kedua belah pihak. Pegawai dapat melihat file pribadinya dan file yang di unggah oleh SDMO. Sehingga mudah untuk mencari file penting terkait kepegawaiannya. SDMO dapat mengunggah file dapat mengolah file menjadi digital, mengunggah, dan kemudian melakukan pemrosesan sesuai dengan kebutuhan kepegawaian
User story Aplikasi 2 : Dalam rangka terselenggaranya good governance diperlukan pengembangan dan penerapan sistem pengelolaan data yang tepat, jelas dan terukur sehingga penyelenggaraan pemerintahan dan pembangunan dapat berlangsung secara berdaya guna, berhasil guna, bersih dan bertanggung jawab. Pembangunan aplikasi evaluasi kegiatan ini bertujuan untuk membantu pengelolaan data kegiatan DIPA yang dilakukan di Badan Pengkajian dan Penerapan Teknologi secara elektronis dengan basis data yang telah tersimpan secara parsial di lingkungan wali data yaitu biro Perencanaan dan Keuangan. Fungsi utama aplikasi Evaluasi kegiatan yang dikembangkan adalah : 1. Menyediakan antarmuka yang dinamis sesuai dengan kebutuhan penginputan data laporan akhir kegiatan yang diterima oleh wali data dari unit atau satuan kerja yang terkait; 2. Menyediakan antarmuka yang dinamis sesuai dengan kebutuhan penampilan informasi kegiatan yang memiliki kata kunci yang sama berdasarkan tahun pelaksanaan kegiatan tersebut; 3. Menyediakan antarmuka untuk master link kegiatan-kegiatan yang sejenis sesuai dengan rencana pelaksanaan kegiatan tersebut.

4. 2. 1. Data Kebutuhan :

Story point aplikasi pertama

Aplikasi ini memiliki tiga modul utama, yaitu :

- Manajemen dokumen, terdiri dari pengunggahan dokumen melalui sistem scan dan/atau pencarian langsung pada direktori file computer oleh pegawai atau operator. Penyimpanan, perubahan, dan penghapusan dokumen juga diatur pada modul ini.
- Validasi dan status dokumen, pada modul ini status dokumen di tentukan untuk mengenali tahapan pemrosesan dokumen. Validasi oleh operator juga diberlakukan pada untuk menentukan dokumen mana yang memenuhi persyaratan untuk diproses lebih lanjut, seperti keaslian dokumen dan kesesuaian format dokumen.
- Kenaikan pangkat, pada modul ini proses pengajuan kenaikan pangkat diproses. Daftar dokumen persyaratan ditampilkan. Pengecekan dilakukan pada data dokumen yang diunggah oleh pegawai yang mengajukan maupun operator.

Dalam proses implementasi terdapat tambahan kebutuhan untuk memperjelas tahapan pelaksanaan teknis. Ketika sebuah siklus pengembangan dimulai, satu per satu detail ditambahkan, diubah atau bahkan dihilangkan. Maka kumpulan *story point* pada kebutuhan tertera pada Tabel 4.2.

Tabel 4.2 Contoh *Story point* Aplikasi Pertama

ID	*Story point*	Depend	Prioritas	Change/Add
M1	Aplikasi ini dikembangkan untuk mengelola file-file penting dan memudahkan biro SDMO dalam pemrosesan hal-hal yang berhubungan dengan kepegawaian. File- file tersebut perlu diubah menjadi bentuk digital dan diatur dengan aplikasi ini dikarenakan permasalahan sebagai berikut : 1. File dalam bentuk kertas susah dikelola dan terkadang terselip atau hilang. 2. Mengumpulkan file dan memproses file kepegawaian yang begitu banyak terkadang akan terlewat dan memakan banyak waktu		High	-
M2	Aplikasi ini akan menyimpan file penting terkait kepegawaian dalam bentuk digital. File ini digunakan oleh kedua belah pihak. Pegawai dapat melihat file pribadinya dan file yang di unggah oleh SDMO. Sehingga mudah untuk mencari file penting terkait kepegawaiannya. SDMO dapat mengunggah file dapat mengolah file menjadi digital, mengunggah, dan kemudian melakukan pemrosesan sesuai dengan kebutuhan kepegawaian		High	-
M3	§ Input file pegawai § Pemutakhiran file pegawai melalui fungsi scan § Pemrosesan file pegawai untuk keperluan kepegawaian		High	-
M4	Scan Dokumen • Memilih nama pegawai • Memilih jenis dokumen yang terkait		High	-

M5	yang diupload ke server e-Filling adalah PPK Final dimana sudah ada tanda tangan dari semua pihak yang terlibat. Usul kami adalah Pegawai akses ke aplikasi SKP -> Cetak PPK -> Ditanda tangani oleh atasan langsung dan atasan dari atasan langsung -> Pegawai upload ke e-filling atau pegawai mengirim PPK tersebut ke SDMO untuk diupload oleh user SDMO ke e-filling	M3,M4	Medium	Add
M6	transkrip dapat ditambahkan. Namun meskipun dapat dicetak dari e-filling, transkip tersebut tetap harus dilegalisir asli sesuai peraturan yang berlaku. Pertanyaan kami : jika yang diupload adalah transkrip yang sudah ada legalisirnya, bagaimana keabsahan dari transkrip tersebut mengingat legalisirnya tidak asli alias hasil print-out ?	M3, M4	Medium	Add
M7	Sebaiknya dokumen resmi yang sudah diterbitkan dan diupload tidak dihapus dari database e-filling. Penghapusan hanya dilakukan untuk dokumen-dokumen yang tidak lolos validasi. Untuk kasus revisi SK, usul kami adalah SK yang sudah direvisi alias tidak berlaku lagi diubah flag-nya menjadi tidak bisa dilihat/dicetak lagi kecuali oleh admin. Namun SK tersebut tetap muncul dalam daftar SK.	M3	High	Change
M8	a. Setiap dokumen pribadi pegawai tersimpan dengan rapi dan terorganisir secara elektronik d. Setiap dokumen pegawai selalu terupdate	M3	High	-
M9	Membuat formulir upload unggah dokumen untuk setiap pegawai Melakukan pembatasan jenis dokumen hanya PDF saja. Membuat formulir penghapusan file. Menghapus atribut file pada database dan repository aplikasi. Memberikan konfirmasi penghapusan file	M7	Medium	Change
M10	Operator mampu memilih atribut pegawai pemilik dokumen dan memilih klasifikasi dokumen yang akan di upload baik melalui scan atau langsung melalui pencarian file pada computer.	M3,M4	High	-

Story point aplikasi kedua

Aplikasi ini memiliki 5 modul utama, yaitu

- Data Master, data ini merupakan data induk seperti nama unit, nama kegiatan, satuan kegiatan, nama mitra, nama bidang teknologi, dan nama bidang sektoral yang terkait dengan kegiatan
- Kaitan Kegiatan, pada data ini dibuat penelusuran kaitan antara satu kegiatan dengan kegiatan yang lain
- Unggah DIPA, merupakan modul untuk dapat mengambil data dari komponen atribut laporan DIPA berupa excel.
- Laporan kegiatan dan laporan anggaran, merupakan pengisian detail pengerjaan kegiatan dan anggaran yang terpakai pada kegiatan tersebut termasuk proses kerjasama dengan mitra terkait
- View laporan, merupakan modul untuk melihat laporan menyeluruh untuk setiap kegiatan terkait berdasarkan pencarian kata kunci, tahun, bidang tekonologi, dan bidang sectoral untuk memudahkan evaluasi.

Seperti pada aplikasi pertama, dalam proses implementasi terdapat tambahan kebutuhan untuk memperjelas tahapan pelaksanaan teknis. Ketika sebuah siklus pengembangan dimulai, satu per satu detail ditambahkan, diubah atau bahkan dihilangkan. Maka kumpulan *story point* pada kebutuhan tertera pada Tabel 4.3.

Tabel 4.3 Contoh *Story point* Aplikasi Kedua

ID	Requirement	Depend	Prioritas	Change/Add
LKR1	Setiap unit kerja akan membuat laporan kegiatan setiap tahunnya untuk kemudian diserahkan kepada biro perencanaan	MS3, MS4,MS5	high	
LKR2	Laporan setiap unit kerja akan mengacu pada kegiatan tahunan yang tercantum pada RKKL		high	change
LKR3	Laporan kegiatan harus ditentukan apakah termasuk kedalam prioritas nasional atau bukan	LKR1	high	add
LKR4	Setiap laporan kegiatan akan terhubung dengan output pada dipa beserta perincian anggarannya	LKR1	medium	add
LKR5	Pada laporan kegiatan, setiap kegiatan yang ada harus dipetakan sesuai dengan bidang teknologi yang mana.	MS4, MS5,	high	
LKR6	Merekam atribut pelaksanaan kegiatan seperti keluaran, kendala, dan hirarki antara satu kegiatan dengan kegiatan yang lain, baik dalam unit yang sama maupun berbeda.	LKR1,LKR4	high	
LKR7	Pada setiap laporan kegiatan menyertakan data dukung, seperti bukti gambar, rancangan prototype, dan lain sebagainya.	LKR1,LKR4,LKR5	medium	add
LKR8	Laporan kegiatan juga mencantumkan kendala yang dihadapi	LKR1,LKR4,LKR5	medium	
LKR9	Pada setiap laporan kegiatan harus mencantumkan posisi tahun berjalan, missal target pelaksanaan adalah 4 tahun maka pada saat laporan dibuat sudah sampai pada tahun keberapa	MS6,LKR1	medium	

4. 2. 2. Data Pengujian

Berdasarkan tiga fase pengujian dan matriks kualitas pengujian maka contoh penerapannya untuk aplikasi pertama tertera pada Tabel 4.4 dan pada aplikasi kedua tertera pada Tabel 4.5 di bawah ini :

Tabel 4.4 Contoh Pengujian Aplikasi Pertama

Kode	Step Detail	Expected Result
P	Membuka menu upload operator	driver scanner muncul pada field
P	Klik tombol pindai	muncul *pop-up* halaman untuk melakukan scan
P	Klik tombol pindai pada pop up	mesin scan melakukan pindai dan hasil dokumen muncul pada kotak tampilan yang tersedia
E	Membuka menu upload operator	field terlihat sesuai dengan gambar yang tercantum
E	Mengetik nip18 pegawai	pada field akan ada autocomplete untuk memilih nip18 pegawai
E	memilih nip18 pegawai	autofill field nip dan nama pegawai
S	Membuka menu upload operator	field terlihat sesuai dengan gambar yang tercantum
S	Mengetik nip18 pegawai	pada field akan ada autocomplete untuk memilih nip18 pegawai
S	memilih nip18 pegawai	autofill field nip dan nama pegawai
U	Melakukan klik menu melihat dokumen	Sub menu sesuai kategori dokumen muncul di bawahnya pada tree list
U	Membuka sub menu sesuai kategori dokumen yang diinginkan	Halaman list terbuka, menampilkan semua data dengan kategori tersebut
U	Menambah dokumen dengan melakukan klik tambah	Membuka halaman tambah

Tabel 4.5 Contoh Pengujian Aplikasi Kedua

Code	Step Detail	Expected Result
KG	Admin masuk ke menu master kaitan kegiatan	Formulir perubahan kaitan kegiatan akan muncul.
KG	Admin memilih tahun berjalannya kegiatan	Terdapat pilihan tahun pada field tahun yang dapat dipilih
KG	Admin mengetik nama kegiatan	Nama kegiatan yang akan ditambahkan akan muncul
KG	Admin mengetik kode unik kegiatan	Kode unik dapat terdiri dari huruf dan angka dan bukan spesial karakter
LK	Operator unit kerja memilih kegiatan yang tersedia	Terdapat pilihan kegiatan yang telah diinputkan sebelumnya pada menu kaitan kegiatan
LK	Operator unit akan melihat tampilan bidang prinas dan bidang teknologi kegiatan tersebut	Bidang prinas dan bidang teknologi untuk kegiatan yang dipilih akan tampil secara otomatis sesuai dengan yang diinputkan pada kaitan kegiatan.
LK	Operator unit memilih jenis prioritas nasional dari kegiatan tersebut	Terdapat pilihan jenis prioritas nasional untuk dipilih
LK	Operator unit memasukkan kata kunci pertama untuk mendeskripsikan kegiatan	Kata kunci dapat diketik dan hanya dapat menyimpan data huruf bukan angka atau spesial karakter

Tabel 4.6 Contoh Matrix Kualitas Pengujian Aplikasi Pertama

Pengujian Plugin Dynamic TWAIN Untuk Scan		
Jumlah Class	Jumlah kasus pengujian	Jumlah *bug*
upload_operator	Membuka menu upload operator	
upload_operator and AcquireImage	Melakukan klik tombol pindai	
AcquireImage	Klik tombol pindai pada jendela pop up modul .	Hasil Scan tidak dapat tampil sempurna
dwtcontainer	Mengetik nama file pada kotak nama file	
imgtype	Memilih format penyimpanan	Ekstensi File tidak berubah
uploadimage and simpanoperator	Menyimpan file pada repository aplikasi	Atribut nama tidak tersimpan Isi file rusak dan tidak terbaca
Total: 6	Total: 6	Total: 4

Beberapa kode yang merupakan penerapan dari kebutuhan dengan ketergantungan besar dipilih. Hasil akhir matriks kualitas untuk pengujian unit, pengujian integrasi, dan pengujian regresi untuk versi pertama dirangkum pada Tabel 4.7.

Tabel 4.7 Contoh Matrix

Testing Stage/ Testing Process	The number class tested	The number of testing cases	The number of *bug*s
Unit Testing	36	38	16
Integration Testing		28	5
Regression Testing		10	2

4. 2. 3. Membuat RTM

RTM dibentuk untuk melihat keterkaitan antara dokumen kebutuhan, dokumen pengujian, tabel penyimpanan pada database, sumber kode, dan *defect* yang dihasilkan di saat produk versi beta dirilis. Keterkaitan ini diperlukan untuk mempermudah menelusuri penerapan kebutuhan yang tidak sesuai. Ketidak sesuaian ini tentunya perlu dikenali untuk menangani *defect* agar aplikasi dapat berjalan dengan baik. RTM ini dijadikan sebagai *ground truth* untuk dibandingkan dengan metode penelusuran menggunakan pemodelan topik.

Pada aplikasi pertama kebutuhan dengan kode M adalah manajemen dokumen berdasarkan *story point* pada Tabel 4.2, prosesnya diuji menggunakan tahapan yang ada pada pengujian dengan kode P,E,S,U yaitu pengujian proses pemindaian seperti yang tertera pada Tabel 4.4. Setiap dokumen pengujian diberikan kode TC di depannya untuk menunjukkan Test Case (dokumen pengujian).

Setiap atribut pada dalam penerapan kebutuhan manajemen dokumen disimpan pada tabel penyimpanan database yang tertera pada Tabel 4.8 dan Tabel 4.9 . sumber kode dan *defect* yang terkait juga akan di identifikasi sehingga RTM dibentuk seperti pada Tabel 4.10 dan Tabel 4.11 .

Tabel 4.8 Tabel Penyimpanan Database Aplikasi Pertama

ID_Table	Table_Name
T1	doc_emp
T2	jenis_doc_pribadi
T3	jenis_naik_pangkat
T4	ms_pegawai
T5	ref_klasifikasi

Tabel 4.9 Tabel Penyimpanan Database Aplikasi Kedua

ID_Table	Table_Name
D1	master_kegiatan
D2	laporan_kegiatan
D3	lap_keg_anggaran
D4	dipa
D5	kait_kegiatan
D6	dokumen_pendukung

Tabel 4.10 Contoh RTM Aplikasi Pertama

Kebutuhan	Pengujian	Tabel	Source Code	*Defect*
M	TC-P	T1 T2 T4 T5	Savetofile.php upload.php	Saat melakukan pemindaian dokumen, terdapat beberapa kejadian kegagalan penyimpanan file
	TC-E TC-S	T1 T2 T4 T5	ambilnama.php ambiljenisdok.php ambildokumen.php ambilnip.php ambilnip18.php SimpanOperator.php Uploadoperator.php	Saat melakukan pemindaian dokumen, terdapat beberapa kejadian kegagalan penyimpanan file
	TC-U	T4 T2	Dok_pribadiadd.php Dok_pribadidelete.php Dok_pribadiedit.php	Pengamanan dokumen kurang karena pegawai dapat mengedit dokumen milik pegawai lain bila mengetahui id file dokumennya

Tabel 4.11 Contoh RTM Aplikasi Kedua

Kebutuhan	Pengujian	Tabel	Code	*Defect*
LK	TC-LK	D1 D2 D5	LaporanKegiatanController.php LaporanKegiatan.php laporan-kegiatan-form (_form.php) BidangSektoralSearch.php	Tampilan laporan kegiatan masih ada yang belum sesuai dengan filter unit dan bidang sektoral yang dipilih.
		D2 D6	dokumen-pendukung-form (_form.php) DokumenpendukungController.php	Dokumen pendukung seringkali tidak dapat diakses karena rusak, meski dapat upload file dalam jumlah banyak ketika dilihat kembali beberapa hanya menampilkan satu saja, meski terdapat dokumen lain yang tersimpan Dokumen pendukung pada laporan kegiatan ada yang tidak dapat dibuka pada saat melihat laporan kegiatan dengan menggunakan filter unit, tahun, bidang sektoral, dan bidang teknologi
	TC-KG	D1 D5	kegiatanController.php masterkaitankegiatanController.php masterkegiatanSearch.php	Ada beberapa kaitan kegiatan yang tidak muncul sesuai yang tersimpan, mengakibatkan kesalahan penelusuran kegiatan beserta laporan kegiatannya.

4. 2. 4. Uji Korelasi

Berdasarkan dokumen kebutuhan ditemukan jumlah *story point* yang muncul, serta berapa banyak *story point* yang berubah atau bertambah. Pada dokumen pengujian jumlah test case yang dilakukan dalam 3 tahap pengujian (pengujian unit, pengujian integrasi, sistem pengujian) juga dilihat korelasinya dengan waktu pengembangan, jumlah *bug* yang muncul pada setiap siklus, dan *defect* setiap waktu *delivery product*. Sehingga data set untuk menentukan korelasi untuk setiap atribut yang telah disebutkan sebelumnya terlihat pada Tabel 4.12 dan Tabel 4.13. Pada aplikasi pertama *defect* yang berhasil teridentifikasi adalah lima, pada siklus pertama tidak ditemukan *defect*, hal ini mungkin terjadi karena fungsi yang diterapkan belum terlalu rumit. Pada siklus terakhir dimana semua kebutuhan lengkap dan sangat kompleks terdapat *bug* dan *defect* paling banyak. Terlihat waktu penerapan cenderung semakin cepat dibandingkan awal pengembangan.

Sementara itu pada aplikasi kedua ditemukan lebih banyak *defect* dengan waktu penerapan yang cenderung meningkat. Kecenderungan penambahan *bug* dan *defect* masih sama dengan aplikasi pertama, ketika kebutuhan semakin bertambah dan memiliki fungsi yang lebih kompleks. Jumlah pengujian yang dilakukan juga diperhitungkan untuk melihat kompleksitas pembangunan aplikasi, pengujian unit cenderung lebih banyak dikarenakan menguji semua nilai variabel yang terlibat dan bagaimana timbal balik nilainya ketika dijalankan sesuai fungsi kecil apakah seperti yang diharapkan. Pengujian integrasi adalah gabungan dari fungsi kecil yang bekerja sebagai satu kesatuan.

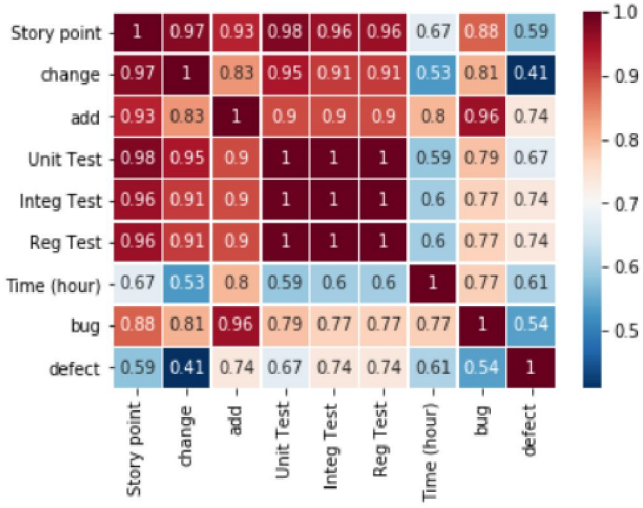
Pada pengujian regresi semua pengujian integrasi di uji ulang, pada penelitian ini tidak memilih pengujian pada fungsi tertentu karena pengujian regresi dilakukan oleh pengguna, bisa jadi memberikan hasil yang berbeda. Meskipun jumlah *story point* pada aplikasi pertama lebih banyak, pengujian unit dan integrasi justru lebih banyak dilakukan pada aplikasi kedua. Terlihat bahwa aplikasi kedua memiliki variabel terlibat yang lebih banyak dan lebih rumit penerapannya. Berdasarkan kedua data tersebut uji korelasi dilakukan dengan menggunakan metode Pearson.

