



UNIVERSITAS
Dinamika

**PENERAPAN *LATENT SEMANTIC ANALYSIS* UNTUK PENCARIAN
KEMIRIPAN BERITA BERDASARKAN RELEVANSI BERITA DENGAN
KATA KUNCI BERBASIS *WEBSITE***



LAPORAN TUGAS AKHIR

**Program Studi
S1 SISTEM INFORMASI**

UNIVERSITAS
Dinamika

Oleh:

YOHANES VELLY SABATTINO

19410100083

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

**PENERAPAN *LATENT SEMANTIC ANALYSIS* UNTUK PENCARIAN
KEMIRIPAN BERITA BERDASARKAN RELEVANSI BERITA DENGAN
KATA KUNCI BERBASIS *WEBSITE***

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana**



**UNIVERSITAS
Dinamika**

Oleh:

Nama : Yohanes Velly Sabattino

NIM : 19410100083

Program Studi : S1 Sistem Informasi

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2023

TUGAS AKHIR

PENERAPAN *LATENT SEMANTIC ANALYSIS* UNTUK PENCARIAN KEMIRIPAN BERITA BERDASARKAN RELEVANSI BERITA DENGAN KATA KUNCI BERBASIS *WEBSITE*

Dipersiapkan dan disusun oleh

Yohanes Velly Sabattino

NIM: 19410100083

Telah diperiksa, dibahas, dan disetujui oleh Dewan Pembahas

Pada: 03 Agustus 2023

Susunan Dewan Pembahas

Pembimbing:

I. **Ayuningtyas, S.Kom., M.MT.**

NIDN. 0722047801

II. **Julianto Lemantara, S.Kom., M.Eng.**

NIDN. 0722108601

Ayuningtyas
cn=Ayuningtyas, o=Universitas
Dinamika, ou=Sistem Informasi,
email=tyas@dinamika.ac.id, c=ID
2023.08.03 17:58:22 +07'00'

Digitally signed by
Julianto
Date: 2023.08.03
16:33:08 +07'00'

Pembahas:

I. **Vivine Nurcahyawati, M.Kom.**

NIDN. 0723018101

Digitally signed
by Vivine
Nurcahyawati
Date: 2023.08.04
13:18:26 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana:

Digitally signed by
Universitas Dinamika
Date: 2023.08.07
07:45:20 +07'00'

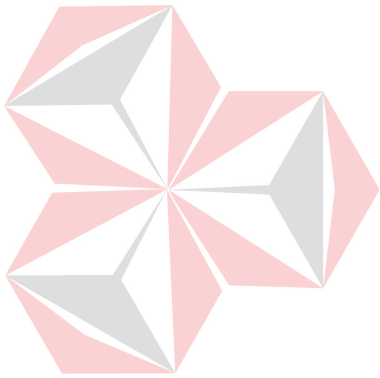
Tri Sagirani, S.Kom., M.MT.

NIDN. 0731017601

Dekan Fakultas Teknologi dan Informatika
UNIVERSITAS DINAMIKA

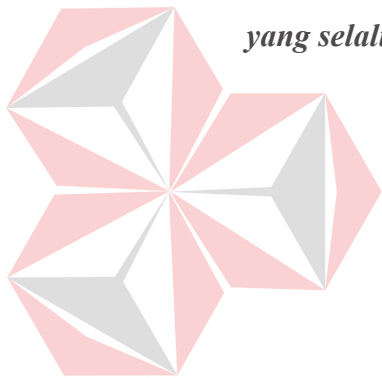
*“Our greatest weakness lies in giving up. The most certain way to succeed is
always to try just one more time.”*

– Thomas Alva Edison



UNIVERSITAS
Dinamika

*Ku persembahkan kepada
Papa, Mama dan Keluarga, serta teman dan sahabat
yang selalu memberikan bantuan dan semangat selama perjuangan ku.*



UNIVERSITAS
Dinamika

**PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa **Universitas Dinamika**, Saya:

Nama : **Yohanes Velly Sabattino**

NIM : **19410100083**

Program Studi : **S1 Sistem Informasi**

Fakultas : **Fakultas Teknologi dan Informatika**

Jenis Karya : **Tugas Akhir**

Judul Karya : **PENERAPAN *LATENT SEMANTIC ANALYSIS* UNTUK
PENCARIAN KEMIRIPAN BERITA BERDASARKAN
RELEVANSI BERITA DENGAN KATA KUNCI BERBASIS
*WEBSITE***

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah Saya tersebut di atas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 27 Juni 2023

Yang Menyatakan



Yohanes Velly Sabattino

NIM: 19410100083

ABSTRAK

Teknologi mesin pencarian merupakan suatu penerapan hasil dari perkembangan teknologi yang memiliki peranan penting untuk mengarahkan pengguna internet menuju daftar-daftar situs yang relevan. Fitur pencarian yang digunakan dapat juga digunakan untuk mencari berita berdasarkan kata kunci namun akan memprioritaskan hasil pencarian dengan *traffic* terbanyak. Studi yang dilakukan mengenai penerapan *clickbait* pada 15000 *headline* berita online di Indonesia. Menghasilkan bahwa terdapat 6290 artikel berita *online* di Indonesia yang menerapkan *clickbait* dalam penulisannya. Hal tersebut memberikan perhatian khusus bahwa terdapat hampir setengah dari total artikel berita yang dapat menyebabkan timbulnya berbagai masalah. Berdasarkan permasalahan yang terjadi, penelitian ini membentuk *website* untuk melakukan pengukuran relevansi berita terhadap kata kunci pencarian dengan menerapkan *Latent Semantic Analysis* (LSA). Dimana LSA mampu untuk mengurangi jumlah dimensi variabel parameter sehingga perhitungan yang dilakukan untuk menentukan topik pada suatu dokumen dapat lebih singkat. Sehingga dengan implementasi yang tepat pada *website* diharapkan mampu menurunkan dan meminimalisir tingkat kesalahan pencarian suatu informasi yang diterima tanpa memperhatikan kesesuaian isi berita. Dilakukan tiga pengujian yang memiliki tujuan yang berbeda dimana uji coba fungsionalitas *website* menggunakan *Black Box Testing* dilakukan demi menghindari kekeliruan pengkodean, menghasilkan bahwa *website* 100% berjalan sesuai dengan 26 skenario uji coba. Uji coba kesesuaian perhitungan yang dikodekan pada sistem dengan menggunakan *Confusion Matrix* menghasilkan bahwa *website* memiliki tingkat *accuracy* sebesar 60,25%, *precision* sebesar 92,75%, dan *recall* sebesar 62,75% berdasarkan 4 *threshold* yang berbeda diantaranya 0,1, 0,2, 0,3, dan *threshold* sistem. Uji coba kebergunaan *website* dengan *System Usability Testing* juga dilakukan dengan target masyarakat umum dengan menggunakan pada 96 responden menghasilkan bahwa *website* diterima oleh masyarakat dengan nilai sebesar 75,28.

Kata Kunci: *Berita, Latent Semantic Analysis, Prototype Modelling, Website, MERN*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa dengan kasih dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir berjudul “PENERAPAN *LATENT SEMANTIC ANALYSIS* UNTUK PENCARIAN KEMIRIPAN BERITA BERDASARKAN RELEVANSI BERITA DENGAN KATA KUNCI BERBASIS *WEBSITE*” dengan bantuan dan dukungan dari pihak-pihak berikut:

1. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika Universitas Dinamika.
2. Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng, selaku Kepala Program Studi S1 Sistem Informasi Universitas Dinamika.
3. Ibu Ayuningtyas, S.Kom., M.MT., selaku Dosen Pembimbing I.
4. Bapak Julianto Lemantara, S.Kom., selaku Dosen Pembimbing II.
5. Ibu Vivine Nurcahyawati, M.Kom., selaku Dosen Pembahas.
6. Kedua orang tua yang memberi restu serta dukungan dan doa untuk penulis agar dapat menyelesaikan Tugas Akhir.
7. Teman-teman, sahabat dan pihak-pihak lain, yang memberi bantuan dan semangat pada proses pembuatan Tugas Akhir.

Semoga kebaikan yang telah diberikan oleh semua pihak yang memberikan dukungan dan bimbingan selama pengerjaan Tugas Akhir akan mendapatkan balasan yang berlimpah dari Tuhan Yang Maha Esa. Penulis juga menyadari bahwa dalam pelaksanaan dan penulisan laporan Tugas Akhir masih terdapat banyak kekurangan, sehingga penulis sangat mengharapkan kritik yang membangun dan saran dari semua pihak agar hasil Tugas Akhir ini dapat diperbaiki lebih lanjut. Semoga laporan Tugas Akhir dapat diterima dengan baik dan memberikan manfaat bagi penulis serta semua pihak yang terlibat.

Surabaya, 26 Juli 2023

Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	4
1.4 Tujuan	4
1.5 Manfaat	5
BAB II LANDASAN TEORI	6
2.1 Penelitian Terdahulu	6
2.2 Portal Berita <i>Online</i>	8
2.3 <i>Text Preprocessing</i>	8
2.4 <i>Term Frequency-Inverse Document Frequency (TF-IDF)</i>	11
2.5 <i>Latent Semantic Analysis (LSA)</i>	11
2.6 <i>Confusion Matrix</i>	14
2.7 <i>MERN Stack</i>	15
2.8 <i>Prototype Model</i>	15
2.9 <i>System Usability Scale (SUS)</i>	16
2.10 <i>Black Box Testing (BBT)</i>	19
BAB III METODOLOGI PENELITIAN	20
3.1 <i>Identify Requirements</i>	20
3.1.1 Studi Literatur	20
3.1.2 Identifikasi Masalah.....	21
3.1.3 Analisis Kebutuhan Pengguna	21
3.1.4 Analisis Kebutuhan Fungsional	22

3.1.5	Analisis Kebutuhan Non-Fungsional	23
3.1.6	Analisis Kebutuhan Perangkat	23
3.2	<i>Prototyping</i>	24
3.2.1	Iterasi <i>Prototyping</i> Pertama	24
3.2.2	Iterasi <i>Prototyping</i> Kedua	26
3.3	<i>Design and Planning</i>	28
3.3.1	Diagram IPO	28
3.3.2	<i>System Flow</i>	29
3.3.3	<i>Context Diagram</i>	34
3.3.4	<i>Data Flow Diagram (DFD)</i>	35
3.3.5	Struktur Dokumen <i>Database</i>	37
3.4	<i>Development</i>	39
3.5	<i>Testing</i>	39
3.6	<i>Maintenance</i>	39
BAB IV HASIL DAN PEMBAHASAN		40
4.1	<i>Development</i>	40
4.1.1	Implementasi Halaman Pencarian.....	40
4.1.2	Menampung Data Pencarian Pengguna	41
4.1.3	Proses Perhitungan LSA Berdasarkan Data Pencarian	41
4.1.4	Hasil Pencarian	52
4.2	<i>Testing</i>	53
4.2.1	<i>Black Box Testing</i>	53
4.2.2	<i>Confusion Matrix</i>	54
4.2.3	<i>System Usability Scale</i>	60
4.3	Evaluasi.....	61
BAB V PENUTUP.....		63
5.1	Kesimpulan	63
5.2	Saran.....	63
DAFTAR PUSTAKA.....		65
LAMPIRAN.....		69



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Kerangka Teori.....	6
Gambar 2.2 Proses <i>Case Folding</i>	9
Gambar 2.3 Proses <i>Tokenizing</i>	9
Gambar 2.4 Proses <i>Filtering</i>	10
Gambar 2.5 Proses <i>Stemming</i>	10
Gambar 2.6 Langkah Penerapan LSA.....	12
Gambar 2.7 MERN <i>Stack Architecture</i>	15
Gambar 2.8 <i>Prototyping Model Life Cycle</i>	16
Gambar 2.9 Peringkat Hasil Akhir Penilaian SUS.....	18
Gambar 3.1 Tahapan Metodologi Penelitian.....	20
Gambar 3.2 Perancangan Desain Antarmuka Pengguna Iterasi Pertama.....	25
Gambar 3.3 Penentuan <i>Prototype</i> Desain.....	25
Gambar 3.4 Halaman Pembentukan <i>User Evaluation</i> dengan Maze.....	26
Gambar 3.5 Perbandingan Desain Iterasi Pertama dan Kedua.....	27
Gambar 3.6 Perbaikan <i>Prototype</i> Desain.....	27
Gambar 3.7 Diagram IPO.....	29
Gambar 3.8 <i>System Flow</i> Pendaftaran Akun.....	30
Gambar 3.9 <i>System Flow</i> Pencarian Berita.....	31
Gambar 3.10 <i>System Flow</i> Tahap <i>Text Preprocessing</i>	32
Gambar 3.11 <i>System Flow</i> Perhitungan LSA.....	33
Gambar 3.12 <i>System Flow</i> Menyimpan <i>History</i> Pencarian.....	34
Gambar 3.13 <i>Context Diagram</i>	34
Gambar 3.14 Diagram Jenjang.....	35
Gambar 3.15 <i>Data Flow Diagram Level 0</i>	36
Gambar 3.16 <i>Data Flow Diagram Level 1</i> Akses Pengguna.....	37
Gambar 3.17 <i>Data Flow Diagram Level 1</i> Pencarian Kemiripan Berita.....	38
Gambar 3.18 Struktur Dokumen <i>Database</i>	38
Gambar 4.1 Tampilan Halaman Pencarian.....	41
Gambar 4.2 Tampilan Halaman Hasil Pencarian.....	53

Gambar 4.3 Hasil Akhir Pengukuran Nilai SUS.....	61
Gambar L7.1 Data Berita Mentah pada Sistem	81
Gambar L7.2 Implementasi <i>Case Folding</i> pada Sistem.....	81
Gambar L7.3 Implementasi <i>Tokenizing</i> pada Sistem.....	82
Gambar L7.4 Implementasi <i>Filtering</i> pada Sistem	82
Gambar L7.5 Implementasi <i>Stemming</i> pada Sistem	82
Gambar L7.6 Hasil Pembobotan Kata dengan TF-IDF pada Sistem.....	83
Gambar L7.7 Hasil Akhir Nilai Singular U	84
Gambar L7.8 Hasil Akhir Nilai Diagonal Matrix	84
Gambar L7.9 Hasil Akhir Nilai Singular V	84
Gambar L7.10 Hasil Perhitungan Nilai Singular U dan Matrix Diagonal.....	85
Gambar L7.11 Hasil Akhir Perhitungan SVD	85
Gambar L7.12 Nilai Kemiripan Setiap Dokumen.....	86
Gambar L7.13 Nilai <i>Threshold</i>	86
Gambar L7.14 Hasil Ekstraksi Topik Pada Berita	86
Gambar L8.1 Halaman Beranda Pengguna.....	87
Gambar L8.2 Halaman Konten Beranda.....	87
Gambar L8.3 Halaman Tentang.....	87
Gambar L8.4 Halaman <i>History</i>	88
Gambar L8.5 Detail <i>History</i> Pencarian.....	88
Gambar L8.6 Halaman Masuk Pengguna	89
Gambar L8.7 Halaman Daftar Pengguna	89
Gambar L9.1 Halaman Masuk Admin	90
Gambar L9.2 Halaman <i>Dashboard</i> Admin.....	90
Gambar L9.3 Halaman <i>Master Users</i>	90
Gambar L9.4 Halaman <i>Master Kategori</i>	91
Gambar L9.5 Halaman <i>Master Berita</i>	91
Gambar L9.6 Halaman <i>Master Pencarian</i>	91

DAFTAR TABEL

	Halaman
Tabel 1.1 Contoh Kasus Pengaduan Ketidaksesuaian Berita	2
Tabel 2.1 Penelitian Terdahulu	7
Tabel 2.2 Gambaran <i>Confusion Matrix</i>	14
Tabel 3.1 Identifikasi Masalah.....	21
Tabel 3.2 Analisis Kebutuhan Pengguna	22
Tabel 3.3 Analisis Kebutuhan Fungsional	22
Tabel 3.4 Sistem Keamanan.....	23
Tabel 3.5 Hak Akses Pengguna	23
Tabel 3.6 Analisis Kebutuhan Perangkat	23
Tabel 3.7 Rangkuman Masukan Responden Terhadap Uji Coba Desain	26
Tabel 3.8 Rangkuman Masukan Iterasi Kedua	28
Tabel 4.1 Data Berita Mentah	41
Tabel 4.2 Implementasi <i>Case Folding</i>	42
Tabel 4.3 Implementasi <i>Tokenizing</i>	42
Tabel 4.4 Implementasi <i>Filtering</i>	43
Tabel 4.5 Implementasi <i>Stemming</i>	43
Tabel 4.6 Pembobotan Kata dengan TF-IDF	44
Tabel 4.7 Hasil Akhir Pencarian	52
Tabel 4.8 Hasil Uji Coba Fungsionalitas dengan <i>Black Box Testing</i>	53
Tabel 4.9 Hasil Rekomendasi Berita pada Uji Coba <i>Confusion Matrix</i>	54
Tabel 4.10 <i>Confusion Matrix</i> dengan <i>Threshold</i> 0,1.....	56
Tabel 4.11 <i>Confusion Matrix</i> dengan <i>Threshold</i> 0,2.....	57
Tabel 4.12 <i>Confusion Matrix</i> dengan <i>Threshold</i> 0,3.....	58
Tabel 4.13 <i>Confusion Matrix</i> dengan <i>Threshold</i> Sistem.....	59
Tabel 4.14 Perhitungan Rata-Rata <i>Accuracy</i> , <i>Precision</i> , dan <i>Recall</i>	60
Tabel 4.15 Hasil <i>Testing System Usability Scale</i>	61
Tabel L1.1 Hasil <i>Refine from User Suggestions</i> Iterasi Pertama	69
Tabel L2.1 Hasil <i>Refine from User Suggestions</i> Iterasi Kedua.....	70
Tabel L3.1 Daftar Skenario Uji Coba <i>Black Box Testing</i>	71

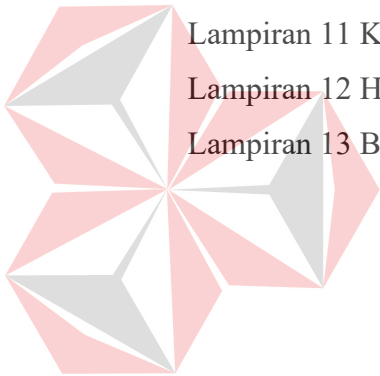
Tabel L3.2 Daftar Pertanyaan Uji Coba <i>System Usability Scale</i>	73
Tabel L4.1 Hasil Uji Fungsionalitas <i>Black Box Testing</i>	75
Tabel L5.1 Sampel Berita <i>Confusion Matrix</i>	76
Tabel L5.2 Hasil Rekomendasi Uji Coba <i>Confusion Matrix</i>	76
Tabel L6.1 Hasil <i>Testing System Usability Scale</i>	79



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Hasil Lengkap <i>Refine from User Suggestions</i> Iterasi Pertama.....	69
Lampiran 2 Hasil Lengkap <i>Refine from User Suggestions</i> Iterasi Kedua.....	70
Lampiran 3 Daftar Skenario Uji Coba	71
Lampiran 4 Hasil Uji Fungsionalitas dengan <i>Black Box Testing</i>	75
Lampiran 5 Pengujian <i>Confusion Matrix</i>	76
Lampiran 6 Hasil <i>Testing System Usability Scale</i>	79
Lampiran 7 Perbandingan Uji Coba Perhitungan	81
Lampiran 8 Implementasi Halaman Pengguna	87
Lampiran 9 Implementasi Halaman Admin.....	90
Lampiran 10 Daftar Jadwal Kerja.....	92
Lampiran 11 Kartu Bimbingan Tugas Akhir	93
Lampiran 12 Hasil Plagiasi/Originalitas Tugas Akhir.....	94
Lampiran 13 Biodata Penulis	95



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi mesin pencarian merupakan suatu penerapan hasil dari perkembangan teknologi yang memiliki peranan penting untuk mengarahkan pengguna internet menuju daftar-daftar situs yang relevan sesuai dengan informasi mengenai suatu topik atau subjek yang dicari seperti, menjawab pertanyaan yang diajukan, melakukan penelitian terhadap suatu produk, melakukan transaksi jual beli, dan sebagainya (Knuth & Masuhr, 2021). Dengan fitur pencarian, pengguna dapat mencari berita berdasarkan kata kunci dan mendapatkan hasil berdasarkan relevansi atau berita terhangat yang berarti bahwa fitur mesin pencari tersebut akan memprioritaskan hasil pencarian dengan *traffic* terbanyak. Hal tersebut dapat menimbulkan suatu permasalahan dimana ketika seseorang ingin mencari kata kunci yang memiliki banyak arti, mereka akan tidak selalu mendapatkan hasil sesuai dengan yang diinginkan (Li & Liu, 2021) dan hasil tersebut bisa mengarah ke *misleading information*.

