

**IMPLEMENTASI TEXT MINING PADA SOSIAL MEDIA
TWITTER ATAU X UNTUK IDENTIFIKASI TRENDING
TOPIC PADA TWEET BAHASA INDONESIA TERKAIT
PEMILU 2024 MENGGUNAKAN METODE LINK-ANOMALY**

TUGAS AKHIR



Oleh:

**MUHAMMAD ERI SETYAWAN
NIM: 2011501778**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

JAKARTA

2024

**IMPLEMENTASI TEXT MINING PADA SOSIAL MEDIA
TWITTER ATAU X UNTUK IDENTIFIKASI TRENDING
TOPIC PADA TWEET BAHASA INDONESIA TERKAIT
PEMILU 2024 MENGGUNAKAN METODE LINK-ANOMALY**

**Diajukan untuk memenuhi salah satu persyaratan memperoleh
gelar Sarjana Komputer (S. Kom)**

TUGAS AKHIR



Oleh:

**MUHAMMAD ERI SETYAWAN
NIM: 2011501778**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

JAKARTA

2024

LEMBAR PENGESAHAN



**PROGRAM STUDI TEKNIK
INFORMATIKA FAKULTAS
TEKNOLOGI INFORMASI UNIVERSITAS
BUDI LUHUR**

PERSETUJUAN TUGAS AKHIR

Nama : Muhammad Eri Setyawan
Nomor Induk Mahasiswa : 2011501778
Program Studi : Teknik Informatika
Bidang Peminatan : Artificial Intelligence
Jenjang Studi : Strata 1
Judul : IMPLEMENTASI TEXT MINING PADA
SOSIAL MEDIA TWITTER ATAU X UNTUK
IDENTIFIKASI TRENDING TOPIC PADA
TWEET BAHASA INDONESIA TERKAIT
PEMILU 2024 MENGGUNAKAN METODE
LINK-ANOMALY

Disetujui untuk dipertahankan dalam sidang Tugas Akhir periode semester Genap
tahun ajaran 2023/2024.

Jakarta, 15 Juli 2024

Dosen Pembimbing

(Dr. Indra, S.Kom., M.T.I.)

ABSTRAK

IMPLEMENTASI TEXT MINING PADA SOSIAL MEDIA TWITTER ATAU X UNTUK IDENTIFIKASI TRENDING TOPIC PADA TWEET BAHASA INDONESIA TERKAIT PEMILU 2024 MENGGUNAKAN METODE LINK-ANOMALY

Oleh : Muhammad Eri Setyawan (2011501778)

Pemilu (pemilihan umum) adalah salah satu proses demokrasi yang dilakukan setiap empat tahun sekali untuk memilih presiden dan wakil presiden. Di era digital saat ini, media sosial telah menjadi sumber informasi utama dalam menyajikan berita terkini yang berkaitan dengan pemilu. Terutama pada platform X, volume informasi yang besar dan cepat seringkali menghasilkan data yang kompleks, sehingga menjadi tantangan untuk mengidentifikasi dan memvalidasi topik yang benar-benar sedang tren. Sebagian besar jurnalis media online di Indonesia memanfaatkan media sosial sebagai sumber informasi berita. Seringkali, diskusi hangat dan relevan muncul dari interaksi antar pengguna. Oleh karena itu, dikembangkan sistem pendeteksi *trending topic* berdasarkan interaksi antar pengguna di media sosial yaitu platform X. Algoritma *link anomaly* mendeteksi penyimpangan dari perilaku mention normal yang ditangkap oleh model probabilitas mention, dan penyimpangan ini dapat digunakan untuk mendeteksi munculnya topik baru di aliran media sosial. Dalam penelitian ini, pendeteksian trending topik dilakukan menggunakan algoritma *link anomaly* dengan data sebanyak 18,012 *tweet* yang dikumpulkan dari tanggal 25 Maret 2024 sampai 14 April 2024 dengan kata kunci “pemilu”. Algoritma LDA kemudian digunakan sebagai topik modelling di dalam sistem trending topik yang dibuat. Hasil pengujian menunjukkan hasil terbaik pada minggu pertama yaitu *topic recall* sebesar 40%, *keyword precision* sebesar 46%, dan *keyword recall* sebesar 56%. Hasil pengujian tersebut membuktikan bahwa algoritma *link anomaly* mampu mendeteksi *trending topic*.

Kata Kunci: pemilu, *trending topic*, *link anomaly*, LDA, twitter.

PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Eri Setyawan
NIM : 2011501778
Program Studi : Teknik Informatika
Bidang Peminatan : Artificial Intelligence
Jenjang Studi : Strata I
Fakultas : Teknologi Informasi

Menyatakan bahwa TUGAS AKHIR yang berjudul :

Implementasi Text Mining Pada Sosial Media Twitter

Atau X Untuk Identifikasi Trending Topic Pada Tweet

Bahasa Indonesia Terkait Pemilu 2024 Menggunakan Metode Link Anomaly

Merupakan :

1. Karya tulis saya sebagai laporan tugas akhir yang asli dan belum pernah diajukan untuk mendapatkan gelar akademik apapun, baik di Universitas Budi Luhur maupun di perguruan tinggi lainnya.
2. Karya tulis ini bukan saduran / terjemahan, dan murni gagasan, rumusan dan pelaksanaan penelitian / implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan pembimbing di organisasi tempat riset.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Saya menyerahkan hak milik atas karya tulis ini kepada Universitas Budi Luhur, dan oleh karenanya Universitas Budi Luhur berhak melakukan pengelolaan atas karya tulis ini sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh berdasarkan karya tulis ini, serta sanksi lainnya sesuai dengan norma di Universitas Budi Luhur dan Undang-Undang yang berlaku.

Jakarta, 13 Juli 2024



Muhammad Eri Setyawan

KATA PENGANTAR

Dengan memanjatkan puji dan syukur atas kehadiran Allah SWT yang telah melimpahkan nikmat, rahmat serta karunia-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “IMPLEMENTASI TEXT MINING PADA SOSIAL MEDIA TWITTER ATAU X UNTUK IDENTIFIKASI TRENDING TOPIC PADA TWEET BAHASA INDONESIA TERKAIT PEMILU 2024 MENGGUNAKAN METODE LINK-ANOMALY”.

Pada kesempatan ini izinkan penulis untuk menyampaikan terima kasih kepada semua pihak yang mendukung dan membimbing dalam menyelesaikan laporan, terutama kepada:

1. Allah SWT yang Maha Pengasih lagi Maha Penyayang.
2. Segenap keluarga, khususnya orang tua, serta kakak, dan adik yang telah memberikan dukungan baik berupa doa, moral maupun material.
3. Bapak Dr. Agus Setyo Budi, M.Sc, selaku Rektor Universitas Budi Luhur.
4. Bapak Dr. Ir. Achmad Solichin, S.Kom., M.T.I, selaku Dekan Fakultas Teknologi Informatika Universitas Budi Luhur.
5. Bapak Dr. Indra S.Kom, M.T.I, selaku Ketua Program Studi Teknik Informatika Universitas Budi Luhur.
6. Bapak Dr. Indra S.Kom, M.T.I, Selaku Dosen Pembimbing yang sudah membantu dan membimbing dalam penulisan penelitian ini.
7. Seluruh Dosen Universitas Budi Luhur Jakarta, khususnya Dosen Teknik Informatika yang telah memberikan ilmu dan bimbingan yang bermanfaat.
8. Sahabat yang selalu ada menemani serta membantu dalam menempuh pendidikan di Universitas Budi Luhur.
9. Semua teman-teman yang sudah membantu membagikan ilmunya serta memberikan bantuan mental dalam menempuh pendidikan di Universitas Budi Luhur.

Dalam penyusunan skripsi ini, disadari masih banyak kekurangan, baik dalam penyajian penulisan maupun keterbatasan ilmu pengetahuan yang dimiliki. Oleh karena itu, sangat diharapkan kritik dan saran yang bersifat membangun dari semua pihak. Harapan terkandung agar penyusunan tugas akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Jakarta, Juli 2024

Penulis

DAFTAR TABEL


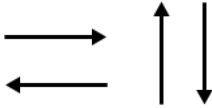



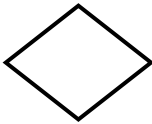
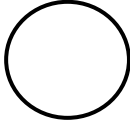
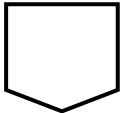
Tabel 2. 1 Taksonomi Penelitian Terdahulu	13
Tabel 3. 1 Data Penelitian.....	15
Tabel 3. 2 Dataset Platform X	24
Tabel 3. 3 Dataset <i>Preprocessing</i>	25
Tabel 3. 4 <i>Slangword</i>	25
Tabel 4. 1 Ilustrasi data <i>twitter</i> yang telah <i>dipreprocessing</i>	34
Tabel 4. 2 Hasil Setelah Pembagian <i>Sequence</i>	35
Tabel 4. 3 Hasil Perhitungan Probabilitas <i>Mention</i>	38
Tabel 4. 4 Hasil Perhitungan Probabilitas <i>User</i>	41
Tabel 4. 5 Hasil Perhitungan Skor <i>Link Anomaly</i>	43
Tabel 4. 6 Perhitungan Agregasi Skor <i>Link Anomaly</i>	45
Tabel 4. 7 Hasil Seleksi Agregasi Skor <i>Link Anomaly</i>	46
Tabel 4. 8 Hasil Perhitungan <i>Cost Function</i>	47
Tabel 4. 9 Data <i>Tweet</i> yang Berada di Waktu Kemunculan <i>Trending Topic</i>	48
Tabel 4. 10 Hasil Setelah Tokenisasi.....	49
Tabel 4. 11 Kamus Kata	49
Tabel 4. 12 Hasil Inisialisasi Topik Pada Kata.....	50
Tabel 4. 13 Hasil Perhitungan <i>Document Topic Counts</i>	50
Tabel 4. 14 Hasil Perhitungan <i>Topic Word Counts</i>	50
Tabel 4. 15 Hasil Perhitungan <i>Topic Counts</i>	51
Tabel 4. 16 Hasil Pengurangan Kata	52
Tabel 4. 17 Hasil Pembobotan Kata Pemilu	52
Tabel 4. 18 Hasil <i>Weighted Random Sampling</i>	53
Tabel 4. 19 Hasil Penambahan Kata.....	54
Tabel 4. 20 Hasil Topik	54
Tabel 4. 21 Tabel Pengujian	75

DAFTAR GAMBAR

Gambar 3. 1 Penerapan Metode.....	17
Gambar 3. 2 Pengumpulan Data	18
Gambar 3. 3 <i>Preprocessing</i>	19
Gambar 3. 4 <i>Link Anomaly</i>	21
Gambar 3. 5 LDA	22
Gambar 3. 6 <i>Class Diagram</i>	23
Gambar 3. 7 <i>Logical Record Structure</i>	24
Gambar 3. 8 Rancangan Menu	26
Gambar 3. 9 Rancangan Layar <i>Dashboard</i>	26
Gambar 3. 10 Rancangan Layar <i>Import</i>	27
Gambar 3. 11 Rancangan Layar <i>Preprocessing</i>	27
Gambar 3. 12 Rancangan Layar <i>Link Anomaly</i>	28
Gambar 3. 13 Rancangan Layar <i>Modelling</i> Topik	28
Gambar 4. 1 <i>Deployment Diagram</i>	29
Gambar 4. 2 Proses Konversi Waktu.....	31
Gambar 4. 3 Proses <i>Case Folding</i>	31
Gambar 4. 4 Proses Penghapusan URL	32
Gambar 4. 5 Proses Hapus <i>Mention</i>	32
Gambar 4. 6 Proses Hapus <i>Hashtag</i>	32
Gambar 4. 7 Proses Hapus Selain Huruf	33
Gambar 4. 8 Proses Hapus Spasi Berlebih	33
Gambar 4. 9 Proses Ubah <i>Slang Word</i>	33
Gambar 4. 10 Proses Hapus <i>Stop Word</i>	34
Gambar 4. 11 Proses <i>Stemming</i>	34
Gambar 4. 12 <i>Flowchart</i> Proses Keseluruhan Sistem	55
Gambar 4. 13 <i>Flowchart</i> Menu <i>Dashboard</i>	55
Gambar 4. 14 <i>Flowchart</i> Proses <i>Import</i> Data	56
Gambar 4. 15 <i>Flowchart</i> Proses <i>Preprocessing</i>	56
Gambar 4. 16 <i>Flowchart</i> Proses Seleksi Atribut	57
Gambar 4. 17 <i>Flowchart</i> Proses Konversi Waktu	57
Gambar 4. 18 <i>Flowchart</i> Proses <i>Case Folding</i>	58
Gambar 4. 19 <i>Flowchart</i> Proses <i>Cleansing</i>	59
Gambar 4. 20 <i>Flowchart</i> Proses Ubah <i>Slang Word</i>	59
Gambar 4. 21 <i>Flowchart</i> Proses Hapus <i>Stop Word</i>	60
Gambar 4. 22 <i>Flowchart</i> Proses <i>Stemming</i>	61
Gambar 4. 23 <i>Flowchart</i> Proses <i>Link Anomaly</i>	61
Gambar 4. 24 <i>Flowchart</i> Proses Hitung Probabilitas Sejumlah <i>Mention</i>	62
Gambar 4. 25 <i>Flowchart</i> Hitung Probabilitas <i>User</i>	63
Gambar 4. 26 <i>Flowchart</i> Hitung Skor <i>Link Anomaly</i>	63
Gambar 4. 27 <i>Flowchart</i> Agregasi Skor <i>Link Anomaly</i>	64
Gambar 4. 28 <i>Flowchart</i> Hitung <i>Cost Function</i>	65

Gambar 4. 29 <i>Flowchart</i> Menentukan Nilai <i>Cost Function</i> Minimum	66
Gambar 4. 30 <i>Flowchart</i> Proses LDA	66
Gambar 4. 32 Grafik Pengujian	78
Gambar 4. 33 Tampilan Layar <i>Dashboard</i>	79
Gambar 4. 34 Tampilan Layar <i>Import Data</i>	79
Gambar 4. 35 Tampilan Layar <i>Preprocessing</i>	80
Gambar 4. 36 Tampilan Layar <i>Link Anomaly</i>	80
Gambar 4. 37 Tampilan Layar <i>Modelling</i> Topik	81

DAFTAR SIMBOL

SIMBOL	NAMA	KETERANGAN
	<i>Terminator</i>	Simbol yang menyatakan awal atau akhir suatu program
	<i>Flow</i>	Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan <i>Connecting Line</i>
	<i>Input/Output</i>	Simbol yang menyatakan proses input atau output tanpa tergantung peralatan
	<i>Process</i>	Simbol yang menyatakan suatu proses yang dilakukan komputer
	<i>Predefine Process</i>	Simbol untuk pelaksanaan suatu bagian (sub-program) atau <i>procedure</i>
	<i>Decision</i>	Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak
	<i>On-Page Reference</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang sama
	<i>Off-Page Reference</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda

DAFTAR ISI

LEMBAR PENGESAHAN	iii
ABSTRAK.....	iv
PERNYATAAN TIDAK PLAGIAT	v
KATA PENGANTAR	vi
DAFTAR TABEL.....	vii
DAFTAR GAMBAR.....	viii
DAFTAR SIMBOL	x
DAFTAR ISI.....	xi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Sistematika Penulisan	4
BAB II LANDASAN TEORI.....	6
2.1. <i>Text Mining</i>	6
2.2. Platform X.....	6
2.3. Pengumpulan Data.....	6
2.4. <i>Preprocessing</i>	7
2.4.1. Seleksi Atribut	7
2.4.2. Konversi Waktu	7
2.4.3. <i>Case Folding</i>	7
2.4.4. <i>Cleansing</i>	7
2.4.5. Mengubah <i>Slang Word</i>	7
2.4.6. Menghapus <i>Stop Word</i>	8
2.4.7. <i>Stemming</i>	8
2.5. <i>Link Anomaly</i>	8
2.5.1. Pemodelan Probabilitas Mention	8
2.5.2. Perhitungan Skor <i>Link Anomaly</i>	9

2.5.3. Agregasi Skor <i>Link Anomaly</i>	9
2.5.4. Deteksi <i>Burst</i>	10
2.6. LDA	10
2.7. Pengujian	11
2.7.1. Data <i>Ground Truth</i>	11
2.7.2. Pengukuran <i>Recall</i> dan <i>Precision</i>	12
2.8. Studi Literatur	13
BAB III METODOLOGI PENELITIAN	15
3.1. Data Penelitian.....	15
3.2. Metode Pembandingan	16
3.3. Penerapan Metode.....	16
3.3.1. Pengumpulan Data.....	17
3.3.2. <i>Preprocessing</i>	18
3.3.3. Algoritma <i>Link Anomaly</i>	21
3.3.4. Algoritma LDA.....	22
3.4. Rancangan Pengujian.....	22
3.5. Rancangan Basis Data	23
3.5.1. <i>Class Diagram</i>	23
3.5.2. <i>Logical Record Structure</i>	23
3.5.3. Spesifikasi Basis Data.....	24
3.6. Rancangan Menu	25
3.7. Rancangan Layar	26
3.7.1. Rancangan Layar Menu <i>Dashboard</i>	26
3.7.2. Rancangan Layar Menu <i>Import</i>	26
3.7.3. Rancangan Layar Menu <i>Preprocessing</i>	27
3.7.4. Rancangan Layar Menu <i>Link Anomaly</i>	27
3.7.5. Rancangan Layar Menu <i>Modelling</i> Topik	28
BAB IV HASIL DAN PEMBAHASAN	29
4.1. Lingkungan Percobaan	29
4.1.1. Spesifikasi <i>Software</i>	29
4.1.2. Spesifikasi <i>Hardware</i>	29
4.1.3. <i>Deployment Diagram</i>	29

4.2. Implementasi Metode	30
4.2.1. Tahap Pengumpulan Data	30
4.2.2. Tahap <i>Preprocessing</i>	30
4.2.3. Tahap <i>Link Anomaly</i>	34
4.2.4. Tahap LDA	48
4.3. <i>Flowchart</i>	55
4.3.1. <i>Flowchart</i> Proses Keseluruhan Sistem	55
4.3.1. <i>Flowchart</i> Menu <i>Dashboard</i>	55
4.3.2. <i>Flowchart</i> Proses <i>Import Data</i>	56
4.3.3. <i>Flowchart</i> Proses <i>Preprocessing</i>	56
4.3.4. <i>Flowchart</i> Proses <i>Link Anomaly</i>	61
4.3.5. <i>Flowchart</i> Proses LDA	66
4.4. Algoritma	67
4.4.1. Algoritma Proses Keseluruhan Sistem	67
4.4.1. Algoritma Menu <i>Dashboard</i>	67
4.4.2. Algoritma Proses <i>Import Data</i>	68
4.4.3. Algoritma Proses <i>Preprocessing</i>	68
4.4.4. Algoritma Proses <i>Link Anomaly</i>	71
4.4.5. Algoritma Proses LDA	74
4.5. Pengujian	75
4.6. Tampilan Layar Aplikasi	78
4.6.1. Tampilan Layar <i>Dashboard</i>	79
4.6.2. Tampilan Layar <i>Import Data</i>	79
4.6.3. Tampilan Layar <i>Preprocessing</i>	80
4.6.4. Tampilan Layar <i>Link Anomaly</i>	80
4.6.5. Tampilan Layar <i>Modelling</i> Topik	81
BAB V PENUTUP	82
5.1. Kesimpulan	82
5.2. Saran	82
DAFTAR PUSTAKA	84

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pemilu yang merupakan singkatan dari Pemilihan Umum merupakan salah satu proses demokrasi yang dilakukan setiap 4 tahun sekali untuk memilih presiden dan wakil presiden. Kegiatan ini melibatkan seluruh tatanan masyarakat Indonesia yang memenuhi syarat untuk memberikan suara kepada calon-calon yang dianggap dapat dipercayai dan mampu melaksanakan aspirasi mereka pada jenjang lokal, regional, dan nasional. Pemilu menjadi ajang yang penting dalam sistem demokrasi di Indonesia karena memberikan rakyat hak suaranya langsung untuk memilih pemimpin yang diinginkan. Maka dari itu, kegiatan ini menjadi salah satu bentuk partisipasi politik dasar yang secara langsung dapat mempengaruhi kebijakan negara.

Di era digital ini, media sosial memegang peran penting dalam memberikan informasi terkini terkait pemilu yang sedang berlangsung. Sebagian besar jurnalis media online di Indonesia memanfaatkan media sosial sebagai sumber informasi berita. Sebanyak 9 dari 10 (92,2%), mengandalkan media sosial sebagai sumber utama berita (Rosemarwati dan Lindawati, 2019). Instagram, platform X, dan facebook merupakan contoh dari sosial media yang sangat populer dan sering digunakan oleh masyarakat Indonesia. Sementara media sosial memberikan akses cepat kepada informasi, cepatnya serta besarnya volume informasi yang ditawarkan oleh platform seperti X sering kali membuat masyarakat umum dihadapkan pada tantangan untuk memfilter dan mengidentifikasi informasi yang paling relevan dan sedang tren. Platform-platform ini seringkali mengandung beragam narasi dan opini yang membuat sulit untuk membedakan mana yang merupakan refleksi akurat dari sentimen umum. Banyaknya opini yang tidak terfilter ini juga menyebabkan masyarakat rentan terhadap opini palsu atau informasi yang menyesatkan, yang dapat mempengaruhi persepsi publik dan keputusan pemilih secara negatif.

Berdasarkan pengamatan terhadap aktivitas masyarakat di media sosial, seringkali terlihat bahwa pembahasan-pembahasan hangat dan relevan muncul dari interaksi antar pengguna. Dari interaksi tersebut, munculah apa yang dikenal sebagai trending topik. *Trending Topic* adalah topik yang mengalami lonjakan popularitas di satu atau lebih platform untuk durasi waktu yang terbatas (Alzubi *et al.*, 2020) .

Deteksi trending topik menjadi kunci penting dalam menggali wawasan dan pemahaman lebih dalam terhadap dinamika masyarakat, memungkinkan kita untuk merespons dengan tepat terhadap perubahan-perubahan yang sedang terjadi serta mengambil keputusan yang lebih berdasarkan data dan informasi aktual. Metode deteksi *trending topic* di media sosial umumnya dibagi menjadi tiga, yaitu *textual content*, *social content* dan *hybrid* (Chen *et al.*, 2012). Metode *textual content* adalah

teknik untuk mendeteksi *trending topic* berdasarkan analisis teks atau pesan yang diposting di media sosial. Metode *social content* adalah teknik deteksi yang didasarkan pada interaksi antar pengguna media sosial, seperti penggunaan mention atau retweet. Metode hibrida menggabungkan pendekatan berbasis *textual content* dan *social content* untuk mendeteksi *trending topic*.

Pada penelitian sebelumnya yang dilakukan oleh (Indra *et al.*, 2019), metode *BN-grams* dan *Doc-p* digunakan untuk mendeteksi *trending topic* dari *tweet* berbahasa Indonesia, dengan fokus pada pembandingan kedua metode tersebut dalam konteks akurasi deteksi topik. Hasil eksperimen menunjukkan bahwa *BN-grams* cenderung memiliki performa yang lebih baik dalam hal *topic recall* dibandingkan dengan *Doc-p*, dengan *BN-grams* mencapai *topic recall* sebesar 20% pada dataset P-4, yang lebih tinggi dibandingkan dengan *Doc-p*. Selain itu, penelitian ini juga mengungkapkan bahwa penggunaan agregasi dalam proses analisis cenderung menurunkan kualitas topik yang dihasilkan, menunjukkan bahwa kedua metode masih memiliki kelemahan dalam menghadapi variasi data yang kompleks dan dinamis.

Penelitian lain dilakukan oleh (Ariwibowo dan Indra, 2021), metode *Maximum Capturing Clustering* dan *Jaccard Similarity* digunakan untuk mendeteksi *trending topic* terkait COVID-19 dari *tweet* berbahasa Indonesia. Tujuan utama dari penelitian ini adalah untuk mengidentifikasi topik yang banyak dibicarakan dalam konteks pandemi, menggunakan dataset berisi 500 *tweet*. Hasil pengujian menunjukkan bahwa sistem ini mampu menghasilkan klaster dengan tingkat *topic recall* sebesar 58.3%, *keyword precision* 66.6%, dan *keyword recall* 83.3%. Penelitian ini mengungkapkan bahwa meskipun metode ini efektif dalam membentuk klaster kata yang relevan, masih terdapat ruang untuk peningkatan dalam hal pemrosesan data dan adaptasi terhadap dinamika konten sosial yang cepat berubah.

Kemudian penelitian berikutnya dilakukan oleh (Fajarudin dan Indra, 2023), penelitian tersebut mengimplementasikan metode *Doc-p* untuk deteksi *trending topic* terkait pemilihan presiden di Twitter. Metode ini menggunakan teknik *Locality Sensitive Hashing* (LSH) untuk klasterisasi, cosine similarity untuk mengukur kesamaan antar dokumen, dan TF-IDF untuk menilai pentingnya sebuah *term* dalam dokumen terhadap koleksi dokumen. Tujuan utama dari penelitian ini adalah untuk menemukan *tweet* yang viral tentang para calon presiden. Hasil penelitian menunjukkan bahwa metode *Doc-p* berhasil mengatasi tantangan dalam mendeteksi topik populer secara manual dengan mencapai skor klasterisasi tertinggi sebesar 98.0 untuk klaster 0. Namun, penelitian ini juga mengakui adanya tantangan dalam penanganan volume data yang besar dan kompleksitas dari data yang tidak terstruktur, yang mempengaruhi efektivitas alat dalam beberapa kasus.

Penelitian sebelumnya berfokus pada penggunaan metode berbasis *textual content* untuk mendeteksi *trending topic* dari *tweet* berbahasa Indonesia, dengan mengutamakan analisis teks dalam mengekstrak informasi dan mengidentifikasi topik populer. Menurut (Takahashi *et al.*, 2014) pendekatan berbasis frekuensi kata konvensional tidak sesuai karena informasi yang ditukar dalam postingan jaringan sosial tidak hanya meliputi teks, tetapi juga gambar, *link*, dan video. Keterbatasan ini muncul karena ambiguitas yang disebabkan oleh sinonim dan homonim, serta kebutuhan *preprocessing* yang rumit tergantung pada bahasa, yang menjadikan metode ini kurang efektif untuk media yang berisi banyak informasi *non-textual*. Sebaliknya, penelitian yang diusulkan berfokus pada pendekatan berbasis *social content* melalui analisis pola interaksi antar *user* seperti *mention*. Pendekatan ini lebih sensitif terhadap bagaimana topik tersebar dan dibicarakan dalam jaringan sosial, memberikan perspektif yang lebih luas dan mendalam tentang popularitas sebuah topik.

Oleh karena itu, dikembangkan sistem deteksi trending topik menggunakan metode *link-anomaly*. Sistem ini menganalisis interaksi pengguna di Platform X, termasuk mention dan hubungan antar konten antar pengguna sehingga dapat mendeteksi trending topik yang sedang terjadi. Penelitian ini akan berfokus pada Pemilu yang dilaksanakan pada tahun 2024. Dengan adanya penelitian ini, diharapkan dapat mengidentifikasi secara akurat topik-topik yang sedang menjadi sorotan di kalangan pengguna Platform X serta memberikan wawasan yang berharga dalam memahami tren dan isu-isu yang sedang hangat dibicarakan terkait pemilu.

1.2. Rumusan Masalah

Berdasarkan uraian yang disampaikan pada latar belakang, rumusan masalah dari penelitian ini adalah seberapa efektifkah metode *link-anomaly* dalam mendeteksi trending topik yang sedang terjadi pada saat pemilu tahun 2024 berlangsung?