Tabel 4.12 Data Set Uji Korelasi Aplikasi Pertama

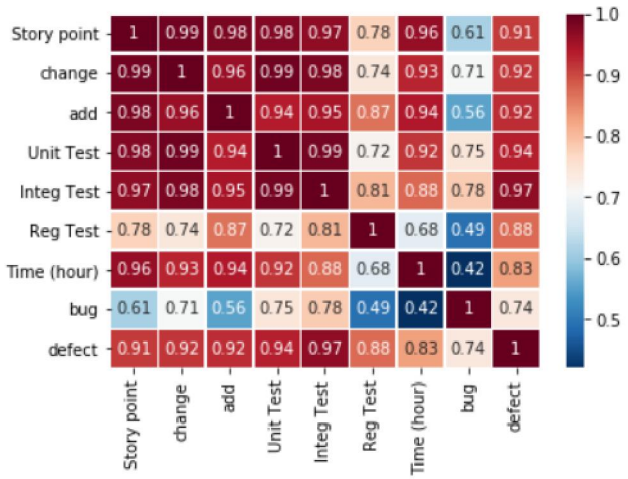
Story point	change	add	Unit Test	Integ Test	Reg Test	Time (hour)	*bug*	*defect*
10	2	2	9	4	4	25	3	0
26	8	3	13	5	5	30	5	1
31	9	3	19	8	8	25	3	1
42	12	6	22	9	9	25	7	1
53	13	9	26	11	11	45	9	2

Tabel 4.13 Data Set Uji Korelasi Aplikasi Kedua

Story point	change	add	Unit Test	Integ Test	Reg Test	Time (hour)	*bug*	*defect*
10	2	1	18	2	2	15	2	0
21	4	5	27	5	5	20	2	1
28	6	6	45	8	8	25	5	2
34	8	7	54	10	10	25	8	2
42	9	9	59	10	10	35	4	2



Gambar 4.1 Korelasi Pearson Heatmap Aplikasi Pertama



Gambar 4.2 Korelasi Pearson Heatmap Aplikasi Kedua

Pada gambar *heatmap* variabel berkorelasi tinggi ditunjukkan dengan warna merah gelap, semakin gelap warnanya maka semakin tinggi korelasinya. Korelasi pada aplikasi pertama terlihat pada Gambar 4.1. *Defect* memiliki korelasi yang erat dengan penambahan *story point* serta jumlah pengujian integrasi dan regresinya dengan nilai 0,74. Kemunculan *defect* juga dipengaruhi dengan kemunculan *bug* sebelumnya dan jumlah *story point*, pengaruh tersebut tidaklah kuat karena nilai korelasinya adalah sedang yaitu 0,54 dan 0,59. Perubahan pada *story point* justru memiliki korelasi yang lemah dengan kemunculan *defect*. Akan

tetapi, penambahan *story point* justru memiliki korelasi yang cukup tinggi dengan kemunculan *defect*. Pada aplikasi pertama penambahan kebutuhan atau tahapan untuk pemenuhan kebutuhan membuat aplikasi menjadi kurang stabil dan berakhir dengan kegagalan fungsi. Meskipun kegagalan fungsi tersebut telah di deteksi sebagai *bug* dan telah diperbaiki, tampaknya pada daftar *defect* kegagalan fungsi tersebut muncul lagi. Selain itu, terdapat kegagalan fungsi baru yang tidak terdeteksi ketika melakukan pengujian dan baru muncul setelah pengujian oleh pengguna pada saat produk rilis pada tiap iterasinya.

Korelasi pada aplikasi kedua terlihat pada Gambar 4.2. Berdasarkan data pada aplikasi kedua, *defect* memiliki korelasi yang sangat erat dengan semua artifak dan waktu pengembangan. Korelasi paling erat untuk *defect* adalah dengan pengujian unit dan pengujian integrasi dengan nilai korelasi 0,94 dan 0,97. Berbeda dengan aplikasi yang pertama, pada aplikasi kedua perubahan dan penambahan kebutuhan memilki pengaruh yang sangat besar terhadap kemunculan *defect*, yaitu 0,92. Nilai korelasi pada sebagian besar artifak untuk aplikasi kedua cenderung sangat tinggi yaitu lebih dari 0,9 berbeda dengan aplikasi pertama yang cenderung sedang.

Berdasarkan hasil penelitian terhadap dua aplikasi dalam siklus pengembangan XP yang berkelanjutan, variabel yang mempengaruhi munculnya *defect* setelah produk rilis pada setiap iterasi adalah sebagai berikut:

Jumlah *defect* yang dipengaruhi oleh perubahan dan penambahan *story point* dalam kebutuhan dan jumlah *story point* itu sendiri. Semakin sering kebutuhan berubah dan bertambah maka semakin banyak juga *defect* yang dihasilkan. Kerumitan aplikasi juga menjadi faktor penentu, perlu dilihat bahwa perubahan pada kebutuhan dapat memberikan pengaruh besar ketika banyak variabel terlibat pada suatu aplikasi dan hanya memberi pengaruh sedang ketika variabel yang terlibat tidak terlalu banyak. Hal tersebut juga berlaku pada pengaruh jumlah *story point* dengan kemunculan *defect*. Disaat aplikasi yang memiliki banyak fungsi mengubah nilai variabel, baik proses penyimpanan dan interaksi antar variabel tentu menjadi sangat tidak stabil ketika ditambahkan satu tahapan

lagi atau proses pada tahapan tersebut diubah sedemikian rupa. Berbeda dengan aplikasi sederhana yang pada satu fungsi jumlah variabel yang terlibat sedikit dan tidak menuntut transaksi antar variabel yang rumit.

Setiap tahapan pengujian memiliki korelasi yang kuat dengan munculnya *bug*, yaitu kesalahan fungsi operasional pada aplikasi. Jumlah pengujian baik unit dan integrasi dipengaruhi oleh kerumitan aplikasi, terutama jumlah variabel dan fungsi yang perlu diuji. Selama periode pengembangan, *bug* segera diperbaiki, sehingga meminimalisir munculnya *defect* setelah dilakukan pengujian regresi. Semakin banyak *bug* yang dikenali dengan semakin banyaknya pengujian yang menyeluruh, maka semakin kecil kemungkinan munculnya *defect* yang fatal.

Pada kedua aplikasi terlihat pengujian yang paling sering mendeteksi *bug* dan memiliki korelasi kuat dengan kemunculan *defect* dan *bug* adalah pengujian integrasi. Kondisi ini terjadi karena unit fungsi yang digabungkan dan dilihat apakah dapat berjalan baik sebagai satu kesatuan utuh. Proses penggabungan fungsi yang sedang berjalan tersebut seringkali menimbulkan ketidakharmonisan pengolahan data dan membuat aplikasi tidak stabil. Beberapa penyesuaian diperlukan untuk membuat aplikasi berjalan sesuai dengan pemenuhan kebutuhannya.Untuk itulah pengujian integrasi menjadi sangat penting.

4. 3. Hasil Pemodelan Topik

4. 3. 1. Aplikasi Pertama

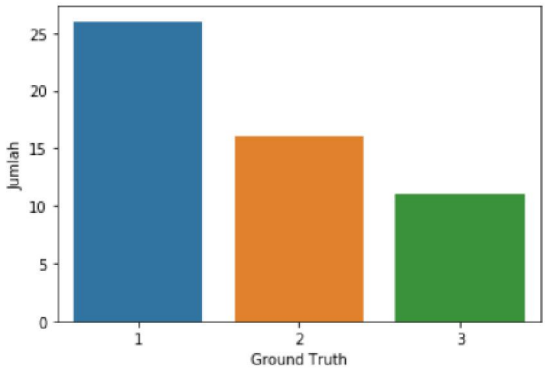
4.3.1.1. Kebutuhan

Pada aplikasi pertama, isi dari kebutuhan dibagi menjadi 3 macam yang kemudian ditentukan sebagai topik yang seharusnya muncul:

1) Manajemen file, yang meliputi penambahan file, pemindaian file, pengunggahan file, penyimpanan, penghapusan, dan perubahan file.
2) Validasi dan status file, yang menjelaskan bagaimana operator memeriksa dan menyatakan dokumen valid dan dapat diproses lebih lanjut atau ditolak untuk dikembalikan kepada pegawai agar dapat diperbaiki.

3) Kenaikan pangkat, yang menjelaskan bagaimana file yang telah tersimpan dapat memenuhi pengajuan kenaikan pangkat serta bagaimana pegawai melengkapinya.

Pada daftar kebutuhan seringkali satu kebutuhan merupakan irisan dari dua topik. Oleh karena itu k topik yang digunakan adalah 4 bukan 3 topik. Kemudian dari topik ke empat dilihat condong kearah manakah hal yang dibahas oleh kebutuhan tersebut. Data yang didapatkan tidak seimbang pada setiap topik seperti pada Gambar 4.4.



Gambar 4.3 Data *Ground truth* Persebaran Topik Aplikasi Pertama

4.3.1.1.1. LSA

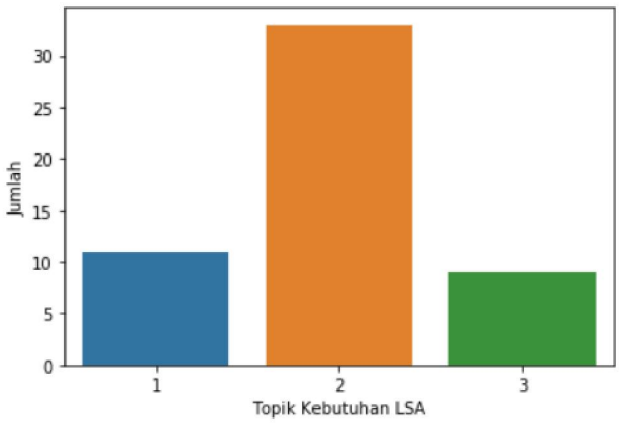
```
TOP 15 untuk setiap topik #0
['upload', 'ditolak', 'berstatus', 'aplikasi', 'validasi', 'pangkat', 'status', 'scan', 'jenis', 'kenaikan', 'file', 'memilih',
'operator', 'pegawai', 'dokumen']


TOP 15 untuk setiap topik #1
['php', 'pemrosesan', 'fisik', 'menghapus', 'konfirmasi', 'formulir', 'kepegawaian', 'doc', 'repository', 'penghapusan', 'atrib
ut', 'database', 'penyimpanan', 'aplikasi', 'file']


TOP 15 untuk setiap topik #2
['pemilihan', 'scan', 'dimanfaatkan', 'struktural', 'fungsional', 'nama', 'mutasi', 'terkait', 'jabatan', 'filling', 'persyarat
an', 'jenis', 'memilih', 'pangkat', 'kenaikan']


TOP 15 untuk setiap topik #3
['divalidasi', 'diubah', 'dihapus', 'kenaikan', 'jabatan', 'mengakses', 'pribadi', 'data', 'berstatus', 'biro', 'pegawai', 'ope
rator', 'persyaratan', 'sdmo', 'filling']
```

Gambar 4.4 Kata Penyusun Topik Unigram-LSA Aplikasi Pertama

Berdasarkan pada Gambar 4.4 topik #0 terdapat kata ditolak, berstatus, validasi, status, yang dapat disimpulkan sebagai pernyataan proses validasi dan status dokumen. Selain itu kata operator juga dapat menandakan proses validasi dan

status dokumen, karena hanya operatorlah yang dapat mengubah status dokumen dan melakukan validasi. Meskipun terdapat kata kenaikan yang menandakan proses kenaikan pangkat dan upload untuk manajemen dokumen, Sebagian besar kata penyusun pada topik 0 atau pertama cenderung membicarakan proses validasi dan status dokumen. Pada topik #1 terdapat kata menghapus, repository, penghapusan, database, penyimpanan yang kemudian dapat disimpulkan sebagai proses manajemen file.

Pada topik #2 terdapat kata struktural dan fungsional yang merupakan kata jenis kenaikan pangkat. Jabatan dan mutasi juga merupakan proses pengecekan jabatan dan pada departemen mana posisi terakhir pegawai berada untuk menunjang proses kenaikan pangkatnya. Selain itu terdapat dua kata pangkat dan kenaikan yang memperkuat bahwa topik ini membicarakan tentang proses kenaikan pangkat. Pada topik #3 terdapat kata kenaikan, jabatan, dan persyaratan yang mengarah pada proses kenaikan pangkat. Kemudian kata-kata biro, pegawai, dan sdmo muncul sebagai wali proses kenaikan pangkat. Maka dapat disimpulkan bahwa topik ini membicarakan kenaikan pangkat. Maka persebaran topik yang ada menjadi seperti yang ditunjukkan pada Gambar 4.5.

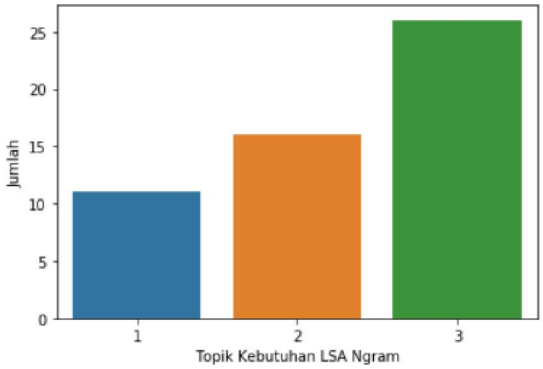


Gambar 4.5 Persebaran Kebutuhan Unigram-LSA Aplikasi Pertama

```
Topic #0: memilih jenis, kenaikan pangkat, jenis kenaikan pangkat, jenis kenaikan, jenis dokumen, pegawai memilih jenis, dokume
n terkait, memilih jenis kenaikan, jenis dokumen terkait, pegawai memilih, dokumen memilih nama, pangkat memilih jenis, memilih
nama pegawai, memilih nama, memilih jenis dokumen
Topic #1: status dokumen, atribut file, atribut file database, file database, penghapusan file, update status, update status do
kumen, repository aplikasi, status dokumen ditolak, aplikasi doc, dokumen ditolak, penyimpanan atribut, penyimpanan atribut fil
e, filtering penanda status, filtering penanda
Topic #2: status dokumen, update status, update status dokumen, status dokumen ditolak, dokumen ditolak, penanda status dokume
n, penanda status, filtering penanda status, filtering penanda, dokumen diterima, status dokumen diterima, diterima valid, doku
men diterima valid, ditolak formulir, ditolak formulir upload
```

Gambar 4.6 Kata Penyusun Topik Kebutuhan LSA-Bigram Trigram Aplikasi Pertama

Pada pencarian kemunculan dua atau tiga kata sekaligus seperti pada Gambar 4.6 dikelompokkan sebagai berikut : Topik #0 sangat jelas membicarakan kenaikan pangkat karena kemunculan kalimat jenis kenaikan pangkat. Topik #1 membicarakan manajemen file dan status dokumen, yang cenderung mengarah pada proses manajemen file. Topik #2 membicarakan validasi dokumen karena terdapat kalimat dokumen diterima valid, dokumen ditolak, dan update status. Maka persebaran topik yang ada untuk Bigram dan Trigram pada pemodelan topik LSA menjadi seperti yang ditunjukkan pada Gambar 4.7.



Gambar 4.7 Persebaran Topik Kebutuhan LSA- Bigram Trigram Aplikasi Pertama

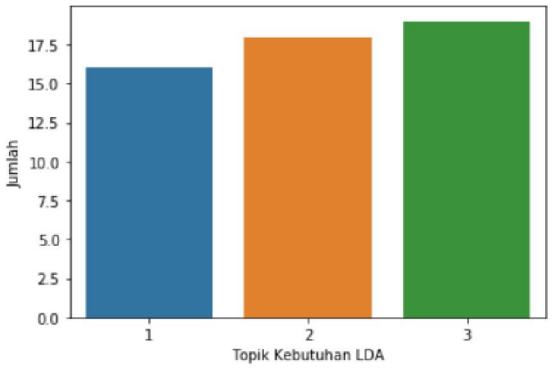
4.3.1.1.2. LDA

```
['filling', 'jabatan', 'formulir', 'terkait', 'kepegawaian', 'penghapusan', 'scan', 'persyaratan', 'pegawai', 'dokumen', 'pangk
at', 'jenis', 'memilih', 'kenaikan', 'file']


TOP 10 untuk setiap topik #1
['repository', 'operator', 'file', 'menghapus', 'dihapus', 'pegawai', 'pribadi', 'sdmo', 'validasi', 'atribut', 'data', 'databa
se', 'aplikasi', 'php', 'dokumen']


TOP 10 untuk setiap topik #2
['ditampilkan', 'tampilan', 'nip', 'validasi', 'scan', 'simpan', 'valid', 'update', 'ditolak', 'pegawai', 'proses', 'berstatu
s', 'operator', 'status', 'dokumen']


TOP 10 untuk setiap topik #3
['langsung', 'fisik', 'operator', 'pegawai', 'file', 'diupload', 'menghubungkan', 'scanner', 'server', 'alasan', 'menolak', 'fi
lling', 'penyimpanan', 'doc', 'aplikasi']
```

Gambar 4.8 Kata Penyusun Topik Kebutuhan Unigram-LDA Aplikasi Pertama

Berdasarkan pada Gambar 4.8, topik #0 membicarakan kenaikan pangkat. Muncul kata jabatan, persyaratan, pangkat, dan kenaikan yang mengarah pada proses kenaikan pangkat yang mengharuskan pegawai melengkapi persyaratan tertentu untuk jabatan tertentu karena proses kenaikan pangkatnya berbeda.

Sementara itu, topik #1 dan #3 membicarakan manajemen file. Terdapat kata repository, database, diupload, scan dan data untuk proses penyimpanan file. Terdapat kata menghapus, dihapus, yang menyatakan proses penghapusan.Topik #2 membicarakan status dan validasi dokumen. Terdapat kata validasi, valid, ditolak, dan status yang merupakan proses validasi dan pernyataan status dokumen. Maka persebaran topik untuk yang ada menjadi seimbang jumlahnya seperti yang ditunjukkan pada Gambar 4.9.



Gambar 4.9 Persebaran Topik Kebutuhan Unigram-LDA Aplikasi Pertama

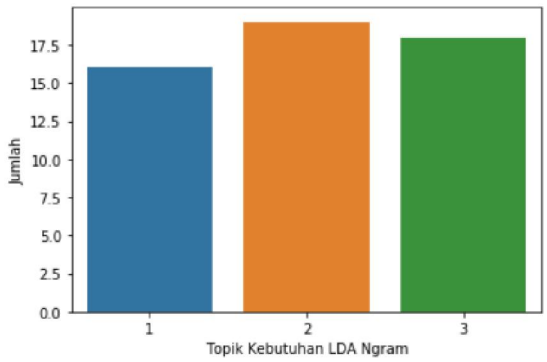
```
Topic #0: update status dokumen, update status, status dokumen, file pegawai, dokumen berstatus, status dokumen ditolak, dokume
n ditolak, status dokumen diterima, dokumen diterima valid, dokumen diterima, diterima valid, pegawai mengakses, divalidasi bir
o, validasi dokumen sesuai, dokumen sesuai peruntukannya
Topic #1: kenaikan pangkat, memilih jenis, jenis dokumen, pegawai memilih, jenis kenaikan, jenis kenaikan pangkat, aplikasi do
c, status dokumen, penanda status dokumen, filtering penanda, filtering penanda status, penanda status, scan dokumen, pembatasa
n jenis dokumen, dokumen pdf
Topic #2: penghapusan file, atribut file, file database, atribut file database, formulir penghapusan, formulir penghapusan fil
e, konfirmasi penghapusan file, konfirmasi penghapusan, repository aplikasi, dokumen data, penyimpanan atribut file, penyimpana
n atribut, database repository, file database repository, menghapus atribut file
```

Gambar 4.10 Kata Penyusun Topik Kebutuhan LDA-Bigram Trigram Aplikasi Pertama

Berdasarkan Gambar 4.10 topik #0 membicarakan validasi dan status dokumen, kata yang sering muncul secara bersamaan seperti update status, diterima valid, validasi dokumen sesuai semuanya mengarah pada bagaimana proses validasi dan status dokumen dilakukan.

Sementara itu, topik #1 membicarakan kenaikan pangkat, kata yang sering muncul bersamaan kenaikan pangkat, memilih jenis, jenis dokumen, jenis kenaikan pangkat menjelaskan bagaimana proses kenaikan pangkat dilakukan.

Topik #2 membicarakan manajemen file, kata yang sering muncul bersamaan seperti penghapusan file, atribut file databasase menjelaskan proses penyimpanan dan penghapusan file. Maka jumlah setiap topik yang ditemukan pada kebutuhan menjadi sedikit berubah seperti pada Gambar 4.11



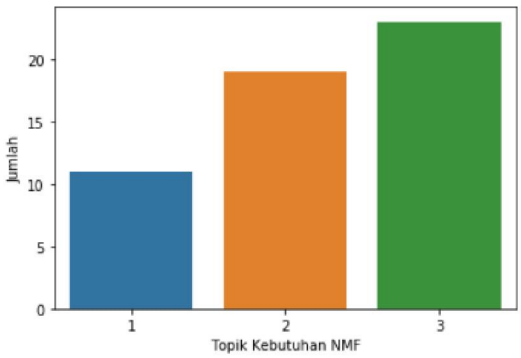
Gambar 4.11 Persebaran Topik Kebutuhan LDA- Bigram Trigram Aplikasi Pertama

4.3.1.1.3. NMF

```
['berasal', 'filtering', 'penanda', 'diubah', 'dihapus', 'validasi', 'valid', 'update', 'proses', 'pegawai', 'ditolak', 'bersta
tus', 'status', 'operator', 'dokumen']


TOP 10 untuk setiap topik #1
['menghubungkan', 'php', 'konfirmasi', 'fisik', 'menghapus', 'kepegawaian', 'formulir', 'doc', 'repository', 'penghapusan', 'at
ribut', 'database', 'penyimpanan', 'aplikasi', 'file']


TOP 10 untuk setiap topik #2
['nip', 'klasifikasi', 'simpan', 'masuk', 'menu', 'nama', 'pegawai', 'terkait', 'upload', 'pangkat', 'kenaikan', 'scan', 'dokum
en', 'jenis', 'memilih']


TOP 10 untuk setiap topik #3
['file', 'pribadi', 'dimanfaatkan', 'sdmo', 'biro', 'struktural', 'fungsional', 'mutasi', 'data', 'jabatan', 'pegawai', 'pangka
t', 'persyaratan', 'filling', 'kenaikan']
```

Gambar 4.12 Kata Penyusun Topik Kebutuhan Unigram -NMF Aplikasi Pertama

Berdasarkan Gambar 4.12, topik #0 membicarakan proses validasi dan status dokumen. Kemunculan kata valid, validasi, ditolak, dan status merupakan kata kunci serangkaian proses validasi dan status dokumen. Sementara itu, topik #1 membicarakan manajemen file. Kemunculan kata menghapus, repository, penghapusan, penyimpanan, dan database menjelaskan bagaimana file diatur pada aplikasi. Topik # 2 dan #3 membicarakan kenaikan pangkat. Kemunculan kata kenaikan dan pangkat adalah kata kunci pada penjelasan seputar proses kenaikan pangkat. Jumlah topik pada kebutuhan pada pemodelan NMF menjadi seperti pada Gambar 4.13.