Disisi lain, masyarakat Indonesia sangat menyukai berita yang cenderung bersifat provokatif atau berita yang sedang hangat namun berita tersebut merupakan berita yang tidak terlalu penting. Hal tersebut yang dimanfaatkan oleh para penulis berita di Indonesia, sehingga membuat beberapa berita yang dipublikasi memiliki berita dengan judul-judul bombastis (Rauf et al., 2020). Pada penelitian yang dilakukan oleh William dan Sari (2020) mengenai studi dalam penerapan *clickbait* pada *headline* berita *online* di Indonesia dengan jumlah sebanyak 15000 berita. Menghasilkan bahwa terdapat 6290 atau sebesar 41,9% artikel berita *online* di Indonesia menerapkan *clickbait* dalam penulisan *headline* berita. Hal tersebut dapat memberikan perhatian khusus bahwa terdapat hampir setengah dari total artikel berita yang dapat menyebabkan timbulnya berbagai masalah, sebagai contoh adalah muncul kelompok-kelompok yang melakukan penyimpangan penyebaran berita atau *buzzer* pada saat pemilu 2019, dan mengenai kerusuhan yang berkaitan dengan gerakan separatis di Papua Barat (McDonnell & MacKinnon, 2022). Dalam penelitian yang dilakukan oleh Hidayat (2019) mengenai dampak *clickbait* pada

media *online* di Indonesia menghasilkan bahwa terdapat beberapa dampak diantaranya terbuangnya waktu dalam mengakses informasi, masyarakat akan terbiasa mengonsumsi informasi yang instan dan sepotong-sepotong, dan efek jangka panjangnya adalah masyarakat akan sulit terliterasi dan masyarakat yang informatif dikarenakan informasi yang didapatkan jauh dari kata berkualitas.

Dampak dari *clickbait* yang telah dipaparkan di atas jika ditarik lebih jauh lagi, *clickbait* memiliki dampak pada kebiasaan masyarakat Indonesia dalam membagikan informasi. Dalam studi mengenai bagaimana *misinformation* di Indonesia terjadi dan berdampak terhadap penyebaran informasi dari 1000 responden yang terlibat menghasilkan bahwa kebiasaan masyarakat Indonesia dalam menyebarkan informasi, sebesar lebih dari 25% terkadang masih membaca suatu berita terlebih dahulu sebelum membagikannya ke orang lain. Sebesar 42% membagikan berita tanpa pernah membaca berita tersebut terlebih dahulu, 17% mengatakan bahwa mereka hanya membaca judulnya saja, dan sebesar 16% mengaku dengan sengaja berbagi berita palsu hanya untuk bersenang-senang. Hal tersebut menunjukkan bahwa literasi mengenai berita di Indonesia masih kurang, sehingga berdampak langsung pada penyebaran berita di Indonesia (McDonnell & MacKinnon, 2022).

Sepanjang tahun 2021 Dewan Pers telah melakukan pencatatan mengenai kasus pengaduan terhadap produk media yang semakin meningkat dibandingkan 3 tahun sebelumnya yaitu pada tahun 2021 tercatat 774 kasus pengaduan (Saptono et al., 2021). Berdasarkan banyaknya kasus pengaduan yang tercatat beberapa diantaranya adalah mengenai kesalahan atau ketidaksesuaian isi berita dengan judul berita yang diterbitkan. Pada Tabel 1.1 di bawah ini merupakan tabel yang berisi contoh pengaduan yang dilayangkan.

Tabel 1.1 Contoh Kasus Pengaduan Ketidakesuaian Berita

No.	Nama Pengadu	Nama Portal Berita Online	Judul Berita	Tanggal Pengaduan	Alasan Pengaduan
1	Haris Suparto Tome	faktanews.com	<i>Flash News!</i> Diduga Oknum Kadis Kabgor Di Pergoki Suami Selingkuhannya Di Kos-kosan	18 Februari 2021	Menilai isi berita tersebut tidak benar

No.	Nama Pengadu	Nama Portal Berita Online	Judul Berita	Tanggal Pengaduan	Alasan Pengaduan
2	Adiyat Nugraha	baritorayapo st.com	Kasus Pemukulan Wartawan, Penyidik Polres Barsel Tak Menjiwai Janji Kapolri	18 Februari 2021	Melanggar Pasal 1 Kode Etik Jurnalistik (KEJ) yakni menurunkan berita yang tidak independen atau mengandung konflik kepentingan

Sumber: (Saptono et al., 2021)

Berdasarkan permasalahan yang telah dipaparkan di atas, maka dibentuklah suatu *website* yang mampu untuk melakukan pengukuran terhadap tingkat kemiripan suatu informasi yang telah tersedia dan akan disajikan berdasarkan kata kunci yang dicari oleh pengguna. Dengan adanya penerapan salah satu metode *text mining* yaitu *Latent Semantic Analysis* (LSA), dimana LSA mampu untuk mengurangi jumlah dimensi variabel parameter sehingga perhitungan yang dilakukan untuk menentukan topik pada suatu dokumen dapat lebih singkat, sehingga dengan implementasi yang tepat pada *website* ini diharapkan mampu menurunkan tingkat kesalahan pencarian suatu informasi dan dapat meminimalisir informasi yang diterima hanya berdasarkan informasi yang sedang hangat tanpa memperhatikan kesesuaian isi berita. Uji kegunaan *website* menggunakan dua metode yaitu *System Usability Scale* (SUS), dan *Black Box Testing* (BBT). Sementara untuk uji keakuratan berita dengan kata kunci berita yang dicari oleh pengguna menggunakan *Confusion Matrix* dengan tujuan agar *website* yang dibangun dapat teruji baik dalam hal akurasi algoritma *text mining* yang digunakan maupun fungsionalitas *website*.

1.2 Rumusan Masalah

Berdasarkan pemaparan permasalahan latar belakang di atas adalah bagaimana penerapan metode LSA pada *website* sehingga dapat mengukur tingkat kemiripan sekaligus menampilkan topik yang diangkat pada suatu berita berdasarkan relevansi hasil pencarian.

1.3 Batasan Masalah

Berdasarkan uraian permasalahan di atas, untuk mencegah meluasnya pokok-pokok bahasan dan tidak menyimpang dari tujuan penyelesaian permasalahan, maka pokok bahasan tersebut dibatasi sehingga hanya meliputi:

1. *Testing website* menggunakan SUS, BBT, dan *Confusion Matrix*.
2. Sumber berita yang dapat diolah hanya melalui 2 sumber yaitu, melalui dokumen, dan *source* portal berita *online*.
3. *Source* portal berita *online* yang dapat digunakan hanya yang menyediakan *Application Programming Interface* (API) secara gratis dan difasilitasi oleh GitHub satyawikananda / berita-indo-api.
4. Tahap *preprocessing case folding*, *stemming* dan *tokenizing* menggunakan bantuan *library Javascript* sastrawijs.
5. Perhitungan *Singular Value Decomposition* menggunakan bantuan *library numericjs* dengan penyesuaian terhadap penelitian.
6. Kamus *stopword* dengan Bahasa Indonesia menggunakan *dataset stopwords* yang dibentuk oleh Kaggle.
7. Kata kunci yang digunakan hanya yang menggunakan basis Bahasa Indonesia atau yang memiliki serapan kata dalam Bahasa Indonesia.

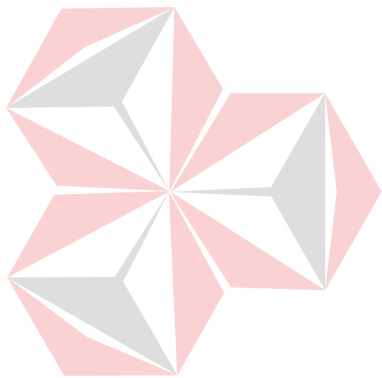
1.4 Tujuan

Berdasarkan hasil rumusan masalah di atas, maka tujuan dari penelitian ini adalah untuk menghasilkan sebuah *website* yang mampu mengukur tingkat kemiripan sekaligus menggali topik berita dengan menerapkan metode LSA dan dilakukan pengukuran keakuratan pencarian berita dengan *Confusion Matrix* untuk meminimalisir kesalahan dalam memperoleh berita.

1.5 Manfaat

Adapun manfaat yang dapat diperoleh dari penelitian ini, sebagai berikut:

1. Membantu untuk mengukur tingkat kemiripan dengan LSA sekaligus keakuratan berita dengan *Confusion Matrix* berdasarkan kata kunci yang dicari oleh pengguna.
2. Memudahkan pengguna untuk menemukan topik yang diangkat pada suatu berita.
3. Meminimalisir kesalahan berita yang diperoleh oleh karena ketidaksesuaian hasil pencarian berita.

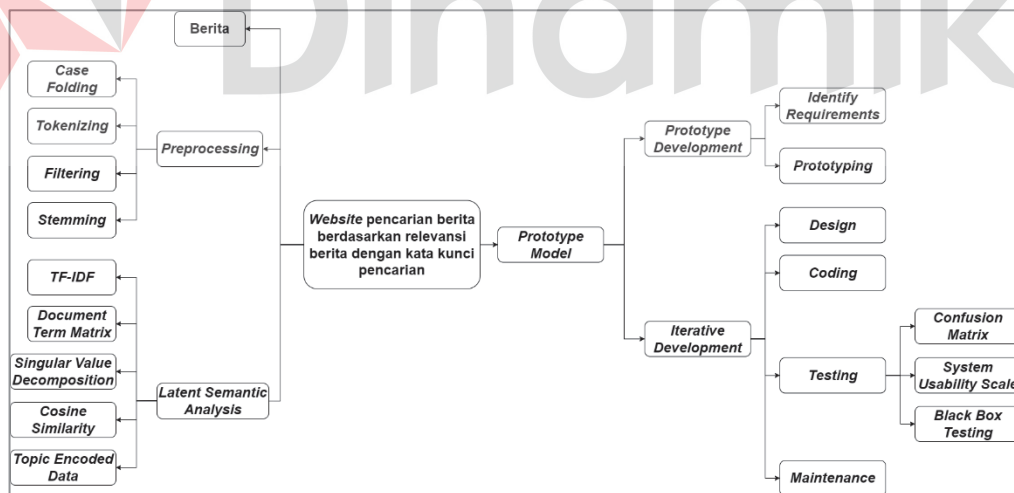


UNIVERSITAS
Dinamika

BAB II LANDASAN TEORI

Landasan teori merupakan kerangka dasar yang membantu dalam menyusun solusi untuk menyelesaikan permasalahan pada *website* yang memiliki fungsi untuk mengukur tingkat kemiripan dan menggali topik berita. Teori yang relevan mencakup *text mining*, terutama metode LSA, dan *confusion matrix* untuk mengukur tingkat keakuratan dan penggalian topik berita berdasarkan kata kunci yang dicari pengguna agar kesalahan dalam memperoleh informasi dapat diminimalisir.

Diperlukan juga pemahaman teori terkait *prototype model* untuk mengembangkan suatu *website* yang digunakan sebagai metodologi penelitian dalam menyelesaikan permasalahan melalui solusi yang ditawarkan dengan tahapan metodologi yang terdiri dari *identify requirements*, *prototyping*, *design*, *coding*, *testing*, dan *maintenance*. Pada Gambar 2.1 di bawah ini merupakan kerangka teori yang dibentuk.



Gambar 2.1 Kerangka Teori

2.1 Penelitian Terdahulu

Dalam melakukan penelitian ini juga dilakukan perbandingan baik persamaan maupun perbedaan terhadap penelitian terdahulu yang telah dilakukan dan dapat digunakan sebagai acuan dalam pengembangan penelitian ini. Pada Tabel 2.1 di

bawah ini disajikan rangkuman penelitian terdahulu yang mengangkat topik mengenai penggunaan *text mining* dengan objek portal berita *online* di Indonesia.

Tabel 2.1 Penelitian Terdahulu

Nama Peneliti	Judul	Hasil Penelitian
Ihsan Ahsanu Amala, Donni Richasdy, dan Mahendra Dwifebri Purbolaksono	<i>Telkom University News Topic Modelling Using Latent Semantic Analysis (LSA) Method on Online News Portal</i> (Amala et al., 2022)	Pemodelan topik berita Universitas Telkom pada portal berita <i>online</i> menggunakan metode LSA, menghasilkan bahwa penggunaan <i>stopwords</i> dan <i>stemming</i> berpengaruh terhadap hasil evaluasi untuk data berupa <i>headline</i> berita terkait Universitas Telkom. Proses <i>stopwords</i> dan <i>stemming</i> menghasilkan jumlah optimal sebanyak enam topik dengan <i>coherence score</i> sebesar 0,524. Hal tersebut dikarenakan pada proses <i>stemming</i> menghilangkan kata imbuhan dan pada proses <i>stopwords</i> menghapus kata yang sering muncul tapi tidak memiliki makna.
Persamaan	Metode <i>text mining</i> yang digunakan penelitian ini adalah sama yaitu menggunakan metode LSA untuk ekstraksi topik berita, dan penggunaan portal berita <i>online</i> sebagai sumber data merupakan kesamaan lainnya.	
Perbedaan	Penelitian sebelumnya terbatas pada topik terkait Universitas Telkom dan fokus pada <i>stemming</i> dan <i>stopwords</i> terhadap ekstraksi topik dan <i>coherence score</i> . Penelitian ini menggunakan data dari API berita dan berfokus pada mengukur kemiripan dan akurasi berdasarkan kata kunci pencarian.	
Dewi Fatmarani Suriyanto, Rezki Angriani Pratiwi Kadir, Fizar Syafaat, M. Miftach Fakhri, dan Dary Mochammad Rifqie	Implementasi Metode <i>Latent Semantic Analysis</i> Pada Peringkasan Artikel Bahasa Indonesia Menggunakan Pendekatan <i>Steinberger Jezek</i> (Suriyanto et al., 2022)	Proses peringkasan teks artikel berita Bahasa Indonesia dengan menerapkan metode LSA dan Teknik <i>Steinberger Jezek</i> untuk proses pemilihan kalimat menghasilkan hasil yang berbeda-beda pada skenario perbedaan nilai <i>compression rate</i> dengan nilai <i>precision</i> tertinggi yaitu 20% pada nilai <i>compression rate</i> 10%, <i>average recall</i> tertinggi yaitu 42% pada nilai <i>compression</i> 22%. Hal tersebut menyatakan bahwa tingkat <i>compression rate</i> sangat berpengaruh terhadap pengujian performa metode yang digunakan.
Persamaan	Penggunaan metode <i>text mining</i> LSA dan sumber data yang berasal dari artikel yang menggunakan Bahasa Indonesia menjadi kesamaan pada penelitian ini.	
Perbedaan	Pada penelitian terdahulu sumber data berasal dari 100 sampel berita, variabel yang digunakan hanya berasal dari teks berita, dan LSA diterapkan untuk menghitung perbandingan nilai <i>precision score</i> , dan <i>average recall</i> . Sedangkan pada penelitian ini sumber data yang digunakan bersifat fluktuatif berdasar pada penyedia berita, variabel penelitian yang digunakan yaitu teks, dan judul berita, dan LSA digunakan untuk ekstraksi topik berita yang akan diimplementasikan pada <i>website</i> .	
Imam Fahrur Rozi, Kadek Suarjuna Batubulan, dan Millenia Rusbandi	Otomatisasi Peringkasan Teks Pada Dokumen Hukum Menggunakan Metode <i>Latent Semantic Analysis</i> (Rusbandi et al., 2021)	Metode LSA digunakan untuk proses peringkasan teks atau dokumen dengan topik hukum. Peringkasan dokumen tersebut akan disajikan pada <i>website</i> untuk memudahkan akses perhitungan. Proses peringkasan dihitung berdasarkan matriks pada setiap baris dan yang terpanjang merupakan dokumen yang terpilih sebagai ringkasan. Untuk uji coba <i>precision</i> , <i>recall</i> , <i>f-measure</i> , dan <i>accuracy</i> menggunakan tiga <i>compression rate</i> yang berbeda

Nama Peneliti	Judul	Hasil Penelitian
		dan tentunya menghasilkan hasil yang berbeda-beda.
Persamaan	Penggunaan metode <i>text mining</i> LSA dan penerapan dari hasil perhitungan LSA tersebut akan disajikan dalam bentuk <i>website</i> .	
Perbedaan	Pada penelitian terdahulu sumber data hanya berasal dari dokumen dengan topik hukum, hasil yang disajikan berupa perhitungan numerik, dan uji coba yang digunakan hanya menggunakan <i>confusion matrix</i> . Sedangkan pada penelitian ini sumber data yang digunakan bersifat fluktuatif berdasar pada penyedia berita, hasil yang disajikan atau ditampilkan pada <i>website</i> berupa hasil perhitungan yang mudah dibaca dan dipahami, dan uji coba <i>website</i> dan metode LSA menggunakan tiga uji coba yaitu <i>confusion matrix</i> , SUS, dan BBT.	