1.3. Batasan Masalah

Berikut adalah ruang lingkup masalah atau batasan yang diterapkan dalam penelitian ini:

- a) Data yang digunakan diambil dari platform media sosial Platform X atau X yang menggunakan Bahasa Indonesia dengan kata kunci "Pemilu".
- b) Data yang diambil terbatas pada tanggal 25 Maret 2024 hingga 14 April 2024 saja.
- c) Metode yang digunakan adalah metode *link-anomaly* dengan pemodelan topik LDA untuk pengujian.
- d) Data yang digunakan sebanyak 18.012 data, dengan minggu pertama sebanyak 3.896 data, minggu kedua sebanyak 7.480 data, dan minggu ketiga sebanyak 6.634 data.

1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem deteksi trending topik menggunakan algoritma *link-anomaly* pada platform media sosial Platform X atau X, terutama terkait Pemilu tahun 2024, dan juga untuk mengevaluasi seberapa efektif algoritma Link-anomaly dalam mengatasi tantangan tersebut.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Meningkatkan pemahaman publik terhadap isu-isu yang tengah ramai dibicarakan terkait Pemilu 2024.
2. Memberikan dukungan yang lebih baik dalam pengambilan keputusan bagi pihak terkait pemilihan umum maupun masyarakat umum, memberikan landasan data yang kuat untuk keputusan yang lebih tepat dan berdasarkan fakta.
3. Menyediakan data dari hasil deteksi trending topik pemilu 2024 untuk penelitian selanjutnya.

1.6. Sistematika Penulisan

Laporan penelitian ini disusun berdasarkan sistematika yang terstruktur sehingga mudah untuk dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Adapun alur sistematika penulisan laporan penelitian ini sebagai berikut:

BAB I PENDAHULUAN

Bagian ini berisi informasi mengenai latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bagian ini memberikan penjelasan mengenai algoritma dan metode yang akan dibahas dalam penelitian, bersama dengan teori-teori yang terkait dengan topik penelitian seperti konsep text mining, twitter, *crawling*, *preprocessing*, *link anomaly*, LDA, dan Pengujian.

BAB III METODOLOGI PENELITIAN

Bagian ini akan membahas mengenai sumber data penelitian serta penerapan atau tahapan metode yang digunakan dan rancangan pengujian yang akan dilakukan.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini akan membahas mengenai lingkungan percobaan sistem yang dikembangkan, implementasi metode, flowchart tahapan metode, dan penjelasan algoritma yang digunakan dalam proses. Selain itu, akan dilakukan analisis pengujian sistem untuk mengevaluasi sistem yang telah kembangkan.

BAB V PENUTUP

Bagian ini berisi kesimpulan yang diperoleh dari hasil penelitian dan pengujian yang telah dilakukan. Selain itu, bab ini juga berisi saran dan kritik untuk pengembangan lebih lanjut dalam konteks penelitian ini.

BAB II

LANDASAN TEORI

2.1. Text Mining

Text mining adalah proses ekstraksi informasi dari data berbentuk teks yang tidak terstruktur. Proses ini berusaha mengatasi masalah kelebihan informasi dengan menerapkan teknik data mining dan melibatkan tahapan pra-pemrosesan dokumen seperti pemilihan teks dan kategorisasi (Ajjiah *et al.*, 2023).

Tujuan dari *Text Mining* adalah untuk mengekstraksi informasi yang berharga dari sekumpulan dokumen. Oleh karena itu, sumber data yang digunakan dalam penambangan teks biasanya terdiri dari teks yang memiliki format tidak terstruktur atau setidaknya semi-terstruktur. Tugas-tugas spesifik dalam penambangan teks meliputi pengkategorian teks dan pengelompokan teks (Ariyanti dan Iswardani, 2020) .

2.2. Platform X

Platform X adalah situs web yang dimiliki dan dioperasikan oleh X Corp, yang menyediakan platform jejaring sosial yang memungkinkan penggunaannya untuk mengirim dan membaca pesan berbasis teks yang disebut kicauan atau *tweets*. Awalnya, pengguna hanya bisa mengirimkan pesan hingga 140 karakter, namun pada 7 November 2017, batas karakter ini diperluas menjadi 280 karakter. Selain teks, pengguna juga dapat mengirim kicauan yang berisi foto atau video, serta membalas kicauan dari pengguna lain (Rosalina *et al.*, 2020).

Platform X telah menjadi alat komunikasi penting dalam berbagai aspek kehidupan, tidak hanya untuk komunikasi sehari-hari tetapi juga dalam penyebaran informasi penting dan berita. Berkat fleksibilitasnya dalam mengirim pesan pendek dan kemampuannya untuk menyertakan media visual, platform ini memungkinkan pesan seseorang untuk menyebar secara luas dan cepat ke berbagai audiens. Ini memperkuat perannya sebagai saluran komunikasi massa yang efektif dan penting di era digital saat ini (Emeraldien *et al.*, 2019).

2.3. Pengumpulan Data

Pengumpulan data (*crawling*) merupakan metode yang dipakai untuk mengambil informasi dari dalam web secara otomatis. Metode ini mengumpulkan data berdasarkan kata kunci yang ditentukan oleh pengguna. Perangkat yang digunakan dalam proses ini disebut *crawler*, yang merupakan sebuah program yang dirancang dengan algoritma khusus untuk memindai halaman web sesuai dengan alamat atau kata kunci yang dimasukkan oleh pengguna (Zuhri dan Saputri, 2020). Pada penelitian ini proses *crawling* dilakukan menggunakan bahasa pemrograman python dengan pustaka *twscrape* untuk mengambil data dari twitter.

2.4. Preprocessing

Dalam melakukan penambahan teks, dokumen teks perlu dipersiapkan terlebih dahulu agar dapat digunakan dalam proses utama. Proses persiapan teks atau data mentah ini dikenal sebagai *preprocessing*. *Preprocessing* berfungsi untuk mengubah data yang tidak terstruktur menjadi terstruktur (Isnain *et al.*, 2021). Adapun tahapan dari *preprocessing* antara lain: Seleksi Atribut, Konversi Waktu, *Case Folding*, *Cleansing*, mengubah *Slang Word*, menghapus *Stop Word*, dan *Stemming*.

2.4.1. Seleksi Atribut

Seleksi atribut dilakukan untuk meminimalisir ukuran dataset dengan menghapus atribut yang tidak sesuai atau tidak relevan dengan tujuan penelitian. Ini membantu dalam menyederhanakan model dan memastikan bahwa analisis lebih fokus pada data yang memang berpengaruh terhadap hasil penelitian (Putri, 2021).

2.4.2. Konversi Waktu

Untuk mengolah data yang diambil dari Platform X, perlu diperhatikan bahwa data dicatat dalam format waktu UTC.. Karena itu, saat menggunakan data dari Platform X untuk keperluan analisis di Indonesia, konversi waktu dari UTC ke WIB (Waktu Indonesia Barat) menjadi langkah penting. WIB berbeda 7 jam lebih cepat dari UTC. Proses konversi ini penting untuk memastikan bahwa data yang disediakan sesuai dengan waktu lokal di Indonesia.

2.4.3. Case Folding

Case Folding adalah proses penyamaan huruf besar dan kecil dalam sebuah dokumen. Proses ini dilakukan untuk mempermudah pencarian. Karena tidak semua dokumen teks konsisten dalam penggunaan huruf kapital, *case folding* berperan penting dalam mengonversi seluruh teks dalam dokumen ke bentuk standar, yaitu huruf kecil (*lowercase*) (Alita dan Isnain, 2020).

2.4.4. Cleansing

Cleansing adalah proses menghapus setiap karakter dalam sebuah *tweet*, kecuali alfabet. Tujuannya adalah untuk mengurangi karakter yang tidak memiliki makna atau tidak diinginkan. Karakter-karakter tersebut termasuk angka, @, #, tautan dari situs web, dan emoji (Wirawan *et al.*, 2024).

2.4.5. Mengubah Slang Word

Slang Word adalah penggunaan bahasa gaul yang menyebabkan bahasa Indonesia menjadi tidak baku. Proses untuk mengubah kata slang ini disebut *convert word*. Caranya adalah dengan mengganti kata slang menjadi kata yang sesuai dengan ejaan yang disempurnakan (EYD) dalam bahasa Indonesia (Rifki dan Imelda, 2022).

2.4.6. Menghapus *Stop Word*

Di dalam dokumen teks terdapat berbagai jenis kata seperti kata depan, kata sambung, kata ganti, kata sifat, dan lainnya. Sebagian dari kata-kata ini tidak cocok untuk dijadikan indeks dokumen karena kemunculannya yang tidak unik untuk dokumen tertentu. Oleh karena itu, dilakukan penyaringan kata-kata tersebut. Langkah ini merupakan proses pembersihan teks dari kata-kata yang tidak relevan untuk dijadikan indeks, yang disebut sebagai langkah penyaringan kata umum atau *Stop Word Filtering* (Rahutomo dan Ririd, 2019).

2.4.7. *Stemming*

Stemming adalah proses mengembalikan kata berimbuhan ke bentuk dasarnya atau kata akar dengan menggunakan aturan yang telah ditentukan. Caranya adalah dengan menghilangkan awalan, sisipan, akhiran, dan konfiks (kombinasi awalan dan akhiran) dari kata berimbuhan. *Stemming* merupakan bagian penting dalam Pengambilan Informasi (Information Retrieval) untuk pencarian web, pengelompokan dokumen, mengurangi jumlah indeks yang berbeda dalam sebuah dokumen, dan penerjemahan (Mustikasari *et al.*, 2021).

2.5. *Link Anomaly*

Link anomaly adalah penyimpangan dari perilaku *mention* normal yang ditangkap oleh model probabilitas *mention*, dan penyimpangan ini dapat digunakan untuk mendeteksi munculnya topik baru di aliran media sosial. Jika seorang *user* yang biasanya jarang disebut (*mention*) tiba-tiba mendapatkan banyak *mention*, atau jika jumlah *mention* dalam sebuah *post* meningkat drastis, ini dianggap sebagai *link anomaly*. Untuk mendeteksi *link anomaly*, dapat digunakan model probabilistik yang mengidentifikasi perilaku *mention* yang normal dari *user* (Takahashi *et al.*, 2014).

2.5.1. Pemodelan Probabilitas Mention

Model ini memodelkan jumlah *mention* k dalam sebuah *post* sebagai distribusi geometris. Probabilitas $P(k | T)$ dari k *mention* dalam sebuah *post* dihitung sebagai:

$$P(k | T) = \frac{(n + \alpha)}{(m + k + \alpha)} \prod_{j=0}^{k-1} \frac{(m + \alpha + j)}{(n + m + \alpha + \beta + j)} \quad \dots(2.1)$$

n adalah jumlah total *post* dalam set *training* T , m adalah jumlah total *mention*, dan α serta β adalah parameter model. Pada penelitian ini nilai yang digunakan pada parameter $\alpha = \beta = 0,5$. Distribusi ini memungkinkan kita untuk memperkirakan seberapa sering pengguna biasanya menyebut pengguna lain dalam postingannya.

Model ini juga menggunakan CRP (*Chinese Restaurant Process*) untuk memodelkan frekuensi *mention* v yang disebut dalam set *training* T . Probabilitas *mention user* v yang diketahui dalam set *training* T adalah:

$$P(v | T) = \frac{m_v}{m + \gamma} \quad \text{for } v: m_v \geq 1 \quad \dots(2.2)$$

m_v adalah jumlah *mention* ke user v dalam dataset T . Untuk *user* yang tidak pernah disebut dalam set pelatihan, probabilitasnya adalah:

$$P(\{v: m_v = 0\} | T) = \frac{\gamma}{m + \gamma} \quad \dots(2.3)$$

γ adalah parameter yang mengatur probabilitas *mention user* baru. Pada penelitian ini nilai parameter yang digunakan adalah $\gamma = 0,5$. Proses ini memungkinkan kita untuk menangani jumlah *user* yang tidak terbatas yang dapat disebutkan dalam *post*.

2.5.2. Perhitungan Skor *Link Anomaly*

Skor anomali untuk sebuah *post* baru $x=(t,u,k,V)$ oleh *user* u pada waktu t dengan k *mention* ke pengguna V , dihitung sebagai:

$$s(x) = -\log P(k | T_u(t)) \prod_{v \in V} P(v | T_u(t)) \quad \dots(2.4)$$

di mana $T_u(t)$ adalah koleksi *post* oleh *user* u dalam periode waktu $[t-T, t]$. Skor ini mengukur seberapa besar penyimpangan jumlah *mention* dan frekuensi *mention user* dari pola normalnya.

2.5.3. Agregasi Skor *Link Anomaly*

Untuk mengukur tren umum dalam perilaku *user*, agregasi skor *link anomaly* dihitung dengan menggabungkan skor anomali dari berbagai *user* menggunakan *window time* diskret τ sebagai berikut:

$$s'_j = \frac{1}{\tau} \sum_{t_i \in [\tau(j-1), \tau j]} s(x_i) \quad \dots(2.5)$$

di mana x_i adalah sebuah *post* pada waktu t_i oleh *user* u_i yang termasuk k_i *mention* ke *user* V_i . Agregasi ini memungkinkan kita untuk mendeteksi perubahan tren umum dalam perilaku *mention user* dari waktu ke waktu.

2.5.4. Deteksi *Burst*

metode ini juga menggunakan cost function untuk mengukur "cost" dari transisi state dalam model probabilistik burst detection. *Cost function* membantu dalam menentukan seberapa signifikan perubahan dari state non-burst ke burst, atau sebaliknya.

Cost function dalam konteks burst detection dinyatakan sebagai berikut:

$$c(q|x) = b \ln \left(\frac{1-p}{p} \right) + \sum_{t=1}^n -\ln f_{it}(x_t) \quad \dots(2.6)$$

di mana b adalah jumlah transisi state, p adalah probabilitas transisi state dan $f_{it}(x_t)$ adalah fungsi densitas probabilitas dari data observasi x_t .

Cost function ini berasal dari model probabilistik burst detection dan digunakan untuk mengukur *cost* dari transisi *state*. Dengan menggabungkan *cost function* dengan skor anomali yang telah dihitung, kita dapat secara efektif mengidentifikasi topik-topik yang sedang muncul berdasarkan pola penyebutan pengguna di media sosial.

2.6. LDA

LDA adalah model probabilistik generatif untuk suatu korpus. Konsep dasarnya adalah bahwa setiap dokumen direpresentasikan sebagai campuran acak dari topik laten, di mana setiap topik dicirikan oleh distribusi kata-kata. *Latent dirichlet allocation* (LDA), pertama kali diperkenalkan oleh Blei, Ng, dan Jordan pada tahun 2003, adalah salah satu metode paling populer dalam pemodelan topik. LDA menggambarkan topik-topik melalui probabilitas kata. Kata-kata dengan probabilitas tertinggi dalam setiap topik biasanya memberikan gambaran yang baik tentang isi dari topik tersebut (Jelodar *et al.*, 2019). Adapun rumus utama dari LDA adalah sebagai berikut:

$$(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) \quad \dots(2.7)$$

$$= \prod_{i=1}^K p(\beta_i) \prod_{d=1}^D p(\theta_d) \left(\prod_{n=1}^N p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right)$$

Keterangan :

- $\beta_{1:K}$: distribusi kata-kata dalam setiap topik
- $\theta_{1:D}$: distribusi topik dalam setiap dokumen
- $z_{1:D}$: topik-topik yang dipilih untuk setiap kata dalam dokumen
- $w_{1:D}$: kata-kata yang muncul dalam dokumen
- $p(\beta_i)$: Probabilitas distribusi kata dalam topik (i)
- $p(\theta_d)$: Probabilitas distribusi topik dalam dokumen (d)
- $p(z_{d,n}|\theta_d)$: Probabilitas topik ($z_{d,n}$) untuk kata ke- (n) dalam dokumen (d)
- $p(w_{d,n}|\beta_{1:K}, z_{d,n})$: Probabilitas kata ($w_{d,n}$) dalam topik ($z_{d,n}$)

2.7. Pengujian

Pengujian adalah proses yang dilakukan untuk mengevaluasi suatu metode yang sudah dilakukan. Dalam penelitian, pengujian bertujuan untuk memastikan bahwa metode yang dikembangkan berfungsi sesuai tujuan yang akan dicapai. Adapun pada pengujian ini terdapat beberapa tahap yang harus dilakukan seperti:

2.7.1. Data *Ground Truth*

Istilah "*ground truth*" merujuk pada label yang diberikan kepada kumpulan data yang digunakan untuk melatih model pembelajaran mesin agar dapat menghubungkan input baru dengan output dan memvalidasi kinerjanya (Lebovitz *et al.*, 2021). Data *ground truth* ini menjadi elemen kunci dalam proses pembelajaran mesin, karena menyediakan dasar untuk mengevaluasi seberapa baik model dapat menerapkan pembelajarannya pada data baru. Data ini juga memiliki peran penting dalam proses pengujian, memungkinkan untuk mengukur efektivitas model dalam kondisi yang dikontrol dan memastikan bahwa model tersebut berfungsi sesuai dengan yang diharapkan sebelum diimplementasikan dalam skenario aplikasi dunia nyata.

2.7.2. Pengukuran Recall dan Precision

Mayoritas pengujian *trending topic* menggunakan *recall* dan *precision*. Pengujian *trending topic* menggunakan tiga metrik pengukuran yaitu *topic recall* (TR), *keyword precision* (KP), dan *keyword recall* (KR). *topic recall* (TR) mengukur sejauh mana *trending topic* mencakup topik-topik yang ada pada *ground truth*. *keyword precision* (KP) menghitung proporsi keyword *trending topic* yang sesuai dengan keyword pada *ground truth* dibandingkan dengan total keyword dalam *trending topic* metode usulan. *Keyword recall* (KR) menghitung proporsi *keyword trending topic* yang sesuai dengan *keyword* pada *ground truth* dibandingkan dengan total *keyword* dalam *ground truth*. Berikut adalah pengukuran TR, KP dan KR:

$$TR = \frac{|GT \cap BT|}{|GT|} \quad \dots(2.8)$$

$$KP = \frac{|KGT \cap KBT|}{|KBT|} \quad \dots(2.9)$$

$$KR = \frac{|KGT \cap KBT|}{|KGT|} \quad \dots(2.10)$$

Keterangan :

GT (*Ground Truth Topic*) : Sekumpulan topik yang ada pada *ground truth*

BT (*Trending Topic*) : Sekumpulan *trending topic* yang dihasilkan dari model

KGT (*Keyword Ground Truth Topic*) : Sekumpulan *keyword* yang ada pada *ground truth*

KBT (*Keyword Trending Topic*) : Sekumpulan *keyword* yang dihasilkan oleh model *trending topic*.

2.8. Studi Literatur

berikut ini disajikan tabel yang merangkum berbagai studi literatur yang relevan.

Tabel 2. 1 Taksonomi Penelitian Terdahulu

No	Penulis	Dataset	Preprocessing	Metode	Pengujian
1	(Takahashi, Tomioka and Yamanishi, 2014)	<i>Tweet</i> dari Twitter	-	Link-Anomaly	Eksperimen pada beberapa dataset <i>real-time</i> ; Akurasi deteksi perubahan: 94%
2	(Ariwibowo and Indra, 2021)	500 tweet bahasa Indonesia dari Juni – Juli 2021	Filtering, spaces cleaning, case folding, slangword correction, stopword removal, stemming	Maximum Capturing, Jaccard Similarity, Algoritme Apriori2	Menggunakan 12 data ground truth dari Kompas.com; Topic recall: 58.3%, Keyword precision: 66.6%, Keyword recall: 83.3%
3	(Rizky Fajarudin and Indra Indra, 2023)	Data tweet terkait calon presiden dari Twitter	Case folding, tokenizing, stopword removal, normalization, stemming	Metode Doc-P dengan LSH, Kesamaan Cosinus, TF-IDF	Penggunaan data ground truth; hasil klasterisasi tertinggi dicapai dengan skor 98.0 untuk klaster 0
4	(Irianto, Maududie and Arifin, 2022)	490 dokumen skripsi Fakultas Ilmu Komputer, Universitas Jember	Cleaning, filtering, stemming, tokenizing	K-Means Clustering dengan Cosine Similarity, PCA, Z-Score Normalization	Pengujian menggunakan algoritma Silhouette Coefficient; nilai Silhouette optimal diperoleh pada $k = 2$ dengan nilai 0.80
5	(Hikmah, Basuki and Azhar, 2020)	Tweet tentang tokoh publik politik seperti Joko Widodo, Ahok, dan lainnya	Stopword Removal, feature normalization, case folding	Agglomerative Hierarchical Clustering, LDA, TF-IDF	Penggunaan Silhouette Coefficient untuk evaluasi klaster; nilai optimal pada $k = 2$ dengan Silhouette score of 0.98
6	(Indra, Winarko and Pulungan, 2019)	dataset tweet berbahasa Indonesia, fokus pada isu politik dan sosial, dikumpulkan berdasarkan	Tokenisasi, <i>stemming</i> , <i>cleansing</i> .	document pivot dan BN-grams	BN-grams menunjukkan recall yang lebih tinggi dengan nilai antara 0.556 hingga 0.824, sementara document pivot menunjukkan nilai

		kata kunci relevan.			yang lebih rendah antara 0.3 hingga 0.6.
7	(Utami and Eka Putra, 2022)	52 judul penelitian dosen tahun 2019-2021 di STMIK Primakara	Tokenization, filtering, stemming	K-Means Clustering dengan cosine similarity	Pengujian menggunakan Silhouette Coefficient; hasil optimal dengan 6 cluster
8	(Karmila and Intan Ardianti, 2022)	300 teks berita dari Detik.com	Case folding, tokenizing, stopwords removal, stemming	Latent Dirichlet Allocation (LDA)	Data dibagi menjadi dataset isi berita dan judul berita; ketiga topik yang dihasilkan mencapai akurasi 67%
9	(Destarani, Slamet and Subanti, 2019)	Data keluhan warga Denpasar dari website Pro Denpasar, Feb-Apr 2018, 508 data keluhan	Case folding, tokenizing, stopwords removal, feature selection	Latent Dirichlet Allocation (LDA) dengan Gibbs Sampling	Pembagian data dengan metode hold-out, evaluasi menggunakan nilai likelihood log terbesar; topik dengan probabilitas tertinggi adalah topik 4 mengenai jalan rusak dengan nilai probabilitas 0.055
10	(MUSHTAQ AHMED, 2023)	Data Twitter	Case folding, tokenizing, filtering	Document Pivot Approach with Fixed Window, TF-IDF	Cross-verification dengan LDA; lebih akurat dalam pemrosesan data microblogging teks pendek

BAB III METODOLOGI PENELITIAN

3.1. Data Penelitian

Penelitian ini menggunakan data dari Platform X yang dikumpulkan menggunakan twscrape, sebuah tool scraping yang tersedia di GitHub pada laman berikut: <https://github.com/vladkens/twscrape>. Pengambilan data dilakukan dengan menggunakan keyword “pemilu” untuk mengidentifikasi dan mengumpulkan *tweet* yang relevan dengan topik pemilihan umum. Data dikumpulkan selama periode 25 Maret 2024 hingga 14 April 2024, dengan jumlah total *tweet* yang berhasil dikumpulkan sebanyak 18,012 *tweet*. Proses pengumpulan data ini memungkinkan pengambilan *tweet* berdasarkan tanggal dan *keyword* tertentu tanpa memerlukan API key Platform X, sehingga memudahkan dalam mendapatkan jumlah data yang cukup besar dalam periode waktu yang singkat. Tabel 3.1 memperlihatkan contoh-contoh *tweet* yang telah dikumpulkan menggunakan metode crawling.

Tabel 3. 1 Data Penelitian

created_at	username	full_text
2024-03-25 10:01:12+00:00	Syarman59	Tak ada Gugatan tanpa Kecurangan @gibran_tweet Kalau Pemilu diulang tanpa paslon 02
2024-03-25 15:55:45+00:00	FarilStudents	@never_alonely setau gw menggugat pemilu udah jd hak paslon presiden, yg aneh kl ada ormas yg menentang gugatan ke MK
2024-03-25 15:57:28+00:00	akmalbariq	@syamfamily @zoelfick Loh yg minta pemilu ulang siapa? Masa sama abah sendiri lupa.. Wahahaha
2024-03-25 16:23:38+00:00	KhendyManggala_	@bengkeldodo @gibran_tweet Kalau kalah yah Iklash lah, secara Pemilu Jurdil.
2024-03-29 17:01:56+00:00	titik013	@aniesbaswedan Keren. Bismillah Pemilu ulang, 02 tanpa Gibran

3.2. Metode Pembandingan

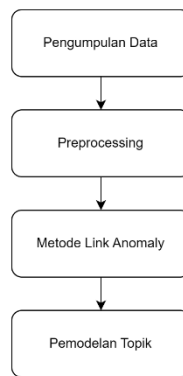
Dalam penelitian yang dilakukan oleh (Fajarudin dan Indra, 2023), digunakan dua metode, BN-grams dan Doc-p, untuk mendeteksi *trending topic* dari *tweet* berbahasa Indonesia. Studi ini bertujuan untuk membandingkan kedua metode tersebut dalam hal akurasi deteksi topik. Hasilnya menunjukkan bahwa metode BN-grams memiliki performa yang lebih baik, dengan topic recall mencapai 20% pada dataset P-4, lebih tinggi dibandingkan dengan Doc-p.

Penelitian kedua oleh (Ariwibowo dan Indra, 2021) pada tahun 2021 fokus pada deteksi *trending topic* terkait COVID-19 dari *tweet* berbahasa Indonesia menggunakan metode *Maximum Capturing Clustering* dan *Jaccard Similarity*. Tujuan utama adalah untuk mengidentifikasi topik yang sering dibicarakan selama pandemi, menggunakan dataset yang terdiri dari 500 *tweet*. Hasil pengujian menunjukkan bahwa metode ini berhasil membentuk klaster dengan topic recall sebesar 58.3%, keyword precision 66.6%, dan keyword recall 83.3%. Meskipun metode ini terbukti efektif dalam membentuk klaster kata yang relevan, masih ada ruang untuk peningkatan dalam pemrosesan data dan adaptasi terhadap perubahan cepat konten sosial.

Penelitian terakhir oleh (Fajarudin dan Indra, 2023) pada tahun 2023 mengimplementasikan metode Doc-p untuk mendeteksi topik tren terkait pemilihan presiden di Twitter. Metode ini memanfaatkan *Locality Sensitive Hashing* (LSH) untuk klasterisasi, *Cosine Similarity* untuk mengukur kesamaan antar dokumen, dan TF-IDF untuk mengevaluasi pentingnya suatu term dalam koleksi dokumen. Tujuan dari penelitian ini adalah untuk membantu masyarakat mengenali calon presiden yang banyak dibicarakan, dengan menemukan *tweet* yang viral tentang para calon. Hasil menunjukkan bahwa metode Doc-p berhasil mengatasi tantangan dalam mendeteksi topik populer secara manual, mencapai skor klasterisasi tertinggi sebesar 98.0 untuk klaster 0.

3.3. Penerapan Metode

Dalam penelitian ini, analisis *trending topic* terkait Pemilu 2024 di Platform X menggunakan algoritma *link anomaly* melibatkan beberapa tahapan, yang dirancang untuk mengolah dan menganalisis data yang diperoleh dari Platform X secara efektif. Rincian tahapan tersebut dapat dilihat pada gambar 3.1 di bawah ini.



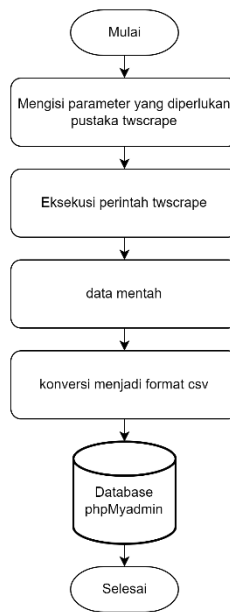
Gambar 3. 1 Penerapan Metode

Pada Gambar 3.1, menggambarkan tahapan yang diaplikasikan dalam penelitian ini. Proses dimulai dengan pengumpulan data, di mana data diambil dari Platform X dan kemudian diubah menjadi format excel dan diimport ke dalam *database*. Selanjutnya, data tersebut diproses melalui tahap *preprocessing*, yang mencakup seperti seleksi atribut, perbaikan waktu, dan pembersihan teks. Setelah data telah disiapkan, Algoritma *link anomaly* diterapkan untuk menganalisis dan mengidentifikasi waktu kemunculan *trending topic*. Akhirnya, pemodelan topik dilakukan untuk mengekstrak dan memodelkan topik-topik utama dari data yang telah diolah.