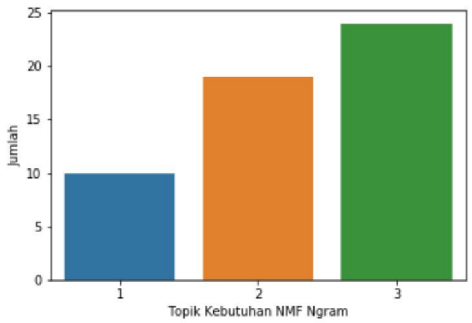


Gambar 4.13 Persebaran Topik Kebutuhan Unigram -NMF Aplikasi Pertama

```
Topic #0: memilih jenis, kenaikan pangkat, jenis kenaikan pangkat, jenis kenaikan, jenis dokumen, jenis dokumen terkait, pegawa
i memilih jenis, memilih jenis kenaikan, dokumen terkait, pegawai memilih, memilih nama pegawai, pangkat memilih, memilih nama,
scan dokumen memilih, pangkat memilih jenis
Topic #1: atribut file, file database, atribut file database, penghapusan file, repository aplikasi, aplikasi doc, penyimpanan
atribut, penyimpanan atribut file, menghapus atribut file, database repository aplikasi, database repository, menghapus atribu
t, file database repository, formulir penghapusan file, konfirmasi penghapusan file
Topic #2: status dokumen, update status, update status dokumen, status dokumen ditolak, dokumen ditolak, penanda status, filter
ing penanda, penanda status dokumen, filtering penanda status, diterima valid, dokumen diterima, dokumen diterima valid, status
dokumen diterima, status dokumen update, ditolak formulir upload
```

Gambar 4.14 Kata Penyusun Topik Kebutuhan Bigram dan Trigram – NMF Aplikasi Pertama

Berdasarkan Gambar 4.14, topik #0 membicarakan kenaikan pangkat, kata yang sering muncul bersamaan adalah jenis kenaikan pangkat. Kemudian, topik #1 membicarakan manajemen file, kata yang sering muncul adalah formular penghapusan file, database repository aplikasi menjelaskan proses penyimpanan dan penghapusan file pada aplikasi. Topik #2 membicarakan status dan validasi dokumen. Kata yang sering muncul secara bersamaan adalah status dokumen, update status, status dokumen ditolak, dan dokumen diterima valid, semuanya mengarah pada bagaimana sebuah file diberi status, ditolak dan dilakukan validasi. Maka jumlah topik yang ditemukan pada dokumen kebutuhan dengan metode ini menjadi seperti pada Gambar 4.15.



Gambar 4.15 Persebaran Topik Kebutuhan Bigram dan Trigram – NMF Aplikasi Pertama

4.3.1.2. Pengujian

4.3.1.2.1. LSA

```
TOP 15 untuk setiap topik #0
['proses', 'nip', 'mencentang', 'dipilih', 'operator', 'menu', 'kenaikan', 'pangkat', 'klik', 'kategori', 'pegawai', 'tipe', 'j
enis', 'memilih', 'dokumen']


TOP 15 untuk setiap topik #1
['rekap', 'dokumen', 'sesuai', 'sub', 'upload', 'validasi', 'operator', 'up', 'pop', 'membuka', 'hapus', 'pindai', 'menu', 'tom
bol', 'klik']


TOP 15 untuk setiap topik #2
['atribut', 'formulir', 'pindai', 'id', 'klik', 'hapus', 'pengajuan', 'masuk', 'mencentang', 'tombol', 'pangkat', 'kenaikan',
'mengetik', 'nip', 'pegawai']


TOP 15 untuk setiap topik #3
['sesuai', 'dokumen', 'lengkap', 'selesai', 'sub', 'pengajuan', 'formulir', 'proses', 'mencentang', 'upload', 'membuka', 'men
u', 'operator', 'kenaikan', 'pangkat']
```

Gambar 4.16 Kata Penyusun Topik Pengujian Pemodelan LSA-Unigram Aplikasi Pertama

Berdasarkan Gambar 4.16, topik #0 dan #3 menggambarkan proses kenaikan pangkat dengan munculnya dua kata kunci kenaikan dan pangkat. Proses pemilihan dokumen untuk kenaikan pangkat tergambar dari kata mencentang dan dokumen lengkap. Topik #1 menggabarkan proses validasi dan status dokumen dengan kata kunci validasi dan upload. Topik #2 menggambarkan proses manajemen file dengan kata kunci pindai dan hapus.

```
Topic #0: tipe dokumen, memilih tipe dokumen, memilih tipe, jenis dokumen, memilih jenis dokumen, memilih jenis, tipe dokumen k
ategori, dokumen kategori dipilih, kategori dipilih, dokumen kategori, tipe dokumen dipilih, dokumen tipe dokumen, dokumen tip
e, jenis dokumen tipe, dokumen dipilih
Topic #1: jenis dokumen, memilih jenis dokumen, memilih jenis, tipe dokumen dipilih, dokumen dipilih, jenis dokumen tipe, dokum
en tipe, dokumen tipe dokumen, kenaikan pangkat, pegawai memilih, pegawai memilih jenis, mencentang dokumen, syarat kenaikan, j
enis dokumen dikirimkan, kenaikan pangkat mencentang
Topic #2: klik tombol, klik tombol pindai, tombol pindai, tombol pindai pop, pindai pop, pop up, pindai pop up, tombol hapus, k
lik tombol hapus, klik tombol validasi, tombol validasi, tombol tolak, klik tombol tolak, operator klik, operator klik tombol
Topic #3: nip pegawai, mengetik nip pegawai, mengetik nip, memilih nip, memilih nip pegawai, format file penyimpanan, file peny
impanan, memilih format, format file, memilih format file, nama file, proses kenaikan, proses kenaikan pangkat, kenaikan pangka
t selesai, lengkap proses
```

Gambar 4.17 Kata Penyusun Topik Pengujian Pemodelan LSA-Bigram dan Trigram Aplikasi Pertama

Kata penyusun seperti pada Gambar 4.17 menunjukkan topik #0 dan #3 merupakan manajemen file karena terdapat proses pemilihan tipe dokumen, jenis dokumen, dan kategori. Kemudian proses penyimpanan juga masuk di dalamnya. Topik #1 merupakan kenaikan pangkat dengan munculnya syarat kenaikan, dan kenaikan pangkat. Proses pemilihan dokumen hampir sama dengan proses validasi dan status dokumen. Topik #2 merupakan gabungan pemindaian dokumen dan proses validasi. Ketika dilakukan pengecekan semua tahapan pengujian yang masuk pada topik ini lebih condong pada proses status dan validasi dokumen.

4.3.1.2.2. LDA

```
TOP 15 untuk setiap topik #0
['muncul', 'upload', 'pop', 'up', 'klik', 'check', 'centang', 'box', 'komputer', 'tombol', 'memilih', 'dokumen', 'mengetik', 'n
ip', 'pegawai']


TOP 15 untuk setiap topik #1
['pengajuan', 'pindai', 'masuk', 'pegawai', 'proses', 'rekap', 'operator', 'mencentang', 'memilih', 'tombol', 'pangkat', 'kenai
kan', 'klik', 'jenis', 'dokumen']


TOP 15 untuk setiap topik #2
['data', 'kategori', 'baris', 'format', 'file', 'penyimpanan', 'sub', 'memilih', 'upload', 'sesuai', 'klik', 'dokumen', 'operat
or', 'membuka', 'menu']


TOP 15 untuk setiap topik #3
['muncul', 'mengetikkan', 'field', 'proses', 'lengkap', 'selesai', 'kenaikan', 'pangkat', 'nama', 'file', 'dipilih', 'kategor
i', 'memilih', 'dokumen', 'tipe']
```

Gambar 4.18 Kata Penyusun Topik Pengujian Pemodelan LDA-Unigram Aplikasi Pertama

Melihat pada Gambar 4.18, maka topik #0 dan #2 menggambarkan manajemen file dengan kata kunci upload dan penyimpanan . Topik #1 muncul kata kenaikan, rekap, dan pindai muncul dalam satu topik yang sama. Menggambarkan proses kenaikan pangkat dan pemindaian dokumen dimana pokok bahasan tidak terpisah seperti yang diinginkan. Topik #3 menggambarkan kenaikan pangkat dengan kata kenaikan, pangkat, lengkap yang menandakan kelengkapan dokumen persyaratan, dan tipe untuk tipe kenaikan pangkat. Untuk proses validasi dan status tidak tergambar pada kelima topik yang dihasilkan oleh LDA.

```
Topic #0: klik tombol, tombol pindai, klik tombol pindai, kategori dokumen, menu dokumen, klik menu, klik menu dokumen, memilih
kategori dokumen, memilih kategori, menu sesuai kategori, sesuai kategori, sesuai kategori dokumen, sub menu, sub menu sesuai,
membuka sub
Topic #1: tipe dokumen, memilih tipe, memilih tipe dokumen, kenaikan pangkat, dokumen kenaikan pangkat, dokumen kenaikan, mence
ntang dokumen kenaikan, operator mencentang dokumen, kategori dipilih, dokumen kategori dipilih, tipe dokumen kategori, dokumen
kategori, operator mencentang, mencentang dokumen, tipe dokumen dipilih
Topic #2: jenis dokumen, memilih jenis dokumen, memilih jenis, menu upload operator, upload operator, menu upload, membuka men
u, membuka menu upload, nama file, check box, centang check box, centang check, proses operator, memilih status dokumen, proses
proses operator
Topic #3: nip pegawai, mengetik nip pegawai, mengetik nip, memilih nip, memilih nip pegawai, memilih dokumen, pangkat selesai,
proses kenaikan, proses kenaikan pangkat, dokumen lengkap, kenaikan pangkat selesai, dokumen lengkap proses, lengkap proses, le
ngkap proses kenaikan, baris data terpilih
```

Gambar 4.19 Kata Penyusun Topik Pengujian LDA- Bigram dan Trigram Aplikasi Pertama

Topik #0 menggambarkan manajemen file dengan munculnya tombol pindai, kategori dokumen, dan proses pemilihan kategori dokumen. Proses memasukkan atribut yang menjelaskan dokumen terdapat pada tahapan pengujian yang tergolong pada topik ini. Topik #1 dan #3 menggambarkan proses kenaikan pangkat. Munculnya frase kenaikan pangkat, dokumen kenaikan, dan lengkap

proses kenaikan pangkat dapat disimpulkan sebagai proses melengkapi dokumen untuk persyaratan kenaikan pangkat. Topik #2 menggambarkan proses validasi dan status dokumen. Centang check box seringkali ditemukan pada tahapan pengujian proses pemilihan dokumen yang akan di validasi dan kemudian berubah statusnya. Terdapat proses upload juga yang sebenarnya masuk pada proses manajemen file.

4.3.1.2.3. NMF

```
TOP 15 untuk setiap topik #0
['komputer', 'pegawai', 'format', 'rekap', 'operator', 'sesuai', 'file', 'sub', 'menu', 'dipilih', 'kategori', 'jenis', 'tipe',
'dokumen', 'memilih']


TOP 15 untuk setiap topik #1
['kategori', 'sesuai', 'sub', 'upload', 'validasi', 'operator', 'membuka', 'pop', 'up', 'hapus', 'dokumen', 'pindai', 'menu',
'tombol', 'klik']


TOP 15 untuk setiap topik #2
['repository', 'aplikasi', 'menyimpan', 'upload', 'penyimpanan', 'nama', 'hapus', 'komputer', 'atribut', 'id', 'file', 'memili
h', 'mengetik', 'nip', 'pegawai']


TOP 15 untuk setiap topik #3
['masuk', 'pegawai', 'muncul', 'diperlukan', 'berisi', 'selesai', 'lengkap', 'pengajuan', 'formulir', 'proses', 'mencentang',
'operator', 'dokumen', 'pangkat', 'kenaikan']
```

Gambar 4.20 Kata Penyusun Topik Pengujian Pemodelan NMF-Unigram

Berdasarkan Gambar 4.20, topik #0 dan #2 menggambarkan manajemen file dengan kata kunci upload, menyimpan , dan kata-kata proses penyimpanan seperti memilih kategori dan jenis dokumen. Topik #1 menggambarkan proses status dan validasi dokumen, karena terdapat kata kunci validasi. Topik #3 menggambarkan proses kenaikan pangkat dengan kata kunci kenaikan, pangkat, dan pengajuan.

```
Topic #0: memilih tipe dokumen, memilih tipe, tipe dokumen, dokumen kategori dipilih, dokumen kategori, kategori dipilih, tipe
dokumen kategori, jenis dokumen tipe, dokumen dipilih, dokumen tipe dokumen, tipe dokumen dipilih, dokumen tipe, klik menu doku
men, klik menu, menu dokumen
Topic #1: memilih jenis dokumen, jenis dokumen, memilih jenis, dokumen tipe, dokumen dipilih, jenis dokumen tipe, tipe dokumen
dipilih, dokumen tipe dokumen, tipe dokumen, kenaikan pangkat, pegawai memilih jenis, pegawai memilih, mencentang dokumen, syar
at kenaikan pangkat, pangkat mencentang
Topic #2: klik tombol, klik tombol pindai, tombol pindai, pop up, pindai pop, tombol pindai pop, pindai pop up, tombol hapus, k
lik tombol hapus, klik tombol validasi, klik tombol tolak, tombol validasi, tombol tolak, operator klik tombol, operator klik
Topic #3: nip pegawai, mengetik nip, mengetik nip pegawai, memilih nip pegawai, memilih nip, kategori dokumen, sesuai kategori,
sub menu, sub menu sesuai, menu sesuai kategori, menu sesuai, membuka sub menu, membuka sub, sesuai kategori dokumen, menu doku
men
```

Gambar 4.21 Kata Penyusun Topik Pengujian Pemodelan NMF-Bigram dan Trigram

Terlihat pada Gambar 4.21, topik #0 dan #3 menggambarkan manajemen file dengan munculnya 2 dan 3 kata bersamaan seperti tipe dokumen, kategori dipilih, dan atribut dokumen lainnya seperti nip untuk identitas pemilik dokumen. Topik #1 menggambarkan kenaikan pangkat dengan munculnya syarat kenaikan

pangkat dan pangkat mencentang. Topik #2 menggambarkan proses status dan validasi dengan munculnya tombol validasi dan tombol tolak. Kedua tombol tersebut untuk menentukan perubahan status dokumen dan dilakukan oleh oleh operator.

4.3.1.3. *Bug* and *Defect*

4.3.1.3.1. LSA

```
TOP 15 untuk setiap topik #0
['muncul', 'ditampilkan', 'file', 'tipe', 'jenis', 'pangkat', 'pegawai', 'kenaikan', 'proses', 'pilihan', 'kategori', 'sesuai',
'berhenti', 'step', 'dokumen']


TOP 15 untuk setiap topik #1
['menampilkan', 'dipilih', 'menu', 'tree', 'bawahnya', 'sub', 'filter', 'ditampilkan', 'kategorinya', 'berdasarkan', 'jenis',
'sesuai', 'kategori', 'tipe', 'pilihan']


TOP 15 untuk setiap topik #2
['operator', 'dihapus', 'scan', 'diubah', 'berubah', 'rekap', 'kategori', 'valid', 'menampilkan', 'terbuka', 'status', 'data',
'muncul', 'list', 'halaman']
```

Gambar 4.22 Kata Penyusun Topik *Bug* dan *Defect* Pemodelan LSA-Unigram Aplikasi Pertama

Berdasarkan Gambar 4.22, dan bila dilihat pada pengelompokan kata pada dokumen kebutuhan dan pengujian sebelumnya, maka pembagian topiknya adalah sebagai berikut :

- Topik #0 untuk kenaikan pangkat
- Topik #1 Manajemen File
- Topik #2 validasi dan status dokumen

4.3.1.3.2. LDA

```
TOP 15 untuk setiap topik #0
['dipilih', 'filter', 'halaman', 'list', 'kategorinya', 'berdasarkan', 'data', 'proses', 'jenis', 'ditampilkan', 'tipe', 'pilih
an', 'sesuai', 'dokumen', 'kategori']


TOP 15 untuk setiap topik #1
['diubah', 'rekap', 'step', 'memilih', 'operator', 'valid', 'berhenti', 'status', 'halaman', 'pegawai', 'field', 'muncul', 'pan
gkat', 'kenaikan', 'dokumen']


TOP 15 untuk setiap topik #2
['autofill', 'tampil', 'atribut', 'memenuhi', 'gagal', 'nip', 'id', 'tersimpan', 'pegawai', 'persyaratan', 'nama', 'berhenti',
'step', 'dokumen', 'file']
```

Gambar 4.23 Kata Penyusun Topik *Bug* dan *Defect* Pemodelan LDA-Unigram Aplikasi Pertama

Berdasarkan Gambar 4.23, dan bila dilihat pada pengelompokan kata pada dokumen kebutuhan dan pengujian sebelumnya, maka pembagian topiknya adalah sebagai berikut :

- Topik #0 manajemen file
- Topik #1 validasi dan status dokumen
- Topik #2 kenaikan pangkat

4.3.1.3.3. NMF

```
TOP 15 untuk setiap topik #0
['gagal', 'nip', 'tersimpan', 'id', 'tampil', 'nama', 'proses', 'persyaratan', 'step', 'berhenti', 'file', 'pegawai', 'pangka
t', 'kenaikan', 'dokumen']


TOP 15 untuk setiap topik #1
['berhenti', 'step', 'dipilih', 'tampil', 'sub', 'filter', 'kategorinya', 'berdasarkan', 'ditampilkan', 'dokumen', 'jenis', 'ka
tegori', 'sesuai', 'tipe', 'pilihan']


TOP 15 untuk setiap topik #2
['kategori', 'diubah', 'terbuka', 'menampilkan', 'rekap', 'berubah', 'operator', 'valid', 'proses', 'data', 'list', 'status',
'muncul', 'dokumen', 'halaman']
```

Gambar 4.24 Kata Penyusun Topik *Bug* dan *Defect* NMF-Unigram Aplikasi Pertama

Berdasarkan Gambar 4.24, dan bila dilihat pada pengelompokan kata pada dokumen kebutuhan dan pengujian sebelumnya, maka pembagian topiknya adalah sebagai berikut :

- Topik #0 kenaikan pangkat
- Topik #1 manajemen file
- Topik #2 Status dan validasi dokumen

Topik pertama adalah masalah promosi dan penyimpanan dokumen. NMF berhasil mengklasifikasikan cacat dengan *bug* yang menyertai dan langkah-langkah pengujian bermasalah dalam proses penyimpanan file, yang gagal menampilkan daftar dokumen promosi dengan benar. Pada topik kedua, pemfilteran tipe dokumen dan kategori dokumen tidak lagi menjadi masalah karena tidak ditemukan cacat. Pada topik 3, masalah dengan tampilan status dokumen. Proses validasi bermasalah. Cacat yang merupakan *bug* terpecahkan sebelumnya terjadi lagi. Saat status dokumen yang telah ditolak masih muncul di halaman operator dan tidak muncul

kembali di halaman karyawan. Metode LSA tidak dapat mengklasifikasikan cacat dengan baik karena semua dikumpulkan pada satu topik saja, dan dua topik lainnya hanya menyajikan pengujian tanpa ada hubungan dengan *bug* dan cacat sama sekali. LDA bekerja dengan baik dalam pengelompokan topik yang berkaitan dengan cacat modul promosi dan manajemen file. Misalnya, penyimpanan file melalui pemindaian tidak dapat muncul dan gagal disimpan, sehingga sistem mengirimkan file kosong untuk validasi. Pada akhirnya, daftar file untuk proses promosi tidak muncul dengan benar.

4.3.1.4. Evaluasi Hasil Pemodelan Topik

NMF dan LSA memberikan hasil yang sedikit lebih baik untuk memisahkan ketiga topik pada kebutuhan. NMF memisahkan dengan baik dan berhasil mengelompokkan hampir semua kebutuhan dan pengujian pada dua topik, yaitu status dan validasi file serta kenaikan pangkat. Topik status dan validasi dokumen semakin tidak dapat dipisahkan pada saat pengujian. Hal ini dikarenakan langkah yang ditempuh sangatlah mirip. Atribut yang terlibat pada tahapan penyimpanan file muncul juga pada saat operator melakukan validasi dokumen dan kemudian merubah status file yang disimpan. Status dokumen menjadi atribut penting dan kata-kata status dokumen muncul pada pengujian mengenai kenaikan pangkat.

Hal ini dikarenakan hanya file dengan status tertentu lah yang dapat digunakan sebagai diproses sebagai persyaratan kenaikan pangkat. Sementara itu proses penyimpanan dokumen meliputi keseluruhan atribut dan kata-kata seputar kenaikan pangkat juga disebutkan seperti pemilihan file untuk kenaikan pangkat oleh operator dan bagaimana dokumen yang belum lengkap dapat di unggah oleh pegawai biasa dalam rangka memenuhi persyaratan. Pada kalimat pengujian banyak kata atribut yang tidak spesifik pada topik tertentu. Sekali lagi LDA tidak dapat memisahkan ketiga topik.