2.2 Portal Berita Online

Internet secara tidak langsung telah merubah cara orang untuk mendapatkan suatu berita. Secara signifikan meningkatkan jumlah informasi yang tersedia sekaligus mengurangi biaya yang diperlukan untuk dapat mengakses informasi tersebut. Hampir seluruh media cetak dan siaran tradisional telah beralih ke platform *online* (Fletcher & Park, 2017). Internet juga menyediakan platform dimana penikmat berita dapat menemukan informasi mereka sendiri, terlepas dari lembaga atau perusahaan penyedia berita, dan dapat mengumpulkan fakta dari berbagai sumber (Sukmayadi, 2019).

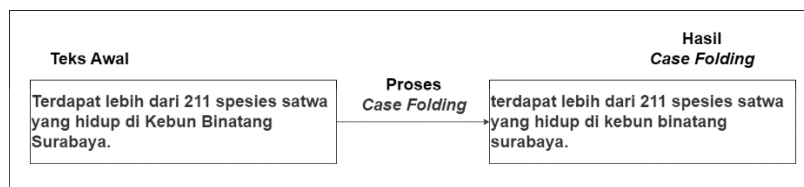
Industri media *online* telah dianggap sebagai salah satu peluang bisnis yang menguntungkan. Di Indonesia sendiri, terdapat 143 juta pengguna individu dari total 262 juta penduduk Indonesia. Selain itu sebanyak 55 persen dari mereka dapat mengakses layanan artikel secara *online*. Jumlah pengguna media *online* di Indonesia akan terus meningkat, dan akan menyebabkan pesatnya pertumbuhan industri media *online* di Indonesia (Dwityas et al., 2020).

2.3 Text Preprocessing

Berdasarkan studi kuantitatif yang menggunakan teks sebagai data, membutuhkan keputusan tentang bagaimana suatu rangkaian kata dapat diubah menjadi nilai dalam bentuk angka. Keputusan tersebut yang dikenal sebagai *preprocessing* yang bertujuan untuk memberikan *input* pada kebutuhan analisis yang dapat membuat proses yang dilakukan menjadi tidak terlalu kompleks (Denny & Spirling, 2018).

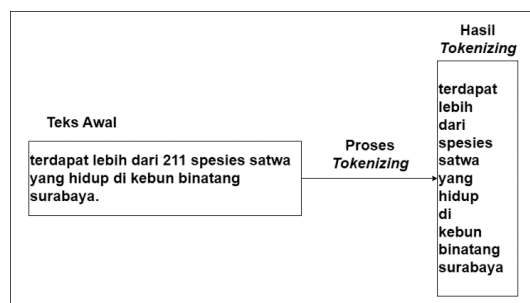
Pada dasarnya analisis kuantitatif mengharuskan untuk mampu menerjemahkan suatu teks menjadi data numerik. Di bawah ini dijelaskan secara rinci bagaimana *text preprocessing* bekerja:

1. *Case Folding*, langkah pertama yang harus dilakukan adalah membuat seluruh huruf dalam suatu kata menjadi huruf kecil. Hal tersebut dilakukan dengan berlandaskan bahwa apakah huruf kapital yang biasa digunakan pada suatu kata akan memiliki arti yang berbeda jika dibandingkan dengan kata yang tidak menggunakan huruf kapital sama sekali (Denny & Spirling, 2018). Pada Gambar 2.2 di bawah ini merupakan sebuah contoh penerapan dari tahap *case folding*.



Gambar 2.2 Proses *Case Folding*

2. *Tokenizing*, langkah selanjutnya adalah menggunakan hasil pada tahap sebelumnya kemudian dilakukan pemotongan suatu teks menjadi potongan-potongan dalam bentuk token, yang dapat berupa baik huruf, kata, ataupun kalimat sebelum dapat lanjut ke tahap berikutnya (Kadhim, 2018). Karakter berupa angka maupun tanda baca yang mampu mempengaruhi pemrosesan data juga dihilangkan pada tahap ini. Pada Gambar 2.3 di bawah ini merupakan contoh penerapan tahap *tokenizing*.



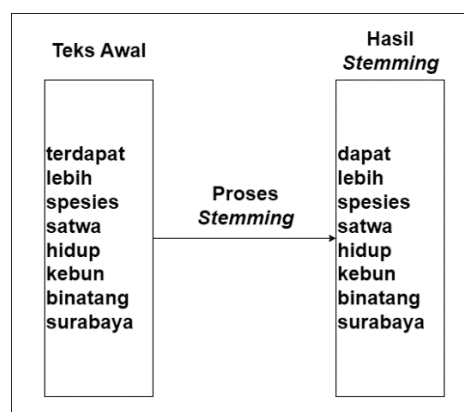
Gambar 2.3 Proses *Tokenizing*

3. *Filtering* atau bisa disebut dengan *Stopword Removal*, *stopword* merupakan serangkaian kata yang sangat umum namun tidak informatif, dan *stopword* terkadang dihilangkan selama *preprocessing text mining*. Sebagai contoh dalam pencarian informasi baik *online* maupun *offline* serangkaian kata yang termasuk kedalam *stopword* jarang memiliki kegunaan dan dengan menghapusnya akan mengurangi waktu perhitungan (Hickman et al., 2022). Berikut pada Gambar 2.4 di bawah ini merupakan contoh penerapan hasil *filtering* yang menggunakan pemrosesan teks pada tahap sebelumnya.



Gambar 2.4 Proses *Filtering*

4. *Stemming*, pada tahapan ini dilakukan perubahan sebuah kata dari hasil pemrosesan dari tahap sebelumnya ke bentuk kata dasar atau yang dikenal dengan kata baku suatu kata. Tahap *stemming* juga menghapus kata imbuhan baik imbuhan di awal dan juga di akhir. Contoh dari penerapan *stemming* dapat dilihat pada Gambar 2.5 di bawah ini.



Gambar 2.5 Proses *Stemming*

2.4 Term Frequency-Inverse Document Frequency (TF-IDF)

Algoritma TF-IDF merupakan metode statistik yang digunakan untuk menilai pentingnya suatu kata dalam dokumen ataupun kumpulan dokumen atau *file* tertentu. Dengan kata lain bahwa jika suatu kata sering muncul pada dokumen dan jarang muncul pada dokumen lain, maka kata tersebut dianggap memiliki keunikan atau pembeda dengan kata lain dan layak untuk diklasifikasi (Liu et al., 2018).

Kata dengan nilai TF yang tinggi berarti kata tersebut memiliki arti penting dalam suatu dokumen. Di sisi lain kata dengan nilai DF yang tinggi berarti kata tersebut sering terlihat pada suatu dokumen dan berarti tidak memiliki arti penting dalam dokumen tersebut. Namun, jika kata tersebut memiliki nilai IDF yang tinggi berarti kata tersebut jarang terlihat pada dokumen, sehingga membuat kata dalam dokumen tersebut memiliki kepentingan yang tinggi (Kim & Gil, 2019).

Rumus perhitungan TF dan IDF dapat dilihat di bawah ini:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \dots (1)$$

Dimana $n_{i,j}$ mewakili banyaknya kata t_i yang keluar pada dokumen d_j dan $\sum_k n_{k,j}$ mewakili banyaknya kata yang keluar pada dokumen d_j . K dan D adalah jumlah kata kunci dan dokumen.

$$IDF_{i,j} = \log \frac{|D|}{|d_j \in D : t_j \in d_j|} \dots (2)$$

Dimana $|D|$ merepresentasikan jumlah total dokumen dan $|d_j \in D : t_j \in d_j|$ merepresentasikan jumlah dokumen dimana kata kunci t_j keluar.co

$$TFIDF = TF \times IDF \dots (3)$$

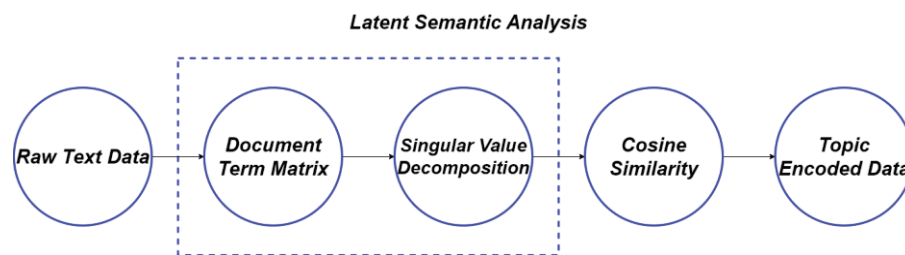
Nilai TF-IDF akan meningkat jika kata kunci tertentu memiliki frekuensi yang tinggi pada suatu dokumen dan dibandingkan dengan frekuensi pada dokumen lain. Sehingga dengan TF-IDF dapat menemukan kata kunci yang penting dalam suatu dokumen.

2.5 Latent Semantic Analysis (LSA)

Metode LSA dikembangkan sejak tahun 1980-an sebagai metode untuk penemuan informasi dari suatu dokumen. Metode ini telah banyak berperan dalam berbagai bidang seperti kecerdasan buatan, psikologi, pendidikan, dan sistem informasi (Wagire et al., 2019).

LSA merupakan metode pemrosesan bahasa yang menggunakan kategorisasi dalam analisisnya untuk dapat mengekstrak istilah yang jarang ditemui dan dibatasi oleh suatu kata kunci dan menghasilkan kata-kata yang memiliki arti yang sama dalam suatu konteks baik dokumen atau teks tertentu (Hutchison et al., 2018).

Di bawah ini merupakan langkah-langkah yang harus diikuti dalam penggunaan metode LSA:



Gambar 2.6 Langkah Penerapan LSA

Sumber: (Cook, 2019)

1. *Raw Text Data*, mengumpulkan berbagai data mentah dengan tujuan nantinya data tersebut akan digunakan sebagai bahan untuk dianalisis.
2. *Document Term Matrix*, merupakan algoritma atau metode perhitungan dalam *text mining* yang mampu untuk merepresentasikan suatu teks dalam dokumen menjadi struktur numerik (Softscients, 2021).
3. *Singular Value Decomposition* (SVD), merupakan cara pemfaktoran matriks untuk mengurai matriks menjadi tiga *unit* matriks U, dan V yaitu matriks ortogonal dan matriks Σ sebagai matriks diagonal yang berisi faktor skala yang berisi nilai singular (Sihombing et al., 2018). SVD memiliki tujuan untuk mengurangi besarnya dimensi dari *document term*, sehingga dokumen yang memiliki tingkat korelasi yang hampir serupa akan dihilangkan dan dapat mempersingkat waktu perhitungan. Di bawah ini merupakan rumus perhitungan SVD.

$$A_{m \times n} = U \Sigma V^T \dots (4)$$

Keterangan:

A = matriks berukuran $m \times n$

U = matriks ortogonal berukuran $m \times m$

Σ = matriks diagonal berukuran $m \times n$

V = matriks ortogonal berukuran $n \times n$

4. *Cosine Similarity*, ketika seluruh perhitungan untuk menentukan bobot dari setiap dokumen berita tersebut telah ditemukan, hal selanjutnya yang dilakukan adalah mengukur dan menghitung rentang kemiripan antar dokumen tersebut dengan menggunakan bantuan algoritma *cosine similarity*. Poin penting dari algoritma *cosine similarity* adalah membandingkan kemiripan antara dokumen *testing* dan dokumen *training* (Imam Riadi et al., 2020). Dalam kasus ini adalah mengukur kemiripan antara kata kunci pencarian dengan beberapa dokumen berita, sehingga dari perhitungan algoritma *cosine similarity* tersebut dapat mengeluarkan sebuah hasil berupa persentase kemiripan dari masing-masing dokumen (Hasugian et al., 2021).

$$\text{cosSim}_{(m,n)} = \frac{m \times n}{\|m\| \times \|n\|} = \frac{\sum_{i=1}^n m_i \times n_i}{\sqrt{\sum_{i=1}^n m_i^2} \times \sqrt{\sum_{i=1}^n n_i^2}} \dots (5)$$

Keterangan:

m = dokumen pertama

n = dokumen kedua

Pada tahap ini juga dilakukan penentuan ambang batas bawah atau *threshold* untuk menentukan nilai minimum kemiripan dari suatu berita, sehingga berita dengan nilai kemiripan di bawah nilai *threshold* tidak akan dilanjutkan ke tahapan selanjutnya dan dengan itu akan lebih mempersingkat waktu pemrosesan untuk tahap selanjutnya. Penentuan nilai *threshold* dihitung berdasarkan penelitian mengenai segmentasi dengan metode *thresholding* (Heryanto et al., 2020).

$$T = \frac{(f_{max} + f_{min})}{2} \dots (6)$$

Keterangan

T = nilai *threshold*

f_{max} = nilai kemiripan tertinggi

f_{min} = nilai kemiripan terendah

5. *Topic Encoded Data*, pada tahap ini merupakan tahapan yang mampu untuk menerjemahkan hasil dari perhitungan SVD ke dalam bentuk tabel dimana pada bagian baris tabel merupakan teks *original* atau data mentah sedangkan

pada bagian kolom berisi hasil menerjemahkan teks ke dalam bentuk *topic encoded data*.

2.6 Confusion Matrix

Dalam konsep *data mining* terdapat suatu metode yang dapat digunakan untuk mengukur keakuratan pada suatu data sehingga nantinya dapat digunakan untuk membantu dalam sistem pendukung keputusan, metode tersebut dikenal dengan *Confusion Matrix* (Rahmad et al., 2020). *Confusion matrix* disajikan dalam bentuk persegi dimana pada bagian kolom menggambarkan prediksi dan bagian baris menggambarkan hal yang sebenarnya. Matriks tersebut disajikan dalam bentuk persegi 2 x 2 dimana setiap matriksnya menggambarkan *True Negative* (TN), *False Negative* (FN), *True Positive* (TP), dan *False Positive* (FP). Tabel 2.2 di bawah ini merupakan gambaran dari matriks 2 x 2 dari *confusion matrix* (Caelen, 2017).

Tabel 2.2 Gambaran *Confusion Matrix*

		<i>True Values</i>	
		<i>True</i>	<i>False</i>
<i>Prediction</i>	<i>True</i>	TP	FP
	<i>False</i>	FN	TN

Confusion matrix berisi seluruh informasi yang masih mentah mengenai prediksi yang dibuat oleh suatu model klasifikasi pada kumpulan data yang diberikan. Banyak indikator yang dapat diekstrak dari *confusion matrix* beberapa diantaranya adalah *precision*, *recall*, *accuracy*, dan *f-score*. Di bawah ini merupakan rumus yang dapat digunakan untuk menghasilkan indikator performa dari *confusion matrix*.

$$Precision = \left(\frac{TP}{TP+FP} \right) \times 100\% \dots (7)$$

Precision, merupakan data yang diambil berdasarkan kurangnya informasi. Dalam klasifikasi biner, *precision* dapat dibentuk sama dengan nilai prediksi positif.

$$Recall = \left(\frac{TP}{TP+FN} \right) \times 100\% \dots (8)$$

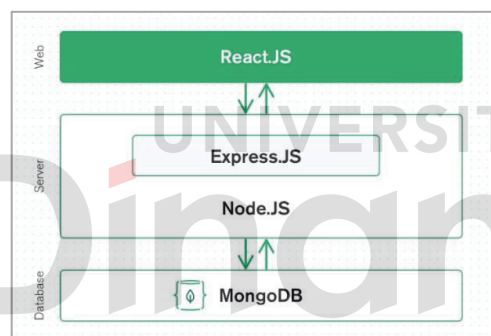
Recall, merupakan hasil penghapusan data yang diambil dari data yang terkait dengan *query*. Dalam klasifikasi biner, *recall* dikenal juga dengan *sensitivity*.

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) \times 100\% \dots (9)$$

Accuracy, merupakan persentase dari total data yang telah diidentifikasi dan dievaluasi.

2.7 MERN Stack

MERN *stack* adalah serangkaian teknologi yang digunakan bersama untuk membangun aplikasi *web*. *Node* dan *Express* berfungsi di bagian *backend*, *MongoDB* sebagai *database NoSQL*, dan *React* di bagian *frontend* untuk interaksi langsung dengan pengguna. Semua teknologi ini gratis, terbuka untuk umum, lintas platform, dan berbasis bahasa pemrograman *Javascript* (Hoque, 2020). Integrasi teknologi ini menghasilkan *tech stack Javascript* yang sederhana dan efektif dalam pengembangan *web*. Berikut adalah ilustrasi arsitektur teknologi dari MERN *stack*.



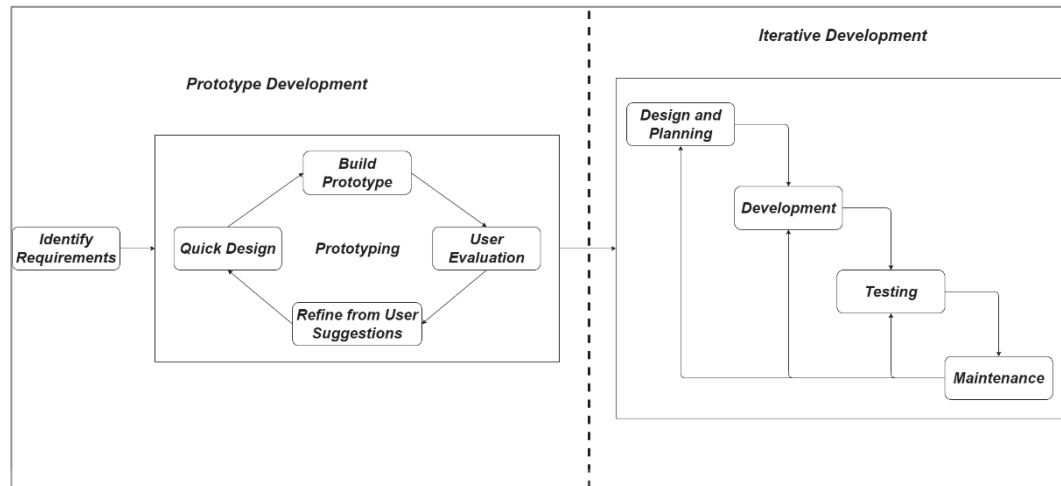
Gambar 2.7 MERN Stack Architecture

Sumber: (Tran, 2022)

2.8 Prototype Model

Model *prototype* dalam pengembangan perangkat lunak merupakan pendekatan yang efektif ketika klien hanya memberikan beberapa tujuan tanpa persyaratan fungsional yang rinci atau ketika *developer* meragukan efisiensi algoritma, adaptasi sistem, dan interaksi dengan pengguna (Pressman & Maxim, 2020). Dalam model ini, desain antarmuka pengguna dibuat secara cepat dan diwujudkan dalam bentuk *prototype* yang diberikan kepada calon pengguna untuk evaluasi dan mendapatkan *feedback* (Saravanan et al., 2020). Proses tersebut berulang hingga mendapatkan *feedback* terbaik. Model *prototype* terdiri dari dua fase, *prototype development* (identifikasi kebutuhan, desain cepat, pembuatan

prototype, evaluasi pengguna, dan penyaringan saran) dan *iterative development* (fase desain dan perencanaan, *development*, uji coba, dan pemeliharaan). Gambar 2.8 menjelaskan cara kerja dari model *prototype*.