3.3.1. Pengumpulan Data

Tahap pengumpulan data merupakan langkah awal dalam metodologi penelitian ini, di mana data dikumpulkan menggunakan pustaka *twscrape*. Pustaka ini memungkinkan pengambilan *tweet* secara efisien dengan menargetkan kata kunci spesifik. Proses pengumpulan data dilakukan dengan mengatur beberapa parameter, mengeksekusi perintah pustaka *twscrape*, kemudian konversi data, dan memasukkannya ke *database*.

Tahapan pertama pada proses pengumpulan data adalah mengatur parameter untuk memastikan data yang dikumpulkan sesuai. Parameter yang diatur meliputi kata kunci, rentang waktu, dan bahasa *tweet*. Kata kunci dipilih sesuai dengan topik pada penelitian ini. Rentang waktu diatur untuk mencakup periode yang diinginkan. Bahasa *tweet* ditetapkan ke Bahasa Indonesia agar sesuai dengan konteks penelitian. Setiap langkah dalam proses ini dirancang untuk memastikan data yang dikumpulkan relevan dan dapat digunakan untuk analisis lebih lanjut. Setelah semua parameter diatur, akan didapatkan sebuah data mentah yang berupa sebuah json. Data json tersebut kemudian akan dikonversi ke format CSV, supaya memudahkan dalam proses penelitian. Rincian tahapan pengumpulan data dapat dilihat pada gambar 3.2 berikut ini.



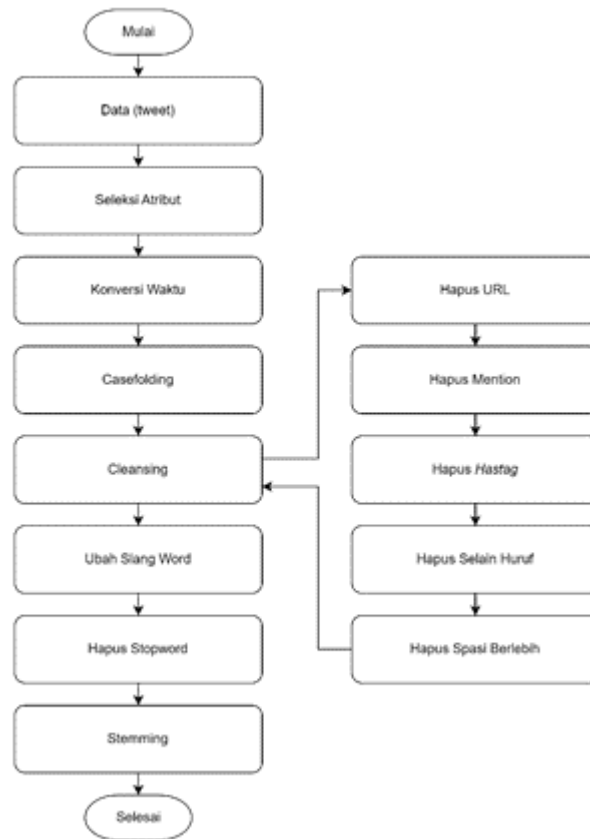
Gambar 3. 2 Pengumpulan Data

3.3.2. *Preprocessing*

Tahap *preprocessing* melibatkan beberapa langkah untuk mempersiapkan data yang telah dikumpulkan menjadi lebih bersih dan terstruktur. Proses ini sangat penting untuk memastikan bahwa data yang akan diproses tidak kotor yang bisa mempengaruhi hasil analisis. Dengan memperbaiki, membersihkan, dan menyusun data secara bersih, *preprocessing* membantu dalam meningkatkan akurasi dan efektivitas algoritma *trending topic* yang akan digunakan pada tahapan berikutnya.

Proses *preprocessing* dimulai dengan pengumpulan data *tweet*, yang kemudian dilanjutkan dengan seleksi atribut untuk menentukan kolom atau fitur yang relevan untuk analisis lebih lanjut. Langkah berikutnya adalah konversi waktu, di mana format waktu dari tweet disesuaikan. Setelah itu, dilakukan proses *case folding* untuk mengubah seluruh teks menjadi huruf kecil.

Langkah selanjutnya adalah *cleansing*, yang bertujuan untuk membersihkan teks dari elemen-elemen yang tidak relevan seperti URL, mention, hashtag, karakter selain huruf, dan spasi berlebih. Kemudian, teks yang mengandung slang atau kata tidak baku diubah ke dalam bentuk yang lebih formal dan standar. Penghapusan stopword juga dilakukan untuk menghilangkan kata-kata umum yang tidak memberikan banyak informasi analitis. Akhirnya, tahap *stemming* diterapkan untuk mengembalikan kata-kata ke bentuk dasar atau akar katanya. tahapan dari proses *preprocessing* dapat dilihat pada gambar 3.3:



Gambar 3. 3 Preprocessing

a) Seleksi Atribut

Dalam seleksi atribut menggunakan untuk data Platform X, prosesnya melakukan pemilihan kolom seperti ‘created_at’, ‘full_text’, dan ‘username’ untuk diambil. Proses ini mengabaikan kolom-kolom lain, dan hanya mengambil data yang relevan.

b) Konversi Waktu

Dalam proses konversi waktu, dilakukan penyesuaian waktu yang diberikan dalam UTC ke format. Ini dilakukan dengan menambahkan tujuh jam ke waktu UTC, sehingga waktu yang dihasilkan sesuai dengan zona waktu Indonesia Barat. Proses ini memastikan bahwa setiap data waktu *tweet* menggunakan waktu lokal.

c) *Case Folding*

Dalam proses *case folding*, dilakukan penyeragaman semua huruf dalam teks ke bentuk huruf kecil. Ini dilakukan untuk mengurangi variasi dalam data, terutama saat melakukan pemrosesan teks. Proses ini memastikan bahwa kata-kata yang sama tidak dianggap berbeda hanya karena perbedaan kapitalisasi.

Misalnya, kata “Pemilu”, “pemilu”, dan “PEMILU” akan dianggap identik sebagai “pemilu” setelah *case folding* dilakukan.

d) *Cleansing*

Dalam proses *cleansing*, dilakukan beberapa langkah untuk membersihkan data teks dari elemen-elemen yang tidak diinginkan atau tidak perlu. Proses ini terdiri dari hapus url, hapus *mention*, hapus *hashtag*, hapus selain huruf, dan hapus spasi berlebih

1. Hapus URL

Menghapus semua tautan web dari teks. Teks seperti ‘<https://t.co/un3zaekvnQ>’ akan dihapus dari kalimat.

2. Hapus Mention

Menghapus semua *mention* yang ada di dalam *tweet*, biasanya diawali dengan simbol ‘@’, seperti contoh “@KudaPusing” akan dihapus dari kalimat.

3. Hapus Hashtag

Menghapus simbol ‘#’ dan teks yang berada setelahnya untuk menghindari pengaruh tagging. Contohnya ‘#Viral’ akan dihapus dari kalimat.

4. Hapus Selain Huruf

Menghapus semua karakter yang bukan merupakan alfabet untuk menyederhanakan teks dan fokus hanya pada kata-kata. Contohnya seperti “pemilu 2024 ricuh!!!”, kata ‘2024’ dan tanda seru akan dihapus dari kalimat sehingga menjadi “pemilu ricuh”.

5. Hapus Spasi Berlebih

Menghapus spasi ekstra yang tidak perlu antar kata dan di akhir kalimat. Contohnya ‘pemilu 2024 ricuh’ akan berubah menjadi ‘pemilu 2024 ricuh’.

f) Ubah *Slang Word*

Dalam proses ubah *slang word*, teks dalam bentuk non-formal diubah menjadi bentuk formal. Misalnya, teks awal seperti “Lu harus coba nih, rasanya nendang banget!” akan diubah menjadi “Anda harus mencoba ini, rasanya sangat lezat!”. Proses pengubahan slang word dilakukan menggunakan kamus yang tersedia secara *open-source* di github pada laman <https://github.com/ramaprakoso/analisis-sentimen>.

g) Hapus *Stop Word*

Proses menghapus *stop word* dalam teks bertujuan untuk menghilangkan kata-kata yang sering muncul tetapi tidak memiliki

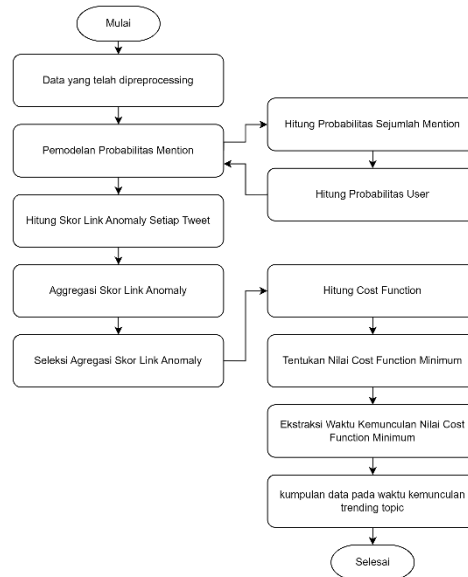
makna, seperti “dan”, “yang”, atau “di”. Penghapusan *stop word* ini dilakukan dengan menggunakan pustaka *sastrawi*.

h) *Stemming*

Proses *stemming* adalah teknik untuk mengubah kata ke bentuk dasar atau akar kata. Hal ini dilakukan untuk menyederhanakan dan meningkatkan kesamaan antara kata-kata yang berhubungan. Contoh kata seperti ‘berlari’ akan diubah menjadi kata dasarnya yaitu ‘lari’. Proses ini dilakukan menggunakan *Sastrawi*.

3.3.3. Algoritma *Link Anomaly*

Dalam tahap ini data yang digunakan adalah data *tweet* yang telah dipreproses. Proses ini dimulai dengan menghitung probabilitas mention setiap *tweet*, yang berfungsi sebagai dasar untuk menghitung skor *link anomaly* untuk setiap *tweet*. Skor ini kemudian diagregasi untuk membentuk sebuah urutan atau sequence yang memberikan gambaran umum tentang aktivitas dalam jaringan. Kemudian urutan ini dianalisis menggunakan *cost function* untuk mengidentifikasi momen dengan aktivitas paling signifikan, yang diindikasikan oleh nilai minimum dari fungsi tersebut. Langkah terakhir adalah mengekstraksi waktu tepat kemunculan nilai minimum ini, yang menunjukkan kapan topik mulai trending. Tahapan dari proses tersebut dapat dilihat pada gambar 3.4 berikut ini.

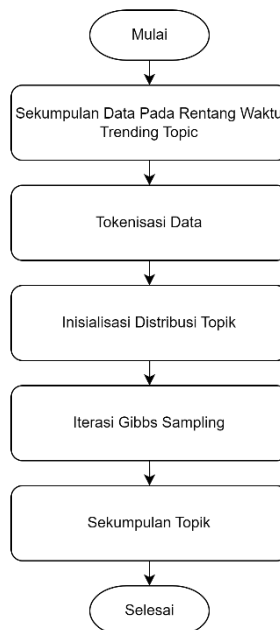


Gambar 3. 4 *Link Anomaly*

Setelah dilakukan algoritma *link anomaly* akan didapatkan Sekumpulan data yang berada pada waktu kemunculan *trending topic*. Data tersebut kemudian akan dilakukan tahapan topik *modelling* untuk mengekstraksi topik yang sedang trending.

3.3.4. Algoritma LDA

Proses *latent dirichlet allocation* (LDA) terdiri dari beberapa tahapan. Pertama, data di-tokenisasi agar mendapatkan seluruh kata pada data. Kemudian, proses LDA dimulai dengan inisialisasi parameter dan distribusi topik secara acak untuk setiap kata dalam dokumen. Tahapan utama LDA adalah iterasi *Gibbs Sampling*, di mana untuk setiap kata dalam setiap dokumen, topiknya dinilai ulang berdasarkan distribusi topik kata sebelumnya dan distribusi kata dalam topik, memperhatikan semua kata lain dan topiknya dalam dokumen. Setiap kata diberikan topik baru berdasarkan probabilitas yang dihitung, yang menggambarkan hubungan kata tersebut dengan topik yang ada dan distribusi kata dalam dokumen. Selanjutnya, menghitung dan menyesuaikan bobot topik untuk setiap kata. Proses ini diulang dalam iterasi yang ditentukan. Setelah proses iterasi selesai, topik untuk setiap kata di dalam dokumen ditentukan dan dapat digunakan untuk mengevaluasi atau mengelompokkan dokumen berdasarkan topik yang dominan. Tahapan dari proses tersebut dapat dilihat pada gambar 3.5 di bawah.



Gambar 3. 5 LDA

3.4. Rancangan Pengujian

Dalam tahap pengujian ini, evaluasi dilakukan untuk menilai kinerja metode yang diusulkan dalam mendeteksi trending topik yang muncul dengan *keyword* “pemilu”. Metode pengujian melibatkan perbandingan antara hasil deteksi trending topik dengan data *ground truth* yang telah ditentukan sebelumnya. Evaluasi ini bertujuan untuk mengukur seberapa akurat model dalam mengidentifikasi trending topik dan *keyword* yang relevan.

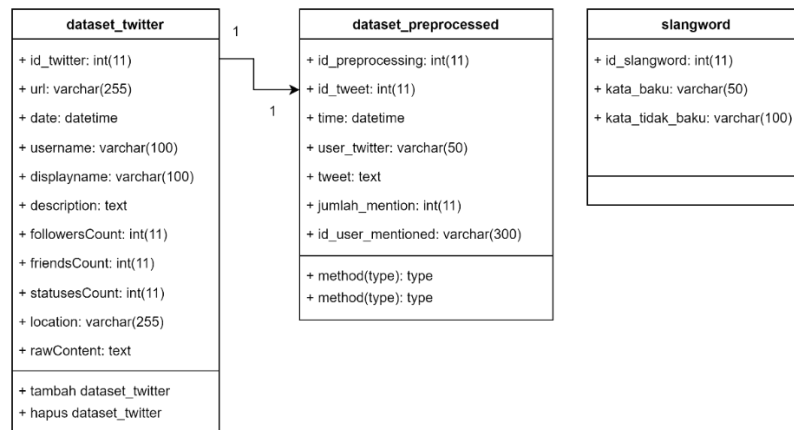
Pengujian dilakukan dengan menggunakan tiga metrik utama: *Topic Recall* (TR), *Keyword Precision* (KP), dan *Keyword Recall* (KR). Ketiga metrik ini memberikan gambaran tentang kemampuan model dalam mendeteksi topik dan *keyword* yang relevan dengan data *ground truth*. TR mengukur sejauh mana trending topik yang terdeteksi mencakup topik *ground truth*. KP menghitung proporsi *keyword* trending topik yang sesuai dengan *keyword* *ground truth* dibandingkan total *keyword* dalam trending topik metode usulan. KR mengukur sejauh mana *keyword* trending topik yang sesuai dengan *keyword* *ground truth* dibandingkan total *keyword* dalam *ground truth*. Dengan ketiga metrik ini, evaluasi memberikan penilaian mengenai kinerja model dalam mendeteksi trending topik.

3.5. Rancangan Basis Data

Dalam pengembangan sistem deteksi *trending topic* mengenai pemilu diperlukan rancangan basis data untuk menyimpan data, melakukan perubahan, dan mengambil data untuk ditampilkan. Perancangan basis data bertujuan untuk memudahkan dalam pengembangan sistem ini.

3.5.1. Class Diagram

Class diagram adalah jenis diagram UML yang menggambarkan struktur sistem dengan menunjukkan kelas-kelas, atribut, metode, dan hubungan antar kelas dalam pengembangan perangkat lunak berorientasi objek. *Class diagram* dapat dilihat pada gambar.



Gambar 3. 6 Class Diagram

3.5.2. Logical Record Structure

Logical Record Structure (LRS) adalah konsep dalam basis data yang mendeskripsikan bagaimana data diatur secara logis di dalam database, tidak bergantung pada penyimpanan fisik data tersebut. LRS menentukan format dan hubungan antar record yang terkait. *Logical record structure* dapat dilihat pada gambar

dataset_twitter	dataset_preprocessed	slangword
+ id_twitter: int(11) + url: varchar(255) + date: datetime + username: varchar(100) + displayname: varchar(100) + description: text + followersCount: int(11) + friendsCount: int(11) + statusesCount: int(11) + location: varchar(255) + rawContent: text	+ id_preprocessing: int(11) + id_tweet: int(11) + time: datetime + user_twitter: varchar(50) + tweet: text + jumlah_mention: int(11) + id_user_mentioned: varchar(300)	+ id_slangword: int(11) + kata_baku: varchar(50) + kata_tidak_baku: varchar(100)

Gambar 3. 7 Logical Record Structure

3.5.3. Spesifikasi Basis Data

- a) Nama Tabel : *dataset_twitter*
Media : *Solid State Drive (SSD)*
Primary_key : *id_twitter*

Tabel 3. 2 Dataset Platform X

No	Nama Field	Jenis	Lebar	Keterangan
1	Id_twitter	Int	11	Id untuk setiap <i>tweet</i>
2	url	Varchar	255	<i>Link tweet</i>
3	date	datetime	-	Tanggal <i>tweet</i>
4	username	varchar	100	<i>Username tweet</i>
5	displayname	varchar	100	<i>Displayname Tweet</i>
6	description	varchar	100	<i>Description user</i>
7	followersCount	int	11	Jumlah <i>follower user</i>
8	friendsCount	int	11	Jumlah <i>friends user</i>
9	statusesCount	int	11	jumlah <i>tweet user</i>
10	location	varchar	255	Lokasi user
11	rawContent	text	-	Isi <i>tweet</i>

- b) Nama Tabel : *dataset_preprocessed*
 Media : *Solid State Drive (SSD)*
 Primary_key : *id_preprocessing*

Tabel 3. 3 Dataset *Preprocessing*

No	Nama Field	Jenis	Lebar	Keterangan
1	Id_preprocessing	Int	11	Id untuk setiap <i>tweet preprocessed</i>
2	id_tweet	int	11	Id untuk setiap <i>tweet</i>
3	time	datetime	-	Tanggal <i>tweet</i>
4	user_twitter	varchar	100	<i>Username tweet</i>
5	tweet	text	-	Teks <i>tweet</i>
6	jumlah_mention	int	11	Jumlah <i>mention</i> pada <i>tweet</i>
7	id_user_mentioned	varchar	300	User <i>mentioned</i> di <i>tweet</i>

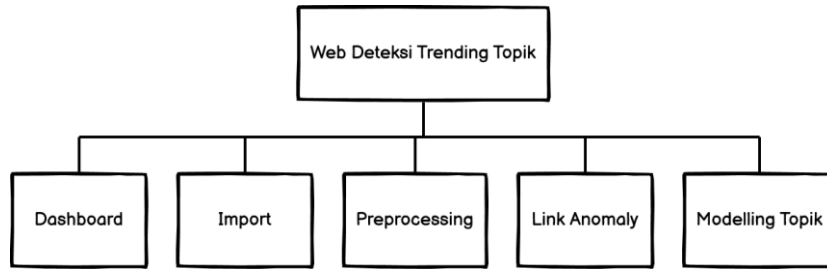
- c) Nama Tabel : *slangword*
 Media : *Solid State Drive (SSD)*
 Primary_key : *id_slangword*

Tabel 3. 4 *Slangword*

No	Nama Field	Jenis	Lebar	Keterangan
1	Id_slangword	Int	11	Id untuk setiap <i>slangword</i>
2	kata_baku	varchar	50	kata baku
3	kata_tidak_baku	varchar	100	kata_tidak_baku

3.6. Rancangan Menu

Rancangan menu adalah tahap penting dalam pengembangan aplikasi atau sistem, yang bertujuan untuk memastikan navigasi yang mudah dipakai oleh pengguna. Rancangan Menu ini terdiri dari menu dashboard, menu *import*, menu *preprocessing*, menu *link anomaly*, dan menu *modelling* topik. Seperti yang tertera pada gambar 3.8



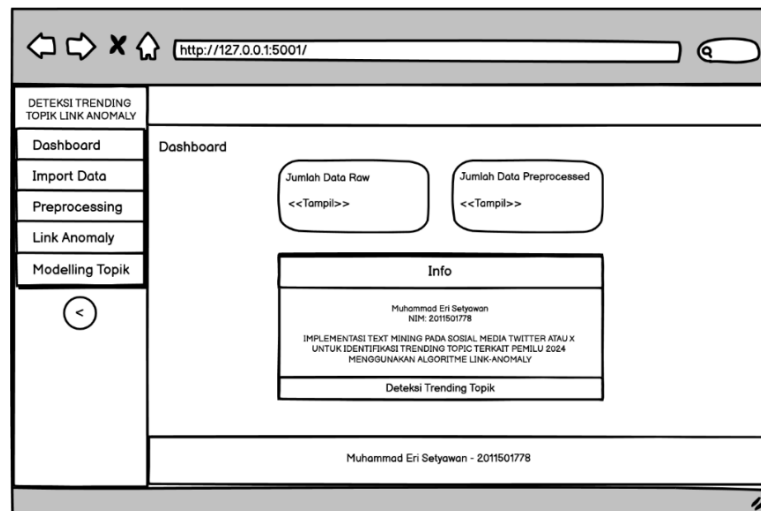
Gambar 3. 8 Rancangan Menu

3.7. Rancangan Layar

Rancangan layar adalah tahap penting dalam pengembangan aplikasi/web yang bertujuan untuk memastikan setiap antarmuka pengguna (UI) dirancang dengan baik. Berikut adalah rancangan layar yang akan ditampilkan:

3.7.1. Rancangan Layar Menu *Dashboard*

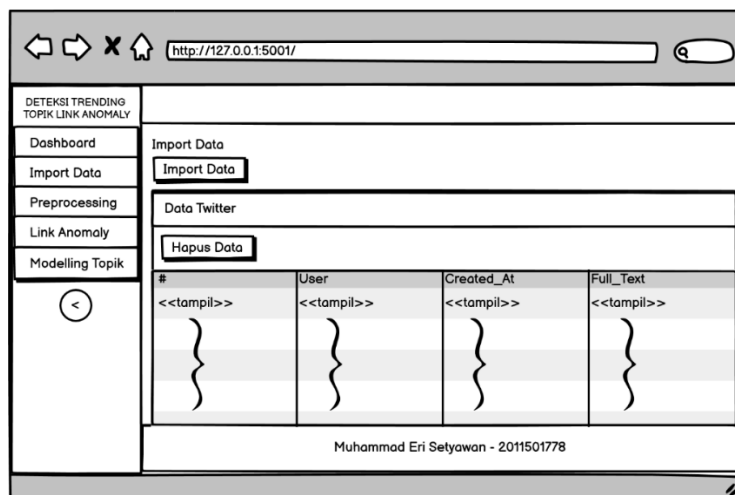
Layar ini menampilkan informasi mengenai jumlah data *raw* dan data *preprocessing* yang tersedia. Selain itu, layar ini juga berfungsi sebagai pengantar yang menjelaskan secara singkat tujuan dan fungsi utama dari aplikasi atau web ini. Rancangan layar menu *dashboard* dapat dilihat pada gambar 3.9.



Gambar 3. 9 Rancangan Layar *Dashboard*

3.7.2. Rancangan Layar Menu *Import*

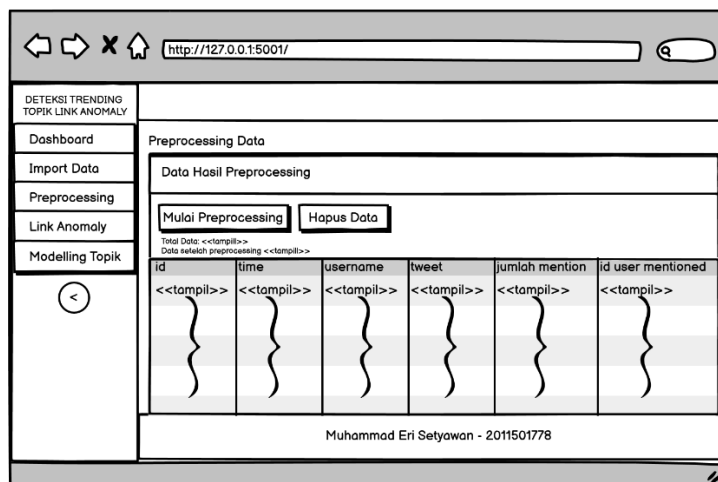
Pada layar ini, pengguna dapat mengimport dataset ke dalam database, menghapus dataset yang tidak diperlukan, dan melihat dataset yang telah diimport. Rancangan layar menu *import* dapat dilihat pada gambar 3.10.



Gambar 3. 10 Rancangan Layar *Import*

3.7.3. Rancangan Layar Menu *Preprocessing*

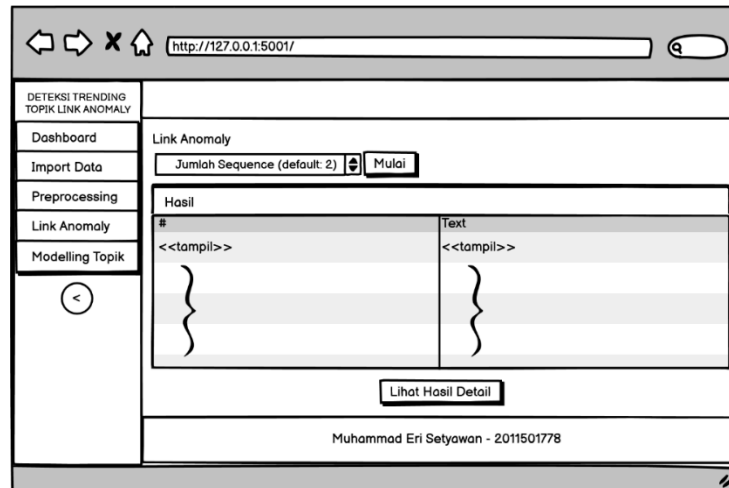
Layar *preprocessing* menyediakan fitur untuk memproses dataset yang telah diimport. Pengguna dapat melihat isi dari database preprocessing untuk memastikan bahwa data telah diolah dengan benar. Selain itu, layar ini juga menyediakan opsi untuk menghapus data *preprocessing* yang sudah tidak diperlukan. Rancangan layar menu *preprocessing* dapat dilihat pada gambar 3.11



Gambar 3. 11 Rancangan Layar *Preprocessing*

3.7.4. Rancangan Layar Menu *Link Anomaly*

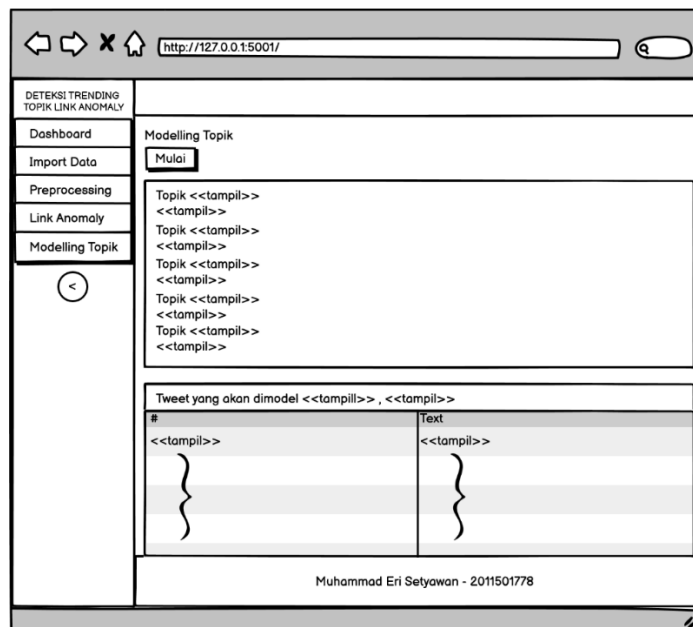
Layar ini dirancang untuk memulai proses *link anomaly* dan melihat hasilnya. Pengguna dapat memilih untuk melihat ringkasan hasil yang menampilkan informasi penting atau melihat detail proses secara mendalam. Rancangan layar menu *link anomaly* dapat dilihat pada gambar 3.12.