Munculnya kata-kata yang kurang spesifik ini membuat tidak hanya kemunculan satu kata unik saja yang dicari, tetapi juga mencari munculnya dua atau tiga kata unik yang muncul secara bersamaan. Beralih pada Bigram dan

Trigram ternyata justru memperkecil akurasi dan bahkan LDA gagal memisahkan topik menjadi yang diharapkan. Sekali lagi NMF memiliki akurasi yang paling tinggi, meskipun nilainya lebih kecil dari sebelumnya.

Tabel 4.14 Nilai Akurasi Kebutuhan dan Pengujian Aplikasi Pertama

Akurasi	Requirement			Test Case		
	LDA	NMF	LSA	LDA	NMF	LSA
No Stem	62 %	67%	56%	29%	59%	34%
Bigram dan Trigram	47%	62%	64%	-	54%	41%

Tabel 4.15 Nilai Cohen Kappa Kebutuhan dan Pengujian Aplikasi Pertama

Cohen kappa	Requirement			Test Case		
	LDA	NMF	LSA	LDA	NMF	LSA
No Stem	0,44	0,54	0,35	-0,02	0,39	0,09
Bigram dan Trigram	0,21	0,46	0,49	-	0,54	0,16

Tabel 4.16 *Precision* dan *Recall* Kebutuhan dan Pengujian Aplikasi Pertama

LDA Unigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	68,8%	42,3%	42,1%	21,1%
2	66,7%	75%	23,8%	23,8%
3	52,6%	90,9%	25%	61,5%

LDA Bigram dan Trigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	62,5%	38,5%		
2	47,4%	56,2%		
3	33,3%	54,5%		
NMF Unigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	100%	42,3%	77,8%	36,8%
2	73,7%	87,5%	41,9%	85,7%
3	47,8%	100 %	100%	84,6%
NMF Bigram dan Trigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	100%	38,5%	76,5%	34,2%
2	63,2%	75%	51,7%	71,4%
3	45,8%	100%	42,3%	84,6%
LSA Unigram				
	Kebutuhan		*Test Case*	

	Precision	*Recall*	*Precision*	*Recall*
1	100%	42,3%	66,7%	26,3%
2	42,4%	87,5%	25%	19%
3	55,6%	45,5%	26,8%	84,6%
LSA Bigram or Trigram				
	Requirement		*Test Case*	
	Precision	Recall	Precision	Recall
1	100%	42,3%	53,3%	21,1%
2	75%	75%	34,4%	52,4%
3	42,3%	100%	44%	84,6%

Secara keseluruhan nilai akurasi menggunakan pencarian kata unigram menunjukkan hasil yang lebih baik dibandingkan dengan N-gram. Begitupula dengan nilai *precision* dan *recall* nya. Pemodelan topik pada pengujian menggunakan LDA bahkan sulit diterjemahkan kedalam topik kenaikan pangkat, status dan validasi dokumen, dan manajemen file. NMF memiliki nilai akurasi yang cukup stabil baik menggunakan unigram maupun Bigram dan Trigram. Sementara itu, LSA berada di tempat kedua dan LDA berada di tempat terakhir. Pada metode LDA dan NMF nilai *precision* dan *recall* nya juga lebih baik ketika menggunakan metode unigram. Hal yang berbeda ditemukan pada penerapan metode LSA dimana meski nilai akurasinya tinggi pada unigram, nilai *precision* lebih baik ketika menggunakan Bigram dan Trigram.

Dikarenakan untuk penelusuran, maka nilai yang diperhatikan adalah akurasi, *precision,* dan *recall*. Terlihat pada Tabel 4.14, metode NMF, nilai akurasi menggunakan unigram mencapai 67% dan menggunakan n-gram hanya mencapai

62% . Disaat menggunakan unigram nilai precision nya pada topik 2 mencapai 73,7% lebih tinggi bila dibandingkan dengan metode n-gram dimana nilai *precision* nya hanya 63,2%. Nilai *precision* diperhatikan karena artinya lebih banyak kebutuhan yang terprediksi benar masuk kedalam topik tersebut dan mengantisipasi jumlah yang terprediksi salah masuk kedalam topik yang tidak seharusnya. Pada kasus aplikasi pertama ini, metode pemodelan topik NMF lah yang bekerja paling baik untuk memisahkan kebutuhan dan pengujian berdasarkan topik yang diinginkan.

Karena unigram bekerja dengan lebih baik bila dibandingkan dengan n-gram, maka untuk menemukan relasi *defect* dengan pengujian hanya menggunakan unigram saja. Ketiga metode pemodelan topik kembali dibandingkan untuk mencari kesamaan topik dari daftar expected result pada dokumen test case, daftar *bug* yang muncul pada saat pengujian berlangsung (bisa saat unit testing atau integratioan testing), dan daftar *defect* ketika aplikasi di uji coba oleh pengguna secara satu kesatuan (regression testing). Hasilnya adalah sekali lagi metode NMF memberikan hasil yang terbaik. Pada topik pertama adalah permasalahan kenaikan pangkat dan penyimpanan dokumen . NMF berhasil mengelompokkan *defect* dengan *bug*s yang menyertainya serta langkah pengujian yang bermasalah pada proses penyimpanan file yang mengakibatkan kegagalan daftar dokumen kenaikan pangkat muncul dengan baik. Pada topik ke 2 adalah permasalahan filter jenis dokumen dan kategori dokumen yang tidak terdapat permasalahan lagi karena memang tidak ada *defect* yang ditemukan kembali. Pada topik ke 3 muncul permasalahan status dokumen dan proses validasi yang bermasalah sehingga *defect* kembali muncul ketika ada status dokumen yang telah ditolak masih muncul pada halaman operator dan tidak kembali muncul pada halaman pegawai.

Metode LSA tidak dapat mengelompokkan *defect* dengan baik karena semua terkumpul pada satu topik saja dan dua topik lainnya hanya menampilkan pengujian tanpa hubungan dengan *bug*s dan *defect* sama sekali. LDA melakukan pengelompokan dengan baik pada topik mengenai *defect* kenaikan pangkat dan penyimpanan dokumen. Dimana penyimpanan file melalui pemindaian tidak dapat muncul sempurna yang kemudian mengakibatkan sistem dapat mengirimkan file

kosong untuk di validasi dan daftar dokumen kenaikan pangkat yang tidak muncul secara sempurna.

Data pengelompokan topik dalam dokumen persyaratan dan dokumen pengujian tidak seimbang untuk setiap representasi modul. Karena data tidak seimbang, nilai akurasi, *precision*, dan perolehan dapat terpengaruh. Misalnya, data pada manajemen file yang lebih banyak dari data dua modul lainnya dikelompokkan secara berbeda dan dapat diidentifikasi sebagai topik yang tumpang tindih atau bahkan sama dengan modul kedua atau ketiga. Oleh karena itu, perlu dilakukan perhitungan koefisien Cohen Kappa untuk melihat apakah ketiga modul tersebut memiliki kesesuaian pola penerapan masing-masing algoritma pemodelan topik. Cohen Kappa mengukur ketergantungan tidak disengaja dari pengukuran pengelompokan yang diprediksi dan hasil aktual untuk menghilangkan karakteristik intrinsik dari data yang ada (Bagli and Visani, 2020).

Pada Tabel 4.15 , kebutuhan pemodelan topik NMF memiliki nilai positif yang tinggi yaitu 0,54 untuk pencarian Unigram dan 0,46 untuk pencarian Bigram dan Trigram. Hasil ini berarti model algoritma NMF berjalan dengan baik pada pengelompokan ketiga modul dibandingkan dengan dua algoritma lainnya yaitu LSA dan LDA. Terlihat bahwa LDA gagal melakukan pengelompokan langkah-langkah pengujian. Nilai Cohen Kappa yang dihasilkan juga bernilai negatif yang berarti model tidak berfungsi. Dan terlihat bahwa NMF memberikan skor Cohen Kappa yang bagus sekali lagi dibandingkan dengan dua algoritma lainnya. Namun, dalam dokumen kasus uji, NMF melakukan pekerjaan yang lebih baik menggunakan pencarian Bigram dan Trigram daripada Unigram. NMF-Unigram hanya mencapai nilai Cohen kappa 0,39, sedangkan NMF - Bigram dan Trigram bisa mencapai 54%. Hasil ini berbeda dari akurasi, *precision*, dan *recall*, di mana Unigram memberikan skor yang lebih baik. Sehingga dapat disimpulkan bahwa pemodelan topik kebutuhan bekerja dengan baik menggunakan algoritma NMF dengan pencarian Unigram. Namun, dalam pengelompokan langkah pengujian, algoritma NMF bekerja lebih baik menggunakan pencarian Bigram dan Trigram.

Berdasarkan Tabel 4.16, Unigram dengan ketiga algoritma tersebut menghasilkan nilai *precision* NMF 73,7% topik dua untuk persyaratan dan 77,8% topik satu untuk kasus uji. Nilai ini lebih tinggi dari metode bigram atau trigram, yang hanya mencapai nilai *precision* 63,2% untuk persyaratan dan 76,5% untuk kasus uji. Dalam aplikasi pertama ini, metode pemodelan topik NMF bekerja paling baik untuk memisahkan persyaratan dan langkah pengujian dalam kasus uji berdasarkan topik yang diinginkan.

Karena Unigram bekerja lebih baik jika dibandingkan dengan bigram atau trigram. Oleh karena itu, pada pemodelan topik *bug* dan *defect* hanya menggunakan metode unigram untuk menemukan *bug* dan tautan cacat dengan menguji hasil yang diharapkan untuk menemukan hubungan cacat. Ketiga metode pemodelan topik tersebut dibandingkan lagi untuk melihat kemiripannya. Dari daftar hasil yang diinginkan dalam dokumen kasus pengujian, daftar *bug* yang muncul selama pengujian (bisa selama pengujian unit atau pengujian integrasi), dan daftar kerusakan saat pengguna menguji fungsi lengkap aplikasi setelah rilis produk (pengujian regresi).

Sekali lagi, metode NMF memberikan hasil nilai akurasi terbaik yang ditunjukkan pada Tabel 4.17 dengan nilai 48% pada posisi pertama berbeda jauh dengan LDA yang hanya mencapai 38% di posisi kedua. Karena data yang dipakai tetaplah tidak seimbang jumlahnya perlu dilihat bagaimana suatu model mencapai kesepakatan penentuan topik. Pada Tabel 4.18, nilai Cohen Kappa masuk dalam kategori positif 0,28 untuk pemodelan NMF. Akan tetapi, posisi kedua justru jatuh pada pemodelan LSA bukan LDA. Nilai *precision* dan *recall* pada Tabel 4.19, menunjukkan bahwa NMF memberikan nilai *precision* yang hampir sama dengan LSA dengan 65% pada topik 1 dan mencapai nilai *recall* tertinggi pada topik 2 dan 3 dengan 68% dan 70,8%. Semakin menguatkan bahwa NMF bekerja dengan baik pada pengelompokan topik *bug* dan *defect* di aplikasi pertama ini.

Tabel 4.17 Nilai Akurasi *Bug* dan *Defect* Aplikasi Pertama

Akurasi *Bug* dan *Defect*	LSA	LDA	NMF
	37%	38%	48%

Tabel 4.18 Nilai Cohen Kappa *Bug* dan *Defect* Aplikasi Pertama

Cohen Kappa *Bug* dan *Defect*	LSA	LDA	NMF
	0,14	0,06	0,28

Tabel 4.19 Nilai *Precision* dan *Recall Bug* dan *Defect* Aplikasi Pertama

	LSA		LDA		NMF	
Topik	Precision	*Recall*	Precision	*Recall*	Precision	*Recall*
1	60%	12,8%	50%	34%	65%	27,7%
2	60%	24%	28,6%	40%	54,8%	68%
3	31,6%	100%	34,5%	41,7%	37,8%	70,8%

Berdasarkan pemodelan topik pada dokumen kebutuhan, test case, dan daftar *defect* setiap modul memiliki kata dominan sebagai identitasnya. Pada aplikasi pertama, kata kunci yang digunakan sebagai alat pencarian pada sumber kode adalah sebagai berikut :

a. Modul pertama adalah manajemen file. Kata kuncinya adalah pindai, scan, simpan, upload, kategori dokumen, dan PDF.
b. Modul kedua adalah status validasi. Kata kunci adalah valid, tolak, proses, operator.
c. Modul ketiga adalah proses promosi. Kata kuncinya adalah syarat, persyaratan, dan pangkat.

Terlepas dari kata-kata yang telah disebutkan, kata-kata lain yang merupakan identitas modul terus berubah. Bahkan, kata-kata tersebut tidak muncul pada sumber kode. Pada aplikasi pertama ini, penelusuran implementasi kebutuhan

dalam sumber kode melalui kata kunci di setiap modul hanya berhasil memetakan 40% kode terkait.

Pada ketiga artifak yang ada, yaitu kebutuhan, kasus uji, dan daftar *defect*, NMF memiliki performa paling baik. NMF dapat memisahkan tiga jenis kebutuhan dan memprediksi dengan tepat kebutuhan yang hanya memiliki sedikit data. Data pada kebutuhan dengan jenis kenaikan pangkat dan status validasi memiliki sedikit data, tapi mampu di identifikasi dengan baik oleh NMF. Pada aplikasi pertama ini, proses terjadi melalui aliran tunggal dimana variabel yang terlibat sama, hanya fungsi berlaku lah yang berbeda. Itulah mengapa tidak banyak kata unik yang menyusun penjelasan pada sebuah modul. Hal ini membuat proses pemisahan dan membedakan antara satu topik yang berkaitan dengan satu modul dengan modul lain menjadi sulit. Selain itu, istilah yang digunakan dalam kebutuhan sangatlah umum, tapi istilah pada dokumen pengujian bersifat spesifik membuat keduanya hampir tidak memiliki kesamaan untuk dapat ditelusuri. Dengan kata lain, data test case pada dokumen pengajian mengandung banyak *noise* dimana banyak istilah teknis dan variabel sama yang disebutkan secara berulang kali.

NMF bekerja lebih baik pada teks pendek dengan *noise* jika dibandingkan dengan dua algoritma lainnya (Chen *et al.*, 2019). Hasil ini didukung oleh temuan sebelumnya di mana kinerja NMF sedikit lebih baik pada data teks pendek dengan noise jika dibandingkan dengan LDA, tetapi LDA menggunakan *bag of words* dan bukan TF-IDF (Curiskis *et al.*, 2020). Sementara itu, banyaknya kata yang sama di setiap modul menyulitkan LSA untuk mengkategorikan topik. Ini karena LSA tidak peduli dengan urutan kata dan bahkan tidak dapat membedakan kalimat dari kumpulan kata saja (Braun and Langen, 2017). LDA cenderung memberikan pandangan umum dan non-spesifik dan fungsi probabilistik yang terus mengubah hasil dan kemunculan kata (Chen *et al.*, 2019)(Ahmadiyah, Sarno and Revindasari, 2020). Hal ini membuat pengelompokan topik tidak stabil karena terkadang ada topik yang hilang saat dibuat ulang dan dengan distribusi kata yang berbeda. Ketidakstabilan hasil pemodelan sebenarnya terjadi juga pada NMF. Namun sebuah penelitian menunjukkan bahwa NMF memberikan hasil yang lebih stabil

dibandingkan LDA berdasarkan nilai Normalized Pointwise Mutual Information (NPMI) sebagai evaluasinya (Belford, Mac Namee and Greene, 2018).

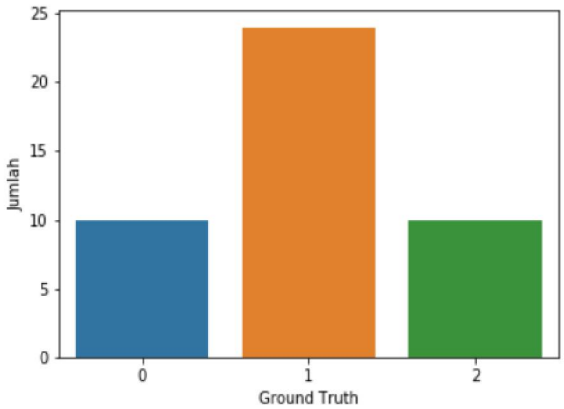
4. 3. 2. Aplikasi Kedua

4.3.2.1. Kebutuhan

Pada aplikasi kedua, isi dari kebutuhan juga dibagi menjadi 3 macam yang kemudian ditentukan sebagai topik yang seharusnya muncul sebagai berikut:

1. Penyimpanan data master, data yang harus ada paling awal agar proses dapat dilanjutkan
2. Pembuatan laporan, laporan kegiatan dan laporan anggaran atas kegiatan tersebut beserta detail atributnya
3. Data pendukung anggaran yaitu DIPA dan proses pencarian detail penyelenggaraan kegiatan terkait.

Terdapat banyak proses DIPA yang dilakukan pada aplikasi ini seperti upload DIPA dari sumber perencanaan, atribut anggaran yang terkait DIPA hingga proses pencarian kegiatan terkait yang diatur pada DIPA. Selain itu laporan kegiatan dan laporan anggaran memiliki penjelasan yang sedikit berbeda sehingga sulit ditemukan kedalam satu topik. Oleh karena itu, jumlah k topik yang dipakai adalah lima bukan tiga. Selanjutnya topik tersebut disusutkan kembali kedalam 3 topik inti yang telah disebutkan diatas. Maka persebaran topik pada aplikasi kedua ini menjadi tidak seimbang seperti yang ada pada Gambar 4.25.



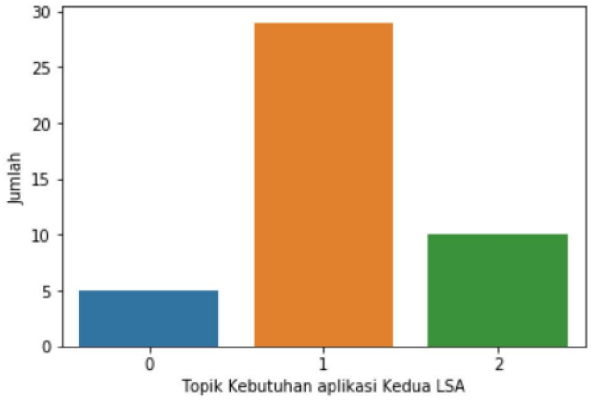
Gambar 4.25 Persebaran *Ground truth* Topik Kebutuhan Aplikasi Kedua

4.3.2.1.1. LSA

```
TOP 15 untuk setiap topik #0
['target', 'komponen', 'kerja', 'dana', 'bidang', 'output', 'master', 'dipa', 'unit', 'mencantumkan', 'pengguna', 'data', 'angg
aran', 'laporan', 'kegiatan']


TOP 15 untuk setiap topik #1
['dana', 'sub', 'sesuai', 'kedalam', 'terhubung', 'output', 'rkkl', 'menampilkan', 'unit', 'komponen', 'pencarian', 'anggaran',
'disetujui', 'dipa', 'data']


TOP 15 untuk setiap topik #2
['bidang', 'teknologi', 'hirarki', 'memiliki', 'sesuai', 'dilaksanakan', 'ditentukan', 'laporan', 'kedalam', 'induk', 'target',
'kegiatan', 'manajemen', 'master', 'data']


TOP 15 untuk setiap topik #3
['mitra', 'disetujui', 'sub', 'target', 'dipa', 'induk', 'teknologi', 'bidang', 'perencanaan', 'disesuaikan', 'lembaga', 'outpu
t', 'kerja', 'master', 'unit']


TOP 15 untuk setiap topik #4
['bidang', 'data', 'rkkl', 'penjelasan', 'dipa', 'target', 'kedalam', 'terlibat', 'dihadapi', 'master', 'disetujui', 'pencaria
n', 'kendala', 'pengguna', 'mitra']
```

Gambar 4.26 Kata Penyusun Topik Kebutuhan LSA-Unigram Aplikasi Kedua

Pada Gambar 4.26, topik #2 dan #3 membicarakan proses input data master. Terdapat kata master pada kedua topik tersebut. Data-data master seperti kegiatan, target, induk, unit, output juga muncul pada kedua topik tersebut. Topik #1 muncul kata dipa, dana, komponen, output, dan rkkl yang mengarah pada penjelasan penyusunan dasar anggaran pada DIPA. Sehingga dapat disimpulkan kebutuhan dengan topik ini mengarah pada penjelasan DIPA. Selain itu terdapat kata pencariang dan data yang merujuk pada pencarian detail anggaran pada kegiatan dengan komponen anggaran yang disesuaikan DIPA. Topik #0 dan #4 membicarakan mengenai penyusunan laporan. Terdapat kata laporan, anggaran, kegiatan, kendala, mitra, dan target yang merujuk pada detail penjelasan kegiatan pada laporan kegiatan, selain itu terdapat kata dana dan anggaran, disetujui bidang yang merupakan detail penjelasan pembuatan laporan anggaran. Maka persebaran topik menjadi berubah menjadi seperti pada Gambar 4.27.