Gambar 2.8 Prototyping Model Life Cycle

Sumber: (Trunkett, 2020)

2.9 System Usability Scale (SUS)

SUS merupakan penilaian kegunaan untuk sistem apapun dengan skala pemeringkatan yang mudah digunakan dan menjadi salah satu kuesioner yang paling populer (Kaya et al., 2019). SUS diciptakan oleh John Brooke pada tahun 1996 yang berisi sepuluh pernyataan dasar dan sederhana mengenai kegunaan suatu sistem. Untuk menyatakan pendapat baik setuju maupun tidak setuju terhadap pernyataan mengenai suatu sistem disajikan dalam bentuk kuesioner dan pada skala likert 5 poin dimana poin 1 menyatakan “Sangat Tidak Setuju” hingga poin 5 menyatakan “Sangat Setuju”. Jika, pengguna tidak yakin akan jawaban mereka, pengguna tetap diharuskan untuk memilih salah satu dari ke 5 poin, meskipun pengguna menjawab poin 3 atau “Ragu-Ragu” (Gronier & Baudet, 2021). Adapun 10 pernyataan pada kuesioner untuk menilai SUS yang telah dibentuk oleh John Brooke dan telah diterjemahkan kedalam Bahasa Indonesia untuk menyesuaikan target responden dapat dilihat pada Lampiran 3 Tabel L3.2.

Skor SUS dapat bernilai 1 hingga 100 dengan memperhitungkan hasil jawaban dari 10 pernyataan, dan hasil skor dihitung melalui tiga langkah:

1. Untuk nomor pernyataan ganjil, penilaian poin skala likert dikurangi satu poin.
2. Untuk nomor pernyataan genap, lima poin dikurangi dengan penilaian poin skala likert.
3. Menjumlahkan hasil perhitungan dua langkah di atas, dan diakhiri dengan dikalikan sebesar 2,5 poin.

Skor akhir penilaian SUS dihitung dari rata-rata skor untuk setiap responden dan dibagi dengan keseluruhan jumlah responden. Rumus dalam menghitung skor akhir penilaian SUS dapat dilihat seperti pada di bawah ini.

$$\bar{x} = \frac{\Sigma x}{n} \dots (10)$$

Keterangan

\bar{x} = skor akhir SUS

Σx = total skor seluruh responden

n = jumlah responden

Dalam menentukan total sampel atau responden masuk kedalam kategori pengambilan sampel dengan jumlah populasi yang tidak diketahui, dan untuk mengetahui secara pasti jumlah sampel digunakanlah Rumus Lemeshow. Rumus Lemeshow dikenal sebagai uji kesesuaian berdasarkan dengan nilai prediksi dan peluang (Yuhadisi, 2021). Penjelasan dari rumus Lemeshow dapat dilihat di bawah ini.

$$n = \frac{Z^2 P(1-P)}{d^2} \dots (11)$$

Keterangan:

n = jumlah sampel

Z = nilai atau skor berdasarkan tingkat kepercayaan

P = estimasi probabilitas maksimal

d = *sampling error*

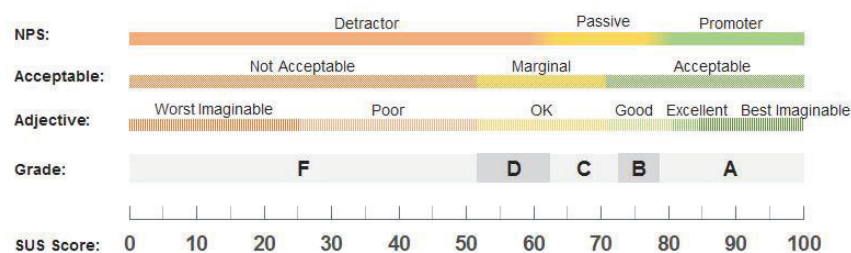
Melalui rumus yang telah dipaparkan di atas, maka dapat dihitung jumlah sampel yang nantinya dapat digunakan juga sebagai patokan untuk menentukan jumlah responden. Tingkat kepercayaan yang ditentukan sebesar 95% dan berdasarkan nilai z tabel, diperoleh nilai z sebesar 1.96. Kemudian estimasi probabilitas sebesar 0.5 dan *sampling error* sebesar 10% atau 0.1.

$$n = \frac{1,96^2 \times 0,5(1 - 0,5)}{0,1^2} = \frac{3,8416 \times 0,25}{0,01} = \frac{0,9604}{0,01} = 96,04$$

Berdasarkan perhitungan yang telah dilakukan dengan menggunakan rumus Lemeshow didapati maka jumlah sampel minimal yang harus didapatkan adalah sebesar 96,04 atau dibulatkan menjadi 96 responden.

Setelah hasil akhir penilaian SUS diperoleh, maka skor tersebut dapat diolah untuk disajikan menjadi beberapa kategori sehingga hasil dari SUS dapat dibaca dengan sebaik mungkin (Sauro, 2018). Beberapa kategori tersebut diantaranya:

1. *Net Promoter Score* (NPS), yaitu nilai untuk mengukur tingkat kepuasan pengguna dan loyalitas pengguna terhadap suatu produk, sehingga *feedback* yang didapatkan berupa kesediaan pengguna untuk merekomendasikan suatu produk kepada orang lain.
2. *Acceptability Range*, merupakan tingkat penerimaan pengguna terhadap suatu sistem dan disajikan menjadi tiga yaitu *not acceptable*, *marginal* dan *acceptable*.
3. *Adjective Rating*, merupakan suatu standar untuk menerjemahkan hasil skor SUS dalam bentuk numerik menjadi enam skala yaitu *worst imaginable*, *poor*, *ok*, *good*, *excellent*, dan *best imaginable*.
4. *Grade Scale*, merupakan pengelompokan penilaian skor akhir SUS dan dibagi menjadi lima skala yaitu F untuk skor kurang dari 51,7, D untuk skor antara 51,7 hingga 62,6, C untuk skor 62,7 hingga 72,5, B untuk skor 72,6 hingga 78,8, dan A untuk skor diatas 78,9.
5. *SUS Score*, yaitu nilai akhir penilaian SUS yang disajikan dalam bentuk skala numerik. Di bawah ini merupakan gambar peringkat hasil penilaian SUS.



Gambar 2.9 Peringkat Hasil Akhir Penilaian SUS

Sumber: (Sauro, 2018)

2.10 *Black Box Testing* (BBT)

BBT merupakan pengujian yang dilakukan hanya dengan mengamati dari hasil data pengujian dan uji fungsionalitas dari suatu perangkat lunak (Verma et al., 2017). Pengujian ini didasarkan pada *requirements and specification* dari suatu perangkat lunak dimana cara kerja pengujian dari *item-item* yang diuji tidak diketahui sebelumnya oleh penguji dan penguji tidak perlu memahami baik struktur maupun penulisan kode dari aplikasi melainkan hanya melihat berdasarkan *input* dan *output* dari suatu aplikasi (Sholeh et al., 2021).

Pengujian aplikasi dengan menggunakan metode BBT juga memiliki kelebihan dan kekurangan dalam penggunaannya (Verma et al., 2017). Di bawah ini dijelaskan baik mengenai kelebihan dan kekurangan dari penggunaan BBT:

Keuntungan penggunaan BBT:

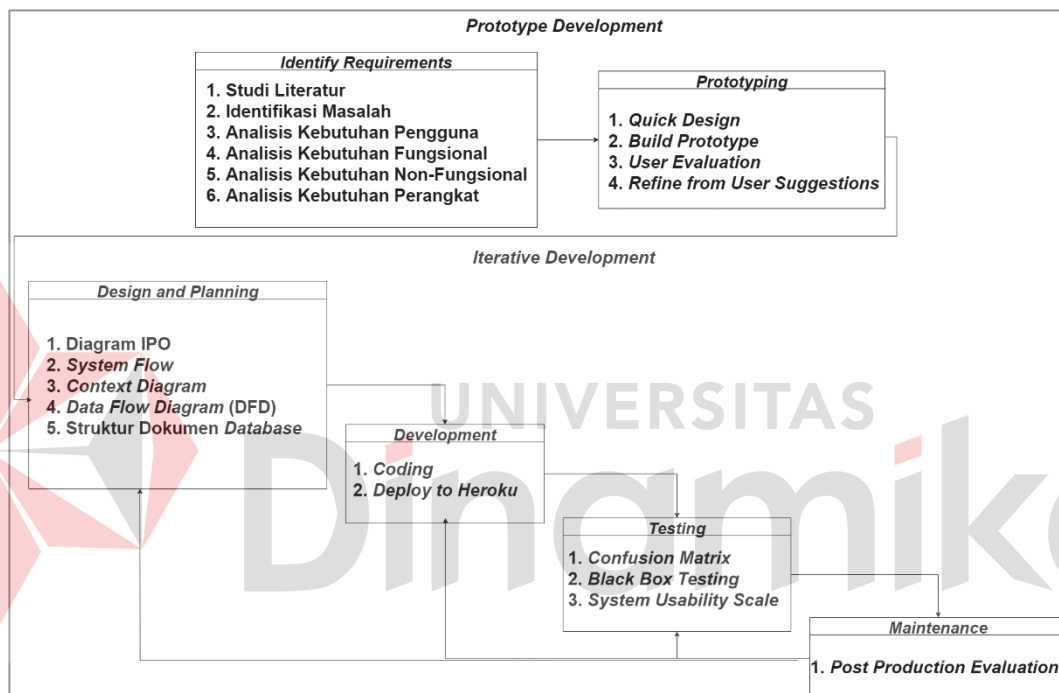
1. Pengujian dilakukan berdasarkan dengan kebutuhan pengguna.
2. Pengujian dilakukan oleh pihak ketiga untuk menghindari bias dari *developer*.
3. Penguji bisa berasal dari orang umum atau yang awam dalam pengembangan aplikasi, dikarenakan metode pengujian ini tidak memerlukan pengetahuan mengenai implementasi pemrograman.
4. Pengujian dapat lebih efisien bila diterapkan pada sistem yang lebih besar.

Kekurangan penggunaan BBT:

1. Pengujian bersifat redundansi jika uji kasus pengujian telah diterapkan pada saat pengembangan perangkat lunak.
2. Pembentukan uji kasus akan menjadi sulit jika persyaratan penggunaan aplikasi tidak jelas dan ringkas.
3. Pengujian akan menjadi tidak efisien jika pengujian dilakukan pada kasus yang memiliki tingkat penulisan kode yang kompleks.
4. Hasil tes sering disampaikan secara berlebihan.

BAB III METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan untuk menyelesaikan dan mengembangkan pada pengerjaan penelitian ini adalah menggunakan *prototype model* yang dibagi menjadi dua tahapan yaitu *prototype development* dan *iterative development*, dengan tahapan penyelesaian seperti pada Gambar 3.1 di bawah ini.



Gambar 3.1 Tahapan Metodologi Penelitian

3.1 Identify Requirements

Pada tahap yang pertama ini merupakan *identify requirements* dimana pada tahap ini dilakukan identifikasi mengenai kebutuhan-kebutuhan dalam mengembangkan solusi yang ditawarkan meliputi 6 topik pembahasan yaitu Studi Literatur, Identifikasi Masalah, Analisis Kebutuhan Pengguna, Analisis Kebutuhan Fungsional, Analisis Kebutuhan Non-Fungsional, Analisis Kebutuhan Perangkat.

3.1.1 Studi Literatur

Studi literatur dilakukan dengan tujuan sebagai referensi pembelajaran baik teori, metode dan penelitian terdahulu mengenai topik yang berhubungan dengan

penelitian. Seluruh referensi yang digunakan pada penelitian ini didapat dan berasal dari buku, jurnal, artikel penelitian dan situs internet. Referensi tersebut bertujuan sebagai landasan teori dan berguna untuk memperkuat topik permasalahan yang telah dijelaskan sebelumnya. Berikut di bawah ini merupakan topik studi literatur yang dilakukan selama penelitian ini:

1. Penelitian Terdahulu
2. Portal Berita *Online*
3. *Text Preprocessing*
4. Pembobotan *TF-IDF*
5. Perhitungan LSA
6. *Confusion Matrix*
7. Pengembangan dengan *MERN Stack*
8. *Prototype Model*
9. Testing menggunakan SUS dan BBT

3.1.2 Identifikasi Masalah

Pada tahap ini merupakan proses identifikasi masalah yang disertai juga solusi yang ditawarkan dan berdasar pada hasil analisis permasalahan yang telah dilakukan pada bab sebelumnya. Hasil dari identifikasi masalah pada tahap ini akan dilanjutkan pada tahap berikutnya sesuai dengan metodologi penelitian yang digunakan. Pada Tabel 3.1 merupakan hasil dari identifikasi masalah dan solusi yang diberikan.

Tabel 3.1 Identifikasi Masalah

Permasalahan	Solusi
Pencarian berita yang terkadang lebih mementingkan <i>traffic</i> pada berita yang sedang hangat dan judul berita yang dibuat cenderung bersifat <i>clickbait</i> tanpa memperhatikan isi berita, dapat menyebabkan kesalahan penerimaan berita atau <i>misleading information</i>	Membangun <i>website</i> yang mampu untuk mengukur tingkat kemiripan dan keakuratan berita berdasarkan kata kunci yang sedang dicari disertai dengan penggalan topik pada suatu berita dengan menerapkan metode LSA.

3.1.3 Analisis Kebutuhan Pengguna

Setelah identifikasi masalah dilakukan pada tahap sebelumnya dan ditemukan solusi dari permasalahan yang terjadi, pada tahap ini dilakukan analisis mengenai

target pengguna yang akan menggunakan sistem yang akan dibentuk disertai dengan kebutuhan dari pengguna. Tabel 3.2 di bawah ini merupakan hasil analisis kebutuhan pengguna.

Tabel 3.2 Analisis Kebutuhan Pengguna

No.	Pengguna	Kebutuhan
1	Pengguna Umum	Pengguna umum merupakan pengguna yang dapat melakukan pencarian berita berdasarkan kata kunci yang dimasukkan ke dalam kolom pencarian dan memperoleh hasil berupa hasil analisis berdasarkan keakuratan berita dengan kata kunci. Pengguna umum juga dapat memilih dari mana dokumen teks didapat. Pengguna umum dapat melakukan pendaftaran akun untuk menyimpan hasil pencarian.
2	Admin	Admin merupakan pengguna yang mengelola di bagian <i>database</i> untuk mencegah penuhnya <i>database</i> . Admin dapat mengelola pengguna yang telah lama tidak menggunakan akunnya dan mengelola hasil pencarian berita yang tersimpan.

3.1.4 Analisis Kebutuhan Fungsional

Dengan dilakukannya analisis kebutuhan fungsional maka diharapkan dapat mengetahui proses dari jalannya suatu sistem yang akan dibangun ke dalam *website*. Aktor, fungsi, dan deskripsi yang dibangun ke dalam sistem nantinya akan digunakan oleh pengguna sesuai dengan kebutuhannya. Pada Tabel 3.3 di bawah ini merupakan hasil dari analisis kebutuhan fungsional yang telah dilakukan.

Tabel 3.3 Analisis Kebutuhan Fungsional

No.	Aktor	Fungsi	Deskripsi
1	Pengguna Umum	Register	Merupakan proses untuk pengguna umum melakukan pendaftaran akun baru mereka ke dalam sistem dengan mengisi beberapa <i>form</i> yang tersedia.
		Login	Merupakan proses yang dilakukan oleh pengguna umum untuk dapat masuk ke dalam akun yang telah terdaftar sebelumnya pada sistem.
		Memilih Sumber Dokumen Teks	Merupakan proses dimana pengguna memilih sumber dokumen teks terlebih dahulu sebelum melakukan pencarian.
		Melakukan Pencarian	Pengguna melakukan pencarian berita berdasarkan kata kunci yang ingin dicari.
2	Admin	Melihat <i>History</i> Pencarian	Pengguna dapat melihat <i>history</i> dari pencarian yang telah dilakukan sebelumnya.
		Login	Merupakan proses yang dilakukan oleh admin untuk dapat masuk ke dalam akun yang telah didaftarkan sebelumnya, agar memperoleh akses untuk mengelola bagian admin.
		Mengelola Akun Pengguna	Merupakan proses dimana admin dapat melihat pengguna yang telah terdaftar dan menghapus pengguna yang telah lama tidak menggunakan akunnya.
		Mengelola Hasil Pencarian	Merupakan proses dimana admin dapat menghapus <i>history</i> hasil pencarian dari pengguna yang belum mendaftarkan akunnya.

3.1.5 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dibentuk untuk mengetahui kebutuhan yang nantinya diterapkan pada sistem dan diluar dari kebutuhan fungsional. Kebutuhan non-fungsional terbagi menjadi dua bagian yaitu sistem keamanan yang dapat dilihat pada Tabel 3.4 dan pengaturan mengenai hak akses terhadap sistem yang dibangun dapat dilihat pada Tabel 3.5.

Tabel 3.4 Sistem Keamanan

No.	Aktor	Username	Password
1	Pengguna Umum	pengguna1	*****
2	Admin	admin	*****

Tabel 3.5 Hak Akses Pengguna

No.	Fungsi	Hak Akses	
		Pengguna Umum	Admin
1	Melakukan pencarian berita	<i>insert, read</i>	-
2	Mengelola akun pengguna	<i>insert, read</i>	<i>Read, update, delete</i>
3	Mengelola <i>history</i> pencarian	<i>read</i>	<i>read</i>

3.1.6 Analisis Kebutuhan Perangkat

Analisis ini dibentuk dengan tujuan sebagai acuan dalam penggunaan baik perangkat lunak dan minimum perangkat keras apa saja yang nantinya digunakan untuk merancang, membangun, serta mengembangkan sistem pada penelitian ini.

Daftar perangkat lunak dan keras akan disajikan pada Tabel 3.6 di bawah ini.