Gambar 3. 12 Rancangan Layar *Link Anomaly*

3.7.5. Rancangan Layar Menu *Modelling Topik*

Pada layar ini, pengguna dapat memulai proses *modelling* topik berdasarkan hasil *link anomaly* yang telah dijalankan sebelumnya. Rancangan layar menu *modelling* topik dapat dilihat pada gambar 3.13.



Gambar 3. 13 Rancangan Layar *Modelling Topik*

BAB IV HASIL DAN PEMBAHASAN

4.1. Lingkungan Percobaan

Agar aplikasi dapat berjalan dengan baik dan tanpa masalah, diperlukan spesifikasi khusus pada perangkat keras dan perangkat lunak yang digunakan.

4.1.1. Spesifikasi *Software*

Di bawah ini adalah spesifikasi *software* yang digunakan dalam pembuatan aplikasi.

- a. IDE : Visual Studio Code
- b. Browser : Google Chrome
- c. Bahasa Pemrograman : Python 3.9.2
- d. DBMS : MySql
- e. OS : Windows 11
- f. Lainnya : XAMPP, Microsoft Excel

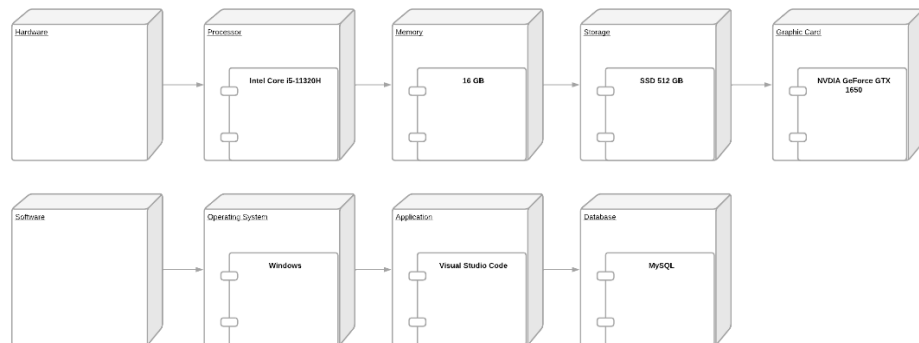
4.1.2. Spesifikasi *Hardware*

Di bawah ini adalah spesifikasi *hardware* yang digunakan dalam pembuatan aplikasi.

- a. *Processor* : Intel Core i5-11320H
- b. RAM : 16 GB
- c. *Storage* : SSD 512 GB
- d. *Graphic Card* : NVIDIA Geforce GTX 1650

4.1.3. *Deployment Diagram*

Gambar 4.1 menunjukkan lingkungan spesifikasi hardware dan software yang digunakan.



Gambar 4. 1 *Deployment Diagram*

4.2. Implementasi Metode

Dalam penelitian ini, terdapat empat tahap utama untuk penerapan metode yang dijalankan secara berurutan, yaitu mencakup pengumpulan data, *preprocessing*, *link anomaly*, dan LDA.

4.2.1. Tahap Pengumpulan Data

Pengumpulan data dimulai dengan menetapkan parameter yang dibutuhkan untuk menjalankan pustaka *twscrape*. Parameter utama yang digunakan adalah kata kunci pencarian yaitu “pemilu”. Selain itu, periode waktu untuk pengambilan data juga ditentukan, yaitu dari tanggal 25 Maret 2024 hingga 14 April 2024. Dengan menggunakan pustaka *twscrape*, proses *crawling* dilakukan dengan memasukkan parameter tersebut. Pustaka ini menghasilkan sekitar 18012 *tweet* yang berkaitan dengan kata kunci yang telah ditentukan. Data yang berhasil didapatkan kemudian disimpan dalam format file teks (.txt).

Setelah proses *crawling* selesai, langkah selanjutnya adalah mengonversi data yang terkumpul dalam file txt ke format CSV. File CSV yang dihasilkan berisi kolom-kolom berikut: *id*, *url*, *date*, *username*, *displayname*, *description*, *followersCount*, *friendsCount*, *statusesCount*, *location*, dan *rawContent*. Setelah konversi ke format CSV, data tersebut akan disimpan ke dalam database melalui halaman import. Data yang telah diimpor ke dalam database selanjutnya diolah melalui tahap *preprocessing*.

4.2.2. Tahap *Preprocessing*

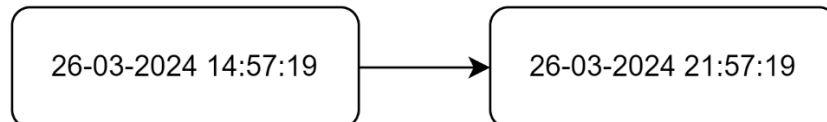
Dalam tahap *preprocessing*, terdapat tujuh langkah yang dilakukan untuk mempersiapkan data, yaitu seleksi atribut, konversi waktu, *case folding*, *cleansing*, merubah *slangword*, menghapus *stopword*, dan *stemming*. Berikut adalah penjabaran dari tahap *preprocessing*:

a. Seleksi Atribut

Dalam tahap seleksi atribut, dilakukan pemilihan kolom yang relevan dari data mentah yang didapatkan. Dari berbagai kolom yang didapatkan seperti *id*, *URL*, *date*, *username*, *displayname*, *description*, *followersCount*, *friendsCount*, *statusesCount*, *location*, dan *rawContent*, hanya beberapa kolom yang dipilih yaitu *date*, *username*, dan *rawContent*. Kolom-kolom ini kemudian dimasukkan ke dalam database dengan kolom-kolom baru: ‘time’ yang diambil dari kolom *date*, ‘user_twitter’ yang diambil dari kolom *username*, dan ‘tweet’ yang berasal dari *rawContent*. Kemudian, kolom ‘jumlah_mention’ diisi dengan jumlah *mention* yang terdapat di *rawContent*, dan ‘id_user_mentioned’ berisi kumpulan *user* yang disebut (*mentioned*) dalam *rawContent*.

b. Konversi Waktu

Dalam tahap konversi waktu, dilakukan perubahan pada data waktu yang awalnya berformat UTC ke WIB. Data mentah yang didapatkan menggunakan waktu UTC, yang merupakan standar waktu global. Waktu ini dikonversi menjadi Waktu Indonesia Barat (WIB). Konversi ini dilakukan dengan menambahkan tujuh jam ke waktu UTC. Contoh konversi waktu dapat dilihat pada gambar 4.2 di bawah.



Gambar 4. 2 Proses Konversi Waktu

c. Case Folding

Dalam tahap *case folding*, proses yang dilakukan adalah mengkonversi semua teks dalam data menjadi huruf kecil. Misalnya, kata "Platform X" diubah menjadi "twitter" sehingga semua kata di dalam data menjadi konsisten. Contoh *case folding* dapat dilihat pada gambar 4.3



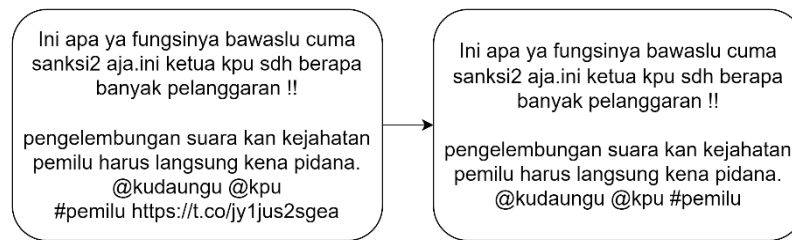
Gambar 4. 3 Proses Case Folding

d. Cleansing

Dalam tahap *cleansing*, proses yang dilakukan beberapa tahap seperti menghapus URL, mention, hashtag, karakter selain huruf, dan spasi berlebih.

1. Hapus URL

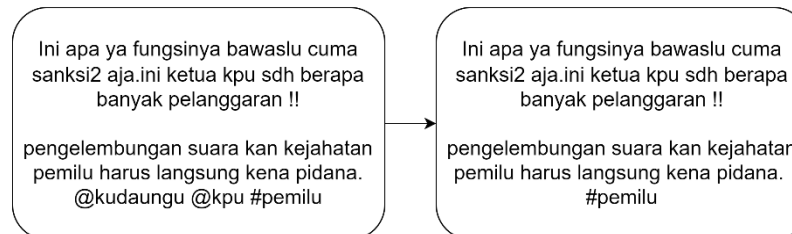
Dalam proses menghapus URL, semua tautan web yang muncul dalam teks akan dihapus. Seperti contoh 'https://link.com' akan dihilangkan. Contoh Penghapusan URL dapat dilihat pada gambar 4.4 di bawah.



Gambar 4. 4 Proses Penghapusan URL

2. Hapus *Mention*

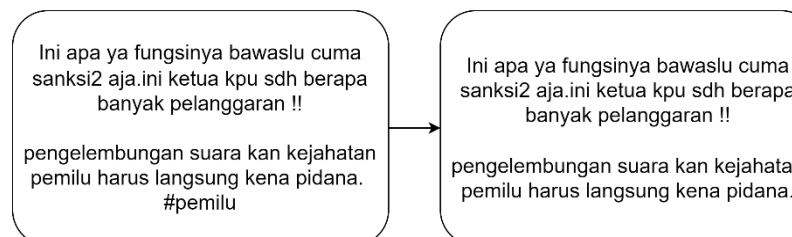
Semua *mention* kepada *user* lain yang diindikasikan dengan simbol '@' dihapus dari teks. Contohnya adalah '@kudaungu' dan '@KPU' akan dihapus. Contoh penghapusan *mention* dapat dilihat pada gambar 4.5 di bawah.



Gambar 4. 5 Proses Hapus *Mention*

3. Hapus *Hashtag*

Semua kata yang diawali dengan simbol '#' akan dihilangkan dari teks. Seperti contoh '#pemilu' akan dihapus. Contoh penghapusan *hashtag* dapat dilihat pada gambar 4.6 di bawah.



Gambar 4. 6 Proses Hapus *Hashtag*

4. Hapus Selain Huruf

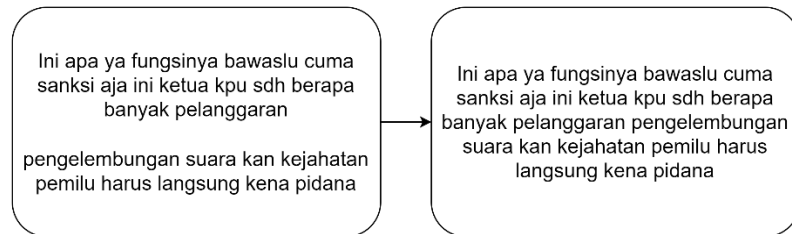
Semua karakter yang bukan merupakan huruf alfabet, seperti angka, simbol, dan tanda baca, dihilangkan dari teks. Contoh penghapusan selain huruf dapat dilihat pada gambar 4.7 di bawah.



Gambar 4. 7 Proses Hapus Selain Huruf

5. Hapus Spasi Berlebih

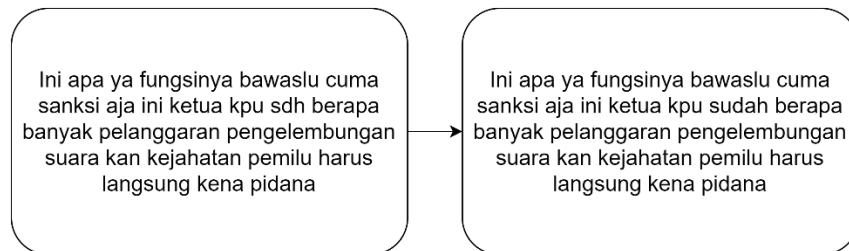
Dalam tahap menghapus spasi berlebih, semua ruang kosong ekstra di dalam teks, termasuk spasi ganda atau lebih, tab, dan baris baru yang tidak perlu, dihilangkan. Contoh penghapusan spasi berlebih dapat dilihat pada gambar 4.8 di bawah.



Gambar 4. 8 Proses Hapus Spasi Berlebih

e. Ubah *Slang word*

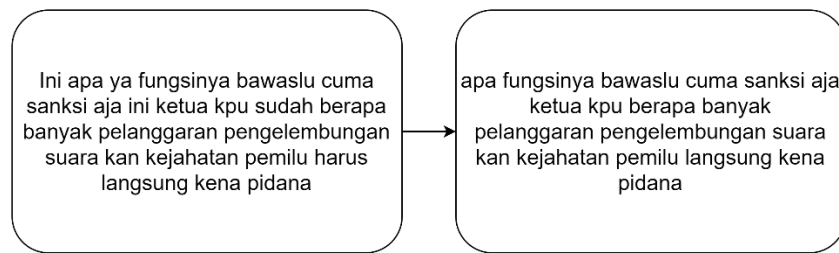
Dalam tahap mengubah *slang word*, kata-kata gaul atau istilah *slang* yang sering digunakan dalam komunikasi informal diubah ke dalam bentuk baku atau lebih formal. Seperti contoh ‘sdh’ akan diubah menjadi ‘sudah’. Contoh penngubahan *Slang word* dapat dilihat pada gambar 4.9 di bawah.



Gambar 4. 9 Proses Ubah *Slang Word*

f. Hapus *Stop Word*

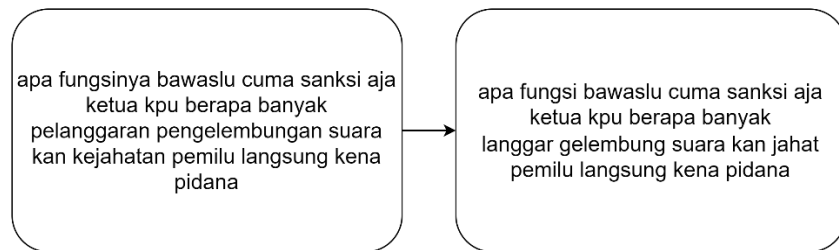
Dalam tahap penghapusan *stop word*, kata-kata umum yang tidak menambahkan informasi penting ke dalam teks, seperti “dan”, “atau”, “tapi”, dan preposisi lainnya akan dihapus dari teks. Contoh Penghapusan *stop word* dapat dilihat pada gambar 4.10 di bawah.



Gambar 4. 10 Proses Hapus Stop Word

g. *Stemming*

Dalam tahap *stemming*, kata-kata di dalam teks diproses untuk diubah ke bentuk dasar atau akar kata (*stem*). Seperti contoh kata “fungsinya” akan diubah menjadi kata dasarnya yaitu “fungsi”. Contoh *stemming* dapat dilihat pada gambar 4.11 di bawah.



Gambar 4. 11 Proses Stemming

4.2.3. Tahap *Link Anomaly*

Tahap *link anomaly* merupakan langkah yang dilakukan setelah proses *preprocessing* selesai. Tujuan utama dari tahap ini adalah untuk mengekstraksi waktu munculnya *trending topic* dengan menggunakan algoritma link anomaly. Berdasarkan Sub-bab 3, tahapan ini terdiri dari enam langkah, yaitu: pemodelan probabilitas *mention*, penghitungan skor *link anomaly*, agregasi skor *link anomaly*, penghitungan *cost function*, penentuan nilai minimum *cost function*, dan ekstraksi waktu kemunculan *trending topic*.

Sebelum memulai langkah pertama, data yang akan digunakan dalam perhitungan tahap *link anomaly* ini berasal dari tabel *tweet_preprocessed*, yang telah melalui proses *preprocessing*. Data ini akan diilustrasikan seperti pada tabel 4.1.

Tabel 4. 1 Ilustrasi data *twitter* yang telah *dipreprocessing*

id	time	user_twitter	tweet	jumlah mention	user mentioned
1	3/25/2024 23:00:49	syamfamily	waduh kok bahas pemilunasi kucing bossmakanya jangan minum air isi ulang	2	@akmalbari q,@zoelfick

2	3/2/2024 23:00:53	awla_zea19	pemilu bersih hormat prinsipprinsip demokrasi kuat	1	@RebeldeF C29
3	3/25/2024 23:01:12	SyahrifulJ	pemilu akan datang	1	@Siswotom o187571

Dalam penelitian ini, sequence yang digunakan adalah 7, sesuai dengan durasi pengambilan data selama tujuh hari. Data per *sequence* ini dibagi lagi menjadi beberapa diskrit. Tabel 4.2 di bawah adalah beberapa data Platform X yang telah dibagi menjadi 7 sequence. Karena data yang digunakan cukup banyak maka yang ditampilkan sebagai contoh ilustrasi dan perhitungan berikutnya hanya beberapa data saja.

Pada tabel 4.2 ditampilkan isi dari data *tweet* yang sudah dipecah menjadi beberapa *sequence*. Dalam tabel, kolom 'id' berfungsi sebagai penanda unik untuk setiap *tweet*, 'time' menunjukkan waktu *tweet* diunggah, dan 'user_twitter' adalah nama *user* yang mengunggah *tweet* tersebut. Kolom 'tweet' berisi teks dari *tweet*, sementara 'jumlah mention' menandakan berapa kali orang atau akun lain disebut dalam *tweet* tersebut. Terakhir, 'user mentioned' mencantumkan nama-nama *user* yang disebutkan dalam masing-masing *tweet*.

Tabel 4. 2 Hasil Setelah Pembagian Sequence

id	time	user_twitter	tweet	jumlah mention	user mentioned
Sequence 1					
1	25-03-2024 10:30:15	Syarman59	adagium fakta pihak satu fakta cukup patah juta opini whats on your mind tetap tahan pecat undur diri percaya tidak ada curang	3	@DKPP_RI,@b awaslu_RI,@off icialMKRI
2	25-03-2024 14:15:17	AthenNana_	cebur tinta pemilu anak jimmy sea avocean	1	@fesssthai
3	25-03-2024 17:01:12	Syarman59	ada gugat curang kalau pemilu ulang paslon	1	@gibran_tweet
4	25-03-2024 19:20:44	lin_conil	klu diulangduit cukong china pasang habis minjem saja kan banyak hsl bumi belum gadai	1	@gibran_tweet
Sequence 2					
558	26-03-2024 05:23:13	ivan_indrajaya4	didik pilih rupa investasi jangka panjang bangun pemilu damai	1	@henrdeyjo
559	26-03-2024 05:23:59	ramona_gidan2 1	pemilu damai jalan wujud ubah positif	1	@marsieles_mo nly
560	26-03-2024 05:24:01	zaky_zakariaa	pemilu damai pintu tuju masa depan cerah	1	@marsieles_mo nly
561	26-03-2024 05:24:05	tedsmagang836 00	naif siapa bantu partai menang pemilu loh bahkan satu partai kursi lebih di senayan kader militan pdip plus grassrootnya salah satu sumber daya oleh capreanya beda sama	3	@hahehehhah, @jecq74,@Bur hanMuhtadi

Sequence 3					
1115	27-03-2024 03:34:53	WongDes90513 142	gwe bngga syukur dkung halal sglacara mnjdi noda hitam sjarah bngsa bhw pemilu ksar brutal tsm kan ingat anak bngsa kelak bukan mslah mnang klah proses demokrasi diInggar plitik gentong babi bagai hrusnya and tolak kcurngan tsm	1	@BennyHarma nID
1116	27-03-2024 03:34:54	emut_sleketep	video korelasi pemilu kan video lama bukti ada kakak perempuan popong tidak malu nyebar video nyebar fitnah kayak begini	1	@MohammadK hafid3
1117	27-03-2024 03:35:02	RudiWah93009 623	hukum tidak ulang maksud ulang pada pemilu lanjut putar dua iya	1	@ZulkifliLubis 69
1118	27-03-2024 03:35:41	SuratmanEman 3	kalah kalah perlu ribut legowo kalau mau urus tidak pemilu sudah menang baru masalah etika	1	@gilang_ahm31 272
Sequence 4					
1672	28-03-2024 03:26:57	samurai_tenro	seperti baru ikut pemilu tuhpdhal masa megawati pasang sama prabowo mrk gugat sby menangbegitu saat jokowi menang prabowo hatta gugat mk prabowo sandi menggugatartinya prabowo berapa kli gugat ml	3	@AMuhammad 6865,@masade pankmuh,@Lag iBeTeAjj
1673	28-03-2024 03:27:23	Arolumb1102	parah kalau tuntutan main pssi boleh serang lewat garis tengah lapang mirip sengketa pemilu	1	@RomeoWalke r19
1674	28-03-2024 03:27:44	hermanaction	sakit sing pemilu pileg	1	@paksiraras
1675	28-03-2024 03:29:21	a_ifa	presiden terlalu fokus ikn tni polisi terlalu fokus pemilu hasil pemilu masalah bangsa indonesia terabaikn gempa banjir tanah longsor kkb celaka rohingya lain tumpuk	1	@tanyakanrl
Sequence 5					
2229	29-03-2024 03:16:12	JunizarT	kalau percaya bawaslu sedang apa park ah nis abah sugiono ikut pemilu	2	@izoelsaka,@C NNIndonesia
2230	29-03-2024 03:16:33	fasya_dera	bagi bansos jelang pemilu alas elnino sekarang banyak warga ny kena musibah daerah gempa banjir longsor mana tu bansos yang joran yang bikin ketawa sekarang bahkan bikin atur pileg boleh ada bansos lawak	1	@CNNIndonesi a
2231	29-03-2024 03:16:35	dicxdicx	tertawa reply dukung abah lucu lucu resmi lantik jadi presiden belum nyalah salah lah kalau orang punya sakit hati bisa nerima kalah hati hitam tutup dengki	1	@susipudjiastut i
2232	29-03-2024 03:16:35	_blacknerd_	pantes lah prabowo kan x pemilu dapat segitu anis baru skrng sabar wkww	1	@tijabar
Sequence 6					

2786	30-03-2024 03:36:47	wongWedokBakoh	obrol sama teman urus partai tk pusat tanggal peb stl pemilu ln sesuai jadwal ruwet bilang syukur kmd mungut suara ulang hasil beda jauh	1	@blue_berets7
2787	30-03-2024 03:37:00	Syahrul6019	personil pks duduk dpr priode sudah orang kalau benar juang sejahtera rakyatjangan munafik yaa bisa tingkat pemilu a dan konsisten bantu penting rakyat agt tangan propinsi ayo pks tunjuk gentlemu	2	@fina_ahmad, @msaid_didu
2788	30-03-2024 03:38:09	tmhalim	sepi job hbs pemilu paman	1	@MasMasBiassaa
2789	30-03-2024 03:49:21	KuliPocok	guru besar wakil semua pikir rakyatkami pilih prabowo gibran murni hati paling dalamdan proses pemilu jalan amandamai penuh kegembiraanbagi kalah mungkin menyakitkanobat sakit hati legowo	1	@AbunawasReturn
Sequence 7					
3343	31-03-2024 06:09:28	jeyl80885	harus masuk rekor muripemilu terkoplak jadi abad	1	@Andria75777
3344	31-03-2024 06:10:02	imam_gung3	partisipasi pemilu bentuk nyata libat dalam bangun negara	1	@Nahnaylaaa
3345	31-03-2024 06:10:17	yunantyo_yas	perintah bukan pihak bukan pihak kait sengketa pemilu	2	@Fauzin15,@g eloraco
3346	31-03-2024 06:10:46	shamia_adzan	harus upaya paham baik penting nasional pemilu tidak bawa arus provokasi politik	1	@Nahnaylaaa

a. Pemodelan Probabilitas *Mention*

Pemodelan probabilitas *mention* melibatkan dua tahap. Pertama, menghitung probabilitas *mention* dalam sebuah tweet, dan kedua, menghitung probabilitas berdasarkan *user* yang melakukan mention tersebut. Kedua hasil perhitungan ini kemudian digunakan untuk menentukan skor *link anomaly* dalam setiap tweet.

1. Hitung Probabilitas Sejumlah *Mention*

Perhitungan probabilitas *mention* dilakukan untuk menentukan kemungkinan munculnya sejumlah *mention* dalam sebuah *tweet*. Perhitungan ini dilakukan untuk setiap *tweet* yang terdapat dalam data. Data *tweet* yang digunakan sebagai ilustrasi dari persamaan (2.1) adalah *tweet* ke-1. Pada *tweet* ke-1 akan dilakukan perhitungan probabilitas *mention* yang dimana *tweet* ke-1 memiliki *mention* yang dituliskan dengan $V_1 = \{@DKPP_RI, @bawaslu_RI, @officialMKRI\}$. Simbol K menandakan total jumlah *mention* dalam *tweet* pertama, yang nilainya adalah 3. Nilai n merupakan jumlah *tweet* pada *sequence tweet* tersebut. Pada perhitungan ini, *tweet* ke-1

berada pada *sequence* 1 yang memiliki jumlah *tweet* sebanyak 557. Simbol m merupakan jumlah *mention* pada *sequence tweet* yang dimana pada *sequence* 1 jumlah *mention* adalah 940. Kemudian nilai α dan β pada penelitian ini masing-masing bernilai 0,5.

$$P(k | T) = \frac{(n + \alpha)}{(m + k + \alpha)} \prod_{j=0}^k \frac{(m + \alpha + j)}{(n + m + \alpha + \beta + j)}$$

$$P(k_1 | T_{\text{Syarman59}}^{10:30:15}) = \frac{(557 + 0.5)}{(940 + 3 + 0.5)} \prod_{j=0}^{556} \frac{(940 + 0.5 + j)}{(557 + 940 + 0.5 + 0.5 + j)}$$

$$P(k_1 | T_{\text{Syarman59}}^{10:30:15}) = 0.5908850026497085 \times 5.295536916778304e - 93$$

$$P(k_1 | T_{\text{Syarman59}}^{10:30:15}) = 3.1290533451021773e - 93$$

Penjabaran persamaan (2.1) menghasilkan nilai probabilitas *mention* untuk k_1 dengan user @Syarman59 adalah $3.1290533451021773e - 91$ % dari total jumlah *mention* di dalam sekumpulan *mention* dan user pada *sequence* pertama. Ini menandakan bahwa probabilitas munculnya *mention* (@DKPP_RI, @bawaslu_RI, @officialMKRI) pada *tweet* ke-1 adalah $3.1290533451021773e - 91$ % dari seluruh *tweet* secara total. Kemudian dilakukan perhitungan yang sama kepada seluruh *tweet* di setiap *sequence*. Tabel 4.3 menunjukkan hasil dari perhitungan probabilitas *mention* untuk beberapa *tweet*.