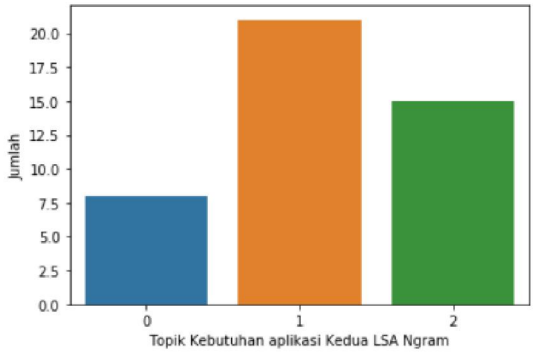


Gambar 4.27 Persebaran Topik Kebutuhan LSA-Unigram Aplikasi Kedua

```
Topic #0: laporan kegiatan, laporan kegiatan mencantumkan, kegiatan mencantumkan, kendala dihadapi, mencantumkan kendala dihada
pi, mencantumkan kendala, kegiatan mencantumkan kendala, kegiatan dilaksanakan, kegiatan mencantumkan lokasi, lokasi dimana keg
iatan, lokasi dimana, mencantumkan lokasi, mencantumkan lokasi dimana, dimana kegiatan, dimana kegiatan dilaksanakan


Topic #1: pencarian anggaran, pengguna pencarian, pengguna pencarian anggaran, disetujui dipa, anggaran disetujui dipa, anggara
n disetujui, pencarian anggaran disetujui, anggaran tercantuk kedalam, pencarian anggaran tercantuk, tercantuk kedalam, tercant
uk kedalam rkkl, kedalam rkkl, anggaran tercantuk, dipa penganggarannya, menampilkan uang


Topic #2: laporan anggaran, laporan anggaran penjelasan, anggaran penjelasan, kegiatan laporan, blu insentif, komponen blu, kom
ponen blu insentif, anggaran penjelasan komponen, penjelasan komponen, penjelasan komponen blu, anggaran mencantumkan, sumber d
ana, sumber dana apbn, dana apbn, mencantumkan sumber dana


Topic #3: sub output, sub output komponen, dipa kegiatan, output komponen, output sub output, output sub, bentuk tampilan berti
ngkat, bentuk tampilan, kegiatan bentuk, renja sesuai, renja sesuai dipa, bertingkat output sub, bertingkat output, tampilan be
rtingkat, sesuai dipa


Topic #4: kegiatan induk, kegiatan dilaksanakan, dilaksanakan kegiatan, dilaksanakan kegiatan induk, master kaitan, kaitan kegi
atan, kaitan kegiatan dilaksanakan, master kaitan kegiatan, kegiatan dilaksanakan kegiatan, induk sesuaikan, kegiatan induk ses
uaikan, kegiatan induk kegiatan, master kegiatan, induk sesuaikan perencanaan, diajukan lembaga
```

Gambar 4.28 Kata Penyusun Topik Kebutuhan LSA-Bigram Trigram Aplikasi Kedua

Semua kata yang terdapat pada Gambar 4.28 dapat disimpulkan bahwa topik #0 dan #2 menjelaskan tentang pembuatan laporan, baik laporan kegiatan maupun laporan anggaran. Detail atribut penyusun kedua laporan juga dijelaskan seperti lokasi kegiatan dan insentif anggaran. Topik #1 dan #3 menjelaskan atribut DIPA seperti komponen, output, dan sub output, kalimat anggaran disetujui dipa menjelaskan proses upload data DIPA dan sinkronisasi dengan atribut anggaran kegiatan lainnya. Kalimat pencarian anggaran dan tampilan bertingkat menyatakan bagaimana pengguna melakukan penelusuran anggaran pada kegiatan dan menampilkannya. Topik #4 membicarakan data master yang diperlukan seperti kalimat master kaitan kegiatan dan master kegiatan. Jumlah data kebutuhan pada setiap topik menjadi terbagi menjadi seperti pada Gambar 4.29.



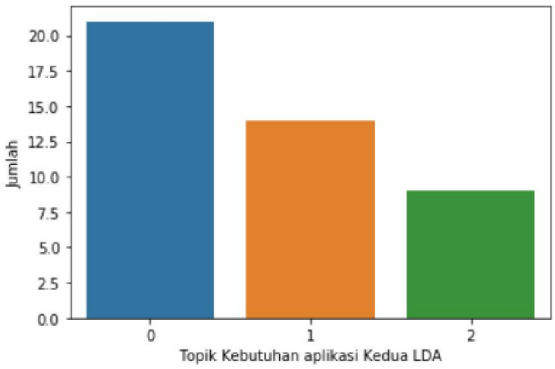
Gambar 4.29 Persebaran Topik Kebutuhan LSA-Bigram Trigram Aplikasi Kedua

4.3.2.1.2. LDA

```
TOP 15 untuk setiap topik #0
['hirarki', 'memiliki', 'unit', 'beserta', 'merekam', 'kendala', 'manajemen', 'pengguna', 'kegiatan', 'dipa', 'master', 'anggar
an', 'target', 'data', 'mitra']


TOP 15 untuk setiap topik #1
['berjalan', 'dihadapi', 'anggaran', 'sesuai', 'lembaga', 'unit', 'bidang', 'teknologi', 'dilaksanakan', 'induk', 'master', 'da
ta', 'laporan', 'mencantumkan', 'kegiatan']


TOP 15 untuk setiap topik #2
['beserta', 'pencarian', 'kedalam', 'perencanaan', 'bidang', 'anggaran', 'rkkl', 'sumber', 'unit', 'kerja', 'total', 'laporan',
'pengguna', 'kegiatan', 'dana']


TOP 15 untuk setiap topik #3
['master', 'target', 'dipa', 'posisi', 'berjalan', 'pelaksanaan', 'kegiatan', 'output', 'disetujui', 'pencarian', 'disesuaika
n', 'kedalam', 'ditentukan', 'unit', 'laporan']


TOP 15 untuk setiap topik #4
['dana', 'pencarian', 'sub', 'bertingkat', 'anggarannya', 'kegiatan', 'penjelasan', 'anggaran', 'disetujui', 'terhubung', 'lapo
ran', 'menampilkan', 'output', 'dipa', 'komponen']
```

Gambar 4.30 Kata Penyusun Topik Kebutuhan LDA-Unigram Aplikasi Kedua

Berdasarkan Gambar 4.30, topik #0 dan #1 membicarakan data master. Terdapat kata master, unit, induk, data, mitra yang merupakan data master utama. Topik #2 dan #3 membicarakan penyusunan laporan kegiatan dan laporan anggaran. Terdapat kata laporan, kegiatan, anggaran, target yang menunjukkan atribut penyusun laporan. Topik #4 membicarakan data DIPA dan pencarian kegiatan berdasarkan detail anggaran. Kata-kata komponen, output, sub, merupakan detail kegiatan yang dijelaskan anggarannya pada DIPA. Kata bertingkat merupakan tampilan saat pencarian. Kali ini persebaran topik pada data menjadi seperti pada Gambar 4.31 dimana ketiga topik menjadi kecil selisihnya satu sama lain.

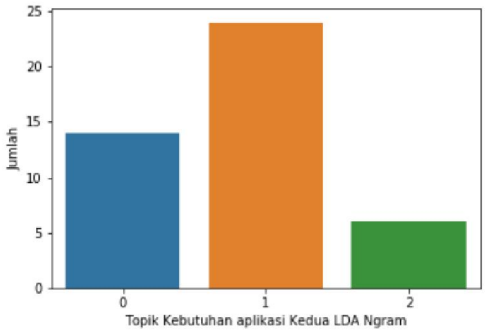


Gambar 4.31 Persebaran Topik Kebutuhan LDA-Unigram Aplikasi Kedua

```
Topic #0: laporan kegiatan, upload data dipa, data dipa, data dipa excel, dipa excel, upload data, dipa disetujui, dipa disetuj
ui unit, disetujui unit, pencarian dipa disetujui, pencarian dipa, kendala mitra bekerjasama, pengguna kendala, mitra bekerjasa
ma dalamnya, mitra bekerjasama


Topic #1: anggaran mencantumkan, laporan anggaran, kegiatan induk, anggaran mencantumkan pagu, ditetapkan realisasinya, mencant
umkan pagu ditetapkan, mencantumkan pagu, pagu ditetapkan realisasinya, pagu ditetapkan, disesuaikan laporan, output disesuaika
n, disesuaikan laporan unit, master output disesuaikan, master output, output disesuaikan laporan


Topic #2: disetujui dipa, anggaran disetujui, pencarian anggaran disetujui, anggaran disetujui dipa, pengguna pencarian anggara
n, pengguna pencarian, pencarian anggaran, uang disetujui, menampilkan uang disetujui, menampilkan uang, penganggarannya kompon
en, disetujui dipa penganggarannya, uang disetujui dipa, dipa penganggarannya komponen, dipa penganggarannya


Topic #3: kegiatan kegiatan, laporan kegiatan, tercantuk kedalam rkkl, tercantuk kedalam, anggaran tercantuk, kedalam rkkl, pen
carian anggaran tercantuk, anggaran tercantuk kedalam, dilaksanakan kegiatan induk, kaitan kegiatan, master kaitan kegiatan, ke
giatan dilaksanakan kegiatan, kaitan kegiatan dilaksanakan, dilaksanakan kegiatan, master kaitan


Topic #4: kegiatan mencantumkan, laporan kegiatan mencantumkan, pengguna kegiatan, pengguna kegiatan laporan, kegiatan laporan
utuh, laporan utuh, mencantumkan kendala dihadapi, mencantumkan kendala, kegiatan mencantumkan kendala, laporan kegiatan, kegia
tan laporan, kendala dihadapi, berjalan posisi, statusnya rutin, kegiatan mencantumkan berjalan
```

Gambar 4.32 Kata Penyusun Topik Kebutuhan LDA-Bigram Trigram Aplikasi Kedua

Berdasarkan pada kata-kata yang tertera pada Gambar 4.32, topik #0, #3, dan #4 membicarakan proses pembuatan laporan kegiatan dan anggaran. Terdapat kalimat laporan kegiatan, kendala mitra, kendala dihadapi, laporan utuh, yang menyatakan atribut pada pembuatan laporan. Topik #1 memunculkan kalimat master output disesuaikan, master output, output disesuaikan laporan lebih mengarah pada memasukkan data master. Topik #2 membicarakan komponen DIPA dan pencarian kegiatan dengan atribut anggarannya dan status penyelesaiannya. Kalimat yang anggaran disetujui dipa, menampilkan uang disetujui, pencarian anggaran menjelaskan proses persetujuan DIPA dan pencarian nya berdasarkan kegiatannya. Persebaran topik dengan metode ini memperbesar selisih pada topik ke dua seperti pada Gambar 4.33.



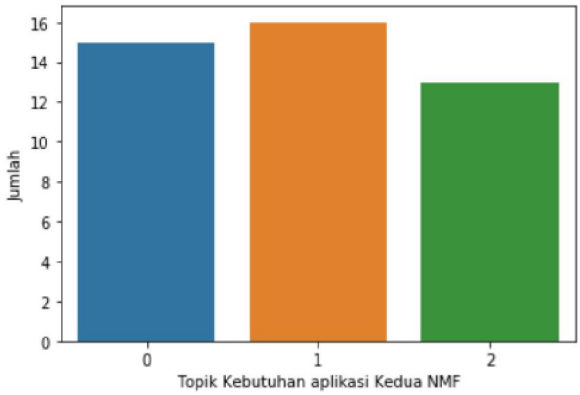
Gambar 4.33 Persebaran Topik Kebutuhan LDA-Bigram Trigram Aplikasi Kedua

4.3.2.1.3. NMF

```
TOP 15 untuk setiap topik #0
['merekam', 'beserta', 'penjelasan', 'anggarannya', 'pelaksanaan', 'target', 'posisi', 'berjalan', 'bidang', 'dilaksanakan', 'a
nggaran', 'pengguna', 'mencantumkan', 'laporan', 'kegiatan']


TOP 15 untuk setiap topik #1
['rkkl', 'sub', 'sumber', 'beserta', 'bertingkat', 'terhubung', 'total', 'pencarian', 'output', 'anggaran', 'disetujui', 'menam
pilkan', 'komponen', 'dana', 'dipa']


TOP 15 untuk setiap topik #2
['disetujui', 'induk', 'dilaksanakan', 'memiliki', 'hirarki', 'ditentukan', 'kegiatan', 'sesuai', 'rkkl', 'kedalam', 'manajeme
n', 'anggaran', 'master', 'dipa', 'data']


TOP 15 untuk setiap topik #3
['hirarki', 'laporan', 'dipa', 'bidang', 'teknologi', 'kegiatan', 'rkkl', 'induk', 'output', 'disesuaikan', 'perencanaan', 'lem
baga', 'master', 'kerja', 'unit']


TOP 15 untuk setiap topik #4
['total', 'manajemen', 'disetujui', 'hirarki', 'kedalam', 'rkkl', 'target', 'pencarian', 'penjelasan', 'terlibat', 'dihadapi',
'master', 'kendala', 'pengguna', 'mitra']
```

Gambar 4.34 Kata Penyusun Topik Kebutuhan NMF-Unigram Aplikasi Kedua

Berdasarkan Gambar 4.34, topik #0 dan #4 membicarakan pembuatan laporan kegiatan dan anggaran. Atribut kegiatan seperti kendala, pengguna,target, dan mitra muncul. Topik #1 membicarakan DIPA dan peneluasuran kegiatan berdasarkan atribut anggaran DIPA. Kata pencarian menandakan proses pencarian, kata dipa, komponen, anggaran dan disetujui mengarah pada penjelasan data DIPA. Topik #2 dan #3 membicarakan data master. Pada kedua topik muncul kata master, dan kata bidang, teknologi, kegiatan, unit yang mengarah pada jenis data master yang dimasukkan. Persebaran jumlah data pada tiga topik menjadi merata seperti pada Gambar 4.35.

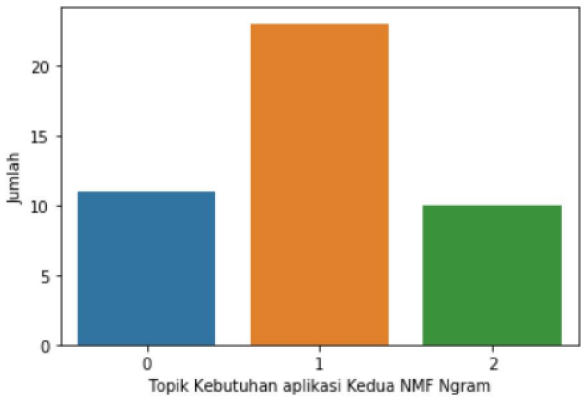


Gambar 4.35 Persebaran Topik Kebutuhan NMF-Unigram Aplikasi Kedua

```
Topic #0: laporan kegiatan, laporan kegiatan mencantumkan, kegiatan mencantumkan, kendala dihadapi, kegiatan mencantumkan kenda
la, mencantumkan kendala, mencantumkan kendala dihadapi, mencantumkan berjalan, berjalan posisi, berjalan posisi statusnya, pos
isi statusnya rutin, mencantumkan berjalan posisi, posisi statusnya, statusnya rutin, kegiatan mencantumkan berjalan


Topic #1: pengguna pencarian, pencarian anggaran, pengguna pencarian anggaran, disetujui dipa, pencarian anggaran disetujui, an
ggaran disetujui, anggaran disetujui dipa, pencarian anggaran tercantuk, anggaran tercantuk kedalam, anggaran tercantuk, tercan
tuk kedalam, tercantuk kedalam rkkl, kedalam rkkl, penganggarannya komponen, dipa penganggarannya komponen


Topic #2: laporan anggaran, anggaran penjelasan, laporan anggaran penjelasan, kegiatan laporan, komponen blu insentif, blu inse
ntif, penjelasan komponen blu, penjelasan komponen, anggaran penjelasan komponen, komponen blu, anggaran mencantumkan, laporan
kegiatan, memiliki lampiran, kegiatan memiliki lampiran, laporan kegiatan laporan


Topic #3: output komponen, sub output komponen, dipa kegiatan, sub output, output sub output, output sub, sesuai dipa kegiatan,
kegiatan bentuk tampilan, kegiatan bentuk, menampilkan renja, bentuk tampilan bertingkat, renja sesuai, sesuai dipa, tampilan b
ertingkat output, renja sesuai dipa


Topic #4: kegiatan induk, kaitan kegiatan, dilaksanakan kegiatan induk, master kaitan kegiatan, kaitan kegiatan dilaksanakan, m
aster kaitan, kegiatan dilaksanakan kegiatan, dilaksanakan kegiatan, kegiatan dilaksanakan, kegiatan induk kegiatan, kegiatan i
nduk sesuaikan, master kegiatan induk, master kegiatan, perencanaan diajukan lembaga, induk kegiatan
```

Gambar 4.36 Kata Penyusun Topik Kebutuhan NMF-Bigram Trigram Aplikasi Kedua

Berdasarkan kata yang tercantum pada Gambar 4.36, topik #0 dan #2 membicarakan laporan kegiatan dan laporan anggaran, terdapat kalimat berjalan posisi statusnya yang menerangkan status kegiatan pada laporan kegiatan. Kalimat komponen blu insentif menerangkan status dana pada laporan anggaran. Topik #1 dan #3 menerangkan masuknya data DIPA dan proses pencarian kegiatan dengan atribut penganggaran DIPA yang telah disetujui. Terdapat kalimat pencarian anggaran, disetujui dipa, menerangkan atribut pencarian. Kalimat output komponen, sub output dan komponen merupakan atribut kegiatan yang dijelaskan penganggarannya pada DIPA. Topik #4 menjelaskan masuknya data master. Terdapat kalimat master kegiatan, induk kegiatan, kaitan kegiatan, yang merupakan jenis data master yang dimasukkan. Persebaran topik dengan menggunakan metode NMF – Bigram dan Trigram mendekati *ground truth* seperti pada Gambar 4.37.



Gambar 4.37 Persebaran Topik Kebutuhan NMF Bigram Trigram Aplikasi Kedua

4.3.2.2. Pengujian

4.3.2.2.1. LSA

```
TOP 15 untuk setiap topik #0
['data', 'satuan', 'rkkl', 'mitra', 'admin', 'pengguna', 'pagu', 'target', 'mengisi', 'memilih', 'kegiatan', 'nilai', 'memasukk
an', 'operator', 'unit']


TOP 15 untuk setiap topik #1
['teknologi', 'mengklik', 'menu', 'upload', 'tombol', 'mengetik', 'prinas', 'master', 'masuk', 'bidang', 'data', 'pengguna', 'a
dmin', 'memilih', 'kegiatan']


TOP 15 untuk setiap topik #2
['tampilan', 'induk', 'prinas', 'mengetik', 'berjalan', 'rutin', 'status', 'operator', 'dimana', 'detail', 'unit', 'kegiatan',
'satuan', 'mengisi', 'target']


TOP 15 untuk setiap topik #3
['rkkl', 'tombol', 'kedalam', 'mengisi', 'file', 'master', 'menu', 'satuan', 'excel', 'sesuai', 'target', 'dipa', 'upload', 'pe
ngguna', 'data']


TOP 15 untuk setiap topik #4
['dialihkan', 'menghapus', 'ikon', 'hapus', 'target', 'ubah', 'master', 'simpan', 'nilai', 'mengetik', 'halaman', 'mengklik',
'tombol', 'data', 'admin']
```

Gambar 4.38 Kata Penyusun Topik Pengujian LSA-Unigram Aplikasi Kedua

Berdasarkan Gambar 4.38, topik #0 dan #2 terdapat atribut yang diisi nilainya untuk laporan seperti satuan, mitra, status, untuk laporan kegiatan dan pagu untuk laporan anggaran. Kata operator dan unit yang merupakan aktor yang melakukan input laporan juga muncul. Topik #3 terdapat kata upload, memilih, rkkl, dan dipa yang mengarah pada proses pencarian kegiatan berdasarkan komponen DIPA dan proses pengunggahan data DIPA sendiri. Topik #1 dan #4 terdapat kata master , dialihkan, halaman yang merupakan proses pertama kali yang dilakukan untuk memulai aplikasi yaitu memasukkan data master. Kata admin yang merupakan aktor pada proses memasukkan data master juga muncul.

```
Topic #0: operator unit memasukkan, unit memasukkan, memasukkan nilai, unit memasukkan nilai, operator unit, memasukkan nilai p
agu, nilai pagu, nilai blu, memasukkan nilai blu, nilai apbnp, memasukkan nilai apbnp, nilai insentif, memasukkan nilai insenti
f, kunci mendeskripsikan kegiatan, memasukkan kunci mendeskripsikan


Topic #1: unit mengisi, operator unit mengisi, satuan target, operator unit, mengisi satuan, unit mengisi satuan, mengisi satua
n target, target satuan, mengisi target, unit mengisi target, satuan target satuan, target satuan target, unit mengisi posisi,
posisi tahun, mengisi posisi tahun


Topic #2: operator unit memilih, unit memilih, operator memasukkan, kontribusi mitra, memasukkan kontribusi, operator memasukka
n kontribusi, memasukkan kontribusi mitra, mitra pengguna, mitra kerja, memilih mitra, unit memilih mitra, mendeskripsikan kegi
atan, kunci mendeskripsikan kegiatan, kunci mendeskripsikan, unit memasukkan kunci


Topic #3: kunci mendeskripsikan, kunci mendeskripsikan kegiatan, unit memasukkan kunci, memasukkan kunci mendeskripsikan, mende
skripsikan kegiatan, memasukkan kunci, operator unit memasukkan, unit memasukkan, memasukkan kendala, unit memasukkan kendala,
sesusai rkkl, memasukkan dampak, memasukkan dampak nya, dampak nya, unit memasukkan dampak


Topic #4: operator memasukkan, memasukkan kontribusi mitra, memasukkan kontribusi, kontribusi mitra, operator memasukkan kontri
busi, kontribusi mitra pengguna, kontribusi mitra kerja, operator memasukkan dokumen, dokumen pendukung, memasukkan dokumen, me
masukkan dokumen pendukung, satuan target, unit mengisi, operator unit mengisi, mitra pengguna
```

Gambar 4.39 Kata Penyusun Topik Pengujian LSA-Bigram Trigram Aplikasi Kedua

Menggunakan metode berbeda menghasilkan kata-kata penyusun setiap topik seperti pada Gambar 4.39. Topik #0 dan #1 terdapat kata-kata yang merupakan atribut laporan anggaran seperti nilai apbnp, nilai pagu, dan nilai blu. Kata-kata untuk atribut laporan kegiatan seperti satuan target, posisi tahun, dan aktor operator unit. Kedua topik membicarakan pembuatan laporan.