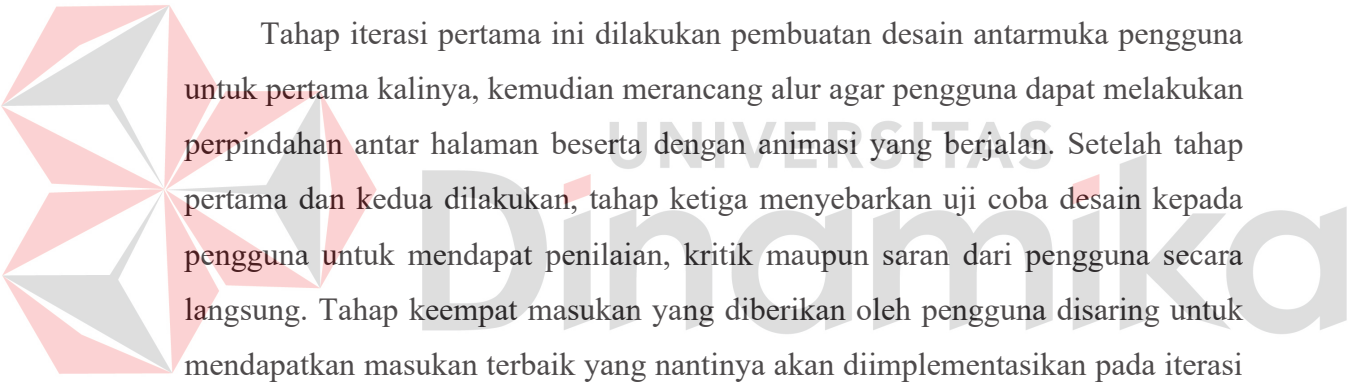
Tabel 3.6 Analisis Kebutuhan Perangkat

No.	Kelompok	Perangkat
1	Perangkat Lunak	Bahasa Pemrograman <i>Javascript</i> <i>Text editor</i> Visual Studio Code Figma sebagai <i>tool ui design</i> Database <i>NoSQL</i> MongoDB <i>Browser</i>
		<i>Processor</i> 2GHz CPU Penyimpanan 32 GB <i>Memory</i> 4 GB RAM
2	Perangkat Keras	<i>OS</i> Windows 8.0 <i>Mouse</i> dan Keyboard Layar Monitor <i>Smartphone</i> yang memiliki <i>browser</i>

3.2 *Prototyping*

Pada tahap yang ketiga yaitu *prototyping* merupakan tahapan dimana dilakukan pembentukan desain antarmuka pengguna secara cepat kemudian dilanjutkan dengan pembuatan *prototype* agar dari desain antarmuka tersebut dapat digunakan secara langsung oleh pengguna. Berdasarkan dari *prototype* tersebut pengguna dapat menilai dan memberikan masukan terhadap desain yang telah dibentuk, kemudian berdasarkan dari masukan-masukan yang diberikan oleh pengguna maka disaringlah masukan tersebut untuk mendapatkan saran atau masukan terbaik untuk fase desain ulang. Proses tersebut akan terus berulang hingga sampai mendapatkan *feedback* yang terbaik. Proses-proses tersebut meliputi *quick design*, *build prototype*, *user evaluation*, dan *refine from user suggestions*.

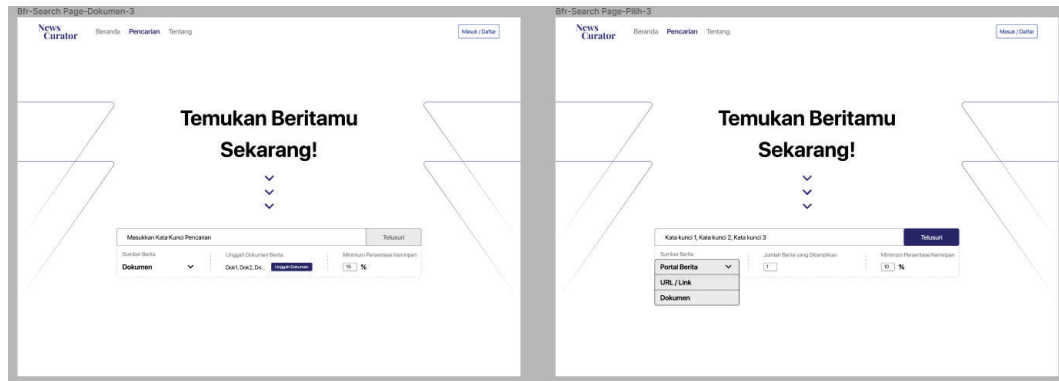
3.2.1 Iterasi *Prototyping* Pertama



Tahap iterasi pertama ini dilakukan pembuatan desain antarmuka pengguna untuk pertama kalinya, kemudian merancang alur agar pengguna dapat melakukan perpindahan antar halaman beserta dengan animasi yang berjalan. Setelah tahap pertama dan kedua dilakukan, tahap ketiga menyebarkan uji coba desain kepada pengguna untuk mendapat penilaian, kritik maupun saran dari pengguna secara langsung. Tahap keempat masukan yang diberikan oleh pengguna disaring untuk mendapatkan masukan terbaik yang nantinya akan diimplementasikan pada iterasi kedua.

A. *Quick Design*

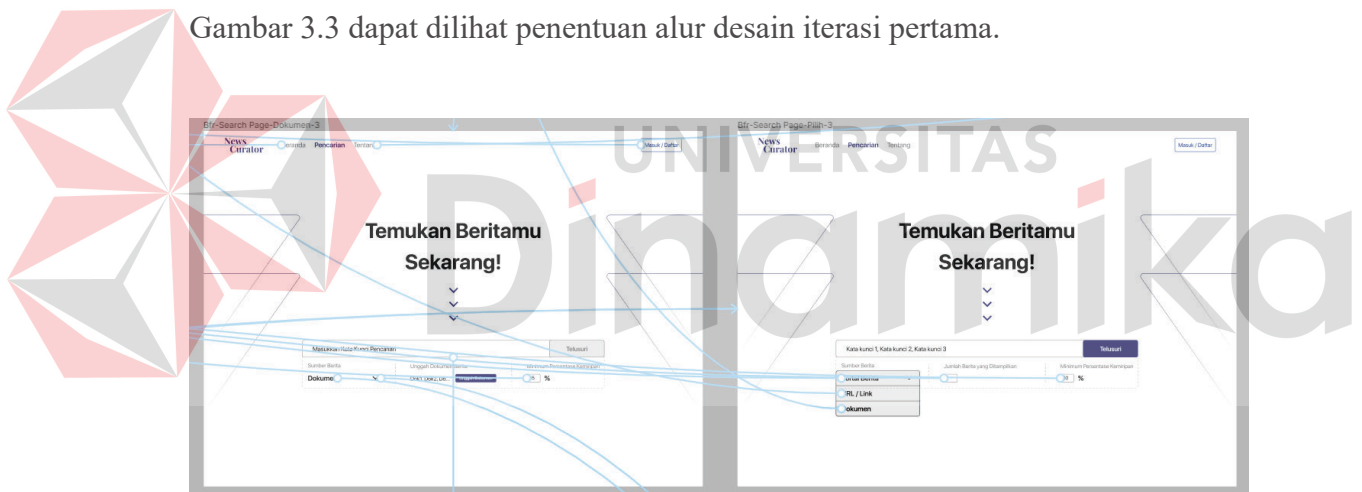
Perancangan awal desain antarmuka pengguna dilakukan dengan menggunakan salah satu perangkat lunak desain yang bernama Figma. Diawali dengan menentukan ide awal untuk pembentukan yang sesuai, pemilihan warna yang sesuai dengan identitas dimulai dari warna primer, sekunder, juga untuk warna latar belakang dan warna tulisan. Akhirnya pembentukan desain dapat dimulai, pada Gambar 3.2 merupakan perancangan desain antarmuka pengguna iterasi pertama.



Gambar 3.2 Perancangan Desain Antarmuka Pengguna Iterasi Pertama

B. Build Prototype

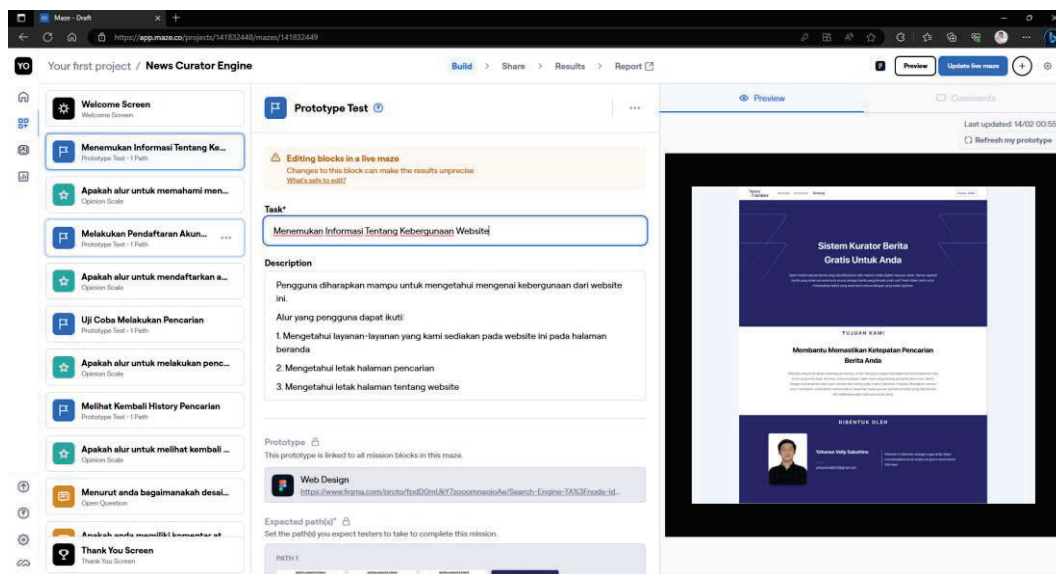
Setelah desain terbentuk, maka hal selanjutnya yang dilakukan adalah menentukan alur desain antar halaman, komponen apa saja yang dapat dilakukan interaksi, dan animasi perpindahan setiap komponen maupun halaman. Pada Gambar 3.3 dapat dilihat penentuan alur desain iterasi pertama.



Gambar 3.3 Penentuan *Prototype* Desain

C. User Evaluation

Pada tahap ketiga ini mencari masukan dari calon pengguna terhadap desain dan alur desain yang telah dibentuk. Terdapat layanan *prototype testing* yang ditawarkan secara gratis yang dapat langsung terhubung dengan Figma bernama Maze. Pada *prototype testing* ini terdapat empat uji coba alur desain, empat pertanyaan mengenai skala terhadap pemahaman akan alur desain, satu pertanyaan mengenai opini terhadap desain secara keseluruhan dan satu pertanyaan terakhir mengenai masukan terhadap desain uji coba yang telah dilakukan. Pada Gambar 3.4 merupakan *user evaluation* yang dibentuk menggunakan *platform* Maze.



Gambar 3.4 Halaman Pembentukan *User Evaluation* dengan Maze

D. Refine from User Suggestions

Tahap terakhir untuk iterasi pertama ini berhasil mendapatkan responden yang telah melakukan uji coba alur desain melalui *link* yang diterima. Responden yang berhasil melakukan uji coba dengan lengkap berjumlah 30 responden, dan masukan ataupun komentar yang diberikan telah didapatkan. Pada Tabel 3.7 di bawah ini merupakan rangkuman dari masukan maupun komentar responden. Hasil lengkap untuk komentar, masukan, kritik, dan saran yang diperoleh pada iterasi pertama dapat dilihat pada Lampiran 1 tepatnya pada Tabel L1.1.

Tabel 3.7 Rangkuman Masukan Responden Terhadap Uji Coba Desain

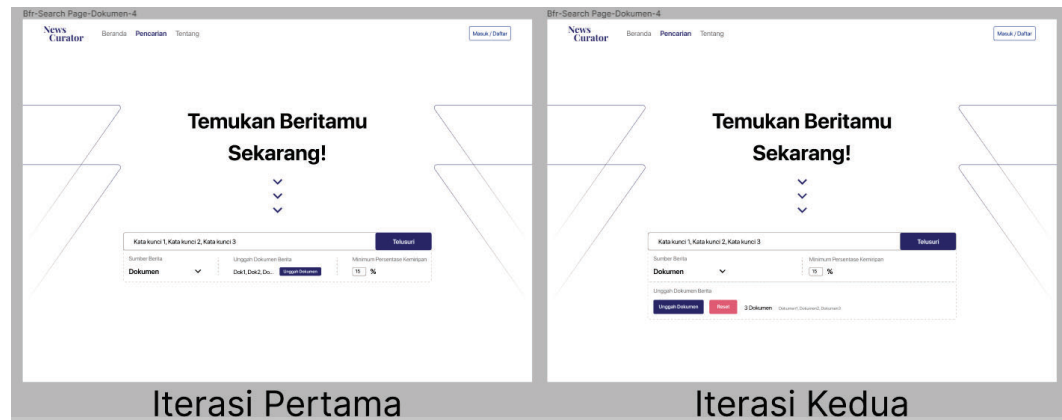
No.	Masukan	Jumlah	Persentase
1.	Bagus	10	33.3%
2.	Warna yang terlalu mendominasi	6	20%
3.	Alur halaman yang masih membingungkan	4	13.3%
4.	Komponen teks, tombol, dan <i>input</i> terlalu kecil	4	13.3%
5.	Tidak ada komentar	4	13.3%
6.	Kurang lengkap, dilengkapi terlebih dahulu	2	6.67%

3.2.2 Iterasi *Prototyping* Kedua

Iterasi kedua dilakukan dikarenakan masih terdapat masukan penting dari responden yang penting dan harus diperbaiki.

A. Quick Design

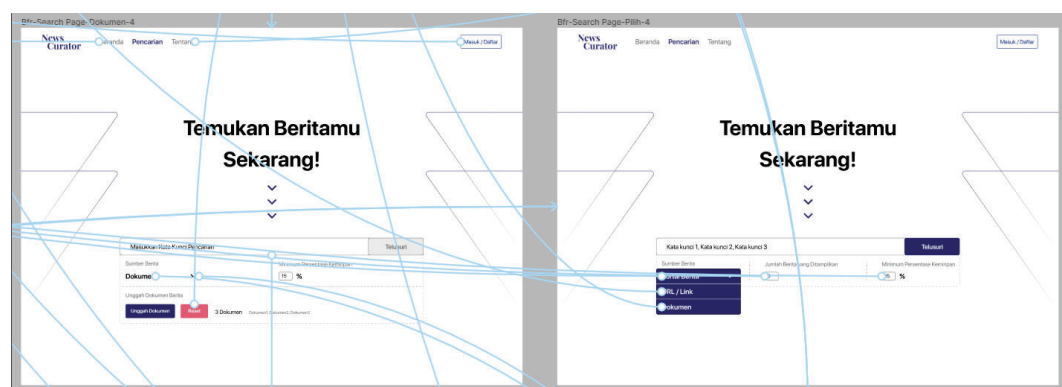
Desain antarmuka pengguna dilanjutkan dan memperbaiki dari iterasi sebelumnya, namun perbaikan yang dilakukan berdasarkan masukan yang diberikan. Gambar 3.5 ini merupakan contoh perbaikan desain antarmuka.



Gambar 3.5 Perbandingan Desain Iterasi Pertama dan Kedua

B. Build Prototype

Penentuan alur desain antarmuka tidak memiliki banyak perubahan, hanya menambahkan dan memfungsikan tombol ataupun komponen sebagai interaksi untuk perpindahan antar halaman atau elemen pendukung. Gambar 3.6 merupakan perbaikan alur desain berdasarkan masukan pengguna.



Gambar 3.6 Perbaikan *Prototype* Desain

C. User Evaluation

Uji coba desain *prototype* dilakukan kembali menggunakan layanan maze, dengan pertanyaan yang sama, dan alur akhir yang sama dengan iterasi pertama.

Namun yang menjadi perbedaan adalah desain yang digunakan merupakan hasil perbaikan desain pada iterasi kedua. Responden yang diperoleh sebanyak 30 responden.

D. Refine from User Suggestions

Masukan yang diperoleh pada tahap sebelumnya disaring kembali menjadi rangkuman komentar dan masukan yang terdapat pada Tabel 3.8. Hasil lengkap komentar, masukan, kritik maupun saran untuk iterasi kedua dapat dilihat pada Lampiran 2 tepatnya pada Tabel L2.1.

Tabel 3.8 Rangkuman Masukan Iterasi Kedua

No.	Masukan	Jumlah	Persentase
1.	Bagus	17	56.67%
2.	Perlu disempurnakan	5	16.67%
3.	Tidak ada komentar	3	10%
4.	Memperbaiki navigasi	3	10%
5.	Memperbaiki animasi	2	6.67%

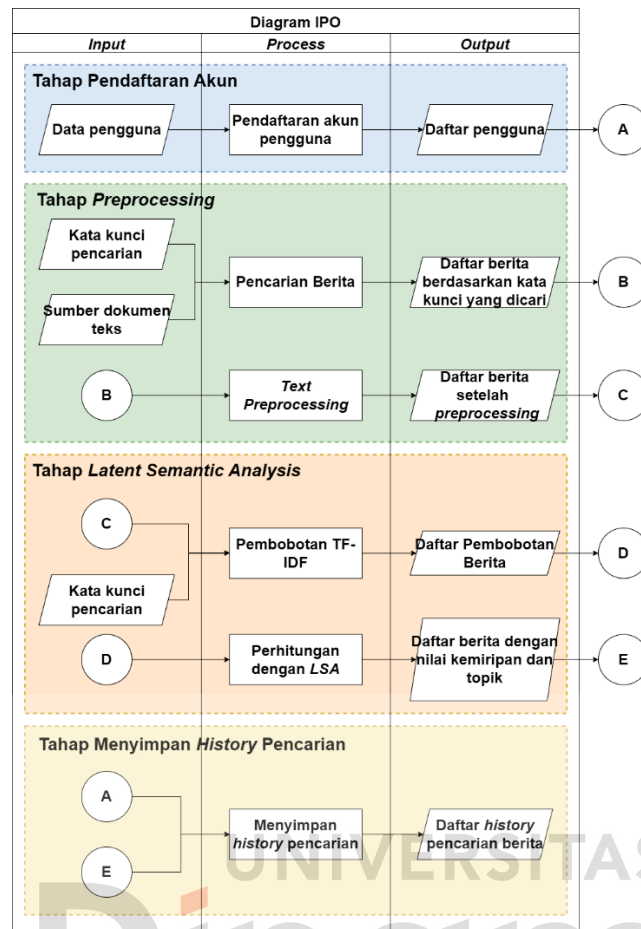
Diperoleh hasil dari iterasi kedua dengan respon dan masukan yang bagus oleh responden lebih dari 50%, sehingga desain beserta alur desain dapat dilanjutkan ke fase selanjutnya.

3.3 Design and Planning

Pada tahap yang ketiga yaitu *design and planning* merupakan tahapan dimana dilakukan perencanaan terhadap pengembangan sistem atau *website* dengan pendekatan terstruktur yang meliputi, pembentukan Diagram IPO, *System Flow*, *Context Diagram*, *Data Flow Diagram (DFD)*, dan Struktur Dokumen *Database*.

3.3.1 Diagram IPO

Diagram IPO dibentuk dengan tujuan untuk mengidentifikasi proses dan alur berjalannya dari sistem yang akan dibangun dan digambarkan pada diagram *input*, *process*, dan *output* yang dapat dilihat pada Gambar 3.7 di bawah ini.