Tabel 4. 3 Hasil Perhitungan Probabilitas *Mention*

Tweet ke	Perhitungan	Hasil
Sequence 1		
1	$\frac{(557 + 0.5)}{(940 + 3 + 0.5)} \prod_{j=0}^{556} \frac{(940 + 0.5 + j)}{(557 + 940 + 0.5 + 0.5 + j)}$	$3.1290533451021773 \times 10^{-93}$
2	$\frac{(557 + 0.5)}{(940 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(940 + 0.5 + j)}{(557 + 940 + 0.5 + 0.5 + j)}$	$3.1357002985702649 \times 10^{-93}$
3	$\frac{(557 + 0.5)}{(940 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(940 + 0.5 + j)}{(557 + 940 + 0.5 + 0.5 + j)}$	$3.1357002985702649 \times 10^{-93}$
4	$\frac{(557 + 0.5)}{(940 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(940 + 0.5 + j)}{(557 + 940 + 0.5 + 0.5 + j)}$	$3.1357002985702649 \times 10^{-93}$
Sequence 2		
558	$\frac{(557 + 0.5)}{(859 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(859 + 0.5 + j)}{(557 + 859 + 0.5 + 0.5 + j)}$	$9.2202807964542389 \times 10^{-99}$
559	$\frac{(557 + 0.5)}{(859 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(859 + 0.5 + j)}{(557 + 859 + 0.5 + 0.5 + j)}$	$9.2202807964542389 \times 10^{-99}$

560	$\frac{(557 + 0.5)}{(859 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(859 + 0.5 + j)}{(557 + 859 + 0.5 + 0.5 + j)}$	$9.2202807964542389 \times 10^{-99}$
561	$\frac{(557 + 0.5)}{(859 + 3 + 0.5)} \prod_{j=0}^{556} \frac{(859 + 0.5 + j)}{(557 + 859 + 0.5 + 0.5 + j)}$	$9.1989004351870983 \times 10^{-99}$
Sequence 3		
1115	$\frac{(557 + 0.5)}{(875 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(875 + 0.5 + j)}{(557 + 875 + 0.5 + 0.5 + j)}$	$1.2922175428630029 \times 10^{-97}$
1116	$\frac{(557 + 0.5)}{(875 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(875 + 0.5 + j)}{(557 + 875 + 0.5 + 0.5 + j)}$	$1.2922175428630029 \times 10^{-97}$
1117	$\frac{(557 + 0.5)}{(875 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(875 + 0.5 + j)}{(557 + 875 + 0.5 + 0.5 + j)}$	$1.2922175428630029 \times 10^{-97}$
1118	$\frac{(557 + 0.5)}{(875 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(875 + 0.5 + j)}{(557 + 875 + 0.5 + 0.5 + j)}$	$1.2922175428630029 \times 10^{-97}$
Sequence 4		
1672	$\frac{(557 + 0.5)}{(1051 + 3 + 0.5)} \prod_{j=0}^{556} \frac{(1051 + 0.5 + j)}{(557 + 1051 + 0.5 + 0.5 + j)}$	$1.2697943963681545 \times 10^{-86}$
1673	$\frac{(557 + 0.5)}{(1051 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1051 + 0.5 + j)}{(557 + 1051 + 0.5 + 0.5 + j)}$	$1.2722073073351248 \times 10^{-86}$
1674	$\frac{(557 + 0.5)}{(1051 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1051 + 0.5 + j)}{(557 + 1051 + 0.5 + 0.5 + j)}$	$1.2722073073351248 \times 10^{-86}$
1675	$\frac{(557 + 0.5)}{(1051 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1051 + 0.5 + j)}{(557 + 1051 + 0.5 + 0.5 + j)}$	$1.2722073073351248 \times 10^{-86}$
Sequence 5		
2229	$\frac{(557 + 0.5)}{(1024 + 2 + 0.5)} \prod_{j=0}^{556} \frac{(1024 + 0.5 + j)}{(557 + 1024 + 0.5 + 0.5 + j)}$	$3.8753671412918562 \times 10^{-88}$
2230	$\frac{(557 + 0.5)}{(1024 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1024 + 0.5 + j)}{(557 + 1024 + 0.5 + 0.5 + j)}$	$3.8791461438674700 \times 10^{-88}$
2231	$\frac{(557 + 0.5)}{(1024 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1024 + 0.5 + j)}{(557 + 1024 + 0.5 + 0.5 + j)}$	$3.8791461438674700 \times 10^{-88}$
2232	$\frac{(557 + 0.5)}{(1024 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1024 + 0.5 + j)}{(557 + 1024 + 0.5 + 0.5 + j)}$	$3.8791461438674700 \times 10^{-88}$
Sequence 6		
2786	$\frac{(557 + 0.5)}{(1123 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1123 + 0.5 + j)}{(557 + 1123 + 0.5 + 0.5 + j)}$	$7.8168478765570719 \times 10^{-83}$
2787	$\frac{(557 + 0.5)}{(1123 + 2 + 0.5)} \prod_{j=0}^{556} \frac{(1123 + 0.5 + j)}{(557 + 1123 + 0.5 + 0.5 + j)}$	$7.8099026540990015 \times 10^{-83}$
2788	$\frac{(557 + 0.5)}{(1123 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1123 + 0.5 + j)}{(557 + 1123 + 0.5 + 0.5 + j)}$	$7.8168478765570719 \times 10^{-83}$
2789	$\frac{(557 + 0.5)}{(1123 + 1 + 0.5)} \prod_{j=0}^{556} \frac{(1123 + 0.5 + j)}{(557 + 1123 + 0.5 + 0.5 + j)}$	$7.8168478765570719 \times 10^{-83}$

Sequence 7		
3343	$\frac{(554 + 0.5)}{(1026 + 1 + 0.5)} \prod_{j=0}^{553} \frac{(1026 + 0.5 + j)}{(554 + 1026 + 0.5 + 0.5 + j)}$	$3.0562157376646374 \times 10^{-87}$
3344	$\frac{(554 + 0.5)}{(1026 + 1 + 0.5)} \prod_{j=0}^{553} \frac{(1026 + 0.5 + j)}{(554 + 1026 + 0.5 + 0.5 + j)}$	$3.0562157376646374 \times 10^{-87}$
3345	$\frac{(554 + 0.5)}{(1026 + 2 + 0.5)} \prod_{j=0}^{553} \frac{(1026 + 0.5 + j)}{(554 + 1026 + 0.5 + 0.5 + j)}$	$3.0532442104525179 \times 10^{-87}$
3346	$\frac{(554 + 0.5)}{(1026 + 1 + 0.5)} \prod_{j=0}^{553} \frac{(1026 + 0.5 + j)}{(554 + 1026 + 0.5 + 0.5 + j)}$	$3.0562157376646374 \times 10^{-87}$

2. Hitung Probabilitas User

Perhitungan probabilitas *user* dilakukan untuk menentukan probabilitas *user* yang disebut (*mention*) dalam sebuah *tweet*. Perhitungan ini dilakukan untuk setiap *tweet* yang ada di dalam data. Persamaan (2.2) digunakan untuk menjabarkan ilustrasi perhitungan pada *tweet* 1 yang mengarah ke *user* @DKPPRI, @bawasluRI, @officialMKRI dengan parameter $\gamma = 0,5$. Simbol v adalah *user* yang di-*mention* pada *tweet* 1 (@DKPPRI, @bawasluRI, @officialMKRI) dan akan dihitung probabilitasnya. m_v merupakan jumlah total *mention* yang ditujukan kepada user dengan total masing-masing adalah @DKPPRI = 1, @bawasluRI = 1, dan @officialMKRI = 6. Sementara m adalah jumlah keseluruhan *mention* yang terjadi pada *sequence* yang berjumlah sebanyak 940.

$$P(v | T) = \frac{m_v}{m + \gamma}$$

$$\begin{aligned} \text{tweet 1} &= P(@DKPPRI | T_{Syarman59}^{10:30:15}) \times \\ &P(@bawasluRI | T_{Syarman59}^{10:30:15}) \times \\ &P(@officialMKRI | T_{Syarman59}^{10:30:15}) \end{aligned}$$

$$\text{tweet 1} = \frac{1}{940 + 0.5} \times \frac{1}{940 + 0.5} \times \frac{6}{940 + 0.5}$$

$$\begin{aligned} \text{tweet 1} &= 0.001063264221158958 \times \\ &0.001063264221158958 \times \\ &0.006379585326953748 \end{aligned}$$

$$\text{tweet 1} = 7.212317728846991 \times 10^{-9}$$

Probabilitas *user* yang di-*mention* dalam *tweet* pertama (@DKPP_RI, @bawaslu_RI, @officialMKRI) adalah $7.212317728846991 \times 10^{-7}\%$ dari semua *mention*

yang terjadi. Kemudian dilakukan perhitungan yang sama kepada seluruh *tweet* di setiap *sequence*. Tabel 4.4 menunjukkan hasil dari perhitungan probabilitas *user* untuk beberapa *tweet*.

Tabel 4. 4 Hasil Perhitungan Probabilitas *User*

Tweet ke	Perhitungan	Hasil
Sequence 1		
1	$\frac{1}{940 + 0.5} \times \frac{1}{940 + 0.5} \times \frac{6}{940 + 0.5}$	0.001063264221158958, 0.001063264221158958, 0.006379585326953748
2	$\frac{1}{940 + 0.5}$	0.001063264221158958
3	$\frac{23}{940 + 0.5}$	0.024455077086656035
4	$\frac{23}{940 + 0.5}$	0.024455077086656035
Sequence 2		
558	$\frac{33}{859 + 0.5}$	0.038394415357766144
559	$\frac{39}{859 + 0.5}$	0.04537521815008726
560	$\frac{39}{859 + 0.5}$	0.04537521815008726
561	$\frac{1}{859 + 0.5} \times \frac{1}{859 + 0.5} \times \frac{1}{859 + 0.5}$	0.0011634671320535194, 0.0011634671320535194, 0.0011634671320535194
Sequence 3		
1115	$\frac{20}{875 + 0.5}$	0.022844089091947458
1116	$\frac{4}{875 + 0.5}$	0.004568817818389492
1117	$\frac{4}{875 + 0.5}$	0.004568817818389492
1118	$\frac{3}{875 + 0.5}$	0.0034266133637921186
Sequence 4		
1672	$\frac{1}{1051 + 0.5} \times \frac{1}{1051 + 0.5} \times \frac{1}{1051 + 0.5}$	0.0009510223490252021, 0.0009510223490252021, 0.0009510223490252021
1673	$\frac{1}{1051 + 0.5}$	0.0009510223490252021
1674	$\frac{1}{1051 + 0.5}$	0.0009510223490252021
1675	$\frac{9}{1051 + 0.5}$	0.008559201141226819
Sequence 5		

2229	$\frac{1}{1024 + 0.5} \times \frac{31}{1024 + 0.5}$	0.0009760858955588092, 0.030258662762323085
2230	$\frac{31}{1024 + 0.5}$	0.030258662762323085
2231	$\frac{2}{1024 + 0.5}$	0.0019521717911176184
2232	$\frac{3}{1024 + 0.5}$	0.0029282576866764276
Sequence 6		
2786	$\frac{1}{1123 + 0.5}$	0.0008900756564307966
2787	$\frac{1}{1024 + 0.5} \times \frac{2}{1024 + 0.5}$	0.0008900756564307966, 0.0017801513128615932
2788	$\frac{1}{1123 + 0.5}$	0.0035603026257231864
2789	$\frac{1}{1123 + 0.5}$	0.0008900756564307966
Sequence 7		
3343	$\frac{3}{1026 + 0.5}$	0.0029225523623964927
3344	$\frac{2}{1026 + 0.5}$	0.001948368241597662
3345	$\frac{1}{1024 + 0.5} \times \frac{5}{1024 + 0.5}$	0.000974184120798831, 0.004870920603994155
3346	$\frac{2}{1026 + 0.5}$	0.001948368241597662

b. Hitung Skor *Link Anomaly* Setiap *Tweet*

Tahapan menghitung skor *link anomaly* dilakukan setelah selesai pemodelan probabilitas *mention*. Perhitungan ini diterapkan untuk setiap *tweet* dengan menggunakan hasil perhitungan dari probabilitas *mention* dan probabilitas *user*.

$$\begin{aligned}
s(x) &= -\log P(k | T_u^t) - \sum_{u \in V} \log P(u | T_u^t) \\
s(x) &= -\log(3.1290533451021773 \times 10^{-93}) - \\
&\quad \log(0.001063264221158958) + \\
&\quad \log(0.001063264221158958) + \\
&\quad \log(0.006379585326953748) \\
&= 100.64651218207501
\end{aligned}$$

Berdasarkan perhitungan di atas, didapatkan skor untuk *tweet* ke – 1 sebesar 100.64. Kemudian menggunakan rumus yang sama dilakukan perhitungan skor *link anomaly* ke semua *tweet* di dalam data. Tabel 4.5 di bawah adalah hasil dari perhitungan skor *link anomaly* untuk beberapa *tweet* di setiap *sequence*.

Tabel 4. 5 Hasil Perhitungan Skor *Link Anomaly*

Tweet ke	Perhitungan	Hasil
Sequence 1		
1	$-\log(3.129 \times 10^{-93}) - \log(0.001063) + \log(0.001063) + \log(0.006379)$	100.64651
2	$-\log(3.136 \times 10^{-93}) - \log(0.001063)$	95.477024
3	$-\log(3.136 \times 10^{-93}) - \log(0.024450)$	94.115296
4	$-\log(3.129 \times 10^{-93}) - \log(0.024450)$	94.115296
Sequence 2		
558	$-\log(3.129 \times 10^{-93}) - \log(0.0383944)$	99.450987
559	$-\log(9.220 \times 10^{-99}) - \log(0.0453752)$	99.378437
560	$-\log(9.220 \times 10^{-99}) - \log(0.0453752)$	99.378437
561	$-\log(9.199 \times 10^{-99}) - \log(0.001163) + \log(0.001163) + \log(0.001163)$	106.83900
Sequence 3		
1115	$-\log(1.292 \times 10^{-97}) - \log(0.0228440)$	98.529890
1116	$-\log(1.292 \times 10^{-97}) - \log(0.0045688)$	99.228860
1117	$-\log(1.292 \times 10^{-97}) - \log(0.0045688)$	99.228860
1118	$-\log(1.292 \times 10^{-97}) - \log(0.0034266)$	99.353799
Sequence 4		
1672	$-\log(1.270 \times 10^{-86}) - \log(0.000951) + \log(0.000951) + \log(0.000951)$	94.961694
1673	$-\log(1.270 \times 10^{-86}) - \log(0.0009510)$	88.917251
1674	$-\log(1.270 \times 10^{-86}) - \log(0.0009510)$	88.917251
1675	$-\log(1.270 \times 10^{-86}) - \log(0.0085592)$	87.963008
Sequence 5		
2229	$-\log(3.875 \times 10^{-88}) - \log(0.000976) + \log(0.030258)$	91.941349
2230	$-\log(3.879 \times 10^{-88}) - \log(0.0302586)$	88.930414
2231	$-\log(3.879 \times 10^{-88}) - \log(0.0019521)$	90.120745

2232	$-\log(3.879 \times 10^{-88}) - \log(0.0029282)$	89.944654
Sequence 6		
2786	$-\log(7.817 \times 10^{-83}) - \log(0.0008900)$	85.157541
2787	$-\log(7.810 \times 10^{-83}) - \log(0.000890) + \log(0.0017801)$	87.907470
2788	$-\log(7.817 \times 10^{-83}) - \log(0.0035603)$	84.555481
2789	$-\log(7.817 \times 10^{-83}) - \log(0.0008900)$	85.157541
Sequence 7		
3343	$-\log(3.056 \times 10^{-87}) - \log(0.0029225)$	89.049053
3344	$-\log(3.056 \times 10^{-87}) - \log(0.0019483)$	89.225144
3345	$-\log(3.053 \times 10^{-87}) - \log(0.000974) + \log(0.004870)$	91.838986
3346	$-\log(3.056 \times 10^{-87}) - \log(0.0019483)$	89.225144

c. Agregasi Skor *Link Anomaly*

Setelah menghitung skor *link anomaly* untuk setiap *tweet*, dilakukan agregasi skor *link anomaly* untuk setiap *tweet* per diskrit. Perhitungan ini dilakukan secara menyeluruh ke semua *tweet* di dalam diskrit masing-masing. Berikut adalah contoh perhitungan agregasi skor *link anomaly* pada diskrit pertama di sequence kedua yang melibatkan *tweet* ke-558 dan ke-559 menggunakan persamaan (2.5). Nilai τ merupakan lebar *window* setiap diskrit yang berarti jarak waktu kemunculan *tweet* pada diskrit. Lebar *window* diskrit untuk *tweet* ke-558 dan ke-559 adalah 46 detik.

$$s'_j = \frac{1}{\tau} \sum_{t_i \in [\tau(j-1), \tau j]} s(x_i)$$

$$s'_1 = \frac{1}{46} \times [s(x_1) + s(x_2)]$$

$$s'_1 = \frac{1}{46} \times [99.45098779 + 99.378437]$$

$$s'_1 = 4.3223788$$

Berdasarkan perhitungan di atas didapatkan agregasi skor *link anomaly* pada *tweet* ke-558 dan 559 di *sequence* 2 adalah 4.3223788. Kemudian menggunakan cara yang sama dilakukan perhitungan untuk keseluruhan *tweet* dan setiap *sequence*. Tabel 4.6 di bawah adalah hasil dari perhitungan

agregasi skor *link anomaly* untuk beberapa diskrit di setiap *sequence*.

Tabel 4. 6 Perhitungan Agregasi Skor *Link Anomaly*

Diskrit ke	Perhitungan	Hasil
Sequence 1		
1	$\frac{1}{13502} \times [100.64651218 + 95.477024]$	0.0145255
2	$\frac{1}{8372} \times [94.115296 + 94.115296]$	0.0224833
Sequence 2		
1	$\frac{1}{46} \times [99.450987 + 99.378437]$	4.3223788
2	$\frac{1}{4} \times [99.378437 + 106.83900]$	51.554359
Sequence 3		
1	$\frac{1}{1} \times [98.529890 + 99.228860]$	197.75875
2	$\frac{1}{39} \times [99.228860 + 99.353799]$	5.0918630
Sequence 4		
1	$\frac{1}{26} \times [94.961694 + 88.917251]$	7.0722671
2	$\frac{1}{97} \times [88.917251 + 87.963008]$	1.8235078
Sequence 5		
1	$\frac{1}{21} \times [91.941349 + 88.930414]$	8.6129411
2	$\frac{1}{159} \times [90.120745 + 89.944654]$	1.7138246
Sequence 6		
1	$\frac{1}{13} \times [85.157541 + 87.907470]$	13.31269
2	$\frac{1}{672} \times [84.555481 + 85.157541]$	0.2525491
Sequence 7		
1	$\frac{1}{34} \times [89.049053 + 89.225144]$	5.2433587

2	$\frac{1}{29} \times [91.838986 + 89.225144]$	6.2435907
---	---	-----------

Setelah dilakukan perhitungan agregasi skor *link anomaly* untuk setiap *tweet*. Dilakukan pemilahan agregasi skor *link anomaly* yang dimana agregasi yang memiliki nilai di bawah dari *threshold burst* (*threshold burst* = 0.9995) akan dibuang dan tidak dipakai. Agregasi yang memiliki nilai di atas *threshold burst* akan menjadi titik/*state* pada proses perhitungan selanjutnya. Seperti contoh pada sequence pertama untuk diskrit yang ke-1 dan ke-2 memiliki nilai agregasi 0.0145255 dan 0.0224833. Keduanya tidak dapat menjadi *state* karena berada di bawah *threshold burst*, oleh karena itu tidak dipakai atau dibuang dari sequence. Tabel 4.7 di bawah menampilkan hasil seleksi agregasi skor *link anomaly* untuk beberapa diskrit.

Tabel 4. 7 Hasil Seleksi Agregasi Skor Link Anomaly

Diskrit ke	Perhitungan Hasil
Sequence 1	
4	8.5252776
5	2.9088391
Sequence 2	
1	4.3223788
2	51.554359
Sequence 3	
1	197.75875
2	5.0918630
Sequence 4	
1	7.0722671
2	1.8235078
Sequence 5	
1	8.6129411
2	1.7138246
Sequence 6	
1	13.31269
3	2.1520766
Sequence 7	
1	5.2433587
2	6.2435907

d. Hitung *Cost Function*

Setelah dilakukan perhitungan agregasi skor *link anomaly* dan penyeleksiannya. Langkah berikutnya adalah

menghitung *cost function* setiap *sequence* menggunakan hasil agregasi skor *link anomaly*. Perhitungan ini dilakukan ke seluruh *sequence* yang ada. Berikut adalah contoh perhitungan *cost function* pada *sequence* pertama:

$$c(q|x) = b \ln \left(\frac{1-p}{p} \right) + \sum_{t=1}^n -\ln f_{it}(x_t)$$

$$c(q_1|x) = b \ln \left(\frac{1-p}{p} \right) + \sum_{t=1}^n -\ln(\alpha \times e^{-\alpha \times x_t})$$

$$c(q_1|x) = 263 \ln \left(\frac{1-0.3}{0.3} \right) + \sum_{t=1}^{263} -\ln(0.01 \times e^{-0.01 \times x_t})$$

$$c(q_1|x) = 222.83933 + 1315.0597$$

$$c(q_1|x) = 1537.0517$$

Nilai b merupakan jumlah *state* yang berada pada *sequence* setelah agregasi skor *link anomaly* diseleksi. Pada *sequence* pertama jumlah *state* adalah 263. Nilai p adalah probabilitas transisi kemunculan yang bernilai 0,3. Pada penelitian ini parameter α digunakan dengan nilai 0.01. Nilai x_t adalah selang waktu di diskrit. Dari perhitungan di atas *sequence* pertama mendapatkan nilai *cost function* sebesar 1537.0517. Dengan menggunakan rumus yang sama dilakukan perhitungan *cost function* untuk setiap *sequence* yang ada. Tabel 4.8 Di bawah menampilkan hasil perhitungan *cost function* untuk setiap *sequence*.

Tabel 4. 8 Hasil Perhitungan Cost Function

Sequence	Perhitungan	Hasil
1	$263 \ln \left(\frac{1-0.3}{0.3} \right) + \sum_{t=1}^{263} -\ln(0.01 \times e^{-0.01 \times x_t})$	1537.0517
2	$258 \ln \left(\frac{1-0.3}{0.3} \right) + \sum_{t=1}^{258} -\ln(0.01 \times e^{-0.01 \times x_t})$	1516.9867
3	$264 \ln \left(\frac{1-0.3}{0.3} \right) + \sum_{t=1}^{264} -\ln(0.01 \times e^{-0.01 \times x_t})$	1568.3015
4	$262 \ln \left(\frac{1-0.3}{0.3} \right) + \sum_{t=1}^{262} -\ln(0.01 \times e^{-0.01 \times x_t})$	1540.9866
5	$258 \ln \left(\frac{1-0.3}{0.3} \right) + \sum_{t=1}^{258} -\ln(0.01 \times e^{-0.01 \times x_t})$	1523.0767

6	$252 \ln\left(\frac{1-0.3}{0.3}\right) + \sum_{t=1}^{252} -\ln(0.01 \times e^{-0.01 \times x_t})$	1506.1519
7	$236 \ln\left(\frac{1-0.3}{0.3}\right) + \sum_{t=1}^{236} -\ln(0.01 \times e^{-0.01 \times x_t})$	1429.3124

e. Menentukan Nilai *Cost Function Minimum*

Nilai *cost function* setiap *sequence* yang telah dihitung kemudian dipilih nilai yang paling kecil sebagai *cost function minimum*. *Sequence 7* memiliki nilai *cost function* terendah dari semua *sequence* yang ada dengan nilai 1429.3124. Hal ini menandakan bahwa *sequence* ke-7 menjadi nilai *cost function minimum* dan kemudian akan di ekstrak waktu kemunculannya.

f. Ekstrasi Waktu Kemunculan

Setelah *sequence* yang memiliki nilai *cost function minimum* diidentifikasi, langkah selanjutnya adalah mengekstrak waktu kemunculan dari *sequence* tersebut. *Sequence* ke-7 memiliki data tweet yang dimulai dari tanggal 31/03/2024 pukul 06:09:28 sampai dengan 01/04/2024 pukul 06:59:31. Hal ini menunjukkan pada waktu tersebut terjadinya kemunculan *trending topic*.

4.2.4. Tahap LDA

Tahap LDA merupakan tahap yang dilakukan setelah melakukan perhitungan *link anomaly*. Setelah didapatkan beberapa *tweet* yang berada di dalam waktu terindikasi *trending topic*, kemudian dilakukan pencarian topik menggunakan algoritma LDA. Tahapan ini bertujuan untuk menemukan topik-topik yang muncul dari beberapa *tweet* di dalam waktu kemunculan *trending topic*. Algoritma LDA memiliki beberapa tahap yaitu: tokenisasi data, inisialisasi distribusi topik, dan iterasi *gibbs sampling*.

Sebelum melakukan tahap LDA, Tabel 4.9 di bawah menampilkan beberapa data *tweet* yang berada pada waktu terindikasinya *trending topic*.

Tabel 4. 9 Data Tweet yang Berada di Waktu Kemunculan *Trending Topic*

id	time	user_twitter	tweet	jumlah mention	user mentioned
3344	31-03-2024 06:10:02	imam_gung3	partisipasi pemilu bentuk nyata libat dalam bangun negara	1	@Nahnaylaa a

3345	31-03-2024 06:10:17	yunantyo_yas	perintah bukan pihak bukan pihak kait sengketa pemilu	2	@Fauzin15, @geloraco
3346	31-03-2024 06:10:46	shamia_adzan	harus upaya paham baik penting nasional pemilu tidak bawa arus provokasi politik	1	@Nahnaylaa a

a. Tokenisasi Data

Tahapan pertama pada proses LDA adalah mentokenisasi teks yang ada pada semua *tweet*. Tujuan dari tahapan ini adalah agar data berubah menjadi per kata dan kemudian data tersebut juga dapat dilakukan pembuatan kamus kata. Tabel 4.10 di bawah menampilkan hasil data setiap tweet yang sudah ditokenisasi.

Tabel 4. 10 Hasil Setelah Tokenisasi

id	tweet	hasil tokenisasi
3344	partisipasi pemilu bentuk nyata libat dalam bangun negara	partisipasi, pemilu, bentuk, nyata, libat, dalam, bangun, negara
3345	perintah bukan pihak bukan pihak kait sengketa pemilu	perintah, bukan, pihak, bukan, pihak, kait, sengketa, pemilu
3346	harus upaya paham baik penting nasional pemilu tidak bawa arus provokasi politik	harus, upaya, paham, baik, penting, nasional, pemilu, tidak, bawa, arus, provokasi, politik

Setelah selesai mentokenisasi setiap data, langkah selanjutnya adalah membuat kamus kata dari semua data *tweet* yang ada. Sebagai contoh, tabel 4.11 di bawah hanya menampilkan kamus data yang didapatkan dari ketiga data yang digunakan sebagai contoh.

Tabel 4. 11 Kamus Kata

Kamus Kata
arus, baik, bangun, bawa, bentuk, bukan, dalam, kait, libat, nasional, negara, nyata, partisipasi, paham, pemilu, penting, perintah, pihak, politik, provokasi, sengketa, tidak, upaya

b. Inisialisasi Distribusi Topik

Setelah dilakukan tokenisasi untuk setiap *tweet* dan pembuatan kamus kata. Langkah selanjutnya adalah menginisialisasi distribusi topik yang dimana variabel untuk perhitungan topik seperti *document topic counts*, *topic word counts*, dan *topic counts* diinisialisasi. Penentuan jumlah topik yang akan dicari juga ditentukan pada tahapan ini. Pada penelitian ini jumlah topik yang ingin dicari adalah 10 topik.

Kemudian setiap kata pada setiap data dimasukkan topik secara acak sebagai inisialisasi. Tabel 4.12 di bawah menampilkan contoh hasil dari setiap kata pada tweet ke-3344 setelah dimasukan topik secara acak.

Tabel 4. 12 Hasil Inisialisasi Topik Pada Kata

id	kata	topik
3344	partisipasi	7
	pemilu	6
	bentuk	3
	nyata	1
	libat	7
	dalam	0
	bangun	6
	negara	6

Setelah selesai melakukan inisialisasi topik pada setiap kata dilakukan perhitungan untuk menghitung jumlah topik pada setiap kata di dalam dokumen. Tabel 4.13 Di bawah menampilkan distribusi untuk *document topic counts* pada data ke-3344.

Tabel 4. 13 Hasil Perhitungan Document Topic Counts

id	Topik	Jumlah
3344	0	1
	1	1
	3	1
	6	3
	7	2

Kemudian dilakukan perhitungan untuk menghitung jumlah topik pada setiap kata di seluruh data. Tabel 4.14 di bawah menampilkan distribusi untuk *topic word counts* pada setiap kata yang berada di dalam data ke-3344.

Tabel 4. 14 Hasil Perhitungan Topic Word Counts

Topik	partisipasi	pemilu	bentuk	nyata	libat	dalam	bangun	negara
0	0	53	0	0	0	1	0	8
1	0	56	0	2	0	0	0	4
2	0	40	1	3	1	0	1	8
3	0	69	1	1	1	0	1	3
4	0	48	1	1	0	0	0	2
5	0	69	0	4	1	1	2	1
6	0	70	0	4	1	1	2	4

7	1	50	0	0	1	0	0	3
8	0	55	0	0	0	0	1	5
9	0	53	0	2	2	0	1	1

Terakhir dilakukan perhitungan untuk menghitung jumlah sebuah topik di seluruh data. Tabel 4.15 di bawah menampilkan distribusi untuk *topic counts*.