Topik #2,#3, dan #4 membicarakan proses memasukkan data master. Seperti memasukkan kunci, mendeskripsikan kegiatan. Akan tetapi, terdapat proses memasukkan dokumen pendukung pada topik #4 yang merupakan proses pembuatan laporan. Tidak terdapat kata-kata yang menjelaskan proses upload DIPA dan pencarian kegiatan berdasarkan komponen DIPA.

4.3.2.2.2. LDA

```
TOP 15 untuk setiap topik #0
['simpan', 'bekerjasama', 'dipa', 'masuk', 'baru', 'memilih', 'operator', 'kerja', 'data', 'memasukkan', 'berjalan', 'kedalam',
'kontribusi', 'mitra', 'pengguna']


TOP 15 untuk setiap topik #1
['pengguna', 'memilih', 'kegiatan', 'nilai', 'mitra', 'detail', 'daftar', 'admin', 'mengklik', 'menghapus', 'ikon', 'hapus', 'd
ialihkan', 'dana', 'halaman']


TOP 15 untuk setiap topik #2
['file', 'komponen', 'masuk', 'data', 'mengetik', 'menu', 'prinas', 'master', 'dipa', 'upload', 'bidang', 'admin', 'pengguna',
'kegiatan', 'memilih']


TOP 15 untuk setiap topik #3
['bekerjasama', 'mendeskripsikan', 'kerja', 'dimana', 'rkkl', 'pagu', 'memilih', 'satuan', 'kegiatan', 'target', 'nilai', 'meng
isi', 'memasukkan', 'operator', 'unit']


TOP 15 untuk setiap topik #4
['master', 'halaman', 'berjalan', 'status', 'rutin', 'nya', 'ubah', 'simpan', 'induk', 'sesuai', 'kegiatan', 'mengklik', 'admi
n', 'tombol', 'data']
```

Gambar 4.40 Kata Penyusun Topik Pengujian LDA-Unigram Aplikasi Kedua

Pada metode LDA kata penyusun yang terbentuk pada setiap topik menjadi seperti pada Gambar 4.40. Topik #0 , #3 memiliki kata target, satuan, mitra, mengisi, pengguna dan pagu yang menerangkan pengisian komponen pengisian laporan kegiatan dan laporan anggaran. Terdapat kata operator dan unit yang merupakan aktor yang memasukkan data laporan dan laporan anggaran. Topik #2 kata memilih , dipa, dan upload adalah proses melakukan proses unggah data DIPA dan pencarian data sesuai komponen DIPA. Topik #1 dan #4 terdapat kata kegiatan, mitra, daftar, yang mengindikasikan proses memasukkan data master. Kemudian kata admin sebagai aktor yang memasukkan data master juga muncul.

```
Topic #0: operator unit, unit memasukkan, operator unit memasukkan, unit memasukkan nilai, memasukkan nilai, nilai pagu, memasu
kkan nilai pagu, nilai insentif, memasukkan nilai insentif, memasukkan dokumen pendukung, dokumen pendukung, memasukkan dokume
n, operator memasukkan dokumen, unit mengisi, operator unit mengisi


Topic #1: operator unit, unit memasukkan, operator unit memasukkan, memasukkan nilai, unit memasukkan nilai, unit memilih, oper
ator unit memilih, admin mengetik, memasukkan kunci, kunci mendeskripsikan kegiatan, kunci mendeskripsikan, unit memasukkan kun
ci, memasukkan kunci mendeskripsikan, mendeskripsikan kegiatan, memilih mitra


Topic #2: admin memilih, memilih kegiatan, pengguna memilih, memilih file, pengguna detail, admin memilih kegiatan, kegiatan pr
inas, memilih kegiatan prinas, memilih berjalannya kegiatan, berjalannya kegiatan, memilih berjalannya, admin memilih berjalann
ya, capai capaian, unit mengisi capai, mengisi capai capaian


Topic #3: memasukkan kontribusi mitra, operator memasukkan kontribusi, memasukkan kontribusi, kontribusi mitra, operator memasu
kkan, admin memilih masuk, masuk bidang, memilih masuk bidang, memilih masuk, admin memilih, pengguna memilih, kontribusi mitra
pengguna, kontribusi mitra kerja, memasukkan nilai realisasi, nilai realisasi pagu


Topic #4: mengklik tombol, admin mengklik tombol, admin mengklik, operator unit, unit mengisi, operator unit mengisi, muncul pe
ngguna, tombol simpan, mengklik tombol simpan, bidang sektoral, unit mengisi satuan, mengisi satuan target, mengisi satuan, sat
uan target, beralih tab memasukkan
```

Gambar 4.41 Kata Penyusun Topik Pengujian LDA-Bigram Trigram Aplikasi Kedua

Berdasarkan kata-kata yang muncul pada Gambar 4.41, maka dapat disimpulkan topik #0,#1 dan Topik #3 muncul kata-kata untuk memasukkan nilai atribut laporan seperti pagu, dokumen pendukung, mendeskripsikan kegiatan, dan aktor operator unit juga muncul. Pada topik #2 muncul kata untuk pencarian kegiatan berdasarkan komponen DIPA. Seperti memilih berjalannya kegiatan, memilih masuk, dan memilih kegiatan prinas. Sementara itu, topik #4 adalah proses memasukkan data master, seperti mengisi satuan, bidang sectoral, terdapat aktor pada proses tersebut yaitu admin. Meski operator unit juga muncul tetapi admin lebih dominan.

4.3.2.2.3. NMF

```
TOP 15 untuk setiap topik #0
['nya', 'output', 'kontribusi', 'kerja', 'tab', 'beralih', 'sesusai', 'mendeskripsikan', 'rkkl', 'mitra', 'pagu', 'nilai', 'uni
t', 'operator', 'memasukkan']


TOP 15 untuk setiap topik #1
['nya', 'operator', 'terpilih', 'induk', 'unit', 'kunci', 'dimana', 'teknologi', 'masuk', 'mengetik', 'prinas', 'bidang', 'admi
n', 'memilih', 'kegiatan']


TOP 15 untuk setiap topik #2
['laporan', 'tombol', 'output', 'komponen', 'berjalan', 'status', 'rutin', 'kegiatan', 'dimana', 'detail', 'operator', 'unit',
'satuan', 'mengisi', 'target']


TOP 15 untuk setiap topik #3
['detail', 'berjalan', 'muncul', 'mitra', 'output', 'komponen', 'rkkl', 'menu', 'excel', 'data', 'file', 'dipa', 'upload', 'mem
ilih', 'pengguna']


TOP 15 untuk setiap topik #4
['tab', 'beralih', 'menghapus', 'ikon', 'hapus', 'sesuai', 'simpan', 'kedalam', 'ubah', 'halaman', 'master', 'mengklik', 'tombo
l', 'admin', 'data']
```

Gambar 4.42 Kata Penyusun Topik Pengujian NMF-Unigram Aplikasi Kedua

Pada Gambar 4.42 kata-kata penyusun yang muncul pada topik #0 dan #2 terdapat kata status, rutin, output, yang merupakan komponen laporan. Kemudian aktor pada proses tersebut juga muncul yaitu kata-kata operator dan unit. Topik #1 dan #4 terdapat kata master, induk, prinas, dan bidang yang merupakan atribut dari data master. Topik #3 terdapat kata detail, memilih, dan output untuk pencarian DIPA dan kata upload untuk proses unggah data DIPA.

```
Topic #0: memasukkan nilai, unit memasukkan nilai, unit memasukkan, operator unit memasukkan, operator unit, nilai pagu, memasu
kkan nilai pagu, nilai apbnp, memasukkan nilai apbnp, memasukkan nilai blu, memasukkan nilai insentif, nilai insentif, nilai bl
u, nilai pagu revisi, pagu revisi


Topic #1: operator unit mengisi, unit mengisi, satuan target, operator unit, mengisi satuan target, mengisi satuan, unit mengis
i satuan, target satuan, mengisi target, unit mengisi target, target satuan target, satuan target satuan, mengisi posisi, posis
i tahun, unit mengisi posisi


Topic #2: operator unit memilih, unit memilih, unit memilih mitra, memilih mitra, operator unit, mitra kerja bekerjasama, pengg
una bekerjasama, memilih mitra kerja, memilih mitra pengguna, mitra pengguna bekerjasama, kerja bekerjasama, mitra pengguna, mi
tra kerja, tombol laporan, tombol laporan anggaran


Topic #3: unit memasukkan kunci, mendeskripsikan kegiatan, memasukkan kunci, memasukkan kunci mendeskripsikan, kunci mendeskrip
sikan, kunci mendeskripsikan kegiatan, operator unit memasukkan, unit memasukkan, operator unit, unit memasukkan kendala, memas
ukkan kendala, sesusai rkkl, dampak nya, memasukkan dampak, unit memasukkan dampak


Topic #4: operator memasukkan, kontribusi mitra, operator memasukkan kontribusi, memasukkan kontribusi mitra, memasukkan kontri
busi, kontribusi mitra pengguna, kontribusi mitra kerja, mitra kerja, mitra pengguna, memasukkan dokumen pendukung, dokumen pen
dukung, memasukkan dokumen, operator memasukkan dokumen, memilih mitra, unit memilih mitra
```

Gambar 4.43 Kata Penyusun Topik Pengujian NMF-Bigram Trigram Aplikasi Kedua

Berdasarkan Gambar 4.43, maka topik #0, #1, dan #3 merupakan proses pembuatan laporan, seperti memilih kegiatan, memasukkan nilai pagu, apbnp, dan blu untuk lepaoran anggaran. Memasukkan posisi tahun dan target untuk laporan kegiatan. Topik #2 terdapat kata operator unit memilih yang merupakan proses

pencarian kegiatan berdasarkan DIPA.Karena atribut DIPA berkaitan erat dengan anggaran, makan laporan anggaran juga muncul pada topik ini tapi sepertinya tampilannya saja dan bukan proses inputnya. Topik #4 memasukkan data master karena condong membicarakan data mitra yang dimasukkan pada awal proses.

4.3.2.3. *Bug* and *Defect*

4.3.2.3.1. LSA

```
TOP 15 untuk setiap topik #0
['kunci', 'rkkl', 'satuan', 'kontribusi', 'sesuai', 'diketik', 'data', 'spesial', 'target', 'nilai', 'karakter', 'angka', 'meny
impan', 'huruf', 'mengetikkan']


TOP 15 untuk setiap topik #1
['teknologi', 'halaman', 'dokumen', 'anggaran', 'list', 'tersimpan', 'sesuai', 'bidang', 'pilihan', 'field', 'dipilih', 'lapora
n', 'muncul', 'data', 'kegiatan']


TOP 15 untuk setiap topik #2
['berpindah', 'formulir', 'terhapus', 'masuk', 'dimasukkan', 'diketik', 'spesial', 'karakter', 'kedalam', 'halaman', 'master',
'database', 'field', 'tersimpan', 'data']
```

Gambar 4.44 Kata Penyusun Topik *Bug* dan *Defect* LSA-Unigram Aplikasi Kedua

Berdasarkan Gambar 4.44 dan bila dilihat pada pengelompokan kata pada dokumen kebutuhan dan pengujian sebelumnya, maka pembagian topiknya adalah sebagai berikut :

- Topik #0 pencarian data berdasarkan anggaran dan kegiatan
- Topik #1 laporan kegiatan dan anggaran
- Topik #2 Memasukkan data master

4.3.2.3.2. LDA

```
TOP 15 untuk setiap topik #0
['master', 'muncul', 'output', 'kegiatan', 'halaman', 'tampilan', 'formulir', 'file', 'database', 'kedalam', 'angka', 'mengetik
kan', 'tersimpan', 'nilai', 'data']


TOP 15 untuk setiap topik #1
['muncul', 'rutin', 'sesuai', 'diketik', 'status', 'data', 'aktif', 'target', 'spesial', 'angka', 'karakter', 'mengetikkan', 'f
ield', 'huruf', 'menyimpan']


TOP 15 untuk setiap topik #2
['ditampilkan', 'unit', 'sektoral', 'teknologi', 'prinas', 'terpilih', 'list', 'muncul', 'anggaran', 'dokumen', 'bidang', 'dipi
lih', 'laporan', 'pilihan', 'kegiatan']
```

Gambar 4.45 Kata Penyusun Topik *Bug* and *Defect* LDA-Unigram Aplikasi Kedua

Berdasarkan Gambar 4.45 dan bila dilihat pada pengelompokan kata pada dokumen kebutuhan dan pengujian sebelumnya, maka pembagian topiknya adalah sebagai berikut :

- Topik #0 memasukkan data master
- Topik #1 membuat laporan kegiatan dan anggaran
- Topik #2 Pencarian kegiatan berdasarkan anggaran DIPA nya

4.3.2.3.3. NMF

```
TOP 15 untuk setiap topik #0
['output', 'kunci', 'rkkl', 'pagu', 'satuan', 'kontribusi', 'diketik', 'spesial', 'target', 'nilai', 'karakter', 'angka', 'meny
impan', 'huruf', 'mengetikkan']


TOP 15 untuk setiap topik #1
['dokumen', 'terpilih', 'prinas', 'anggaran', 'unit', 'sesuai', 'sektoral', 'teknologi', 'muncul', 'list', 'laporan', 'piliha
n', 'bidang', 'dipilih', 'kegiatan']


TOP 15 untuk setiap topik #2
['tampilan', 'terhapus', 'masuk', 'berpindah', 'dimasukkan', 'formulir', 'sesuai', 'kedalam', 'muncul', 'halaman', 'master', 'd
atabase', 'tersimpan', 'field', 'data']
```

Gambar 4.46 Kata Penyusun Topik *Bug* and *Defect* NMF-Unigram Aplikasi Kedua

Berdasarkan Gambar 4.46 dan bila dilihat pada pengelompokan kata pada dokumen kebutuhan dan pengujian sebelumnya, maka pembagian topiknya adalah sebagai berikut :

- Topik #0 Pembuatan lapioran kegiatan dan anggaran
- Topik #1 Pencarian kegiatan berdasarkan anggaran DIPA
- Topik #2 Memasukkan data master

4.3.2.4. Evaluasi Hasil Pemodelan Topik

Pada aplikasi kedua ini, terdapat lima modul oleh karena itu inisiasi jumlah topik nya juga adalah lima. Lima modul yang disebutkan adalah master data, manajemen data DIPA, laporan kegiatan, laporan anggaran, dan view laporan secara keseluruhan. Ketika pemodelan topik diberlakukan, baik menggunakan LDA,NMF, dan LSA terdapat pola pengelompokan yang hampir sama. Hasil pemisahan kelima topik dari ketiga algoritma dapat diperkecil lagi menjadi dua topik. Menggunakan pembobotan kata unigram, topik pertama adalah seputar dana

atau anggaran dan topik kedua seputar kegiatan beserta atributnya. Pada hasil pemodelan topik dengan jumlah topik sebanyak lima pada metode LDA, topik pertama adalah seputar master data yang terkait dengan detail kegiatan seperti pengukuran target, penghapusan dan penambahan kegiatan dan mitra pelaksanaan kegiatan.

Topik kedua juga masih seputar kegiatan seperti penambahan master data pada bidang teknologi, unit kerja dimana kegiatan dilaksanakan dan penambahan nama kegiatan itu sendiri. Laporan kegiatan seperti kendala, lokasi kegiatan dilaksanakan dan data dukung kegiatan juga masuk kedalam topik kedua. Terlihat topik pertama dan topik kedua sama-sama membahas atribut kegiatan. Sementara itu laporan anggaran beserta atribut sumber dana dan pembagian laporan anggaran pada setiap unit Masuku pada topik ke tiga. Topik kelima tidak dapat disimpulkan sementara itu topik ke lima mengenai acuan pokok anggaran dan proses pengunggahannya kedalam apikasi. Sehingga topik ke tiga dan kelima membahasn dana atau anggaran. Hasil pemodelan topik menggunakan NMF memisahkan komponen dana menjadi satu topik pada topik kedua, sisanya merupakan atribut pelaksanaan kegiatan.

Sementara itu LSA menghasilkan kombinasi baru, dimana semua hal yang terkait dengan master data masuk kedalam topik ketiga. Sebagian besar atribut anggaran masuk pada topik kedua sementara yang sebagian kecil lainnya masuk kedalam topik pertama dengan atribut utama laporan kegiatan. Topik ke empat bercampur tapi lebih dominan membicaran seputar kegiatan. Ketika pemisahan topik diperkecil menjadi dua dari lima topik dikarenakan kesamaan pola, nilai akurasi nya cukup tinggi. NMF pada tempat pertama mencapai 79%, LDA dan LSA berada di tempat kedua mencapai 75%.

Metode pencarian dan pembobotan kata menggunakan bigram dan trigram juga masih memiliki pola yang sama. Pada metode LDA topik ke 3 susah dikenali karena konten yang bercampur dengan cenderung mengarah pada anggaran karena mengandung kata-kata seputar acuan pokok anggaran. Topik satu dan dua secara jelas mengenai dana anggaran. Topik tiga dan empat mengenai kegiatan dan detail

pelaksanaan kegiatan. Metode NMF menghasilkan setiap topik yang lebih mudah di identifikasi daripada LDA, Topik satu dan lima mengenaik kegiatan dan sisanya mengenai anggaran tidak ada topik yang berisi campuran keduanya. LSA masih memiliki hasil pemisahan topik yang lebih baik daripada LDA, topik dua, tiga, dan empat mengenai anggaran tidak ada kebutuhan mengenai kegiatan pada ketiga topik tersebut. Sementara itu kebutuhan mengenai kegiatan terkumpul pada topik pertama. Beberapa kebutuhan pada topik kelima mengenai anggaran tapi lebih banyak yang merupakan penjelasan akan penambahan master kegiatan dan kaitan kegiatan. Aplikasi bigram dan trigram berhasil memisahkan dengan sangat baik pada metode NMF yang mencapai nilai akurasi 81% dan LSA di tempat kedua dengan 77%. Akan tetapi, LDA justru mengalami penurunan akurasi menjadi hanya 61%.

Permasalahan muncul ketika pola pengelompokan kegiatan dan anggaran tidak dapat diterapkan pada pengujian. Pengujian laporan kegiatan dengan pengujian laporan anggaran justru seringkali masuk kedalam satu topik yang sama. Pola pengelompokan harus berubah karena terdapat banyak kesalahan, terutama menentukan ke kelompok yang manakah kombinasi laporan kegiatan dan anggaran ini. Untuk itu melihat pola yang ada pada pengujian, pengelompokan pada kebutuhan juga berubah. Semua hal yang membicarakan modul master masuk pada satu kelompok. Semua hal yang membicarakan modul laporang kegiatan dan laporan anggaran masuk kedalam satu kelompok. Semua hal yang membicarakan acuan anggaran (DIPA) dan view laporan masuk pada satu kelompok. Terdapat total tiga kelompok yang disimpulkan dari hasil lima topik yang dihasilkan dari pemodelan topik. Nilai akurasi pada kebutuhan berubah seperti yang tertera pada Tabel 4.20, NMF tidak lagi merupakan metode terbaik. Hal ini dikarenakan pada aplikasi pencarian dan pembobotan kata unigram nilai akurasi LSA adalah yang terbaik yaitu 79%. NMF berada pada tempat kedua 70% dan LDA berada pada tempat terakhir. NMF kembali menghasilkan nilai akurasi terbaik pada pencarian dan pembobotan bigram dan trigram yaitu 68%.

Nilai Cohen Kappa yang tinggi pada LSA baik dalam dokumen kebutuhan maupun dokumen pengujian yang ditunjukkan pada Tabel 4.21 semakin

memperkuat bahwa LSA adalah pemenang kali ini. LSA mencapai nilai koefisien Cohen Kappa 0,62 dalam dokumen persyaratan dan 0,72 dalam dokumen kasus uji dengan pencarian kata unigram. Namun, LSA lebih rendah dari NMF dalam dokumen persyaratan saat menggunakan pencarian kata bigram dan trigram. Dimana LSA hanya mencapai 45% sedangkan NMF mencapai 47%.

Seperti yang ditunjukkan pada Tabel 4.22, nilai *precision* menggunakan Unigram sekali lagi lebih unggul daripada menggunakan Bigram dan Trigram. Pemodelan menggunakan algoritma LSA memberikan skor sempurna pada pengelompokan topik pertama dan 79,3% pada topik kedua dokumen persyaratan. Dokumen uji LSA dan NMF memberikan hasil yang hampir sama baiknya. NMF berhasil memberikan nilai *precision* yang sangat baik pada topik kedua, sedangkan LSA hanya memberikan hasil *precision* sebesar 94,6%. Namun pada dua topik lainnya, NMF tidak dapat memberikan nilai *precision* yang lebih baik dari LSA. LDA datang terakhir. Pada semua topik, nilai *precision* bahkan tidak bisa menyentuh 70%,

Tabel 4.20 Nilai Akurasi Kebutuhan dan Pengujian Aplikasi Kedua

Akurasi	Requirement			Test Case		
	LDA	NMF	LSA	LDA	NMF	LSA
No Stem	52%	70%	79%	75%	79%	84%
Bigram dan Trigram	56%	68%	65%	55%	56%	60%

Tabel 4.21 Nilai Cohen Kappa Kebutuhan dan Pengujian Aplikasi Kedua

Cohen Kappa	Requirement			Test Case		
	LDA	NMF	LSA	LDA	NMF	LSA
No Stem	28%	55%	63%	60%	68%	72%

Bigram dan Trigram	27%	47%	45%	19%	31%	37%

Tabel 4.22 *Precision* dan *Recall* Kebutuhan dan Pengujian Aplikasi Kedua

LDA Unigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	38,1%	80%	90%	50%
2	64,3%	37,5%	94,1%	82,1%
3	66,7%	60%	44%	91,7%
LDA Bigram or Trigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	42,9%	60%	45,5%	27,8%
2	62,5%	62,5%	64,4%	74,4%
3	66,7%	40%	30,8%	33,3%
NMF Unigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	53,3%	80%	62,1%	100%
2	87,5%	58,3%	100%	69,2%
3	69,2%	90%	76,9%	83,3%

NMF Bigram or Trigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	45,5%	50%	60%	33,3%
2	87%	83,3%	82,1%	59%
3	50%	50%	29%	75%
LSA Unigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	100%	50%	68%	94,4%
2	79,3%	95,8%	94,6%	89,7%
3	70%	70%	85,7%	50%
LSA Bigram dan Trigram				
	Kebutuhan		*Test Case*	
	Precision	*Recall*	*Precision*	*Recall*
1	50%	40%	55,6%	55,6%
2	85,7%	75%	78,1%	64,1%
3	46,7%	70%	36,8%	58,3%

Berdasarkan hasil dari pemodelan topik pada dokumen kebutuhan dan pengujian dimana metode unigram menghasilkan nilai akurasi, koefisien Cohen Kappa, dan bahkan *precision* atau *recall* yang lebih baik, maka untuk pemodelan

topik *bug* dan *defect* metode yang digunakan hanyalah unigram dan tidak membandingkannya dengan Bigram atau Trigram lagi. Seperti yang ditunjukkan pada Tabel 4.23, LSA masih unggul dalam menemukan topik yang sesuai dibandingkan dengan dua metode lainnya. Akurasi LSA adalah 56% di tempat pertama dengan selisih yang sangat sedikit dengan NMF yang menghasilkan nilai 54% di tempat kedua dan LDA di tempat terakhir dengan 53%.