Gambar 3.7 Diagram IPO



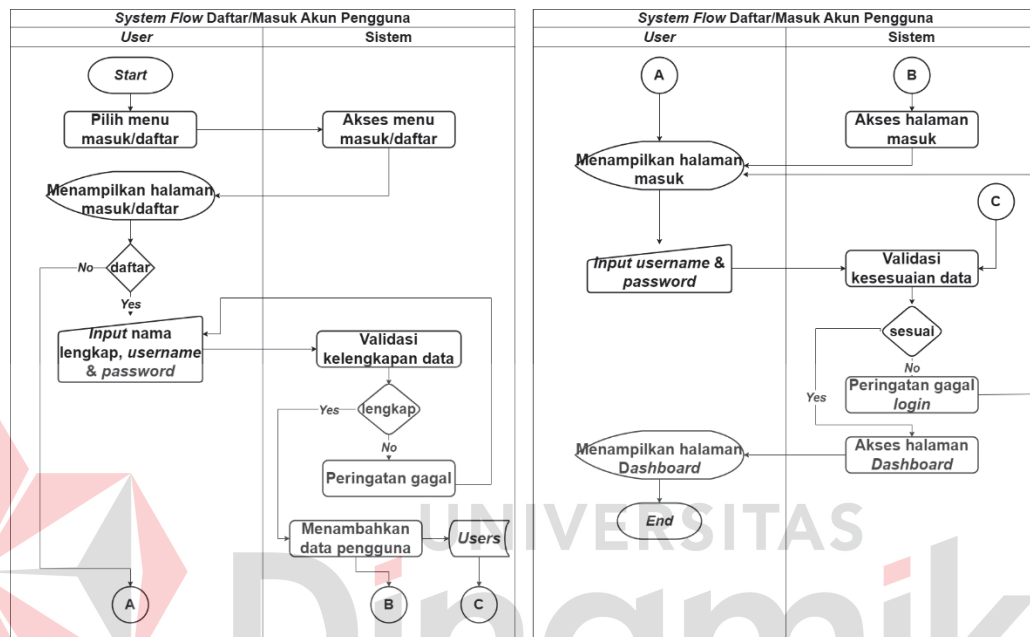
3.3.2 System Flow

System flow merupakan gambaran visual dari serangkaian Langkah atau aktivitas yang perlu dibentuk dalam suatu rangkaian sistem guna mencapai tujuan yang diinginkan. Diagram yang akan dipaparkan nantinya berguna untuk membantu dan meningkatkan pemahaman tentang sistem, alur sistem, serta kemungkinan masalah yang dapat terjadi.

A. System Flow Pendaftaran Akun

Alur tahap pendaftaran akun pada Gambar 3.8 di bawah diawali dengan pengguna memilih halaman *register* kemudian sistem menampilkan halaman *register* tersebut. Setelah halaman *register* tampil dengan sempurna maka pengguna dapat mengisi *form* yang harus diisi dengan data kebutuhan pengguna untuk melakukan pendaftaran akun. Setelah *form* terisi maka pengguna dapat langsung mendaftarkan akun dan akun baru tersimpan pada tabel *users* dan pengguna akan

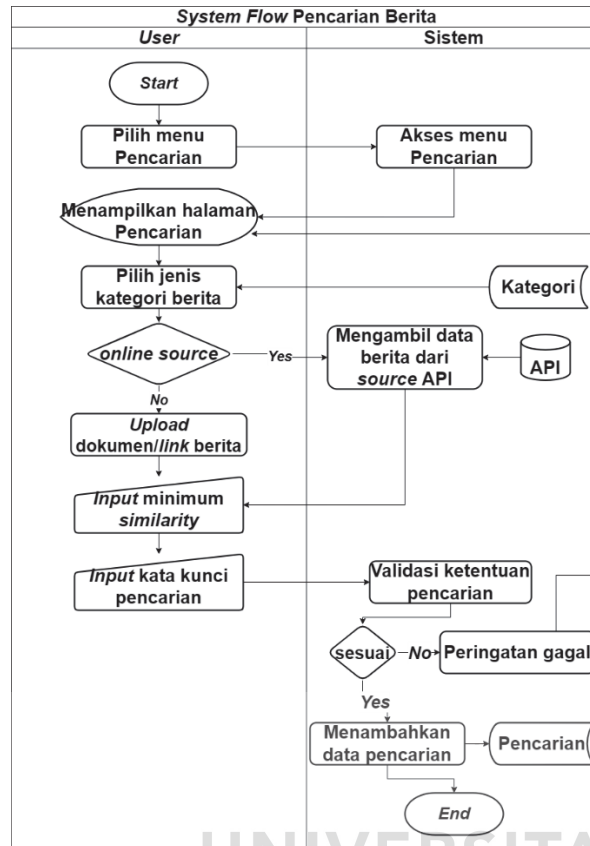
diarahkan menuju halaman *login* dimana pengguna dapat memasukkan *username* dan juga *password* yang telah dibuat sebelumnya dengan benar. Lalu sistem akan melakukan validasi terhadap data dari *form login* dengan data pengguna yang telah didaftarkan sebelumnya. Jika hal tersebut dinyatakan benar maka pengguna diarahkan menuju halaman *dashboard*.



Gambar 3.8 System Flow Pendaftaran Akun

B. System Flow Pencarian Berita

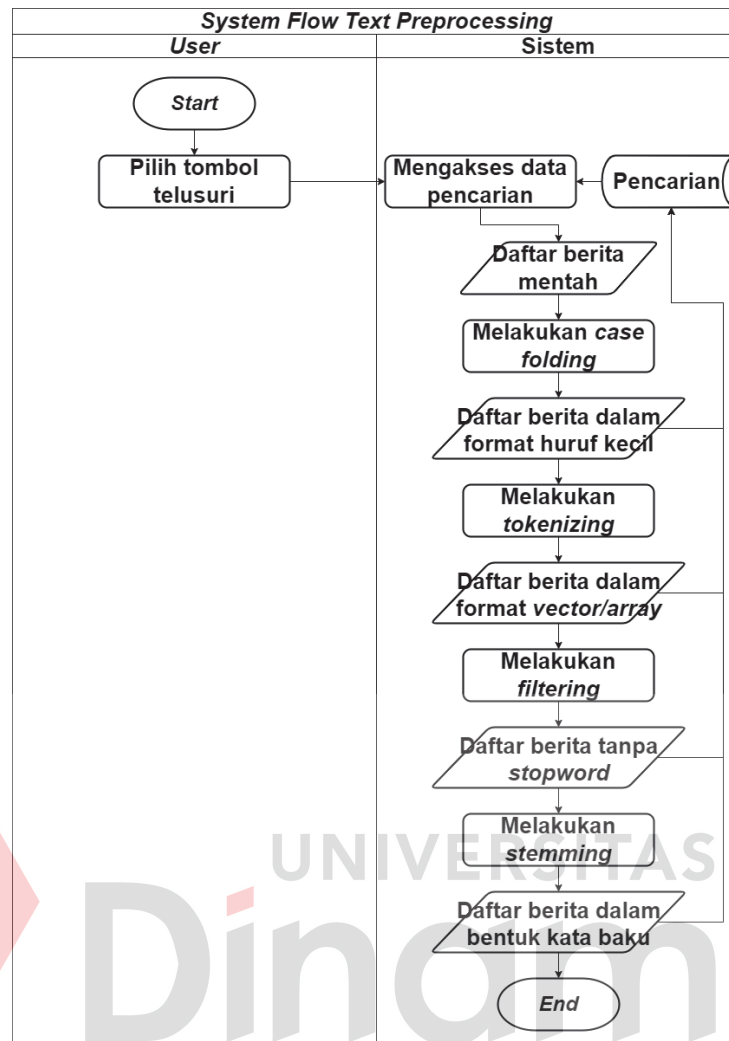
Alur sistem pencarian berita pada Gambar 3.9 yang akan disajikan di bawah diawali dengan memilih menu pencarian, kemudian dilanjutkan memilih jenis atau kategori sumber berita apakah itu berasal dari sumber *online* atau melakukan unggah secara manual dokumen berita. Jika memilih sumber dari *online* maka data berita diambil dari API yang telah disediakan. Namun jika memilih sumber berasal dari dokumen berita yang dimiliki, maka pengguna wajib mengunggah dokumen berita tersebut terlebih dahulu. Kemudian pengguna diharuskan untuk mengisi keperluan pencarian seperti kata kunci pencarian, dan kesamaan terkecil yang diinginkan. Jika sistem telah memvalidasi dan memenuhi seluruhnya maka informasi tersebut disimpan untuk tahap perhitungan selanjutnya.



Gambar 3.9 System Flow Pencarian Berita

C. System Flow Tahap Text Preprocessing

Alur sistem untuk tahap *text preprocessing* disajikan pada Gambar 3.10 dimana pengguna menekan tombol telusuri, selanjutnya perhitungan *text preprocessing* dilakukan oleh sistem diawali dari daftar berita mentah dilakukan *case folding* atau membuat berita mentah memiliki huruf kecil secara keseluruhan. Selanjutnya *tokenizing* membuat data berita tersebut dalam *format* array secara satuan, kemudian *filtering* dimana menghilangkan kata-kata yang tidak memiliki pengaruh terhadap penelitian seperti kata hubung atau kata sambung. Terakhir merupakan *stemming* dimana menjadikan kata dalam data berita menjadi kata baku secara keseluruhan.

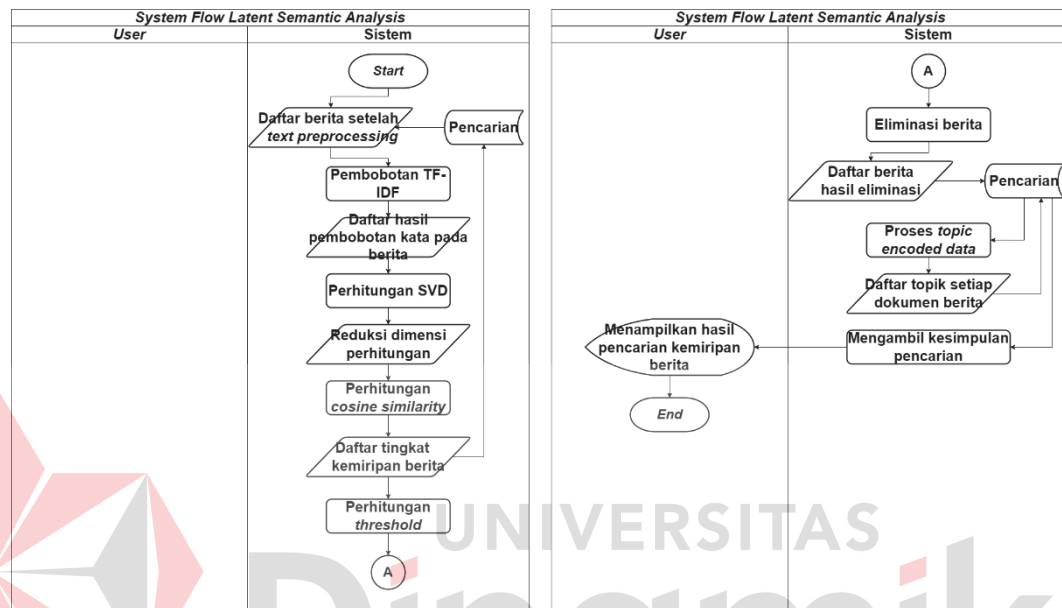


Gambar 3.10 System Flow Tahap Text Preprocessing

D. System Flow Tahap Latent Semantic Analysis

Alur perhitungan utama pada pembuatan *website* ini merupakan perhitungan LSA yang dapat dilihat pada Gambar 3.11. Diawali dengan menggunakan daftar berita setelah tahap teks preprocessing kemudian dilakukan pembobotan dengan menggunakan TF-IDF. Setelah hasil TF-IDF keluar maka dilakukan perhitungan dengan SVD, dan hasil perhitungan SVD tersebut digunakan untuk tahap selanjutnya yaitu perhitungan kesamaan antara berita dengan kata kunci pencarian atau *cosine similarity*. Setelah hasil kemiripan berita keluar maka dilakukan perhitungan untuk menentukan ambang batas bawah atau *threshold* sebelum dapat dilanjutkan untuk mengekstrak dari topik suatu berita. Nilai *threshold* tersebut digunakan untuk mengeliminasi berita yang nilai kemiripannya tidak mencukupi

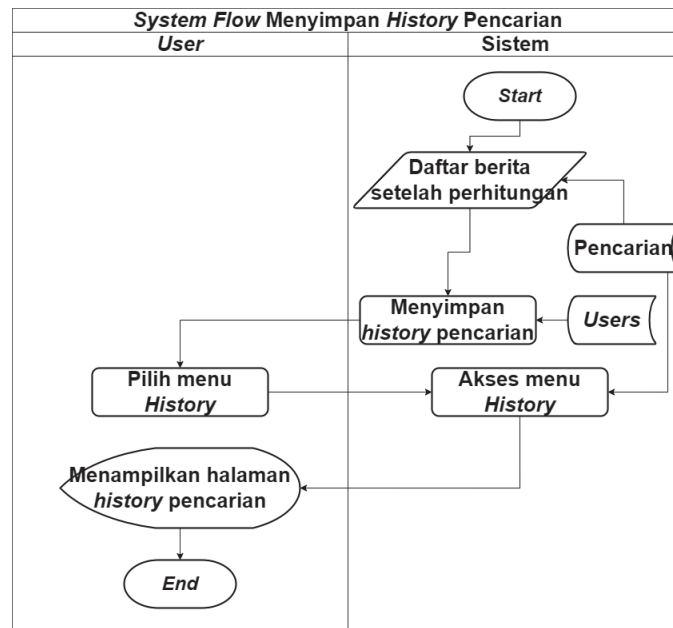
atau tidak melebihi dari nilai *threshold*. Berita yang tidak tereliminasi akan dilanjutkan ke proses ekstraksi topik dari suatu berita dengan menggunakan kembali hasil perhitungan SVD. Setelah keseluruhan proses telah berhasil dilakukan, maka berita tersebut akan ditampilkan sejumlah permintaan atau pilihan pengguna.



Gambar 3.11 System Flow Perhitungan LSA

E. System Flow Tahap Menyimpan *History* Pencarian

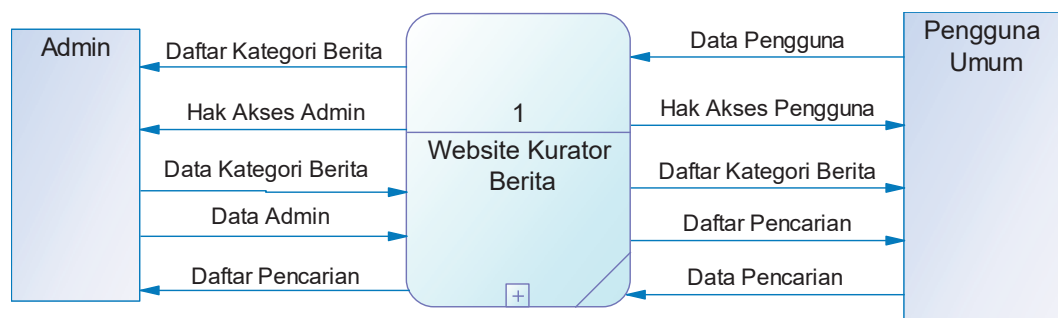
Pada alur tahapan menyimpan *history* pencarian pada Gambar 3.12 di bawah ini diawali dengan menggunakan daftar berita hasil perhitungan LSA. Berita yang memiliki nilai kesamaan yang lebih tinggi dari *threshold* akan dapat disimpan pada tabel *history* jika pengguna telah mendaftarkan akunnya ke dalam sistem.



Gambar 3.12 System Flow Menyimpan History Pencarian

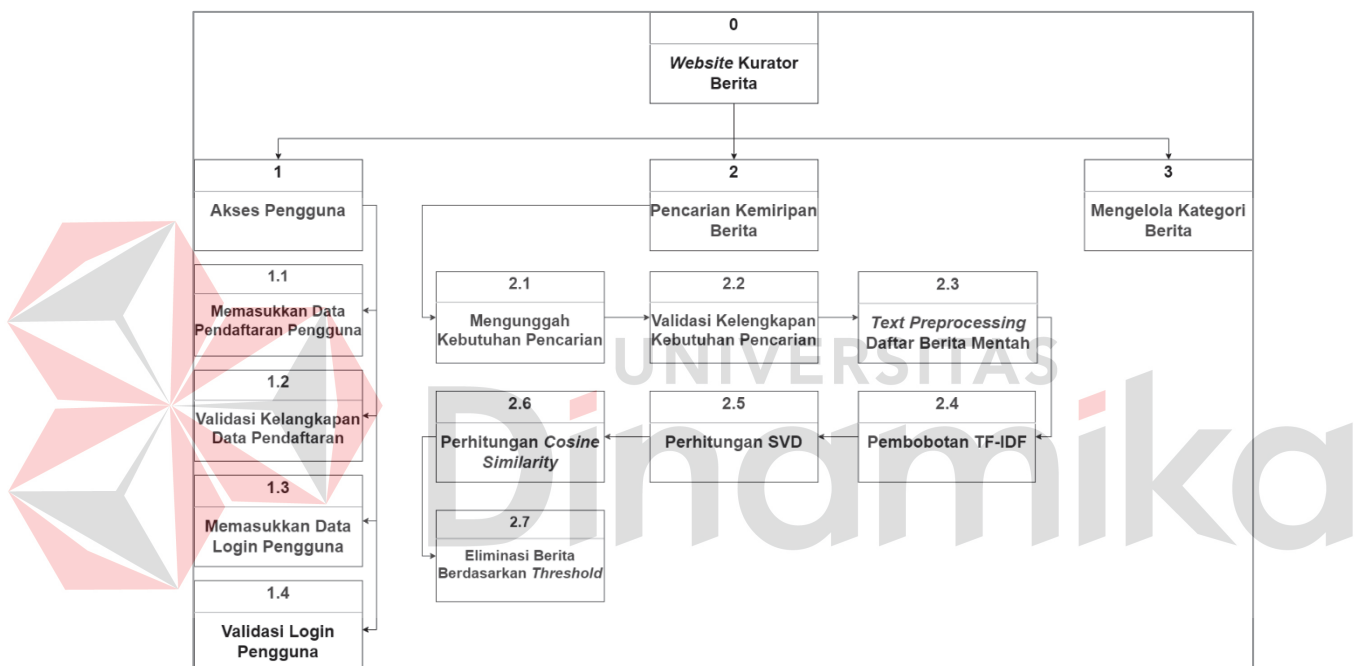
3.3.3 Context Diagram

Context diagram mengilustrasikan sistem sebagai suatu entitas pusat dengan entitas eksternal lainnya seperti pengguna atau alur *input* dan *output* dengan tujuan untuk memvisualisasikan aliran informasi yang saling terlibat. Pada Gambar 3.13 terdapat satu proses inti yaitu *website* kurator berita itu sendiri, terdapat dua aktor yaitu admin dan pengguna umum. Admin melakukan *input data* yaitu data kategori berita dan data admin. Admin juga mendapatkan *output* dari proses yaitu daftar kategori berita, hak akses admin, dan daftar pencarian. Aktor pengguna umum melakukan *input* berupa data pengguna dan data pencarian. Pengguna umum juga mendapatkan *output* berupa hak akses pengguna, daftar kategori berita, dan daftar pencarian.