Tabel 4. 15 Hasil Perhitungan *Topic Counts*

Topic	Count
0	873
1	844
2	895
3	881
4	872
5	845
6	913
7	848
8	912
9	879

c. Iterasi *Gibbs Sampling*

Setelah setiap data selesai dilakukan inisialisasi distribusi topik secara acak dan inisialisasi *document topic counts*, *topic word counts*, dan *topic counts* telah dilakukan. Langkah selanjutnya adalah tahap iterasi *gibbs sampling*. Pada iterasi ini terdapat beberapa langkah diantaranya adalah menghapus kata serta jumlah dari kata yang terpilih, menghitung bobot topik untuk semua kata, memilih topik baru, dan menambah kata serta jumlah dari kata yang terpilih. Pada penelitian ini iterasi maksimum yang digunakan adalah 2000 iterasi.

Tahapan yang pertama dilakukan adalah penghapusan kata yang terpilih pada iterasi dan mengurangi jumlah di setiap *counts* yang ada. Misal kata ‘pemilu’ akan dihilangkan dari setiap data *counts* yang ada. Karena pada inisialisasi kata ‘pemilu’ diidentifikasi sebagai topik 6 maka jumlah topik 6 pada *document topic counts* dikurang 1 dan menjadi 2. Pada *topic word counts*, kata pemilu pada topik 6 dikurang 1 sehingga menjadi 69. Pada *topic counts*, topik 6 dikurang 1 sehingga menjadi 912. Tabel 4.16 di bawah menampilkan hasil pengurangan kata untuk *document topic counts*, *topic word counts*, dan *topic counts*.

Tabel 4. 16 Hasil Pengurangan Kata

Document Topic Counts		
id	Topik	Jumlah
3344	6	2
Topic Word Counts		
id	Topik	Jumlah
3344	6	69
Topic Counts		
id	Topik	Jumlah
3344	6	912

Setelah dilakukan pengurangan, tahapan selanjutnya adalah pembobotan setiap topik yang ada. Perhitungan dilakukan menggunakan data dari *document topic counts*, *topic word counts*, dan *topic counts*. Tabel 4.17 di bawah menampilkan hasil pembobotan setiap topik yang ada untuk kata ‘pemilu’.

Tabel 4. 17 Hasil Pembobotan Kata Pemilu

Topik	Perhitungan Bobot	Hasil
0	$\frac{53 + 0.1}{873 + 2508 \times 0.1} \times \frac{1 + 0.1}{7 + 10 \times 0.1}$	0.006496
1	$\frac{56 + 0.1}{844 + 2508 \times 0.1} \times \frac{1 + 0.1}{7 + 10 \times 0.1}$	0.007045
2	$\frac{40 + 0.1}{895 + 2508 \times 0.1} \times \frac{0 + 0.1}{7 + 10 \times 0.1}$	0.000437
3	$\frac{69 + 0.1}{881 + 2508 \times 0.1} \times \frac{1 + 0.1}{7 + 10 \times 0.1}$	0.008394
4	$\frac{48 + 0.1}{872 + 2508 \times 0.1} \times \frac{0 + 0.1}{7 + 10 \times 0.1}$	0.000535
5	$\frac{69 + 0.1}{845 + 2508 \times 0.1} \times \frac{0 + 0.1}{7 + 10 \times 0.1}$	0.000788
6	$\frac{69 + 0.1}{912 + 2508 \times 0.1} \times \frac{2 + 0.1}{7 + 10 \times 0.1}$	0.015599
7	$\frac{50 + 0.1}{848 + 2508 \times 0.1} \times \frac{2 + 0.1}{7 + 10 \times 0.1}$	0.011968
8	$\frac{55 + 0.1}{912 + 2508 \times 0.1} \times \frac{0 + 0.1}{7 + 10 \times 0.1}$	0.000592
9	$\frac{53 + 0.1}{879 + 2508 \times 0.1} \times \frac{0 + 0.1}{7 + 10 \times 0.1}$	0.000587

Setelah pembobotan setiap topik dilakukan, dilakukan *weighted random sampling* untuk mengambil bobot yang paling sesuai dengan topik. *Weighted random sampling* akan memilih salah satu bobot dari seluruh bobot yang ada. Bobot yang nilainya lebih besar memiliki kemungkinan terpilih lebih besar dibandingkan bobot yang lebih kecil. Langkah pertama dari *weighted random sampling* adalah dengan menjumlahkan seluruh bobot dan juga menentukan nilai acak.

$$\begin{aligned} \text{Total bobot} &= 0.006496 + 0.007045 + 0.000437 \\ &\quad + 0.008394 + 0.000535 + 0.000788 \\ &\quad + 0.015599 + 0.011968 + 0.000592 \\ &\quad + 0.000587 \end{aligned}$$

$$\text{Total bobot} = 0.052441$$

Pada ilustrasi ini nilai acak yang dipilih bernilai 0.7321101202045186. Kemudian nilai acak tersebut dikalikan dengan total bobot.

$$\text{Nilai acak} = 0.7321101202045186 \times 0.052441$$

$$\text{Nilai acak} = 0.03839258681364516$$

Setelah nilai acak sudah ditentukan, langkah berikutnya adalah melakukan pengecekan untuk setiap bobot apakah terpilih atau tidak. Dilakukan perhitungan pengurangan nilai acak dengan bobot yang dihitung. Apabila hasil akhirnya nilai tersebut di bawah 0 atau negatif, maka bobot tersebut terpilih. Ilustrasi perhitungan *weighted random sampling* dapat dilihat pada tabel 4.18 di bawah.

Tabel 4. 18 Hasil Weighted Random Sampling

Topik	Perhitungan	Hasil
0	0.032857764659085194 – 0.006496	0.026361764659085192
1	0.026361764659085192 – 0.007045	0.019316764659085193
2	0.019316764659085193 – 0.000437	0.018879764659085193
3	0.018879764659085193 – 0.008394	0.010485764659085193
4	0.010485764659085193 – 0.000535	0.009950764659085192
5	0.009950764659085192 – 0.000788	0.009162764659085191
6	0.009162764659085191 – 0.015599	-0.006436235340914809

Topik 6 memiliki hasil di bawah dari 0 dan negatif sehingga kata ‘pemilu’ dianggap sebagai topik 6. Kemudian setelah itu dilakukan penambahan kata ke dalam topik yang terpilih yaitu topik 6. Tabel 4.19 menampilkan hasil dari

penambahan kata untuk *document topic counts*, *topic word counts*, dan *topic counts* telah dilakukan.

Tabel 4. 19 Hasil Penambahan Kata

Document Topic Counts		
id	Topik	Jumlah
3344	6	3
Topic Word Counts		
id	Topik	Jumlah
3344	6	70
Topic Counts		
id	Topik	Jumlah
3344	6	913

Kemudian dengan cara yang sama tahapan ini dilakukan sebanyak maksimum iterasi yang dimana pada penelitian ini adalah 2000 iterasi.

d. Hasil Topik

Setelah dilakukan iterasi *gibbs sampling* sebanyak 2000 iterasi didapatkan 10 hasil topik dari data *tweet* yang muncul pada rentang waktu *trending topic*. Tabel 4.20 di bawah ini menampilkan 10 topik yang didapatkan.

Tabel 4. 20 Hasil Topik

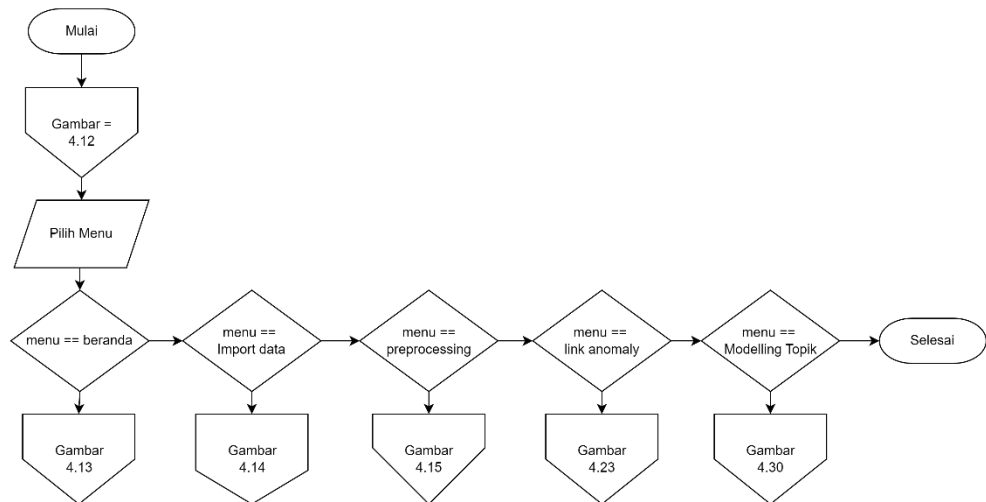
Topik	Kata Kunci
1	pemilu, curang, langgar, bawaslu, pilpres, kpu
2	pemilu, ulang, kalah, adil, jujur, curang, sejarah, jalan, indonesia
3	pemilu, menang, presiden, pilih, suara, ulang, jokowi, prabowo
4	pemilu, curang, bodoh, isu, alih, bikin
5	pemilu, bansos, kemarin
6	pemilu, buzzer, orang, dukung
7	pemilu, partai, jokowi, uang, bilang
8	pemilu, rakyat, negara, suara
9	pemilu, satu, politik, pui, negara, pascapemilu, hasil, jaga
10	mk, pemilu, hasil, uu, putus, kpu, sengketa, gugat, atur, sidang

4.3. Flowchart

Flowchart adalah representasi visual dari rangkaian proses atau algoritma, yang digambarkan menggunakan simbol-simbol grafis. Berikut adalah deskripsi tentang aplikasi flowchart dalam penelitian ini.

4.3.1. Flowchart Proses Keseluruhan Sistem

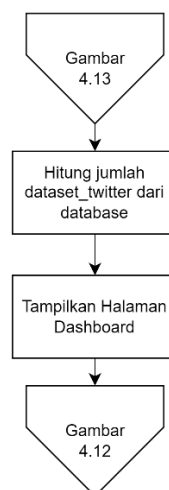
Flowchart ini menggambarkan proses pemilihan menu dalam sebuah aplikasi yang telah dikembangkan. Proses pemilihan menu tersebut dijelaskan melalui *Flowchart* yang ditampilkan pada gambar 4.12.



Gambar 4. 12 Flowchart Proses Keseluruhan Sistem

4.3.1. Flowchart Menu Dashboard

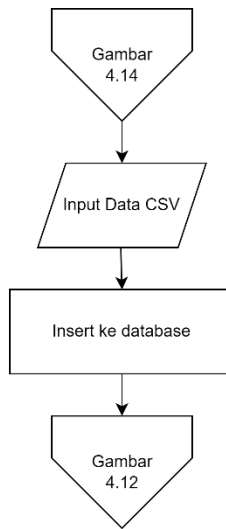
Flowchart ini menjelaskan tentang proses menu *dashboard* pada aplikasi yang dibuat. *Flowchart* Menu *Dashboard* dapat dilihat pada gambar 4.13 di bawah.



Gambar 4. 13 Flowchart Menu Dashboard

4.3.2. Flowchart Proses Import Data

Flowchart ini menjelaskan proses dari *Import Data*.

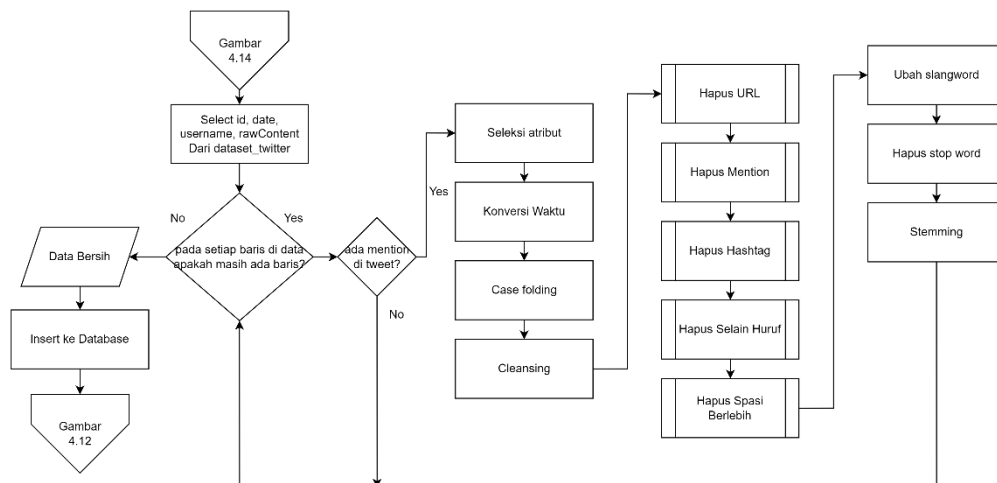


Gambar 4. 14 Flowchart Proses Import Data

Gambar 4.14 menjelaskan proses *Import Data* yang dimana tahapan pertama adalah melakukan input data berupa file CSV. File CSV tersebut kemudian akan dimasukan ke dalam database yaitu *table dataset_twitter*.

4.3.3. Flowchart Proses Preprocessing

Flowchart ini menjelaskan proses dari *preprocessing*. Proses dimulai dengan pengambilan data dari *table dataset_twitter* di database kemudian diakhiri dengan memasukan hasil data yang sudah bersih ke dalam *table dataset_preprocessed*.

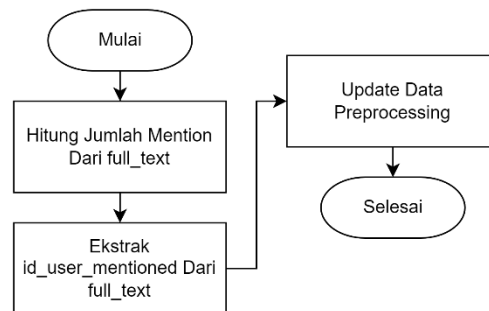


Gambar 4. 15 Flowchart Proses Preprocessing

Gambar 4.14 menjelaskan proses *preprocessing* yang diawali dengan pengambilan data dari *table dataset_twitter* di database. Kemudian data tersebut akan melewati beberapa tahapan *preprocessing* seperti seleksi atribut, konversi waktu, *case folding*, *cleansing* yang didalamnya terdapat beberapa proses lagi, ubah *slang word*, hapus *stop word*, dan *stemming*. Setelah data tersebut melewati semua tahapan *preprocessing* maka akan didapatkan data yang sudah bersih. Kemudian data yang sudah bersih tersebut akan dimasukkan ke dalam *table dataset_preprocessed*.

a. *Flowchart* Proses Seleksi Atribut

Flowchart ini menjelaskan proses dari seleksi atribut pada tahapan proses *preprocessing*.

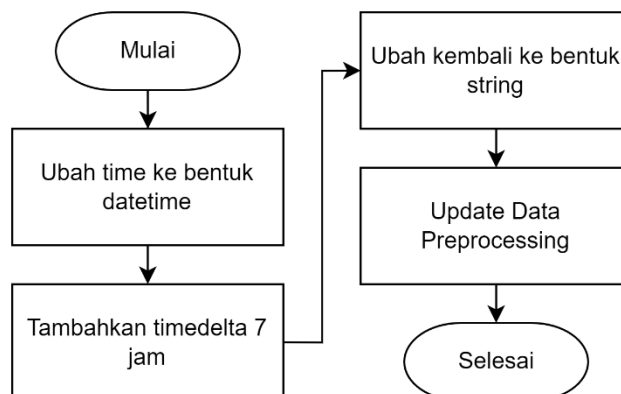


Gambar 4. 16 *Flowchart* Proses Seleksi Atribut

Gambar 4.16 menjelaskan proses dari seleksi atribut. Tahapan ini diawali dengan melakukan perhitungan jumlah mention di dalam *tweet*. Kemudian dilakukan juga pengestrakan id dari *user* yang dimention. Setelah semua data sudah dipilih maka data *preprocessing* akan diperbarui sedemikian rupa.

b. *Flowchart* Proses Konversi Waktu

Flowchart ini menjelaskan proses dari konversi waktu pada tahapan proses *preprocessing*.

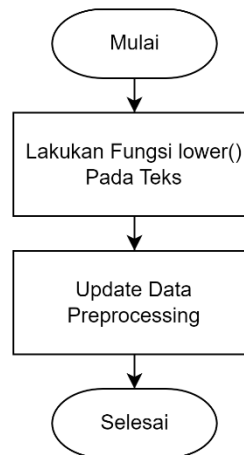


Gambar 4. 17 *Flowchart* Proses Konversi Waktu

Gambar 4.17 menjelaskan proses konversi waktu. Proses ini diawali dengan mengubah tipe data waktu pada dataset `twitter` ke dalam bentuk `'datetime'`. Kemudian data yang telah diubah akan ditambahkan 7 jam kedepan. Setelah itu data yang sudah diubah ke tipe data `'datetime'` akan kembali diubah ke tipe data `'string'`. Setelah rangkaian proses konversi waktu dilakukan, maka data *preprocessing* akan diperbarui.

c. *Flowchart Proses Case Folding*

Flowchart ini menjelaskan proses dari *case folding* pada tahapan proses *preprocessing*.

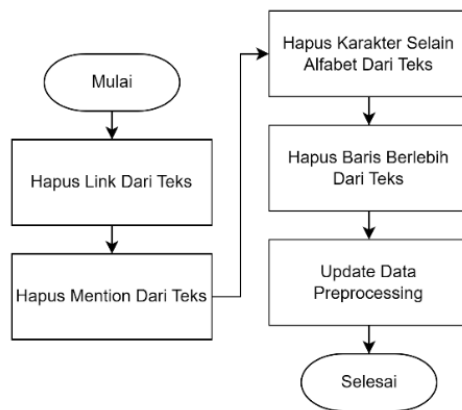


Gambar 4. 18 Flowchart Proses Case Folding

Gambar 4.18 menjelaskan proses *case folding*. Proses ini akan mengubah teks menggunakan huruf kecil sehingga membuat data menjadi konsisten. Proses ini melibatkan penggunaan fungsi `'lower()'` dari python. Setelah teks sudah menggunakan huruf kecil semua, data *preprocessing* akan diperbarui. Penggunaan huruf kecil dalam seluruh teks bertujuan untuk menghilangkan perbedaan yang timbul akibat penggunaan huruf besar, sehingga data menjadi konsisten. Setelah teks sudah dikonversi ke dalam huruf kecil semua, data *preprocessing* akan diperbarui dan dilanjutkan ke tahapan selanjutnya. Dengan melakukan *case folding*, teks menjadi lebih mudah untuk dianalisis, karena perbedaan yang disebabkan oleh kapitalisasi dapat diabaikan. Proses ini merupakan langkah awal yang penting dalam *preprocessing* teks sebelum masuk ke tahap-tahap berikutnya seperti tokenisasi, penghapusan tanda baca, dan yang lainnya.

d. *Flowchart Proses Cleansing*

Flowchart ini menjelaskan proses dari *cleansing* pada tahapan proses *preprocessing*.

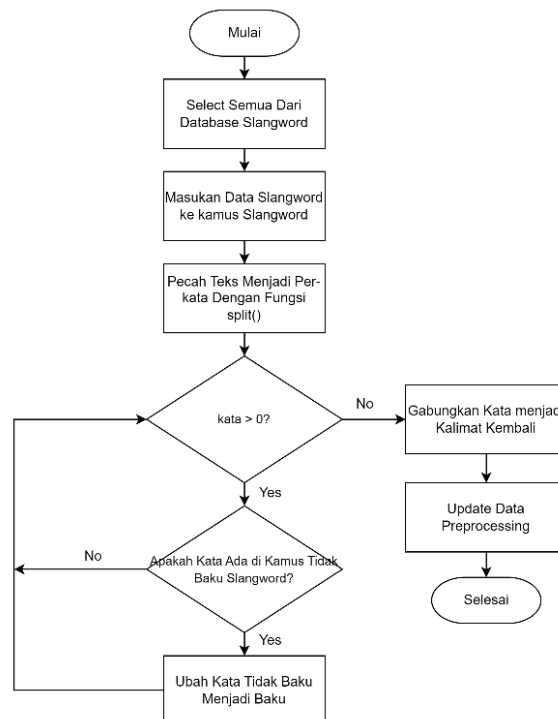


Gambar 4. 19 Flowchart Proses Cleansing

Gambar 4.19 menjelaskan proses *cleansing*. Proses ini terdiri dari beberapa tahapan dan diawali dengan proses menghapus link dari teks. Setelah itu *mention* yang ada di teks akan dihapus dan semua karakter selain alfabet akan dihapus dari teks. Terakhir, baris yang berlebih juga akan dihapus agar membuat data menjadi lebih bersih. Jika tahapan *cleasing* sudah dilakukan maka data preprocessing akan diperbarui.

e. *Flowchart Proses Ubah Slang Word*

Flowchart ini menjelaskan proses dari ubah *slang word* pada tahapan proses *preprocessing*.

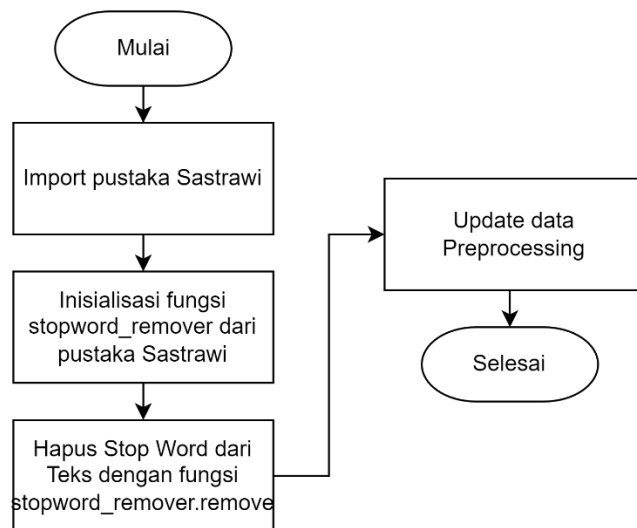


Gambar 4. 20 Flowchart Proses Ubah Slang Word

Gambar 4.20 menjelaskan proses dari pengubahan *slang word*. Pada tahapan ini setiap kata yang merupakan kata *slang word* akan diubah dari yang tidak baku menjadi kata baku. Langkah pertama adalah mengambil data kata baku dan tidak baku dari *table* 'slangword' yang ada di dalam database. Setelah itu data *slang word* yang sudah diambil akan diubah bentuknya menjadi kamus agar memudahkan proses pengubahan. Kemudian teks akan dipecah menjadi kata per kata. Tahap yang berikutnya adalah melakukan perulangan untuk setiap kata di dalam teks, jika ada kata tidak baku di dalam teks maka akan diubah menjadi kata baku. Jika semua teks pada data sudah dilakukan proses ubah *slang word* maka data *preprocessing* akan diperbarui.

f. *Flowchart Proses Hapus Stop Word*

Flowchart ini menjelaskan proses dari hapus *stop word* pada tahapan proses *preprocessing*.

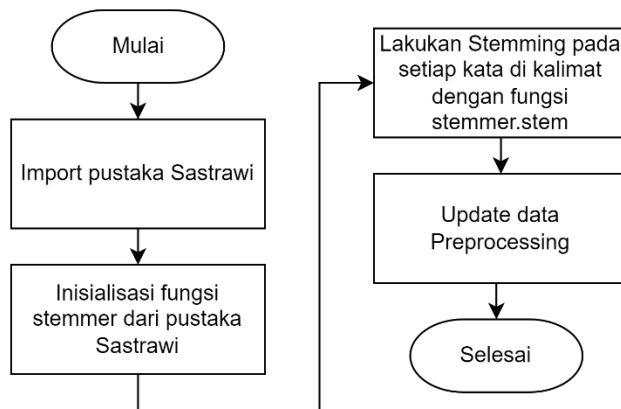


Gambar 4. 21 Flowchart Proses Hapus Stop Word

Gambar 4.21 menjelaskan proses penghapusan *stop word*. Pada setiap teks di dalam data, akan dilakukan penghapusan *stop word* agar data menjadi lebih bersih dari kata yang tidak memiliki makna. Langkah pertama dari proses ini adalah mengimport pustaka Sastrawi. Kemudian perlu untuk menginisialisasi fungsi untuk menghapus *stop word* yang sudah disediakan oleh pustaka Sastrawi. Setelah itu barulah dapat dilakukan proses penghapusan *stop word*. Jika semua data sudah dilakukan penghapusan *stop word* maka data *preprocessing* akan diperbarui.

g. *Flowchart Proses Stemming*

Flowchart ini menjelaskan proses dari *stemming* pada tahapan proses *preprocessing*.

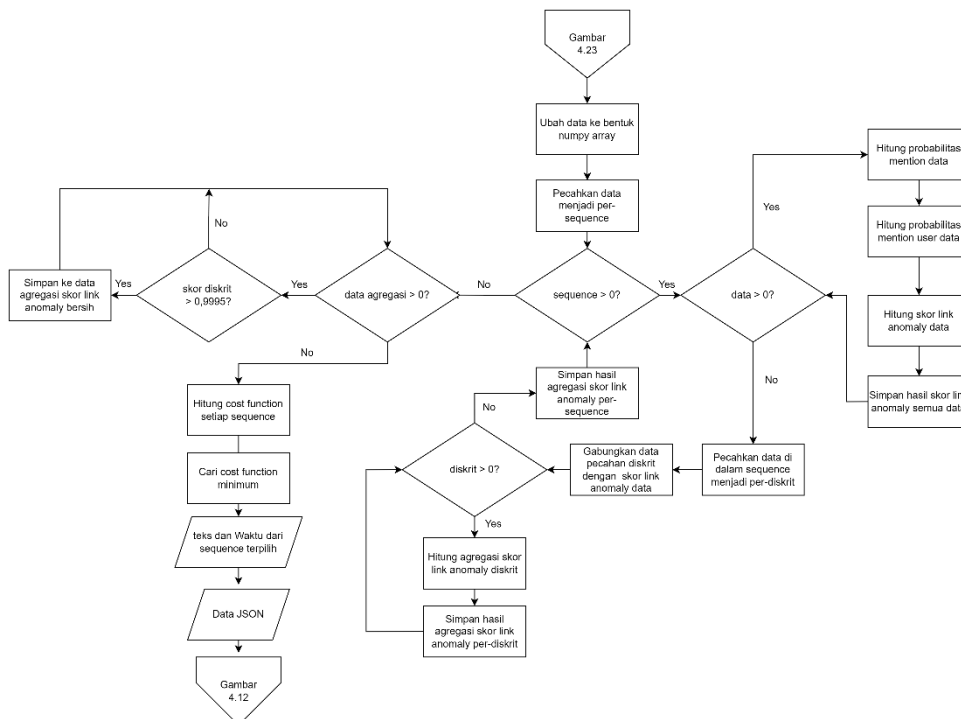


Gambar 4. 22 Flowchart Proses Stemming

Gambar 4.22 menjelaskan proses *stemming*. Proses ini dilakukan menggunakan pustaka Sastrawi. Langkah pertama adalah mengimport pustaka Sastrawi, kemudian inisialisasi fungsi *stemming* yang sudah disediakan oleh pustaka Sastrawi. Setelah itu dilakukan proses *stemming* pada setiap teks yang ada di dalam data. Jika semua data sudah dilakukan proses *stemming* maka data *preprocessing* akan diperbarui.

4.3.4. Flowchart Proses Link Anomaly

Flowchart ini menjelaskan proses *link anomaly*.