Sedangkan LSA menunjukkan bahwa model bekerja cukup baik dengan nilai yang tercantum pada Tabel 4.24 untuk memisahkan topik dengan koefisien Cohen Kappa tertinggi sebesar 0,33 dibandingkan dengan dua metode lainnya yang sekali lagi hanya memiliki selisih yang sangat sedikit 0,32 untuk NMF dan 0,29 untuk LDA. Melihat nilai *precision* dan *recall* pada ketiga topik dari Tabel 4.25, LSA memberikan hasil yang baik dalam pengelompokan kedua topik tersebut. Nilai *precision* LSA tertinggi pada topik 2 yaitu 96,3% dibandingkan NMF 95,8% dan LDA 80%, Nilai *recall* LSA juga merupakan hasil tertinggi pada topik 3 yaitu 85,7% dibandingkan NMF yang hanya 60,7% dan LDA 53,6%. Metode pemodelan topik LSA berkinerja terbaik dalam menemukan topik yang sesuai dalam aplikasi kedua ini.

Tabel 4.23 Nilai akurasi Pemodelan Topik *Bug* dan *Defect* Aplikasi Kedua

Akurasi *Bug* dan *Defect*	LSA	LDA	NMF
	56%	53%	54%

Tabel 4.24 Nilai Cohen Kappa *Bug* dan *Defect* Aplikasi Kedua

Cohen Kappa *Bug* dan *Defect*	LSA	LDA	NMF
	0,33	0,29	0,32

Tabel 4.25 Nilai *Precision* dan *Recall Bug* dan *Defect* Aplikasi Kedua

	LSA		LDA		NMF	
Topik	Precision	*Recall*	Precision	*Recall*	Precision	*Recall*
1	36,4%	17,4%	40%	52,2%	40%	52,2%
2	96,3%	56,5%	80%	52,2%	95,8%	50%
3	40,7%	85,7%	40,5%	53,6%	39,5%	60,7%

Berdasarkan topik pemodelan pada dokumen requirement, test case, dan *defect* list, ditemukan bahwa setiap modul memiliki kata dominan sebagai identitasnya. Pada aplikasi kedua ini, kata kunci yang digunakan sebagai alat pencarian sumber kode adalah sebagai berikut:

a. Sebuah. Modul pertama adalah master data. Kata kunci yang digunakan adalah "master", "kegiatan kegiatan", "data awal", "data satuan", "nama kegiatan"
b. Modul kedua adalah laporan kegiatan dan anggaran. Kata kuncinya adalah pnbp, blu, status, target, dokumen pendukung
c. Modul ketiga adalah pengelolaan data DIPA. Kata kuncinya adalah sub, keluaran, komponen, DIPA, unggah, impor
d. Modul keempat adalah menelusuri dan menampilkan laporan berdasarkan data kegiatan DIPA. Kata kuncinya adalah cari, cari, pilih, tampil, daftar, bidang teknologi, bidang prinas

Terlepas dari kata-kata yang telah disebutkan, kata-kata lain adalah identitas modul, tetapi istilah-istilahnya terus berubah. Istilah-istilah ini tidak pernah muncul di kode sumber. Pelacakan implementasi persyaratan dalam kode sumber melalui kata kunci di setiap modul menunjukkan hasil yang lebih baik daripada aplikasi pertama. Dalam aplikasi kedua ini, kode sumber terkait dapat dilacak hingga 62%.

Berbeda dengan aplikasi pertama, aplikasi kedua memiliki kata-kata unik yang secara konsisten disebutkan mulai dari persyaratan tiap modul sehingga kendala yang dihadapi dalam penelitian (Nanang, Penelusuran and Penelusuran, 2018) tidak ditemukan. Satu-satunya perubahan adalah hal-hal yang dibahas oleh persyaratan secara umum. Itu dibagi menjadi dua kategori, kegiatan dan dana. Kemudian test case menjelaskan secara lebih detail melalui tahapan proses dalam aplikasi. Kondisi ini menyebabkan banyaknya topik tentang kebutuhan perlu diubah sehingga kebutuhan tersebut dapat dipetakan dengan kasus uji yang ada secara tepat

Berdasarkan konsistensi kemunculan kata-kata konstituen unik di setiap modul, LSA dapat berkinerja lebih baik daripada NMF dan LDA. Namun, LSA tidak bisa membedakan arti kata dalam kalimat. Oleh karena itu perlu dilakukan penyesuaian seperti letak subjek dan objek serta jenis kata seperti kata benda dan kata sifat menurut penelitian sebelumnya (Suleman and Korkontzelos, 2021). Sebuah kata memiliki banyak arti tergantung pada penempatannya dalam sebuah kalimat, sehingga proses pemisahan ini diperlukan untuk pemodelan topik LSA yang lebih baik. LSA bisa berhasil dengan data yang berisi kata-kata unik yang konsisten digunakan atau dengan menyesuaikan sinonim untuk kata-kata (A Amalia et al, 2019). Hasil ini juga didukung oleh penelitian sebelumnya dimana metode LSA dengan TF-IDF memiliki hasil terbaik ketika digunakan dalam teks pendek dari tweet, dan email berisi masalah perawatan kesehatan daripada LSA yang dikombinasikan dengan Doc2Vec dan metode lain seperti LDA dengan TF-IDF atau Doc2vec (Lossio-Ventura *et al.*, 2021).

4. 3. 3. Analisa Hasil Pemodelan Topik Kedua Aplikasi

Proses penelusuran menggunakan pemodelan topik bergantung pada kemunculan istilah dan frase yang digunakan. Ketika suatu istilah atau frase dicantumkan secara konsisten dan unik untuk mewakili modul tertentu, maka akan semakin baik pula *trace link* yang tercipta. Aplikasi kedua lebih mudah ditelusuri dikarenakan kata dan istilah pada deskripsi kebutuhan unik pada setiap modulnya berbeda dengan aplikasi pertama dimana modul pada setiap aplikasi di deskripsikan

dengan istilah dan kata yang disebutkan berulang pada setiap modul. Istilah dan frase yang dijadikan kata kunci setiap modul pada aplikasi kedua juga muncul secara konsisten pada pengujian dan sumber kode. Sementara itu, aplikasi pertama tidak muncul secara konsisten dan bahkan beberapa tidak disebutkan atau berganti istilah. Secara keseluruhan penelusuran menggunakan pemodelan topik menghasilkan performa lebih baik dengan istilah satu kata dibandingkan dengan frase dan istilah dua kata.

4. 4. Klasifikasi *Defect*

Data *defect* dikumpulkan sepanjang pengembangan dan pasca rilis produk baik pada aplikasi pertama dan kedua. *Defect* kemudian dikategorikan sebagai mayor dan minor berdasarkan pendapat ahli yang terdiri dari 4 orang yang merupakan tim pengembang termasuk di dalamnya perwakilan dari pengguna. Atribut yang dianggap berpengaruh dalam menghasilkan *defect* adalah sebagai berikut :

1. Jumlah *story point* pada penjelasan kebutuhan
2. Konsistensi pencantuman istilah atau kata pada dokumen kebutuhan yang dicantumkan kembali pada dokumen pengujian
3. Konsistensi pencantuman istilah atau kata pada dokumen kebutuhan dan pengujian yang tercantum pada sumber kode.
4. Jenis perubahan kebutuhan, apakah penambahan dari yang awalnya tidak ada atau perubahan sebagian.
5. Jumlah pengujian diulang hingga *defect* tertangani.

Atribut 1, 4, 5 diketahui bukan berdasarkan *trace link* menggunakan pemodelan topik dikarenakan nilai akurasi, *precision*, dan *recall* yang hanya mampu menyentuh sekitar 60% secara keseluruhan. Oleh karena itu, data tersebut diperoleh dari RTM yang dilakukan secara manual.

Sementara itu, atribut nomor 2 dan 3 adalah berdasarkan tahapan sebelumnya yaitu pemodelan topik. Dimana kata penyusun menjadi kunci terciptanya *link* penelusuran. Berdasarkan kondisi kemunculan kata-kata tersebut, maka terdapat 5 kondisi sebagai berikut :

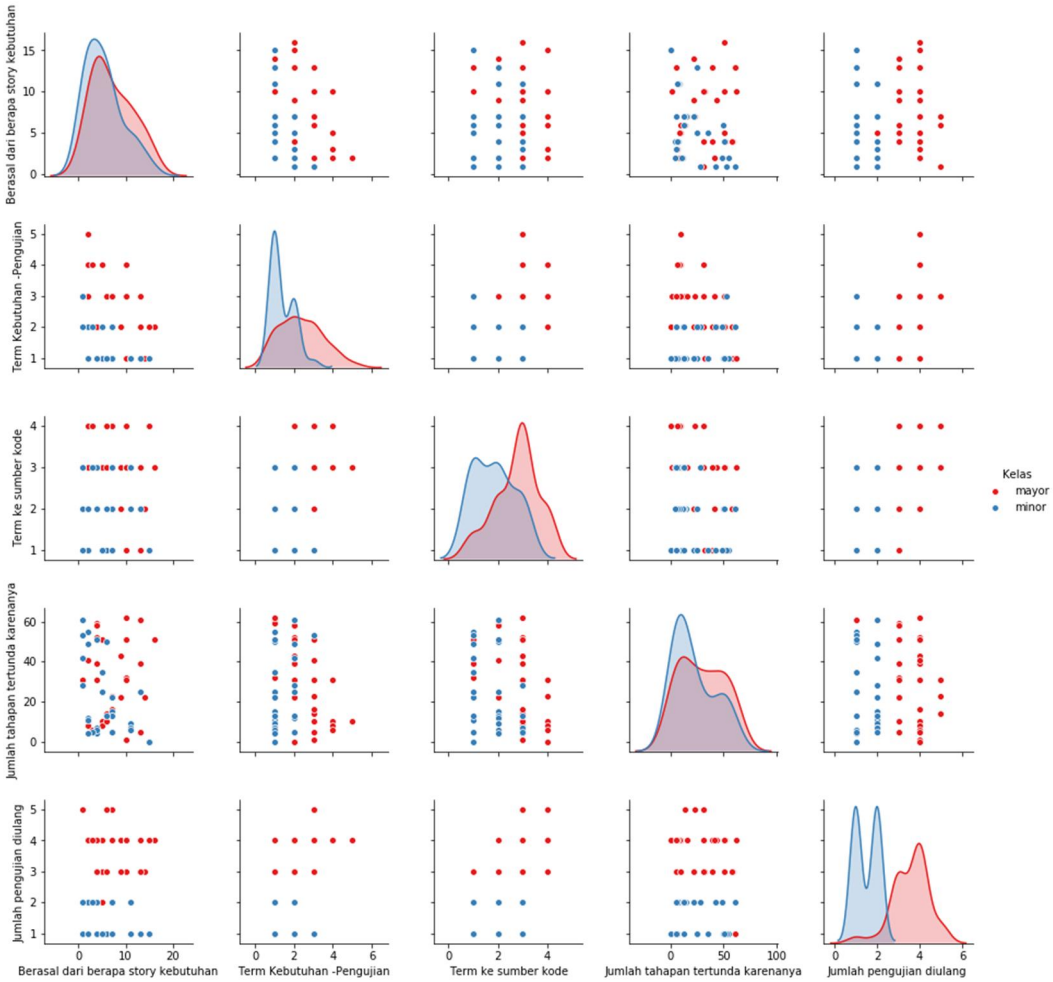
1. Ditemukan tidak berubah Sebagian besar
2. Ditemukan berubah Sebagian kecil
3. Ditemukan berubah Sebagian besar
4. Tidak ditemukan sebagian kecil
5. Tidak ditemukan sebagian besar

Sementara itu, atribut data set *defect* nomor 1, 4, dan 5 berasalah dari *link* penelusuran yang tercipta oleh pemodelan topik yang menggolongkan kebutuhan, *test-case* dan *defect* pada topik yang sama dari metode pemodelan terbaik, yaitu NMF pada aplikasi pertama dan LSA pada aplikasi kedua. Data *defect* pada kedua aplikasi akan digabung pada dataset ini semua kecacatan fungsi akan diidentifikasi sebagai *defect* seperti pada Tabel 4.26.

Tabel 4.26 Contoh Data *Defect*

No	*Defect*	Berasal dari berapa story kebutuhan	Term Kebutuhan -Pengujian	Term ke sumber kode	Jumlah tahapan tertunda karenanya	Perubahan kebutuhan	Jumlah pengujian diulang	Kelas
1	Hasil pemindaian hanya tampil Sebagian, berhenti pada tahap ketiga	13	1	2	61	add	1	mayor
2	variabel checked true, Http upload tidak tereksekusi. Tidak mengirim variabel pada perintah penyimpanan untuk mengubah ekstensi, berhenti pada tahap kelima.	4	1	2	59	change	3	mayor
3	File tersimpan dengan nama untitled, atribut nama file tidak terkirim variabelnya pada proses penyimpanan. Tahap terakhir gagal	4	2	2	58	change	3	mayor
4	Tipe dokumen masih tampil semua belum sesuai kategori yang dipilih pada sub menunya berhenti pada step ke 3	1	2	3	28	-	2	minor
5	Terdapat data dengan nama dokumen kosong, dokumen tidak dapat dibuka, dengan status proses / proses operator berhenti pada step 6	13	1	2	25	-	1	minor

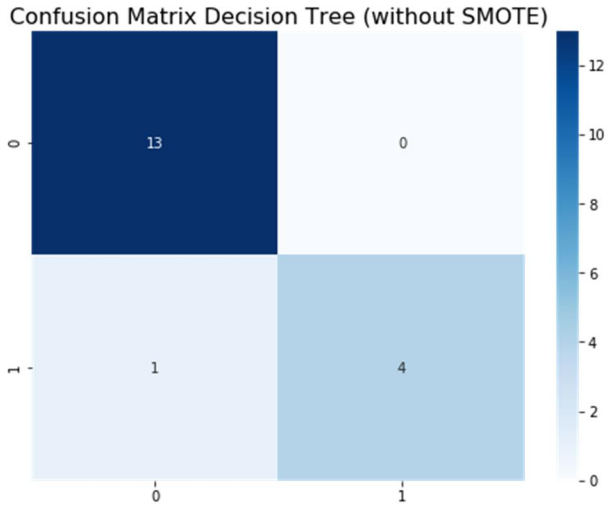
Dataset yang terbentuk berdasarkan 5 atribut memiliki persebaran seperti pada Gambar 4.47 untuk kedua kelas, yaitu mayor dan minor. Persebaran ini dilihat hanya berdasarkan atribut selain jenis perubahan kebutuhan yang bersifat kuantitas. Terlihat kelas mayor dan minor terpisah dengan baik dalam pada atribut jumlah pengujian diulang. Kemunculan term pada kebutuhan, pengujian, dan sumber kode juga tampak memisahkan kedua kelas, meskipun tidak terlihat terdapat data yang beririsan. Kemudian 2 atribut lainnya, yaitu jumlah story kebutuhan dan jumlah tahapan yang tertunda justru tidak terlihat memisahkan kedua kelas *defect*.



Gambar 4.47 Persebaran Data *Defect*

Data kemudian dipisah menjadi 70% training dan 30% testing. Algoritma *decision tree* belajar menggunakan data training untuk menentukan apakah sebuah *defect* termasuk mayor atau minor. Berdasarkan pembelajaran menggunakan data

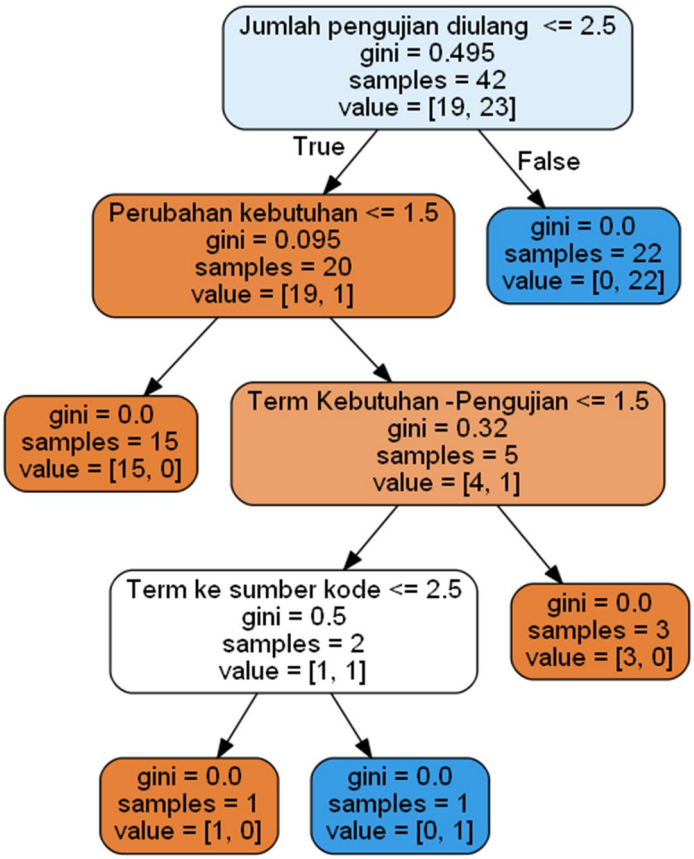
training maka hasil evaluasi prediksi algoritma *decision tree* pada data testing menghasilkan akurasi sebesar 94% . Akurasi yang besar tersebut harus di evaluasi kembali dengan nilai *recall* nya. Nilai *recall* yang dicapai adalah sebesar 80%. Karena hasil yang diharapkan adalah lebih baik terjadi false negatif alias salah memprediksi sebuah minor *defect* sebagai major *defect*, maka yang perlu diperhatikan adalah nilai *recall* agar dapat ditingkatkan. Dikarenakan data kelas mayor dan minor tidaklah seimbang dapat dilihat pada confusion matrix dibawah ini.



Gambar 4.48 Confusion Matrix *Decision tree* Awal

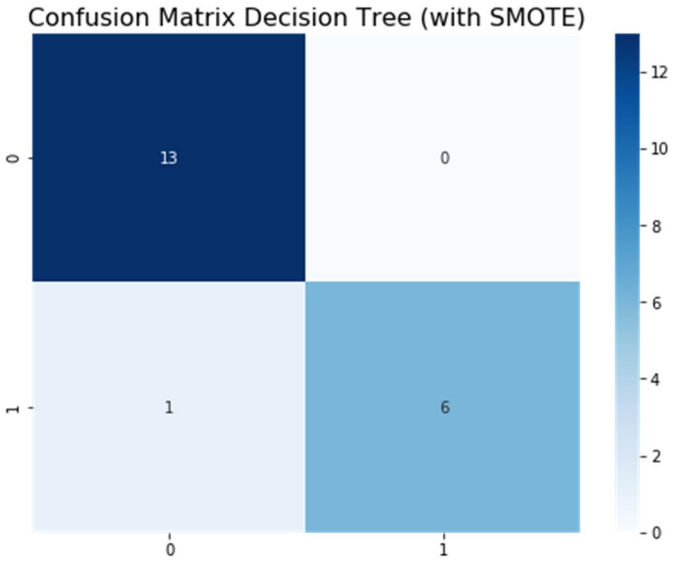
Decision tree yang terbentuk ditampilkan pada gambar Gambar 4.49. Sesuai perkiraan di awal bahwa atribut yang berperan adalah jumlah pengujian diulang, jenis perubahan kebutuhan, term kebutuhan yang muncul pada pengujian, dan term yang muncul pada sumber kode. Jumlah pengujian diulang <=2.5 mengikuti panah true di sebelah kiri dan sisanya mengikuti panah false di sebelah kanan. Nilai gini menunjukkan kualitas pembagian dimana gini = 0,495 yang artinya terdapat perbedaan signifikan dari 42 sampel. Value [19,23] menunjukkan 19 melanjutkan dan 23 merupakan mayor. Kemudian berdasarkan jenis perubahan kebutuhan <=1,5 terdapat 20 pada cabang ini. Karena inisiasi =1 , penambahan = 2, dan perubahan = 3. Maka bila tidak terdapat perubahan dan state kebutuhannya adalah inisiasi seringkali *defect* yang ditemukan adalah minor. Akan tetapi bila

tidak ditemukan term pada pada kebutuhan yang muncul kembali pada pengujian maka *defect* seringkali mayor. Sehingga 15 adalah minor dan 5 berada pada cabang term kebutuhan – pengujian. Bila term pada pengujian dan kebutuhan ditemukan pada sumber kode maka *defect* yang muncul adalah minor. Sehingga 3 adalah mayor dan 2 kemudian berada pada cabang term ke sumber kode. Pada cabang ini ketika dibagi nilai gini yang dihasilkan pada masing-masing sample adalah nol. Sehingga tidak terdapat lagi perbedaan.