Gambar 3.13 Context Diagram

Setelah terbentuknya *context diagram* dilanjutkan dengan membentuk diagram jenjang merupakan bentuk diagram yang digunakan untuk menggambarkan hubungan secara hierarki antar komponen sistem. Tujuannya untuk memvisualisasikan struktur dengan teratur dalam bentuk tingkatan-tingkatan yang nantinya akan diimplementasikan pada *data flow diagram* untuk alur data yang lebih kompleks. Terdapat tiga proses utama yaitu Akses Pengguna, Mengelola Kemiripan Berita dan Pencarian Kemiripan Berita yang dapat dilihat pada Gambar 3.14.



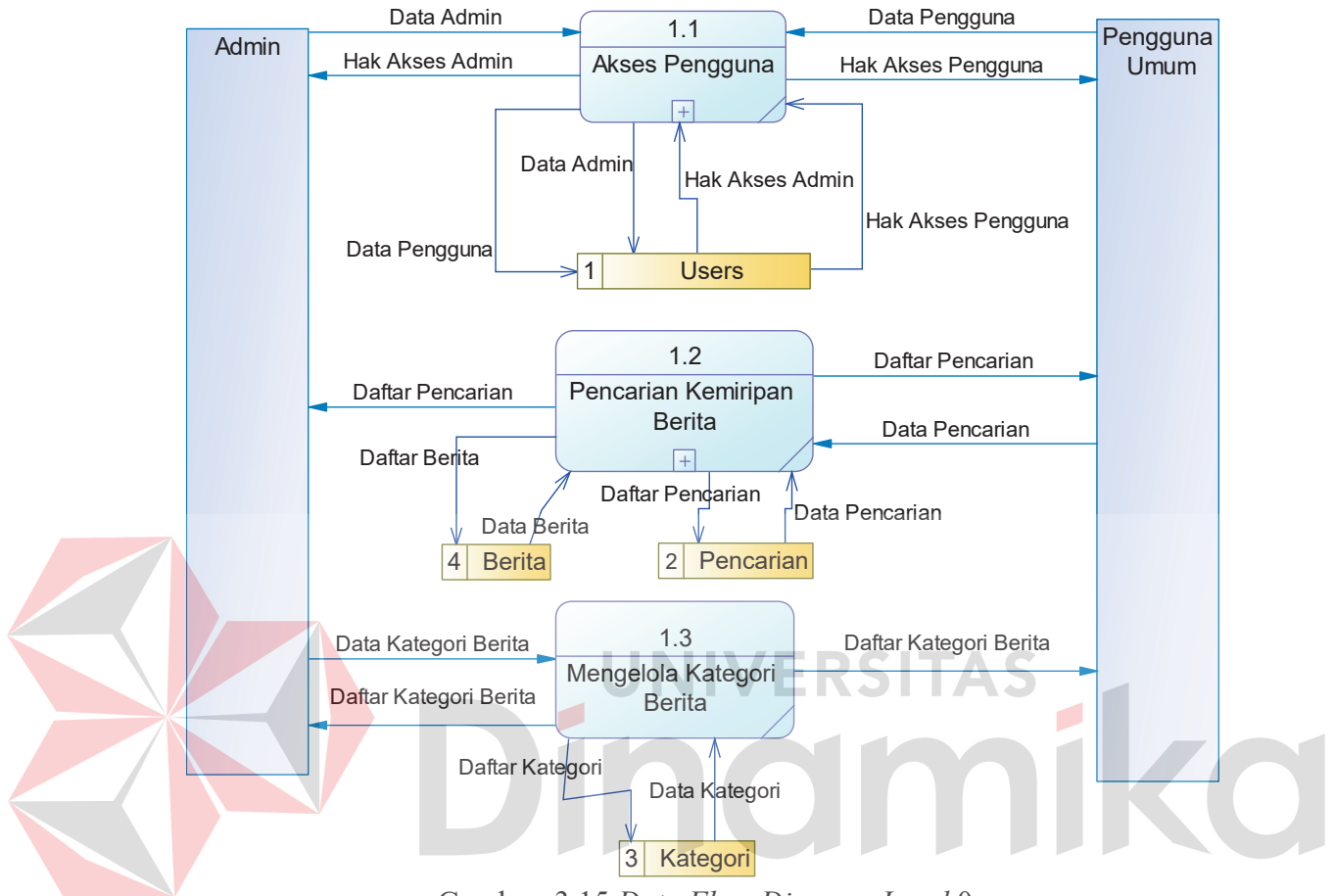
Gambar 3.14 Diagram Jenjang

3.3.4 Data Flow Diagram (DFD)

A. Data Flow Diagram Level 0

Pada Gambar 3.15 merupakan gambaran aliran data utama yang digambarkan secara *general* dimana terdapat entitas utama dan entitas eksternal lainnya yang dihubungkan dengan panah sebagai gambaran data yang bergerak dalam sistem. Pada hal ini terdapat tiga proses utama yang sama dengan yang telah dijelaskan sebelumnya pada diagram jenjang yaitu, akses pengguna, pencarian kemiripan berita, dan mengelola kategori berita. Terdapat juga dua aktor yang berperan dalam proses pertukaran data yaitu admin dan pengguna umum. Terdapat empat tempat

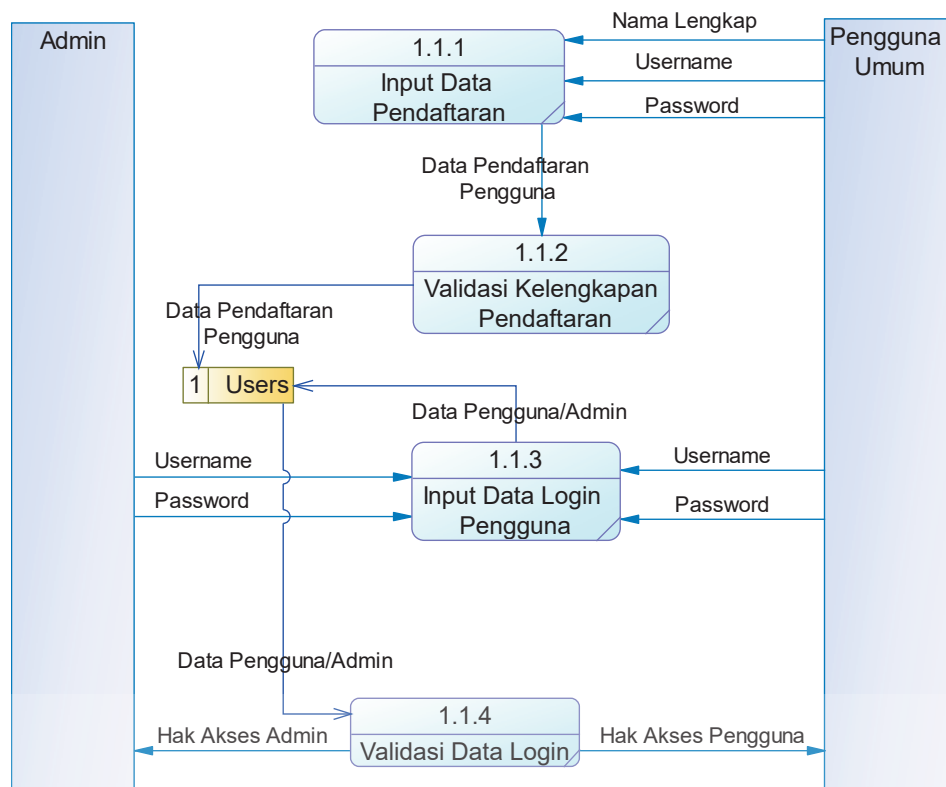
penyimpanan data sebagai tempat penyimpanan daftar hasil proses yang telah dilakukan.



Gambar 3.15 Data Flow Diagram Level 0

B. Data Flow Diagram Level 1 Akses Pengguna

DFD *Level 1* untuk akses pengguna menggambarkan bagaimana dua aktor atau pengguna memasukkan data pengguna, kemudian data tersebut divalidasi berdasarkan data pengguna yang telah tersimpan sebelumnya. Setelah seluruh data tervalidasi maka pengguna mendapatkan hak akses untuk *login*. DFD *Level 1* untuk akses pengguna dapat dilihat pada Gambar 3.16.



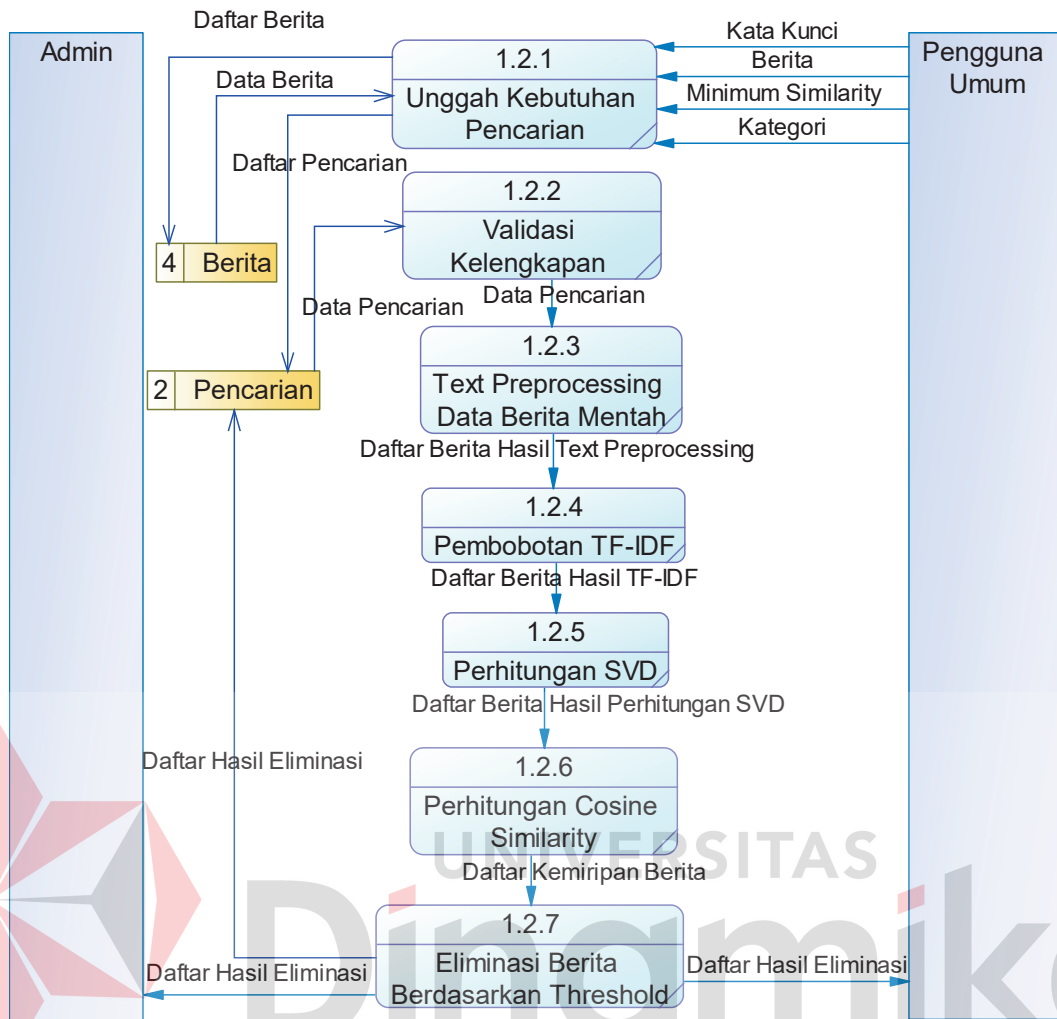
Gambar 3.16 Data Flow Diagram Level 1 Akses Pengguna

C. Data Flow Diagram Level 1 Pencarian Kemiripan Berita

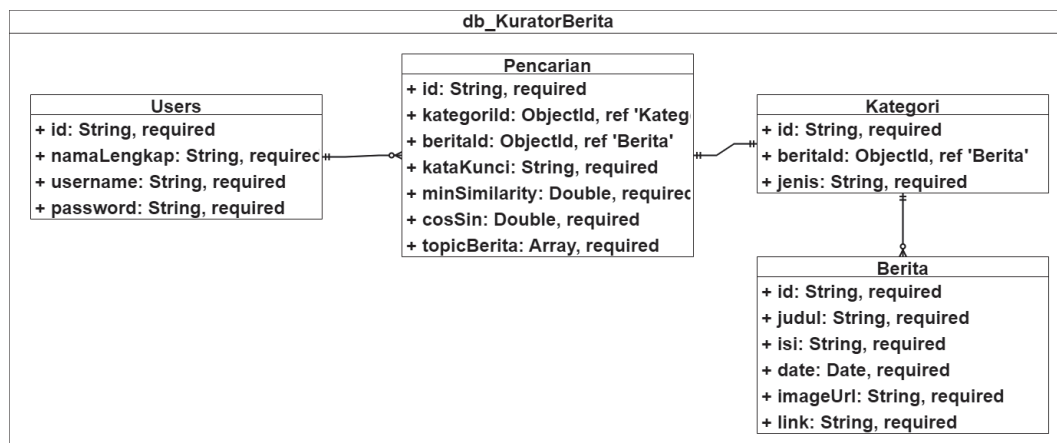
Pada DFD *Level 1* pencarian kemiripan berita, pengguna umum mengambil peran penting dimana kebutuhan untuk pencarian seperti kata kunci, data berita, tingkat kemiripan minimum, dan kategori berita diproses untuk perhitungan pencarian. Setelah perhitungan pencarian berhasil dilakukan maka hasil tersebut disimpan pada data pencarian dan dapat dilihat oleh pengguna umum dan juga admin. Pada Gambar 3.17 merupakan DFD *Level 1* pencarian kemiripan berita.

3.3.5 Struktur Dokumen Database

Database sebagai media penyimpanan data memiliki jenis NoSQL atau non-relasional *database*. Sehingga tanpa adanya relasi antar dokumen, *database* tetap dapat terhubung dan terkoneksi. Bentuk penyimpanan data menggunakan format JSON yang tentunya fleksibel tanpa perlu memperhatikan skema lainnya. Pada Gambar 3.18 dipaparkan bagaimana *database* pada sistem ini berjalan.



Gambar 3.17 Data Flow Diagram Level 1 Pencarian Kemiripan Berita



Gambar 3.18 Struktur Dokumen Database

3.4 *Development*

Tahap keempat adalah *development* dimana pada tahap ini merupakan tahapan dimana pengembangan *website* yang merupakan solusi pada penelitian ini dilakukan. Dilakukan juga *deployment* dari *website* yang telah dibangun kepada pihak penyedia *hosting* secara gratis dengan tujuan agar *website* tidak hanya dapat digunakan oleh pengembang saja melainkan dapat digunakan oleh target pengguna yaitu masyarakat umum.

3.5 *Testing*

Pada tahap yang kelima merupakan tahapan *testing* dimana pada tahap ini merupakan tahapan yang dilakukan untuk melakukan uji coba terhadap akurasi keakuratan hasil pencarian berita dengan kata kunci pencarian menggunakan *confusion matrix*, fungsionalitas *website* maupun algoritma yang digunakan untuk metode perhitungannya menggunakan BBT, dan uji kegunaan *website* dengan melibatkan sebanyak minimal 96 responden untuk mengisi kuesioner berdasarkan pada ketentuan SUS dan juga disediakan pertanyaan khusus terkait pengalaman dan masukan yang dapat disampaikan selama menggunakan *website*. *Testing* dilakukan baik oleh pengembang sendiri maupun kepada calon pengguna untuk memperoleh pandangan dan tanggapan lain terhadap *website* yang telah dibentuk.

Daftar skenario uji coba untuk BBT yang harus diuji setelah *website* berhasil dibentuk dan untuk keseluruhan skenario uji coba dapat dilihat pada Lampiran 3 pada Tabel L3.1.

3.6 *Maintenance*

Pada tahap terakhir merupakan tahap *maintenance* dimana setelah mendapatkan hasil dari *testing* yang telah dilakukan pada tahap sebelumnya. Maka dari hasil *testing* tersebut dilakukan evaluasi kembali dan hasil yang terbaik dari evaluasi tersebut akan diterapkan pada *website*.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab sebelumnya, telah dilakukan tahapan-tahapan penting dalam proses pengembangan *website* ini. Tahapan-tahapan tersebut mencakup *identify requirements*, *prototyping*, dan *design and planning*. Setiap tahapan ini memainkan peran yang sangat vital dalam memastikan kelancaran dan kesuksesan dari implementasi *website* yang menjadi fokus penelitian ini. Pada bab ini berfokus untuk menganalisis hasil dan kontribusi penelitian dalam mengembangkan *website* yang efektif dan berkualitas yang dimana melanjutkan pada tahapan sebelumnya. Tahapan-tahapan tersebut meliputi *development*, dan *testing* yang dijabarkan secara jelas di bawah ini.

4.1 *Development*

Pada fase ini merupakan tahap implementasi dari rancangan-rancangan yang telah dibentuk pada tahap sebelumnya berupa *website* pencarian kemiripan berita dengan menerapkan metode LSA didalamnya. Implementasi *website* yang dibentuk akan dipaparkan di bawah ini.

4.1.1 Implementasi Halaman Pencarian

Halaman pencarian merupakan halaman inti dari *website* yang dibangun pada penelitian ini. Pada halaman pencarian, pengguna dapat melakukan pengecekan terhadap berita yang pernah dibaca sebelumnya ataupun melakukan pencarian awal pada *website* ini secara langsung. Terdapat dua pembagian sumber berita yang dapat menjadi pilihan pengguna untuk menentukan sumber berita yang diinginkan, terdapat portal berita, dan dokumen. Pengguna dapat memasukkan kata kunci yang diinginkan, menentukan minimum kemiripan, dan banyaknya hasil berita yang ditampilkan. Gambar 4.1 merupakan implementasi untuk halaman pencarian.

Gambar 4.1 Tampilan Halaman Pencarian

4.1.2 Menampung Data Pencarian Pengguna

Data berita mentah yang telah ditentukan oleh pengguna akan dikumpulkan menjadi satu untuk nantinya dapat diolah. Kata kunci pencarian juga termasuk kedalam data mentah yang memiliki peran penting untuk proses pencarian. Data berita mentah dan kata kunci pencarian yang telah dikumpulkan dapat dilihat pada Tabel 4.1 di bawah ini, dimana simbol Q menggambarkan *query* atau kata kunci pencarian.