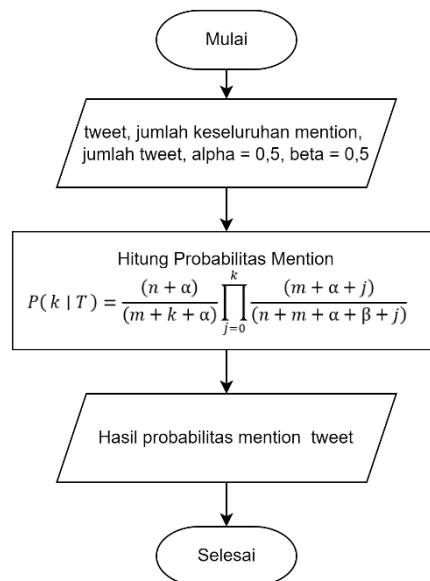


Gambar 4. 23 Flowchart Proses Link Anomaly

Gambar 4.23 menjelaskan proses menyeluruh dari tahapan *link anomaly*. Proses ini diawali dengan pengubahan data ke dalam bentuk *array*. Setelah itu data akan dipecah menjadi beberapa *sequence*. Akan dilakukan perulangan untuk setiap *sequence* dan setiap data yang dimana akan dilakukan perhitungan probabilitas *mention* data, probabilitas *mention* user, dan skor *link anomaly*. Setelah itu data yang sudah dipecah menjadi beberapa *sequence* akan dipecah lagi menjadi beberapa diskrit. Kemudian dilakukan perulangan di dalam diskrit yang sudah dibuat untuk melakukan perhitungan agregasi skor *link anomaly*. Data agregasi skor *link anomaly* yang sudah didapatkan akan dilakukan perulangan untuk memilah agregasi skor yang memiliki nilai skor diskrit di atas 0.99995. Berikutnya akan dilakukan perhitungan *cost function* setiap *sequence* dan dan setelahnya akan dipilih *sequence* yang memiliki nilai *cost function* terendah. Terakhir akan diambil semua *tweet* yang berada di *sequence* yang memiliki nilai *cost function* terendah dan kemudian akan diubah serta disimpan ke dalam json.

a. *Flowchart* Proses Hitung Probabilitas Sejumlah Mention

Flowchart ini menjelaskan proses perhitungan probabilitas sejumlah *mention*.



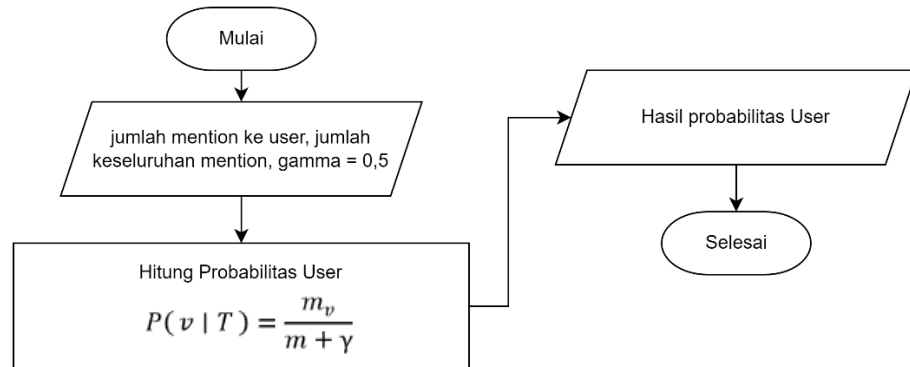
Gambar 4. 24 *Flowchart* Proses Hitung Probabilitas Sejumlah Mention

Gambar 4.24 menjelaskan proses perhitungan probabilitas sejumlah *mention*. Pada proses ini data input yang digunakan berupa *tweet*, jumlah keseluruhan *mention*, jumlah *tweet*, nilai *alpha* dengan nilai *default* adalah 0.5, serta nilai *beta* dengan nilai *default* 0.5. Setelah memasukkan data input yang dibutuhkan maka akan dilakukan perhitungan probabilitas

mention. Hasil dari perhitungan ini akan disimpan ke dalam *variabel* untuk digunakan pada perhitungan berikutnya.

b. *Flowchart* Proses Hitung Probabilitas User

Flowchart ini menjelaskan proses perhitungan probabilitas *user*.

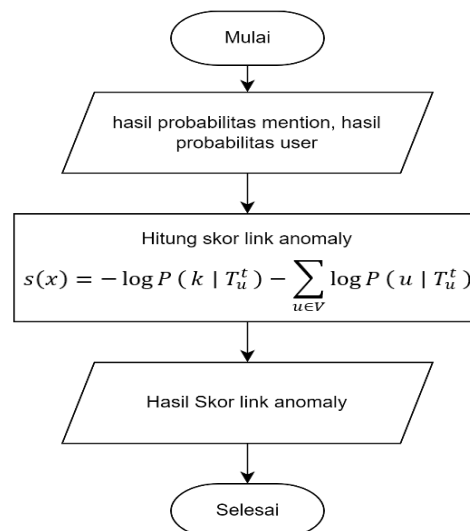


Gambar 4. 25 Flowchart Hitung Probabilitas User

Gambar 4.25 menjelaskan proses perhitungan probabilitas *user*. Data input yang dibutuhkan adalah jumlah *mention* ke *user*, jumlah keseluruhan *mention*, dan nilai *gamma* yang memiliki nilai default 0.5. Dengan menggunakan data input yang dibutuhkan maka akan dilakukan perhitungan probabilitas *user* dan hasilnya akan disimpan ke dala *variabel* untuk digunakan pada perhitungan berikutnya.

c. *Flowchart* Proses Hitung Skor Link Anomaly

Flowchart ini menjelaskan proses perhitungan skor *link anomaly*.

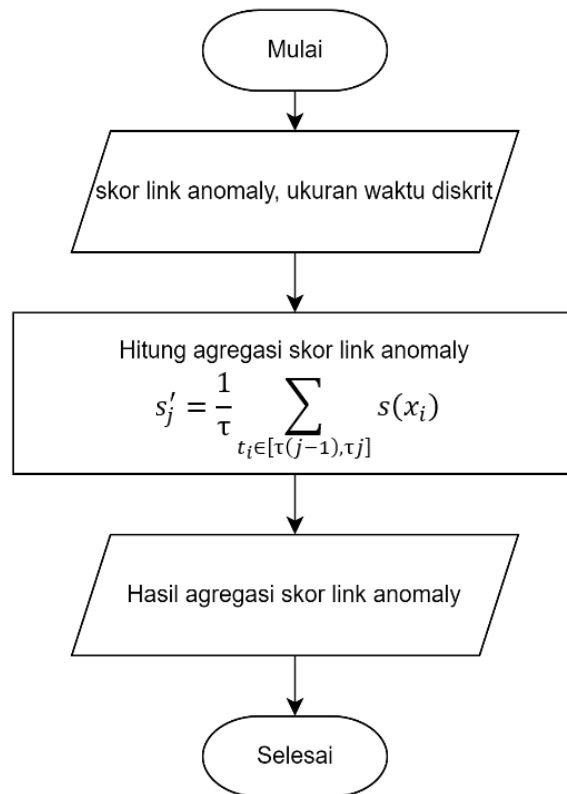


Gambar 4. 26 Flowchart Hitung Skor Link Anomaly

Gambar 4.26 menjelaskan proses perhitungan skor *link anomaly*. Data input yang dibutuhkan adalah hasil perhitungan probabilitas *mention* dan *user*. Setelah menerima input tersebut maka akan dilakukan perhitungan skor *link anomaly* yang dimana hasilnya akan disimpan ke dalam *variabel* untuk perhitungan berikutnya.

d. *Flowchart* Proses Agregasi Skor *Link Anomaly*

Flowchart ini menjelaskan proses agregasi skor *link anomaly*.

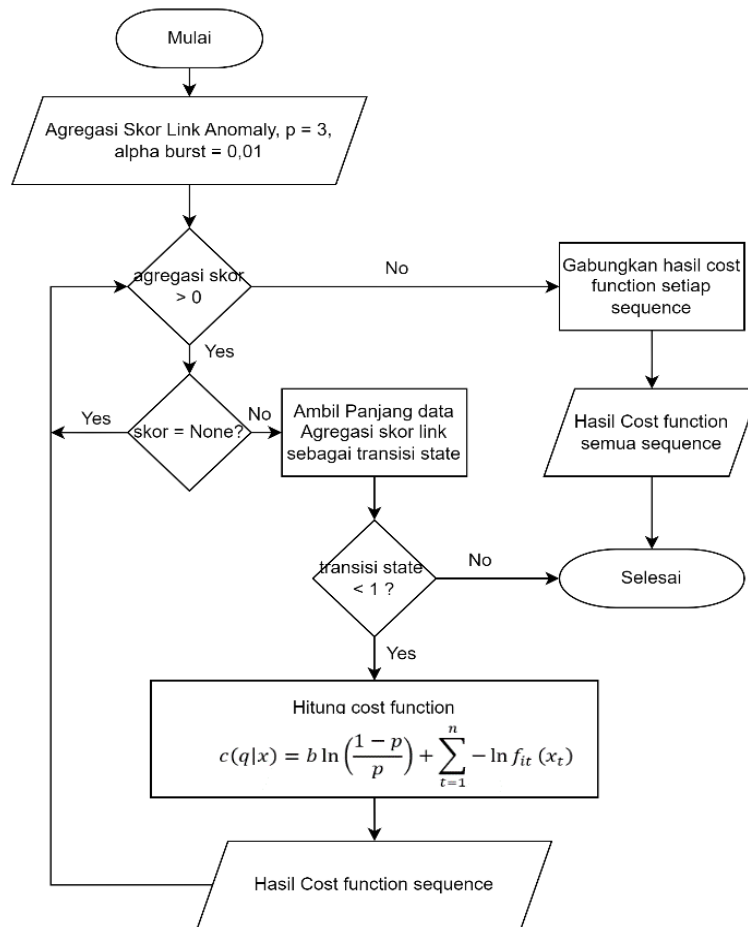


Gambar 4. 27 Flowchart Agregasi Skor Link Anomaly

Gambar 4.27 menjelaskan proses perhitungan agregasi skor *link anomaly*. Data input yang dibutuhkan adalah skor *link anomaly* dan ukuran waktu setiap diskrit. Setelah data input disiapkan maka akan dilakukan proses perhitungan agregasi skor *link anomaly*. Kemudian hasil dari perhitungan tersebut akan disimpan ke dalam sebuah *variabel* untuk digunakan pada tahap selanjutnya.

e. *Flowchart* Proses Hitung *Cost Function*

Flowchart ini menjelaskan proses perhitungan *cost function*.

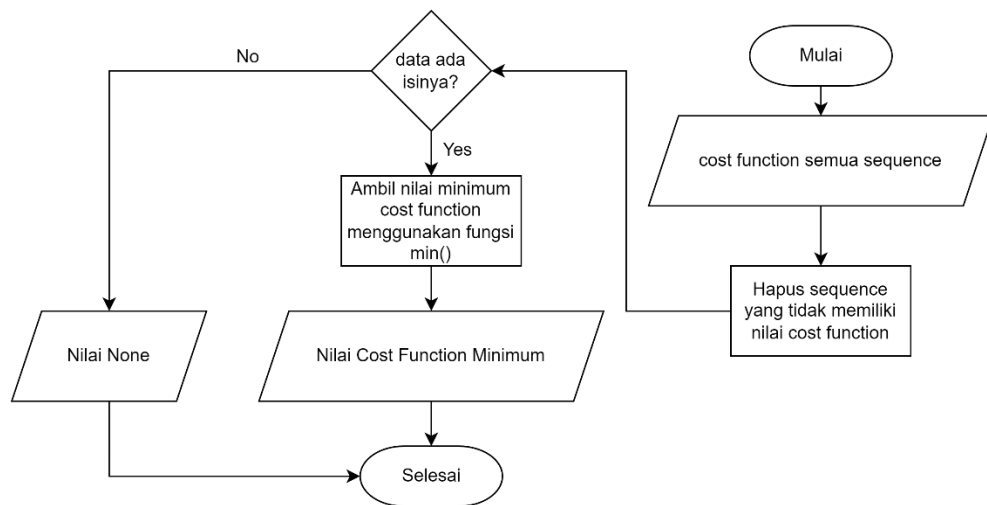


Gambar 4. 28 Flowchart Hitung Cost Function

Gambar 4.28 menjelaskan proses perhitungan *cost function*. Data input yang dibutuhkan adalah hasil agregasi skor *link anomaly*, nilai probabilitas transisi yang bernilai 0.3, dan nilai *alpha burst* dengan nilai 0.01. Kemudian akan dilakukan perulangan pada setiap skor agregasi yang ada. Jika skor agregasi tidak memiliki nilai maka agregasi tersebut akan dilewati. Selanjutnya akan diambil panjang data setiap agregasi skor *link anomaly* sebagai *transisi state*. Jika nilai *transisi state* di bawah 1 maka perhitungan tidak bisa dilanjutkan. Jika nilai *transisi state* di atas 1 maka akan dilakukan perhitungan *cost function* dan kemudian hasilnya akan disimpan ke dalam *variabel* untuk tahap berikutnya.

f. *Flowchart* Proses Tentukan Nilai *Cost Function* Minimum

Flowchart ini menjelaskan proses penentuan nilai *cost function* minimum.

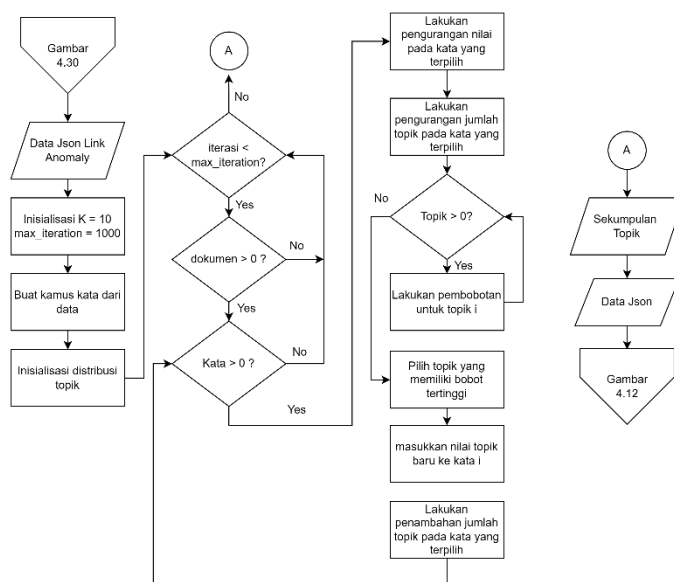


Gambar 4. 29 Flowchart Menentukan Nilai Cost Function Minimum

Gambar 4.29 menjelaskan proses penentuan nilai *cost function* minimum. Data input yang dibutuhkan adalah hasil perhitungan *cost function* setiap *sequence*. Langkah pertama akan dilakukan pengecekan apakah data ada isinya atau tidak. Jika data tidak ada isinya maka perhitungan tidak dapat dilakukan dan akan mengembalikan nilai *None*. Jika data ada maka akan dilakukan penentuan nilai *cost function minimum*. Nilai *cost function* yang terpilih akan disimpan ke dalam variabel untuk proses berikutnya.

4.3.5. Flowchart Proses LDA

Flowchart ini menjelaskan proses LDA.



Gambar 4. 30 Flowchart Proses LDA

Gambar 4.30 menjelaskan proses LDA secara menyeluruh. Data input yang dibutuhkan adalah data json hasil perhitungan *link anomaly*. Langkah pertama adalah melakukan inisialisasi paramter seperti nilai K dan iterasi maksimum. Kemudian dibuatkan kamus kata dari data yang berada di dalam json. Langkah berikutnya adalah melakukan inisialisasi distribusi topik. Perhitungan proses LDA dijalankan melalui beberapa perulangan yang terdiri dari perulangan iterasi maksimum, setiap data, dan setiap kata. Pada iterasi tersebut akan dilakukan pengurangan jumlah kata yang terpilih pada distribusi topik. Kemudian dilakukan perulangan untuk setiap topik dan akan dilakukan pembobotan untuk setiap topik. Setelah melakukan.

4.4. Algoritma

Algoritma merupakan kumpulan langkah atau prosedur yang dijabarkan dalam bentuk teks. Ini juga berfungsi sebagai representasi dari *flowchart* yang telah dirancang sebelumnya. Berikut adalah penjelasan mengenai algoritma berdasarkan *flowchart* yang telah dikembangkan.

4.4.1. Algoritma Proses Keseluruhan Sistem

Algoritma ini mendeskripsikan proses sistem yang telah dibuat pada *flowchart* sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam sistem. Berikut adalah penjabaran dari algoritma keseluruhan proses sistem.

Algoritma 4. 1 Proses Keseluruhan Sistem

```

1  Mulai
2  Tampilkan Menu Dashboard (DB)
3  Pilih Menu
4  if menu == beranda:
5      Tampilkan Menu Dashboard (DB)
6  elif menu == Import data:
7      Tampilkan Menu Import Data (ID)
8  elif menu == preprocessing:
9      Tampilkan Menu Preprocessing (PR)
10 elif menu == link anomaly:
11     Tampilkan Menu Link Anomaly (LA)
12 elif menu == Modelling Topik:
13     Tampilkan Menu Modelling Topik (MT)
14 end if
15 Selesai

```

4.4.1. Algoritma Menu *Dashboard*

Algoritma ini mendeskripsikan proses menu *dashboard* yang telah dibuat pada *flowchart* sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses menu *dashboard*. Berikut adalah penjabaran dari algoritma proses menu *dashboard*.

Algoritme 4. 2 Menu *Dashboard*

1	Mulai
2	Tampilkan Menu Dashboard (DB)
3	Hitung jumlah dataset Platform X dari database
4	Tampilkan Halaman Dashboard
5	Selesai

4.4.2. Algoritma Proses *Import* Data

Algoritma ini mendeskripsikan proses *import* data yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses *import* data. Berikut adalah penjabaran dari algoritma proses *import* data.

Algoritma 4. 3 Proses *Import* Data

1	mulai
2	Input Data CSV
3	Insert ke database
4	selesai

4.4.3. Algoritma Proses *Preprocessing*

Algoritma ini mendeskripsikan proses *preprocessing* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses *preprocessing*. Berikut adalah penjabaran dari algoritma proses *preprocessing*.

Algoritma 4. 4 Proses *Preprocessing*

1	mulai
2	Select id, date, username, rawContent dari dataset_twitter
3	Perulangan: Pada setiap baris data
4	Seleksi atribut
5	Konversi Waktu
6	Case folding
7	Cleansing
8	Hapus URL
9	Hapus Mention
10	Hapus Hashtag
11	Hapus Selain Huruf
12	Hapus Spasi Berlebih
13	Ubah slangword
14	Hapus stop word
15	Stemming
16	end Perulangan
17	Data Bersih
18	Insert ke Database
19	selesai

a. Algoritma Proses Seleksi Atribut

Algoritma ini mendeskripsikan proses seleksi atribut pada *preprocessing* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses seleksi atribut. Berikut adalah penjabaran dari algoritma proses seleksi atribut.

Algoritma 4. 5 Proses Seleksi Atribut

1	mulai
2	Hitung Jumlah Mention Dari full_text
3	Ekstrak id_user_mentioned Dari full_text
4	Update Data Preprocessing
5	selesai

b. Algoritma Proses Konversi Waktu

Algoritma ini mendeskripsikan proses konversi waktu pada *preprocessing* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses konversi waktu. Berikut adalah penjabaran dari algoritma proses konversi waktu.

Algoritma 4. 6 Proses Konversi Waktu

1	mulai
2	Ubah data time ke bentuk datetime
3	Tambahkan timedelta 7 jam
4	Ubah data kembali ke bentuk string
5	Update Data Preprocessing
6	selesai

c. Algoritma Proses *Case Folding*

Algoritma ini mendeskripsikan proses *case folding* pada *preprocessing* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses *case folding*. Berikut adalah penjabaran dari algoritma proses *case folding*.

Algoritma 4. 7 Proses *Case Folding*

1	mulai
2	Lakukan Fungsi lower() pada Teks
3	Update Data Preprocessing
4	selesai

d. Algoritma Proses *Cleansing*

Algoritma ini mendeskripsikan proses *cleansing* pada *preprocessing* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses *cleansing*. Berikut adalah penjabaran dari algoritma proses *cleansing*.

Algoritma 4. 8 Proses *Cleansing*

```
1  mulai
2  Hapus Link Dari Teks
3  Hapus Mention Dari Teks
4  Hapus Karakter Selain Alfabet Dari Teks
5  Hapus Baris Berlebih Dari Teks
6  Update Data Preprocessing
7  selesai
```

e. Algoritma Ubah *Slang Word*

Algoritma ini mendeskripsikan proses ubah *slang word* pada *preprocessing* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses ubah *slang word*. Berikut adalah penjabaran dari algoritma proses ubah *slang word*.

Algoritma 4. 9 Proses Ubah *Slang Word*

```
1  mulai
2  Select Semua Dari Database Slangword
3  Masukkan Data Slangword ke kamus Slangword
4  Pecah Teks Menjadi Perkata Dengan Fungsi
   split()
5  Perulangan: Selama masih ada kata di dalam
   teks (kata > 0)
6     if kata ada di Kamus Tidak Baku
7       Ubah Kata Tidak Baku Menjadi Baku
8     else
9       Lanjutkan tanpa perubahan
10    endif
11  end Perulangan
12  Gabungkan Kata menjadi Kalimat Kembali
13  Update Data Preprocessing
14  selesai
```

f. Algoritma Hapus *Stop Word*

Algoritma ini mendeskripsikan proses hapus *stop word* pada *preprocessing* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses hapus *stop word*. Berikut adalah penjabaran dari algoritma proses hapus *stop word*.

Algoritma 4. 10 Proses Hapus *Stop Word*

```
1  mulai
2  Import pustaka Sastrawi
3  Inisialisasi fungsi stopwords_remover dari
   pustaka Sastrawi
4  Hapus Stop Word dari Teks dengan fungsi
   stopwords_remover.remove()
5  Update data Preprocessing
6  selesai
```

g. Algoritma Proses *Stemming*

Algoritma ini mendeskripsikan proses *stemming* pada *preprocessing* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses *stemming*. Berikut adalah penjabaran dari algoritma proses *stemming*.

Algoritma 4. 11 Proses *Stemming*

```
1  mulai
2  Import pustaka Sastrawi
3  Inisialisasi fungsi stemmer dari pustaka
   Sastrawi
4  Lakukan Stemming pada setiap kata di kalimat
   dengan fungsi stemmer.stem()
5  Update data Preprocessing
6  selesai
```

4.4.4. Algoritma Proses *Link Anomaly*

Algoritma ini mendeskripsikan proses *link anomaly* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses *link anomaly*. Berikut adalah penjabaran dari algoritma proses *link anomaly*.

Algoritma 4. 12 Proses *Link Anomaly*

```
1  Mulai
2  Ubah data ke bentuk numpy array
3  Pecahkan data menjadi per-sequence
4  Perulangan (sequence > 0):
5    Perulangan (data > 0):
6      Hitung probabilitas mention data
7      Hitung probabilitas mention user data
8      Hitung skor link anomaly data
9      Simpan hasil skor link anomaly semua      data
10   end Perulangan (data)
11  Pecahkan data di dalam sequence menjadi per-
   diskrit
12  Gabungkan data pecahan diskrit dengan skor link
   anomaly data
13  Perulangan (diskrit > 0):
14    Hitung agregasi skor link anomaly diskrit
15    Simpan hasil agregasi skor link anomaly per-
   diskrit
16  end Perulangan (diskrit)
17  Simpan hasil agregasi skor link anomaly per-
   sequence
18 end Perulangan (sequence)
19 Perulangan (data agregasi > 0):
20   If skor diskrit > 0.9995:
21     Simpan ke data agregasi skor link anomaly bersih
22   end if
23 end Perulangan (data agregasi)
24 Hitung cost function setiap sequence
```

25	Cari cost function minimum
26	Teks dan waktu dari sequence terpilih diambil
27	Data disimpan dalam format JSON
28	Selesai

a. Algoritma Proses Hitung Probabilitas Sejumlah *Mention*

Algoritma ini mendeskripsikan proses hitung probabilitas sejumlah *mention* pada *link anomaly* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses hitung probabilitas sejumlah *mention*. Berikut adalah penjabaran dari algoritma proses hitung probabilitas sejumlah *mention*.

Algoritma 4. 13 Proses Hitung Probabilitas Sejumlah Mention

1	Mulai
2	Input: tweet = [data tweet], jumlah_keseluruhan_mention = [jumlah total mention], jumlah_tweet = [jumlah total tweet], alpha = 0.5, beta = 0.5
3	Hitung Probabilitas Mention
4	Output hasil probabilitas mention tweet
5	Selesai

b. Algoritma Hitung Probabilitas *User*

Algoritma ini mendeskripsikan proses hitung probabilitas *user* pada *link anomaly* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses hitung probabilitas *user*. Berikut adalah penjabaran dari algoritma proses hitung probabilitas *user*.

Algoritma 4. 14 Proses Hitung Probabilitas User

1	Mulai
2	Input: jumlah_mention_ke_user = [input jumlah mention ke user tertentu], jumlah_keseluruhan_mention = [input jumlah total mention], gamma = 0.5
3	Hitung Probabilitas User
4	Output hasil probabilitas User
5	Selesai

c. Algoritma Hitung Skor *Link Anomaly*

Algoritma ini mendeskripsikan proses hitung skor *link anomaly* pada *link anomaly* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses hitung skor *link anomaly*. Berikut adalah penjabaran dari algoritma proses hitung skor *link anomaly*.

Algoritma 4. 15 Proses Algoritma Hitung Skor *Link Anomaly*

```
1  Mulai
2  Input:  hasil probabilitas mention,  hasil
         probabilitas user
3  Hitung skor link anomaly
4  Output hasil skor link anomaly
5  Selesai
```

d. Algoritma Agregasi Skor *Link Anomaly*

Algoritma ini mendeskripsikan proses agregasi skor *link anomaly* pada *link anomaly* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses agregasi skor *link anomaly*. Berikut adalah penjabaran dari algoritma proses agregasi skor *link anomaly*.

Algoritma 4. 16 Proses Agregasi Skor *Link Anomaly*

```
1  Mulai
2  Input: skor_link_anomaly = [array skor link
                             anomaly], ukuran_waktu_diskrit = [periode waktu
                             diskrit untuk agregasi]
3  Hitung agregasi skor link anomaly
4  Output hasil agregasi skor link anomaly
5  Selesai
```

e. Algoritma Hitung *Cost Function*

Algoritma ini mendeskripsikan proses hitung *cost function* pada *link anomaly* yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses hitung *cost function*. Berikut adalah penjabaran dari algoritma proses hitung *cost function*.

Algoritma 4. 17 Proses Hitung *Cost Function*

```
1  Mulai
2  Input: Agregasi Skor Link Anomaly, p = 3,
         alpha_burst = 0.01
3  If agregasi skor > 0:
4      If skor = None:
5          Continue
6      Else:
7          Ambil Panjang data Agregasi skor link
            sebagai transisi state
8          If transisi state < 1:
9              Selesai
10         Else:
11             Hitung cost function
12             Simpan hasil cost function untuk sequence
              tersebut
13         EndIf
14     EndIf
15 EndIf
```

16	Gabungkan hasil cost function setiap sequence
17	Output hasil cost function semua sequence
18	Selesai

f. Algoritma Proses Tentukan Nilai *Cost Function* Minimum

Algoritma 4. 18 Proses Tentukan Nilai *Cost Function* Minimum

1	Mulai
2	Input: cost function semua sequence
3	Hapus sequence yang tidak memiliki nilai cost function
4	IF masih ada data:
5	Ambil nilai minimum cost function menggunakan fungsi min()
6	Output nilai Cost Function Minimum
7	ELSE:
8	Output nilai None
9	EndIf
10	Selesai

4.4.5. Algoritma Proses LDA

Algoritma ini mendeskripsikan proses LDA yang telah dibuat pada flowchart sebelumnya. Algoritma ini memuat langkah-langkah yang terdapat dalam proses LDA. Berikut adalah penjabaran dari algoritma proses LDA.