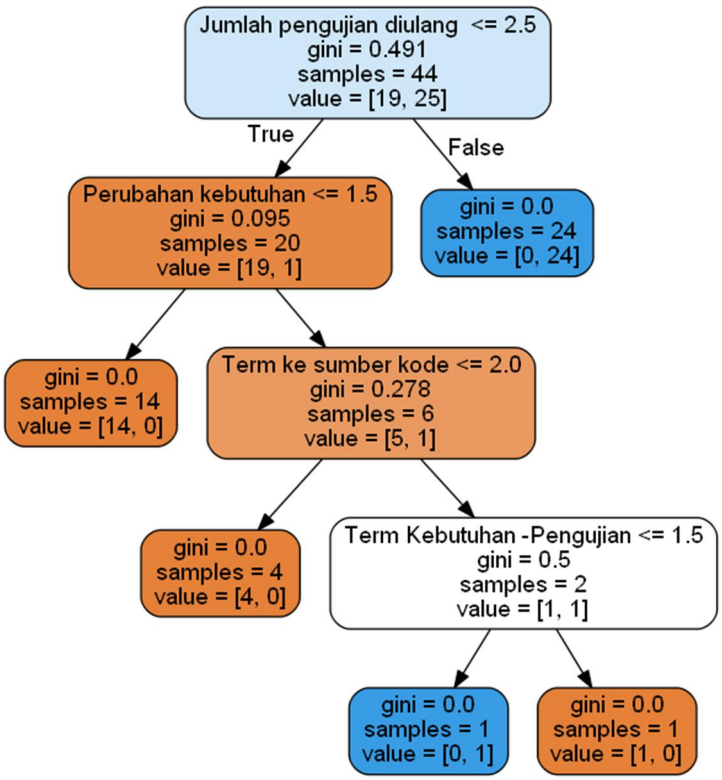


Gambar 4.49 Pohon Keputusan Klasifikasi *Defect*

Untuk menyeimbangkan data maka diberlakukan SMOTE, SMOTE tidak memberikan dampak yang signifikan dan hanya meningkatkan sedikit nilai akurasi dan *recall*. Nilai akurasi naik menjadi 95% dan nilai *recall* naik menjadi 86%. Confusion matrix berubah menjadi seperti pada Gambar 4.51 . Sementara itu *decision tree* juga mengalami perubahan dimana atribut perubahan kebutuhan muncul menjadi cabang dimana sebelumnya atribut ini tidak diperhitungkan.



Gambar 4.50 Confusion Matrix *Decision tree* dengan SMOTE



Gambar 4.51 Pohon Keputusan Klasifikasi *Defect* Akhir

Klasifikasi *Defect* menggunakan *Decision tree* menghasilkan hasil yang sangat baik dengan 5 atribut yang telah disebutkan sebelumnya. Meskipun cabang dari pohon keputusan yang dihasilkan berubah-ubah, tapi nilai akurasi dan *recall*

nya sangat tinggi. *Defect* major cenderung terjadi ketika kebutuhan mengalami perubahan dan tidak konsisten penyebutan istilah nya pada setiap dokumen. Jumlah kelas yang tidak seimbang tidak terlalu berpengaruh pada kesalahan klasifikasi terbukti dari nilai *False* pada confusion matrix nya. Penerapan SMOTE juga tidak berpengaruh karena sedari awal nilai akurasinya sudah sangat besar.

Hasil pohon keputusan yang ditampilkan adalah hasil generate yang paling sering muncul setelah running model *decision tree* sebanyak 10 kali. Nilai akurasi yang tinggi ini perlu dibandingkan kembali dengan dataset yang berasal dari pengembangan aplikasi dengan karakteristik yang berbeda dari dua aplikasi yang digunakan saat ini.

(Halaman ini sengaja dikosongkan)

BAB 5
PENUTUP

Pada bab ini diuraikan beberapa hal yang dapat disimpulkan dari hasil pengujian dan beberapa saran dengan harapan untuk lebih menyempurnakan perancangan yang telah dibuat.

5.1 Kesimpulan

Identifikasi kebutuhan pada proses pengembangan XP dilihat dari awal penjelasan kebutuhan pada *story point* yang terjadi saat *business analyst* dan perwakilan pengguna bertemu. Kebutuhan yang pada saat itu bersifat umum akan dijelaskan kembali kedalam tahapan teknis oleh *system analyst* dan *programmer*. Penjelasan ini menjadi sangat penting, terutama tingkat ketergantungan dan status urgensi penerapannya pada aplikasi.

Setiap tahapan pada proses pengujian merupakan proses penting untuk mendeteksi *bug* selama periode pengembangan. Hal ini merupakan upaya untuk meminimalkan munculnya *defect* ketika produk dirilis dalam versi beta. Oleh karena itu, pendokumentasian tahapan pengujian dan pembuatan *trace link* mulai dari kebutuhan, pengujian, hingga ke sumber kode menjadi sangat penting.

Variabel yang memiliki keterkaitan dengan kemunculan *defect* adalah banyaknya perubahan dan penambahan kebutuhan, tingkat kerumitan aplikasi dilihat dari banyaknya jumlah pengujian integrasi yang terjadi, dan juga berapa banyak cabang tahapan penting dalam aplikasi yang harus dilakukan pengujian regresi untuk melihat tahapan yang tertunda atau terpengaruh karenanya.

Penelusuran terhadap aplikasi dapat dilakukan dengan cara manual menggunakan RTM mulai dari kebutuhan, pengujian, hingga ke sumber kode. Akan tetapi, proses manual menggunakan RTM akan sangat memakan waktu, maka pemodelan topik dapat sedikit membantu untuk mempercepat proses tersebut. Metode pemodelan topik yang diuji pada dua dataset aplikasi berbeda menunjukkan metode NMF bekerja paling stabil. Meskipun, pemodelan topik menggunakan

NMF hanya dapat menyentuh akurasi sekitar 60% pada aplikasi pertama dan 70% pada aplikasi kedua. Berdasarkan nilai akurasi, *precision* dan *recall* pada kedua aplikasi terdapat kelas yang sangat baik hingga mencapai sekitar 80% ada yang hanya menyentuh sekitar 60%.

Kata-kata yang mewakili modul-modul tertentu melalui pemodelan topik NMF dilihat kemunculannya pada setiap tahapan untuk dilihat konsistensi penerapan kebutuhan pada pengujian hingga sumber kode. Hal-hal teknis perlu dibahas secara jelas pada setiap tahapan pengembangan bahkan di awal pembuatan kebutuhan. Sementara itu, dalam dokumen pengujian, istilah tersebut perlu muncul kembali dan tidak diganti. Komentar programmer pada kode sumber diperlukan untuk memudahkan menemukan hubungan pada kesamaan istilah selain memberikan nama variabel yang telah tercantum pada dua dokumen sebelumnya (kebutuhan dan pengujian).

Klasifikasi *defect* menggunakan metode *decision tree* dengan data *bug* dan *defect* dari kedua aplikasi yang dijadikan satu. *Trace link* yang digunakan bukan berasal dari pemodelan topik melainkan yang berasal dari RTM yang dibuat secara manual. Hasil dari pemodelan topik hanya akan menjadi nilai atribut konsistensi kemunculan istilah dari kebutuhan pada pengujian dan konsistensi kemunculan istilah dari pengujian ke sumber kode. Atribut yang lain akan dimasukkan adalah banyaknya *story point* atau kebutuhan yang terkait, jumlah tahapan tertunda karena *defect* atau *bug* tersebut, fungsi terakhir yang gagal berasal dari perubahan atau penambahan kebutuhan, dan berapa banyak pengujian diulang hingga *defect* tertangani. Kelas *defect* hanya terbatas pada dampaknya yaitu *minor* dan *major*.

Hasil dari klasifikasi *defect* memberikan hasil akurasi yang mendekati sempurna, dengan melibatkan empat cabang penting yaitu jumlah pengujian diulang hingga *defect* teratasi, berapa banyak perubahan kebutuhan yang terjadi pada proses tersebut, konsistensi istilah yang di dapatkan dari hasil pemodelan topik dari kebutuhan ke pengujian, dan juga dari pengujian ke sumber kode. Semakin besar jumlah pengulangan pengujian, perubahan terhadap kebutuhan maka semakin besar pula kemungkinan *defect* major muncul. Begitupula dengan semakin

tidak konsistennya istilah yang dipakai pada saat menyatakan kebutuhan, melakukan pengujian, dan pencantumannya dalam sumber kode maka membuat kemungkinan *defect* yang tercipta adalah major.

5.2 Saran

Dalam penelitian ini artifak yang dilibatkan belumlah terlalu mendetail terutama pada fase desain dimana diagram Unified Modeling Language (UML) dan tampilan desain antar muka. Artifak ini mungkin dapat memberikan gambaran penelusuran awal yang lebih baik bila disertakan.

Kemudian pada saat membuat penelusuran semi otomatis dengan pemodelan topik ditemukan beberapa kelemahan, antara lain tidak memperhatikan struktur kalimat pada data yang ada, tidak memberikan sinonim, dan bergantung pada penilaian ahli. Kedepannya, data tersebut akan diolah oleh struktur kalimat dan sinonim kata sehingga kata-kata yang diperiksa kesamaannya memiliki arti yang sama. Selain itu, kesepakatan dari expert judgement yang nantinya akan dibandingkan sebagai *ground truth* harus diukur tidak hanya berdasarkan cohen kappa tetapi juga menggunakan alat ukur lain seperti menghitung “Area Under the Curve” (AUC) dari “Karakteristik Penerima Operator" (ROC) untuk menekankan seberapa baik pemisahan setiap grup.

Ketika pemodelan topik tersusun dengan baik, maka atribut pada proses klasifikasi *defect* akan berisi nilai yang lebih baik dan mampu melakukan identifikasi *defect major* dan *minor* dengan lebih baik lagi

DAFTAR PUSTAKA

A Amalia et al (2019) 'Automated Bahasa Indonesia essay evaluation with latent semantic analysis Automated Bahasa Indonesia essay evaluation with latent semantic analysis', *Journal of Physics: Conference Series 1235 012100*, pp. 0–8. doi: 10,1088/1742-6596/1235/1/012100,

Agarwal, R., Patyal, A. and Nindrajog, A. (2015) '*Defect* Prediction in Agile Software Development', *International Journal of Scientific & Engineering Research*, 6(2), pp. 75–77.

Ahmadiyah, A. S., Sarno, R. and Revindasari, F. (2020) 'Adopted topic modeling for business process and software component conformity checking', *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(6), pp. 2939–2947. doi: 10,12928/TELKOMNIKA.v18i6.13381.

Albalawi, R., Yeap, T. H. and Benyoucef, M. (2020) 'Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis', *Frontiers in Artificial Intelligence*, 3(July), pp. 1–14. doi: 10,3389/frai.2020,00042.

Bagli, E. and Visani, G. (2020) 'Metrics for Multi-Class Classification : an Overview', *arXiv*, abs/2008.0, pp. 1–17.

Belford, M., Mac Namee, B. and Greene, D. (2018) 'Stability of topic modeling via matrix factorization', *Expert Systems with Applications*. Elsevier Ltd, 91, pp. 159–169. doi: 10,1016/j.eswa.2017.08.047.

Braun, D. and Langen, M. (2017) 'Evaluating Natural Language Understanding Services for Conversational Question Answering Systems', *Proceedings of the 18th Annual {SIG}dial Meeting on Discourse and Dialogue*, (August), pp. 174–185.

Chen, Y. *et al.* (2019) 'Experimental explorations on short text topic mining between LDA and NMF based Schemes', *Knowledge-Based Systems*. Elsevier B.V., 163, pp. 1–13. doi: 10,1016/j.knosys.2018.08.011.

Curiskis, S. A. *et al.* (2020) 'An evaluation of document clustering and topic modelling in two online social networks : Twitter and Reddit', *Information Processing and Management*. Elsevier, 57(2), p. 102034. doi: 10,1016/j.ipm.2019.04.002.

Dalianis, H. and Dalianis, H. (2018) 'Evaluation Metrics and Evaluation', *Clinical Text Mining*, (1967), pp. 45–53. doi: 10,1007/978-3-319-78503-5_6.

Elamin, R. and Osman, R. (2018) 'Implementing traceability repositories as graph databases for software quality improvement', *Proceedings - 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security, QRS 2018*. IEEE, (1), pp. 269–276. doi: 10,1109/QRS.2018.00040,

Furtado, F. and Zisman, A. (2016) 'Trace++: A Traceability Approach to Support Transitioning to Agile Software Engineering', *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference, RE 2016*. IEEE, pp. 66–75. doi: 10,1109/RE.2016.47.

Ghiasi, M. M., Zendehboudi, S. and Mohsenipour, A. A. (2020) '*Decision tree*-based diagnosis of coronary artery disease: CART model', *Computer Methods and Programs in Biomedicine*. Elsevier B.V., 192, p. 105400, doi: 10,1016/j.cmpb.2020,105400,

Guo, A. and Yang, T. (2016) 'Research and improvement of feature words weight based on TFIDF algorithm', *Proceedings of 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2016*. IEEE, pp. 415–419. doi: 10,1109/ITNEC.2016.7560393.

Hien, T. D. *et al.* (2015) 'Novel algorithm for non-negative matrix factorization', *New Mathematics and Natural Computation*, 11(2), pp. 121–133. doi: 10,1142/S1793005715400013.

Janković, R. (2019) 'Classifying cultural heritage images by using *decision tree* classifiers in WEKA', *CEUR Workshop Proceedings*, 2320, pp. 119–127.

Jeong, S., Cho, H. and Lee, S. (2018) 'Agile requirement traceability

matrix', *Proceedings - International Conference on Software Engineering*, pp. 187–188. doi: 10,1145/3183440,3195089.

Jinzenji, K. *et al.* (2013) 'An experience report for software quality evaluation in highly iterative development methodology using traditional metrics', *2013 IEEE 24th International Symposium on Software Reliability Engineering, ISSRE 2013*. IEEE, pp. 310–319. doi: 10,1109/ISSRE.2013.6698884.

Kanmani, K. and Murugan, D. A. (2020) 'Prognosticate and diagnosis of diabetes using data preprocessing and null value removal on the modified data set with possible outcome of a *decision tree* construction through R- programming', *Materials Today: Proceedings*. Elsevier Ltd, (xxxx). doi: 10,1016/j.matpr.2020,10,631.

Kumar, S. and Solanki, K. (2015) 'Research Paper on Analysis of Correlation between Prerelease and Post release *Defect*s', 1(2), pp. 45–49.

Lossio-Ventura, J. A. *et al.* (2021) 'Evaluation of clustering and topic modeling methods over health-related tweets and emails', *Artificial Intelligence in Medicine*. Elsevier B.V., 117(March), p. 102096. doi: 10,1016/j.artmed.2021.102096.

Murtazina, M. S. and Avdeenko, T. V. (2019) 'An ontology-based approach to support for requirements traceability in agile development', *Procedia Computer Science*. Elsevier B.V., 150, pp. 628–635. doi: 10,1016/j.procs.2019.02.044.

Nanang, D., Penelusuran, P. L. and Penelusuran, P. L. (2018) 'PEMBANGUNAN LINK PENELUSURAN KEBUTUHAN FUNGSIONAL DAN METHOD PADA KODE SUMBER DENGAN METODE PENGAMBILAN INFORMASI', *ELTEK*, 16, pp. 151–165. Available at: https://doi.org/10,33795/eltek.v16i2.106.

Ngah, A., Munro, M. and Abdallah, M. (2017) 'An overview of regression testing', *Journal of Telecommunication, Electronic and Computer Engineering*, 9(3-5 Special Issue), pp. 45–49. doi: 10,1145/308769.308790,

Noor, R. and Fahad Khan, M. (2014) '*Defect* Management in Agile Software Development', *International Journal of Modern Education and Computer Science*, 6(3), pp. 55–60, doi: 10,5815/ijmecs.2014.03.07.

Obilor, E. I. and Amadi, E. C. (2018) 'Test for Significance of Pearson ' s Correlation Coefficient ( )', (March).

Paper, C. (2016) 'Preprocessing Techniques for Text Mining Preprocessing Techniques for Text Mining', 5(October 2014), pp. 7–16.

Prihatini, P. M. *et al.* (2017) 'Indonesian text feature extraction using gibbs sampling and mean variational inference latent dirichlet allocation', *QiR 2017 - 2017 15th International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering*, 2017-Decem, pp. 40–44. doi: 10,1109/QIR.2017.8168448.

Putra, I. P. E. S., Yuliawati, A. and Mursanto, P. (2012) 'Industrial extreme programming practice's implementation in rational unified process on agile development theme', *2012 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2012 - Proceedings*. IEEE, pp. 137–142.

Putro, H. P. and Wibowo, A. F. (2018) 'Software verification and validation on object oriented software development using traceability matrix', *Proceedings of the 3rd International Conference on Informatics and Computing, ICIC 2018*. IEEE, pp. 1–5. doi: 10,1109/IAC.2018.8780518.

Rani, S. and Kumar, M. (2021) 'Topic modeling and its applications in materials science and engineering', *Materials Today: Proceedings*. Elsevier Ltd., 45, pp. 5591–5596. doi: 10,1016/j.matpr.2021.02.313.

Rath, M., Lo, D. and Mäder, P. (2018) 'Analyzing requirements and traceability information to improve *bug* localization', *Proceedings - International Conference on Software Engineering*, pp. 442–453. doi: 10,1145/3196398.3196415.

Rempel, P. and Mader, P. (2017) 'Preventing *defect*s: The impact of

requirements traceability completeness on software quality', *IEEE Transactions on Software Engineering*, 43(8), pp. 777–797. doi: 10,1109/TSE.2016.2622264.

Sadath, L., Karim, K. and Gill, S. (2018) 'Extreme programming implementation in academia for software engineering sustainability', *2018 Advances in Science and Engineering Technology International Conferences, ASET 2018*. IEEE, pp. 1–6. doi: 10,1109/ICASET.2018.8376925.

Sharma, H. and Kumar, S. (2016) 'A Survey on *Decision tree* Algorithms of Classification in Data Mining', *International Journal of Science and Research (IJSR)*, 5(4), pp. 2094–2097. doi: 10,21275/v5i4.nov162954.

Suhartoyo, H. and Rochimah, S. (2016) 'Membangun Hubungan Kerunutan Artifak Pada Lingkungan Pengembangan Cepat', *SYSTEMIC*, 02(01), pp. 1–17.

Suleman, R. M. and Korkontzelos, I. (2021) 'Extending latent semantic analysis to manage its syntactic blindness', *Expert Systems with Applications*. Elsevier Ltd, 165(January 2020), p. 114130, doi: 10,1016/j.eswa.2020,114130,

Suri, P. and Roy, N. R. (2017) 'Comparison between LDA & NMF for event-detection from large text stream data', *3rd IEEE International Conference on* . IEEE, pp. 1–5. doi: 10,1109/CIACT.2017.7977281.

Tarhini, A., Ismail, Z. and Mansour, N. (2008) 'Regression testing web applications', *Proceedings - 2008 International Conference on Advanced Computer Theory and Engineering, ICACTE 2008*. IEEE, pp. 902–906. doi: 10,1109/ICACTE.2008.82.

Vale, T. *et al.* (2017) 'Software product lines traceability: A systematic mapping study', *Information and Software Technology*. Elsevier B.V., 84, pp. 1–18. doi: 10,1016/j.infsof.2016.12.004.

Vanbelle, S. (2017) 'Comparing dependent kappa coefficients obtained on multilevel data', *Biometrical Journal*, 59(5), pp. 1016–1034. doi: 10,1002/bimj.201600093.

Vanfretti, L. and Arava, V. S. N. (2020) '*Decision tree*-based classification of multiple operating conditions for power system voltage stability assessment', *International Journal of Electrical Power and Energy Systems*. Elsevier, 123(June), p. 106251. doi: 10,1016/j.ijepes.2020,106251.

Wang, B. *et al.* (2018) 'Requirements traceability technologies and technology transfer decision support: A systematic review', *Journal of Systems and Software*. Elsevier Inc., 146, pp. 59–79. doi: 10,1016/j.jss.2018.09.001.

Yao, Z. and Ze-Wen, C. (2011) 'Research on the construction and filter method of stop-word list in text preprocessing', *Proceedings - 4th International Conference on Intelligent Computation Technology and Automation, ICICTA 2011*. IEEE, 1, pp. 217–221. doi: 10,1109/ICICTA.2011.64.

Zhao, J. *et al.* (2019) 'Using topic modeling via non-negative matrix factorization to identify relationships between genetic variants and disease phenotypes: A case study of Lipoprotein(a) (LPA)', *PLoS ONE*, 14(2), pp. 1–15. doi: 10,1371/journal.pone.0212112.

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Nuraisa Novia Hidayati, lahir di Cilacap, 23 November 1989. Anak pertama dari dua bersaudara. Penulis menempuh pendidikan formal di SDN Al-Khairat Ternate, SMPN 2 Cilacap, SMAN 1 Gedangan, dan melanjutkan pendidikan S1 di Jurusan Sistem Informasi, Institut Teknologi Sepuluh Nopember (ITS). Pada Tahun 2014, penulis menjadi perekayasa pada Badan Pengkajian dan Penerapan Teknologi (BPPT) bertugas sebagai pengembang perangkat lunak untuk kebutuhan internal hingga sekarang. Pada tahun 2019 penulis melakukan tugas belajar pendidikan S2 di jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS). Penulis dapat dihubungi melalui email : nunohida@gmail.com