Algoritma 4. 19 Proses LDA

1	Mulai
2	Input: Data JSON untuk Link Anomaly
3	Inisialisasi: K = 10, max_iteration = 2000
4	Buat kamus kata dari data
5	Inisialisasi distribusi topik
6	Perulangan: dari 1 sampai max_iteration
7	Perulangan: setiap dokumen dalam data
8	Perulangan: setiap kata dalam dokumen
9	Kurangi nilai pada kata yang terpilih
10	Kurangi jumlah topik pada kata yang terpilih
11	Perulangan: setiap topik i
12	Lakukan pembobotan untuk topik i
13	end Perulangan (topik i)
14	Pilih topik dengan bobot tertinggi dan masukkan nilai topik baru ke kata i
15	Tambahkan jumlah topik pada kata yang terpilih
16	end Perulangan (kata dalam dokumen)
17	end Perulangan (dokumen dalam data)
18	end Perulangan (iterasi)
19	Output: Kumpulkan dan simpan hasil topik dalam format Data JSON
20	Selesai

4.5. Pengujian

Pengujian dilakukan untuk menilai dan menganalisis metode yang telah diajukan. Dalam penelitian ini, pengujian dilakukan dengan membandingkan data *ground truth*. *Ground truth* diperoleh dari situs berita valid yang dikumpulkan secara manual dalam rentang waktu terdeteksinya *trending topic* yaitu pada tanggal 31 Maret 2024 sampai 1 April 2024. Serta satu hari setelah terdeteksi *trending topic* yaitu 2 April 2024 karena butuh waktu untuk media berita membuat *headline* yang sesuai dengan *trending topic*. Pertimbangan pemilihan waktu ini adalah karena *trending topic* dapat muncul sebagai hasil dari sebuah berita, dan sebaliknya, berita juga dapat muncul akibat adanya *trending topic* di Platform X.

Data *ground truth* didapatkan secara manual dari portal situs berita kompas yang relevan dengan hasil kata kunci dan waktu kemunculan dari *trending topic*. Kemudian *headline* setiap berita yang dipilih akan diidentifikasi dan dilakukan ekstraksi kata kunci yang mencerminkan isi utama berita tersebut. Kata kunci ini akan digunakan sebagai data *ground truth* untuk kemudian dilakukan evaluasi terhadap kata kunci hasil dari sistem yang telah dibuat.

Pengujian dilakukan menggunakan Persamaan (2.7), (2.8), dan (2.9). Kata kunci pada *ground truth* diambil dari judul dan isi berita. Hasil evaluasi dapat dilihat pada Tabel 4.21:

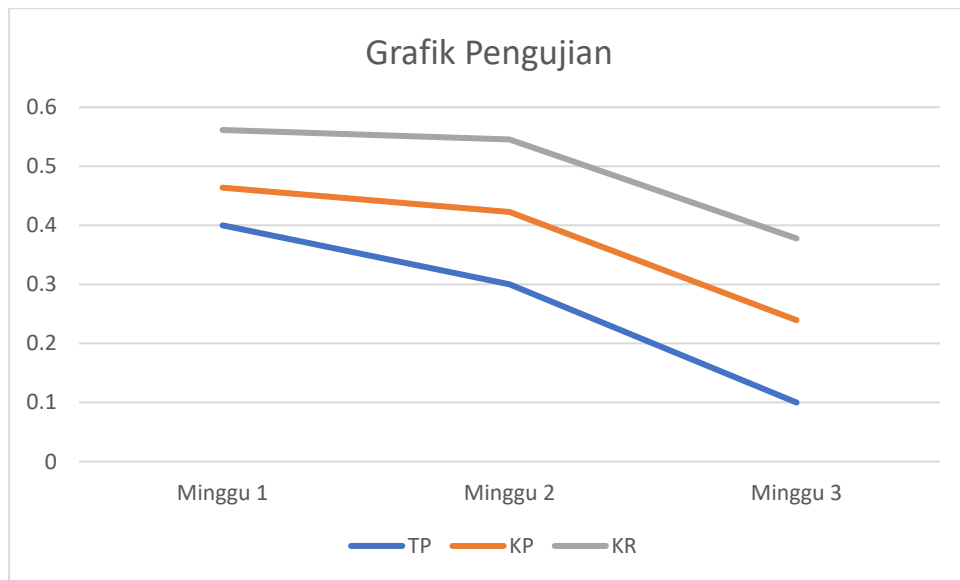
Tabel 4. 21 Tabel Pengujian

Ground Truth	Sumber	Tanggal	HasilTrending Topic
mk, anies, mobilisasi, kades, dukung, prabowo, gibran, jatim	https://nasional.kompas.com/read/2024/04/01/16461201/di-sidang-mk-kubu-anies-ungkap-ada-mobilisasi-kades-untuk-dukung-prabowo	01 April 2024	pemilu, curang, langgar, bawaslu, pilpres, kpu
gugat, kpu, ptun, pdip	https://nasional.kompas.com/read/2024/04/02/17103731/gugat-kpu-ke-ptun-pdi-p	02 April 2024	pemilu, ulang, kalah, adil, jujur, curang, sejarah, jalan, indonesia

	sampaikan-4-petitum-ini		
pemilu, ulang, mungkin, bivitri, adil, sejarah, jalan, indonesia	https://nasional.kompas.com/read/2024/04/01/14150851/pilpres-ulang-dinilai-masih-dimungkinkan-bivitri-enggakan-serumit-bikin	01 April 2024	pemilu, menang, presiden, pilih, suara, ulang, jokowi, prabowo
saksi, mk, bukti, duga, curang, pemilu	https://nasional.kompas.com/read/2024/04/02/04060021/gaya-para-saksi-saat-diminta-mk-beri-bukti-dugaan-kecurangan-pilpres?page=all	02 April 2024	pemilu, curang, bodoh, isu, alih, bikin
muhajidir, airlangga, risma, sri, bansos, pemilu	https://www.kompas.com/tren/read/2024/04/02/160000165/kata-muhadjir-airlangga-risma-dan-sri-mulyani-soal-bansos-saat-pemilu-2024?page=all	02 April 2024	pemilu, bansos, kemarin
wapres, pemilu, kondusif, mk, panas	https://nasional.kompas.com/read/2024/04/02/05291421/wapres-sebut-kondisi-pasca-pemilu-2024-lebih-kondusif-sidang-mk-panas-tapi	02 April 2024	pemilu, buzzer, orang, dukung
ganjar, bansos, kampanye, jokowi, menang	https://nasional.kompas.com/read/2024/04/02/11	02 April 2024	pemilu, partai, jokowi, uang, tambang, dpr,

	102131/ahli-kubu-ganjar-pemberian-bansos-kampanye-terselubung-jokowi-menangkan		menang, bilang, alir
ganjar, selisih, kpu	https://nasional.kompas.com/read/2024/04/02/15581511/saksi-ganjar-mahfud-klaim-ada-jutaan-selisih-suara-di-sirekap-kpu	02 April 2024	pemilu, rakyat, negara, suara dukung, indonesia
pemilu, pascapemilu, kondusif, mk, hasil, poiltik	https://nasional.kompas.com/read/2024/04/02/05291421/wapres-sebut-kondisi-pasca-pemilu-2024-lebih-kondusif-sidang-mk-panas-tapi	02 April 2024	pemilu, satu, politik, pui, negara, pascapemilu, hasil, jaga
sidang, mk, ui, suara, prabowo, dukung	https://nasional.kompas.com/read/2024/04/01/11465091/di-sidang-mk-ekonom-ui-sebut-suara-prabowo-hanya-42-persen-jika-tak-didukung	01 April 2024	mk, pemilu, hasil, uu, putus, kpu, sengketa, gugat, atur, sidang
$Topic Recall = \frac{4}{10} = 0.4$ $Keyword Precision = \frac{32}{69} = 0.46$ $Keyword Recall = \frac{32}{57} = 0.56$			

Dengan cara yang sama pengujian dilakukan sebanyak 3 kali untuk 3 minggu. Periode minggu pertama dari tanggal 25 Maret 2024 sampai 31 Maret 2024 seperti yang sudah dihitung dan dijabarkan pada tabel 4.20. Periode minggu kedua dari tanggal 1 April 2024 sampai 7 April 2024. Terakhir periode minggu ketiga dari tanggal 8 April 2024 sampai 14 April 2024. Hasil pengujian untuk ketiga minggu dapat dilihat pada gambar 4.32 di bawah.



Gambar 4. 31 Grafik Pengujian

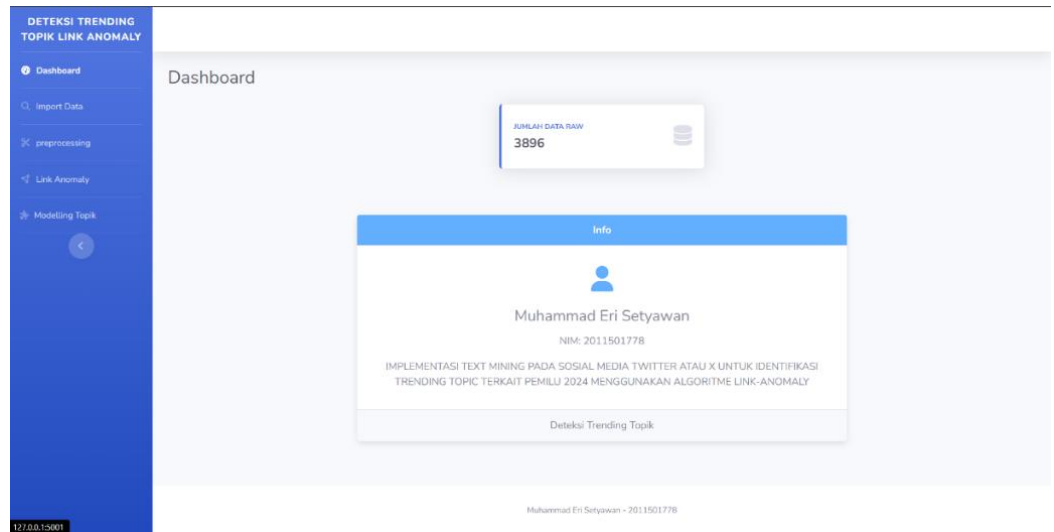
Grafik pengujian menunjukkan hasil perhitungan performa sistem dalam mendeteksi trending topik dari minggu ke minggu. Pada minggu pertama, nilai TP (Topic Recall) adalah 0.4, KP (Keyword Precision) adalah 0.5, dan KR (Keyword Recall) adalah 0.6 sebagaimana tercantum dalam Tabel 4.32. Grafik menunjukkan bahwa performa sistem mengalami penurunan pada minggu kedua, dengan nilai TP sebesar 0.3, KP sebesar 0.4, dan KR sebesar 0.5. Penurunan performa ini disebabkan oleh variasi data yang lebih besar atau perubahan dalam pola trending topik yang terdeteksi oleh sistem. Pada minggu ketiga, performa sistem menunjukkan penurunan lebih lanjut dengan nilai TP hanya sebesar 0.1, KP sebesar 0.3, dan KR sebesar 0.4. Penurunan performa ini disebabkan oleh kurangnya trending topik yang muncul pada minggu tersebut, membuat sistem kesulitan dalam mendeteksi pola yang jelas akibat data yang sangat beragam.

4.6. Tampilan Layar Aplikasi

Hasil dari penelitian ini telah diimplementasikan dalam bentuk sebuah program aplikasi, yang mencakup berbagai tampilan layar sebagai berikut.

4.6.1. Tampilan Layar *Dashboard*

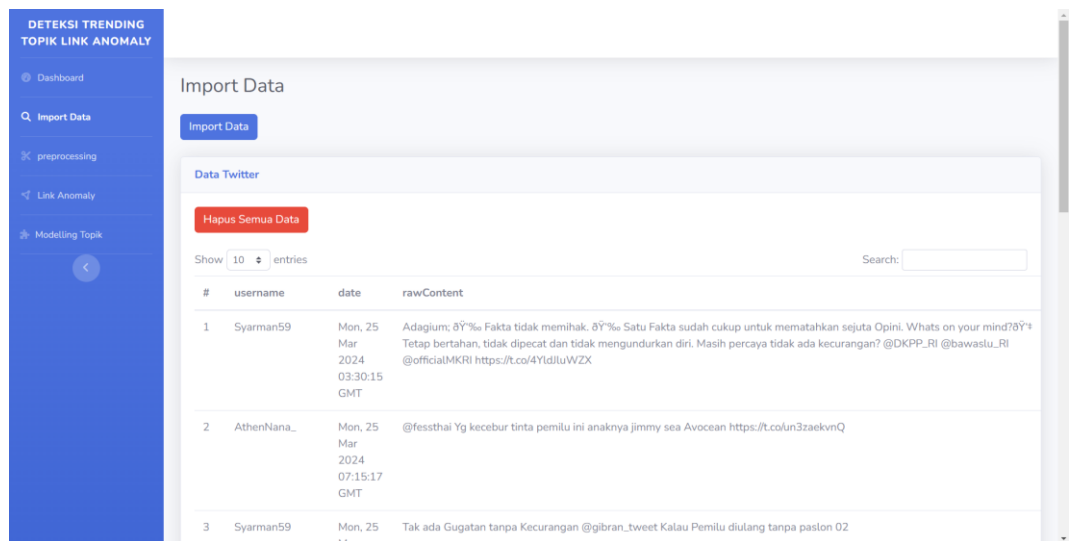
Gambar 4.33 menunjukkan Tampilan Layar Beranda dari program aplikasi yang telah dibuat.



Gambar 4. 32 Tampilan Layar *Dashboard*

4.6.2. Tampilan Layar *Import Data*

Gambar 4.34 menunjukkan Tampilan Layar *Import Data* dari program aplikasi yang telah dibuat.



Gambar 4. 33 Tampilan Layar *Import Data*

4.6.3. Tampilan Layar *Preprocessing*

Gambar 4.35 menunjukkan Tampilan Layar *Preprocessing* dari program aplikasi yang telah dibuat.

DETEKSI TRENDING TOPIK LINK ANOMALY

Dashboard

Import Data

preprocessing

Link Anomaly

Modelling Topik

Preprocessing Data

Data Hasil Preprocessing

Mulai Preprocessing Hapus Data

Total Data: 3896

Show 10 entries

Search:

id	time	username	tweet	jumlah mention	id user mentioned
1	25-03-2024 10:30:15	Syarman59	adagium fakta fakta patah juta opini whats on your mind tahan pecat undur percaya curang	3	@DKPP_RI.@bawaslu_RI.@officiaIMKR
2	25-03-2024 14:15:17	AthenNana	cebur tinta pemilu anak jimmy sea avocean	1	@fessthai
3	25-03-2024 17:01:12	Syarman59	gugat curang pemilu ulang paslon	1	@gibran_tweet
4	25-03-2024 19:20:44	lin_conil	klu diulanduit cukong china pasang habis minjem hsl bumi gadai	1	@gibran_tweet
5	25-03-2024 20:59:13	NenkMonica	tukang obat bermetamorfosa amat politik dekat dialektika dagang obat	1	@HaikalHassanbrs
6	25-03-2024 22:53:38	HengkyCh1	rencana tuntutan pemilu ulang maaf yee tu ajak	2	@uasister.@gibran_tweet
7	25-03-2024 22:53:42	nka_muslimin84	sukses pemilu damai prestasi kuat fondasi demokrasi	1	@marsieles_monly

Gambar 4. 34 Tampilan Layar *Preprocessing*

4.6.4. Tampilan Layar *Link Anomaly*

Gambar 4.36 menunjukkan Tampilan Layar *link anomaly* dari program aplikasi yang telah dibuat.

DETEKSI TRENDING TOPIK LINK ANOMALY

Dashboard

Import Data

preprocessing

Link Anomaly

Modelling Topik

Link Anomaly

Jumlah Sequence (default: 2) Mulai

Hasil

Sequence yang terpilih (sequence ke-1)

Pada waktu Senin 25/03/2024 10:30:15 - Kamis 28/03/2024 06:13:50

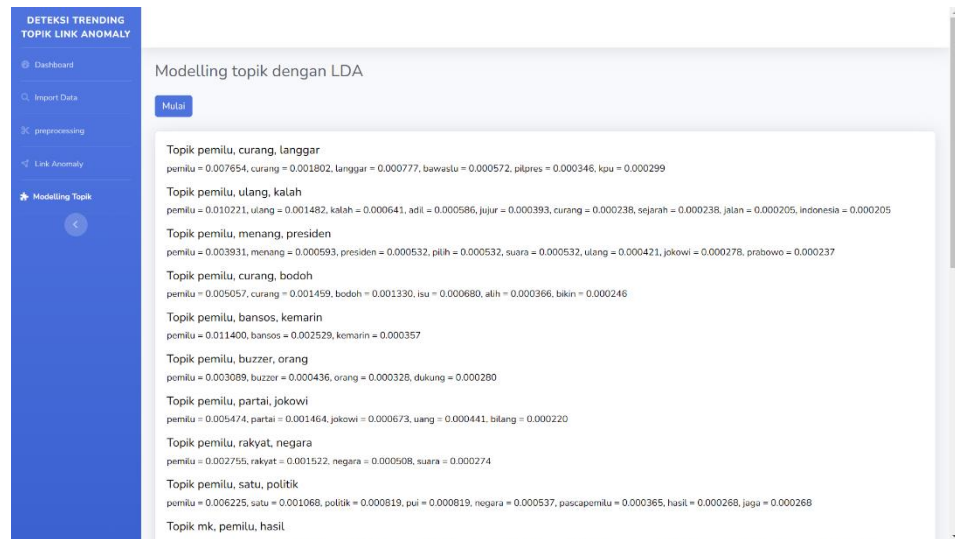
Nilai bobot sequence adalah 5714.46

#	Text
1	adagium fakta pihak satu fakta cukup patah juta opini whats on your mind tetap tahan pecat undur diri percaya tidak ada curang
2	cebur tinta pemilu anak jimmy sea avocean
3	ada gugat curang kalau pemilu ulang paslon
4	klu diulanduit cukong china pasang habis minjem saja kan banyak hsl bumi belum gadai
5	tukang obat bermetamorfosa jadi amat politik saja dekat dialektika dagang obat
6	kami rencana tuntutan pemilu minta ulang tapi maaf yee kamu tu akan ajak

Gambar 4. 35 Tampilan Layar *Link Anomaly*

4.6.5. Tampilan Layar *Modelling* Topik

Gambar 4.37 menunjukkan Tampilan *Modelling* Topik dari program aplikasi yang telah dibuat.



Gambar 4. 36 Tampilan Layar *Modelling* Topik

BAB V

PENUTUP

5.1. Kesimpulan

Setelah melakukan penelitian mengenai implementasi algoritma *link anomaly* untuk mendeteksi trending topik, penulis dapat mengambil kesimpulan yang berhubungan dengan tujuan penelitian. Berikut ini kesimpulan yang dapat penulis jabarkan:

- a. Penelitian ini berhasil melakukan deteksi *trending topic* menggunakan algoritma *link anomaly*. Hasil penelitian menunjukkan bahwa algoritma ini mampu mengidentifikasi topik-topik yang sedang tren dengan akurasi yang memadai.
- b. Pendeteksian *trending topic* menggunakan algoritma *link anomaly* dan pengujian menggunakan *Latent Dirichlet Allocation* (LDA) memberikan hasil yang cukup. Hasil pengujian menunjukkan bahwa *topic recall* mencapai 40%, *keyword precision* mencapai 46%, dan *keyword recall* sebesar 56%.
- c. Hasil deteksi *trending topic* menunjukkan bahwa pada ajang pemilu 2024, topik yang paling banyak dibicarakan oleh pengguna Platform X atau X adalah mengenai kecurangan pemilu.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan mengenai implementasi algoritma *link anomaly* untuk mendeteksi trending topik, penulis mengajukan beberapa saran untuk pengembangan lebih lanjut. Saran-saran ini bertujuan untuk meningkatkan akurasi dan efektivitas dari deteksi trending topik di masa depan. Berikut ini saran yang dapat penulis jabarkan:

- a. Dapat melakukan pengujian dengan algoritma topik modelling yang lain seperti *Non-negative Matrix Factorization* (NMF) atau *Dynamic Topic Modeling* (DTM).
- b. Integrasi dengan langsung mendapatkan data berita secara langsung dari Google menggunakan pustaka Google News dari hasil trending topik yang dideteksi.
- c. Melakukan proses *Latent Dirichlet Allocation* (LDA) dengan lebih baik lagi sehingga hasilnya lebih bagus. Misalnya, melakukan hyperparameter tuning.

- d. Gunakan data yang lebih banyak dan lebih beragam sehingga hasilnya dapat lebih akurat dan mencakup keseluruhan isu yang sedang tren.

DAFTAR PUSTAKA

- Ajjiah, N., Kurniawan, A. and Susilawati, S. (2023) 'Klasifikasi teks mining terhadap analisa isu kegiatan tenaga lapangan menggunakan algoritmaa k-nearest neighbor (knn)', *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, 7(1), pp. 254–262.
- Alita, D. and Isnain, A.R. (2020) 'Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier', *Jurnal Komputasi*, 8(2), pp. 50–58. Available at: <https://doi.org/10.23960/komputasi.v8i2.2615>.
- Alzubi, S. *et al.* (2020) 'Whats Trending? An Efficient Trending Research Topics Extractor and Recommender', *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, pp. 191–196. Available at: <https://doi.org/10.1109/ICICS49469.2020.239519>.
- Ariwibowo, P. and Indra, I. (2021) 'Deteksi Trending Topik Terkait Covid-19 Pada Tweet Bahasa Indonesia Menggunakan Metode Maximum Capturing', *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*, 2(2), pp. 288–299.
- Ariyanti, D. and Iswardani, K. (2020) 'Teks Mining untuk Klasifikasi Keluhan Masyarakat Pada Pemkot Probolinggo Menggunakan Algoritma Naïve Bayes', *Jurnal IKRA-ITH Informatika*, 4(3), pp. 125–132.
- Destarani, A.R., Slamet, I. and Subanti, S. (2019) 'Trend Topic Analysis using Latent Dirichlet Allocation (LDA) (Study Case: Denpasar People's Complaints Online Website)', *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 5(1), pp. 50–58. Available at: <https://doi.org/10.26555/jiteki.v5i1.13088>.
- Hikmah, F.N., Basuki, S. and Azhar, Y. (2020) 'Deteksi Topik Tentang Tokoh Publik Politik Menggunakan Latent Dirichlet Allocation (LDA)', *Jurnal Repositor*, 2(4), pp. 415–426. Available at: <https://doi.org/10.22219/repositor.v2i4.52>.
- Indra, Winarko, E. and Pulungan, R. (2019) 'Trending topics detection of Indonesian tweets using BN-grams and Doc-p', *Journal of King Saud University - Computer and Information Sciences*, 31(2), pp. 266–274. Available at: <https://doi.org/10.1016/j.jksuci.2018.01.005>.
- Irianto, M.R., Maududie, A. and Arifin, F.N. (2022) 'Implementation of K-Means Clustering Method for Trend Analysis of Thesis Topics (Case Study: Faculty of Computer Science, University of Jember)', *Berkala Sainstek*, 10(4), p. 210. Available at: <https://doi.org/10.19184/bst.v10i4.29524>.
- Isnain, A.R. *et al.* (2021) 'Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma Svm', *Jurnal Data Mining dan Sistem Informasi*, 2(1), p. 31. Available at: <https://doi.org/10.33365/jdmsi.v2i1.1021>.

- Jelodar, H. *et al.* (2019) ‘Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey’, *Multimedia Tools and Applications*, 78(11), pp. 15169–15211. Available at: <https://doi.org/10.1007/s11042-018-6894-4>.
- Karmila, S. and Intan Ardianti, V. (2022) ‘METODE LATENT DIRICHLET ALLOCATION UNTUK MENENTUKAN TOPIK TEKS SUATU BERITA’, 16(April), pp. 5–24.
- Kencana Putri, khaerunissa anggraeni (2021) ‘PENGELOMPOKAN DAERAH DI JAWA TENGAH BERDASARKAN INDIKATOR KESEJAHTERAAN MENGGUNAKAN METODE ENSEMBLE CLUSTER ROCK’.
- Lebovitz, S., Levina, N. and Lifshitz-Assaf, H. (2021) ‘Is AI ground truth really true? The dangers of training and evaluating AI tools based on experts’ know-what’, *MIS Quarterly: Management Information Systems*, 45(3), pp. 1501–1525. Available at: <https://doi.org/10.25300/MISQ/2021/16564>.
- MUSHTAQ AHMED, R.K.N.S.M.M.K.A.S.T.S.I.A.I. (2023) ‘Topic Modeling Using Document Pivot Approach With Fixed Window’, pp. 127–144. Available at: <https://doi.org/10.17605/OSF.IO/S6N58>.
- Mustikasari, D. *et al.* (2021) ‘Comparison of Effectiveness of Stemming Algorithms in Indonesian Documents’, *Proceedings of the 2nd Borobudur International Symposium on Science and Technology (BIS-STE 2020)*, 203, pp. 154–158. Available at: <https://doi.org/10.2991/aer.k.210810.025>.
- Rahutomo, F. and Ririd, A.R.T.H. (2019) ‘Evaluasi Daftar Stopword Bahasa Indonesia’, *Jurnal Teknologi Informasi dan Ilmu Komputer*, 6(1), pp. 41–48. Available at: <https://doi.org/10.25126/jtiik.2019611226>.
- Rifki, M. and Imelda, I. (2022) ‘Borobudur Menggunakan Multinomial Naïve Bayes Analysis of Discourse Sentiment of Borobudur Temple Ticket’, *jurnal Informatika dan Komputer*, 5(2), pp. 156–163. Available at: <https://doi.org/10.33387/jiko>.
- Rizky Fajarudin and Indra Indra (2023) ‘Penerapan Metode Doc-P untuk Deteksi Trending Topik Pemilihan Presiden Pada Twitter’, *KRESNA: Jurnal Riset dan Pengabdian Masyarakat*, 3(2), pp. 156–165. Available at: <https://doi.org/10.36080/kresna.v3i2.92>.
- Rosalina, R., Auzar and Hermendra (2020) ‘Penggunaan Bahasa Slang di Media Sosial Twitter’, *JURNAL TUAH Pendidikan dan Pengajaran Bahasa*, 2(1), pp. 77–84.
- Rosemarwati, T.U. and Lindawati, L. (2019) ‘Penggunaan Media Sosial sebagai Sumber Berita oleh Jurnalis Media Daring di Indonesia’, *Jurnal Studi Komunikasi dan Media*, 23(2), p. 101. Available at: <https://doi.org/10.31445/jskm.2019.1744>.
- Takahashi, T., Tomioka, R. and Yamanishi, K. (2014) ‘Discovering emerging topics in social streams via link-anomaly detection’, *IEEE Transactions on*

- Knowledge and Data Engineering*, 26(1), pp. 120–130. Available at: <https://doi.org/10.1109/TKDE.2012.239>.
- Utami, N.W. and Eka Putra, I.G.J. (2022) ‘Text Minig Clustering Untuk Pengelompokan Topik Dokumen Penelitian Menggunakan Algoritma K-Means Dengan Cosine Similarity’, *Jurnal Informatika Teknologi dan Sains*, 4(3), pp. 255–259. Available at: <https://doi.org/10.51401/jinteks.v4i3.1907>.
- Wirawan, R., Krisnanik, E. and Arista, A. (2024) ‘INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION Text Mining for News Forecasting on The Turnback Hoax Website’, 8(March), pp. 96–106.
- Zahria Emeraldien, F., Jefri Sunarsono, R. and Alit, R. (2019) ‘Twitter Sebagai Platform Komunikasi Politik Di Indonesia’, *Jurnal Teknologi dan Informasi*, 14(1), pp. 21–30.
- Zuhri, K. and Saputri, N.A.O. (2020) ‘Analisis Sentimen Masyarakat Terhadap Pilpres 2019 Berdasarkan Opini Dari Twitter Menggunakan Metode Naive Bayes Classifier’, *Journal of Computer and Information Systems Ampera*, 1(3), pp. 185–199. Available at: <https://doi.org/10.51519/journalcisa.v1i3.45>.