Tabel 4.1 Data Berita Mentah

No	Contoh Berita
1	Daftar Harga BBM Terbaru 2022 November, Cek di Sini!
2	Daftar BBM Terbaru 15 November 2022, Ada yang Turun Lagi?
3	Keluh Kesah Setelah Harga BBM Naik, Antrean Peralite di SPBU Mengular hingga Beralih dari Pertamina
4	Sudah Diturunkan, Ini Daftar Harga Pertamina Turbo Hingga Peralite 18 November 2022
5	Ada yang Naik, Ini Daftar Terbaru Harga BBM Pertamina di SPBU
Q	Kenaikan Harga BBM Terbaru

4.1.3 Proses Perhitungan LSA Berdasarkan Data Pencarian

Dalam menentukan hasil pencarian yang relevan diperlukan beberapa tahapan perhitungan diantaranya sebagai berikut:

1. *Case Folding*

Data berita mentah yang telah dikumpulkan terlebih dahulu dilakukan pemrosesan kata, agar kata yang diproses nantinya tidak menjadi rancu dan dapat

menyebabkan kesalahan dalam perhitungan. Pada tahap ini dilakukan *case folding* dimana data-data berita dilakukan pemrosesan dengan mengubah menjadi ke dalam bentuk huruf kecil secara keseluruhan. Tahap *case folding* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Implementasi *Case Folding*

No	Sebelum	Sesudah
1	Daftar Harga BBM Terbaru 2022 November, Cek di Sini!	daftar harga bbm terbaru 2022 november, cek di sini!
2	Daftar BBM Terbaru 15 November 2022, Ada yang Turun Lagi?	daftar bbm terbaru 15 november 2022, ada yang turun lagi?
3	Keluh Kesah Setelah Harga BBM Naik, Antrean Peralite di SPBU Mengular hingga Beralih dari Pertamina	keluh kesah setelah harga bbm naik, antrean peralite di spbu mengular hingga beralih dari pertamina
4	Sudah Diturunkan, Ini Daftar Harga Pertamina Turbo Hingga Peralite 18 November 2022	sudah diturunkan, ini daftar harga pertamax turbo hingga peralite 18 november 2022
5	Ada yang Naik, Ini Daftar Terbaru Harga BBM Pertamina di SPBU	ada yang naik, ini daftar terbaru harga bbm pertamina di spbu
Q	Kenaikan Harga BBM Terbaru	kenaikan harga bbm terbaru

2. *Tokenizing*

Data berita hasil *case folding* digunakan pada tahap ini, dimana pada tahap ini data berita diolah menjadi dalam bentuk token-token atau yang dikenal dengan istilah tipe data *array* sehingga setiap kata terpisah dari kata lainnya. *Tokenizing* selain menjadikan data menjadi bentuk token, *tokenizing* juga menghilangkan angka maupun tanda baca yang tidak memiliki pengaruh dalam pemrosesan data. Pada Tabel 4.3 merupakan implementasi dari *tokenizing*.

Tabel 4.3 Implementasi *Tokenizing*

No	Sebelum	Sesudah
1	daftar harga bbm terbaru 2022 november, cek di sini!	[daftar, harga, bbm, terbaru, november, cek, di, sini]
2	daftar harga bbm terbaru 15 november 2022, ada yang turun lagi?	[daftar, bbm, terbaru, november, ada, yang, turun, lagi]
3	keluh kesah setelah harga bbm naik, antrean peralite di spbu mengular hingga beralih dari pertamina	[keluh, kesah, setelah, harga, bbm, naik, antrean, peralite, di, spbu, mengular, hingga, beralih, dari, pertamina]
4	sudah diturunkan, ini daftar harga Pertamina Turbo hingga Peralite 18 November 2022	[sudah, diturunkan, ini, daftar, harga, pertamax, turbo, hingga, peralite, november]
5	ada yang naik, ini daftar terbaru harga BBM Pertamina di SPBU	[ada, yang, naik, ini, daftar, terbaru, harga, bbm, pertamina, di, spbu]
Q	kenaikan harga bbm terbaru	[kenaikan, harga, bbm, terbaru]

3. *Filtering*

Pada tahap *filtering* dilakukan penghapusan pada data berita yang memiliki serangkaian kata yang umum namun tidak informatif, seperti kata hubung, kata sambung, dan sejenisnya. Implementasi *filtering* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Implementasi *Filtering*

No	Sebelum	Sesudah
1	[daftar, harga, bbm, terbaru, november, cek, di, sini]	[daftar, harga, bbm, terbaru, november, cek]
2	[daftar, harga, bbm, terbaru, november, ada, yang, turun, lagi]	[daftar, bbm, terbaru, november, turun]
3	[keluh, kesah, setelah, harga, bbm, naik, antrean, pertalite, di, spbu, mengular, hingga, beralih, dari, pertamina]	[keluh, kesah, harga, bbm, naik, antrean, pertalite, spbu, mengular, beralih, pertamina]
4	[sudah, diturunkan, ini, daftar, harga, pertamax, turbo, hingga, pertalite, november]	[diturunkan, daftar, harga, pertamax, turbo, pertalite, november]
5	[ada, yang, naik, ini, daftar, terbaru, harga, bbm, pertamina, di, spbu]	[naik, daftar, terbaru, harga, bbm, pertamina, spbu]
Q	[kenaikan, harga, bbm, terbaru]	[kenaikan, harga, bbm, terbaru]

4. *Stemming*

Tahap akhir pada proses *text preprocessing* adalah *stemming*. *Stemming* berperan untuk menjadikan kembali setiap kata pada berita ke dalam bentuk baku kata tersebut. Hasil implementasi *stemming* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Implementasi *Stemming*

No	Sebelum	Sesudah
1	[daftar, harga, bbm, terbaru, november, cek]	[daftar, harga, bbm, baru, november, cek]
2	[daftar, bbm, terbaru, november, turun]	[daftar, bbm, baru, november, turun]
3	[keluh, kesah, harga, bbm, naik, antrean, pertalite, spbu, mengular, beralih, pertamina]	[keluh, kesah, harga, bbm, naik, antre, pertalite, spbu, ular, alih, pertamina]
4	[diturunkan, daftar, harga, pertamax, turbo, pertalite, november]	[turun, daftar, harga, pertamax, turbo, pertalite, november]
5	[naik, daftar, terbaru, harga, bbm, pertamina, spbu]	[naik, daftar, baru, harga, bbm, pertamina, spbu]
Q	[kenaikan, harga, bbm, terbaru]	[naik, harga, bbm, baru]

5. TF-IDF

TF-IDF awalnya melakukan pencarian term atau kata yang terdapat pada setiap dokumen. Kemudian dilakukan perhitungan TF yaitu dengan menghitung banyaknya term yang muncul pada setiap dokumen seperti pada contoh kata

“daftar” muncul sebanyak 1 kali pada dokumen 1, 2, 4, dan 5. Kemudian dilanjutkan dengan perhitungan DF yaitu menjumlahkan keseluruhan kata yang muncul seperti pada contoh kata “daftar” muncul sebanyak 4 kali dari total 6 dokumen. Dilanjutkan lagi dengan menghitung IDF berdasarkan rumus (2).

$$\text{Dimana } IDF = \log\left(\frac{6}{4}\right) = 0,1761$$

Perhitungan diakhiri dengan menghitung TF-IDF pada masing-masing dokumen terhadap suatu kata disini menggunakan contoh kata “daftar” yaitu dengan mengkalikan TF dengan IDF.

Dimana berdasarkan rumus (3):

$$TFIDF_{d1} = 1 \times 0,1761 = 0,1761$$

$$TFIDF_{d2} = 1 \times 0,1761 = 0,1761$$

$$TFIDF_{d4} = 1 \times 0,1761 = 0,1761$$

$$TFIDF_{d5} = 1 \times 0,1761 = 0,1761$$

Tabel 4.6 Pembobotan Kata dengan TF-IDF

term	tf					df	n/df	idf=1 log(n/df)	tf-idf						
	d1	d2	d3	d4	d5				d1	d2	d3	d4	d5	d6	
daftar	1	1	0	1	1	4	1,5	0,1761	0,17	0,17	0,00	0,17	0,17	0,00	0,00
harga	1	0	1	1	1	5	1,2	0,0792	0,07	0,00	0,07	0,07	0,07	0,07	0,07
bbm	1	1	1	0	1	5	1,2	0,0792	0,07	0,07	0,07	0,00	0,07	0,07	0,07
baru	1	1	0	0	1	4	1,5	0,1761	0,17	0,17	0,00	0,00	0,17	0,17	0,00
november	1	1	0	1	0	3	2	0,3010	0,30	0,30	0,00	0,30	0,00	0,00	0,00
ceker	1	0	0	0	0	1	6	0,7782	0,77	0,00	0,00	0,00	0,00	0,00	0,00
turun	0	1	0	1	0	2	3	0,4771	0,00	0,47	0,00	0,47	0,00	0,00	0,00
kelu	0	0	1	0	0	1	6	0,7782	0,00	0,00	0,77	0,00	0,00	0,00	0,00
kesa	0	0	1	0	0	1	6	0,7782	0,00	0,00	0,77	0,00	0,00	0,00	0,00
naik	0	0	1	0	1	3	2	0,3010	0,00	0,00	0,30	0,00	0,30	0,30	0,00
antr	0	0	1	0	0	1	6	0,7782	0,00	0,00	0,77	0,00	0,00	0,00	0,00
perte	0	0	1	1	0	2	3	0,4771	0,00	0,00	0,47	0,47	0,00	0,00	0,00
spbu	0	0	1	0	1	2	3	0,4771	0,00	0,00	0,47	0,00	0,47	0,00	0,00
ular	0	0	1	0	0	1	6	0,7782	0,00	0,00	0,77	0,00	0,00	0,00	0,00

term	tf						df	n/df	idf=1 log(n/df)	tf-idf					
	d1	d2	d3	d4	d5	q				d1	d2	d3	d4	d5	d6
alih	0	0	1	0	0	0	1	6	0,778	0,00	0,00	0,77	0,00	0,00	0,00
									2	00	00	82	00	00	00
pertami	0	0	1	0	1	0	2	3	0,477	0,00	0,00	0,47	0,00	0,47	0,00
									1	00	00	71	00	71	00
pertama	0	0	0	1	0	0	1	6	0,778	0,00	0,00	0,00	0,77	0,00	0,00
									2	00	00	00	82	00	00
x															
turb	0	0	0	1	0	0	1	6	0,778	0,00	0,00	0,00	0,77	0,00	0,00
									2	00	00	00	82	00	00

6. SVD

SVD merupakan tahapan pada LSA dimana SVD ini melakukan reduksi dimensi atau variabel yang dijadikan sebagai penelitian, dimana pada banyaknya variabel tersebut memiliki korelasi yang tidak terpaut jauh dengan variabel lainnya.

Sehingga oleh karena hal tersebut SVD melakukan reduksi dimensi atau menghilangkan beberapa variabel yang memiliki hubungan dekat dengan variabel lain dengan tujuan untuk memangkas waktu perhitungan yang terhitung cukup rumit. Pada perhitungan yang akan disajikan di bawah ini berdasarkan rumus (4) melanjutkan data perhitungan dari TF-IDF dan dikarenakan pada kasus ini akan menentukan topik yang diangkat pada masing-masing dokumen maka reduksi yang dilakukan hanya sebatas jumlah total dokumen.

$$A = (A \times A^T) \times \Sigma \times (A^T \times A)$$

$$A \times A^T =$$

0,1761	0,1761	0,0000	0,1761	0,1761	0,0000
0,0792	0,0000	0,0792	0,0792	0,0792	0,0792
0,0792	0,0792	0,0792	0,0000	0,0792	0,0792
0,1761	0,1761	0,0000	0,0000	0,1761	0,1761
0,3010	0,3010	0,0000	0,3010	0,0000	0,0000
0,7782	0,0000	0,0000	0,0000	0,0000	0,0000
0,0000	0,4771	0,0000	0,4771	0,0000	0,0000
0,0000	0,0000	0,7782	0,0000	0,0000	0,0000
0,0000	0,0000	0,7782	0,0000	0,0000	0,0000
0,0000	0,0000	0,3010	0,0000	0,3010	0,3010
0,0000	0,0000	0,7782	0,0000	0,0000	0,0000
0,0000	0,0000	0,4771	0,4771	0,0000	0,0000
0,0000	0,0000	0,4771	0,0000	0,4771	0,0000
0,0000	0,0000	0,7782	0,0000	0,0000	0,0000
0,0000	0,0000	0,7782	0,0000	0,0000	0,0000
0,0000	0,0000	0,4771	0,0000	0,4771	0,0000
0,0000	0,0000	0,0000	0,7782	0,0000	0,0000
0,0000	0,0000	0,0000	0,7782	0,0000	0,0000

X

0,1761	0,07	0,07	0,17	0,30	0,7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	92	92	61	10	78	00	00	00	00	00	00	00	00	00	00	00	00	00
					2	0	0	0	0	0	0	0	0	0	0	0	0	0
0,1761	0,00	0,07	0,17	0,30	0,0	0,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	00	92	61	10	00	77	00	00	00	00	00	00	00	00	00	00	00	00
					0	1	0	0	0	0	0	0	0	0	0	0	0	0
0,0000	0,07	0,07	0,00	0,00	0,0	0,0	0,7	0,7	0,3	0,7	0,4	0,4	0,7	0,7	0,4	0,0	0,0	0,0
	92	92	00	00	00	00	78	78	01	78	77	77	78	78	77	00	00	00
					0	0	2	2	0	2	1	1	2	2	1	0	0	0
0,1761	0,07	0,00	0,00	0,30	0,0	0,4	0,0	0,0	0,0	0,0	0,4	0,0	0,0	0,0	0,0	0,7	0,7	0,7
	92	00	00	10	00	77	00	00	00	00	77	00	00	00	00	78	78	78
					0	1	0	0	0	0	1	0	0	0	0	2	2	2
0,1761	0,07	0,07	0,17	0,00	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,4	0,0	0,0	0,4	0,0	0,0	0,0
	92	92	61	00	00	00	00	00	01	00	00	77	00	00	77	00	00	00
					0	0	0	0	0	0	0	1	0	0	1	0	0	0
0,0000	0,07	0,07	0,17	0,00	0,0	0,0	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	92	92	61	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00
					0	0	0	0	0	0	0	0	0	0	0	0	0	0

$A \times A^T =$

0,1	0,0	0,0	0,0	0,1	0,1	0,1	0,0000	0,0000	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1
24	41	41	93	59	37	68			530	000	840	840	00	00	84	37	37
0	8	8	0	0	0	0							0	0	0	0	0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0616	0,0616	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
41	31	25	41	47	61	37			715	616	756	756	61	61	75	61	61
8	3	1	8	7	6	8							6	6	6	6	6
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0616	0,0616	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
41	25	31	55	47	61	37			715	616	378	756	61	61	75	00	00
8	1	3	8	7	6	8							6	6	6	0	0
0,0	0,0	0,0	0,1	0,1	0,1	0,0	0,0000	0,0000	0,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
93	41	55	24	06	37	84			060	000	000	840	00	00	84	00	00
0	8	8	0	0	0	0							0	0	0	0	0
0,1	0,0	0,0	0,1	0,2	0,2	0,2	0,0000	0,0000	0,0	0,0	0,1	0,0	0,0	0,0	0,0	0,2	0,2
59	47	47	06	71	34	87			000	000	436	000	00	00	00	34	34
0	7	7	0	9	2	3							0	0	0	2	2
0,1	0,0	0,0	0,1	0,2	0,6	0,0	0,0000	0,0000	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
37	61	61	37	34	05	00			000	000	000	000	00	00	00	00	00
0	6	6	0	2	5	0							0	0	0	0	0
0,1	0,0	0,0	0,0	0,2	0,0	0,4	0,0000	0,0000	0,0	0,0	0,2	0,0	0,0	0,0	0,0	0,3	0,3
68	37	37	84	87	00	55			000	000	276	000	00	00	00	71	71
0	8	8	0	3	0	3							0	0	0	3	3
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,6055	0,6055	0,2	0,6	0,3	0,3	0,6	0,6	0,3	0,0	0,0
00	61	61	00	00	00	00			342	055	713	713	05	05	71	00	00
0	6	6	0	0	0	0							5	5	3	0	0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,6055	0,6055	0,2	0,6	0,3	0,3	0,6	0,6	0,3	0,0	0,0
00	61	61	00	00	00	00			342	055	713	713	05	05	71	00	00
0	6	6	0	0	0	0							5	5	3	0	0
0,0	0,0	0,0	0,1	0,0	0,0	0,0	0,2342	0,2342	0,2	0,2	0,1	0,2	0,2	0,2	0,2	0,0	0,0
53	71	71	06	00	00	00			719	342	436	873	34	34	87	00	00
0	5	5	0	0	0	0							2	2	3	0	0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,6055	0,6055	0,2	0,6	0,3	0,3	0,6	0,6	0,3	0,0	0,0
00	61	61	00	00	00	00			342	055	713	713	05	05	71	00	00
0	6	6	0	0	0	0							5	5	3	0	0
0,0	0,0	0,0	0,0	0,1	0,0	0,2	0,3713	0,3713	0,1	0,3	0,4	0,2	0,3	0,3	0,2	0,3	0,3
84	75	37	00	43	00	27			436	713	553	276	71	71	27	71	71

0	6	8	0	6	0	6							3	3	6	3	3
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3713	0,3713	0,2	0,3	0,2	0,4	0,3	0,3	0,4	0,0	0,0
84	75	75	84	00	00	00			873	713	276	553	71	71	55	00	00
0	6	6	0	0	0	0							3	3	3	0	0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,6055	0,6055	0,2	0,6	0,3	0,3	0,6	0,6	0,3	0,0	0,0
00	61	61	00	00	00	00			342	055	713	713	05	05	71	00	00
0	6	6	0	0	0	0							5	5	3	0	0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,6055	0,6055	0,2	0,6	0,3	0,3	0,6	0,6	0,3	0,0	0,0
00	61	61	00	00	00	00			342	055	713	713	05	05	71	00	00
0	6	6	0	0	0	0							5	5	3	0	0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3713	0,3713	0,2	0,3	0,2	0,4	0,3	0,3	0,4	0,0	0,0
84	75	75	84	00	00	00			873	713	276	553	71	71	55	00	00
0	6	6	0	0	0	0							3	3	3	0	0
0,1	0,0	0,0	0,0	0,2	0,0	0,3	0,0000	0,0000	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,6	0,6
37	61	00	00	34	00	71			000	000	713	000	00	00	00	05	05
0	6	0	0	2	0	3							0	0	0	5	5
0,1	0,0	0,0	0,0	0,2	0,0	0,3	0,0000	0,0000	0,0	0,0	0,3	0,0	0,0	0,0	0,0	0,6	0,6
37	61	00	00	34	00	71			000	000	713	000	00	00	00	05	05
0	6	0	0	2	0	3							0	0	0	5	5

Perhitungan untuk vektor singular U belum berhenti sampai disini, langkah selanjutnya adalah setelah mendapatkan hasil perkalian matriks. Maka selanjutnya adalah menghitung untuk menentukan nilai eigen dan vektor eigen yang bertujuan untuk normalisasi dari vektor singular U. Di bawah ini akan dilakukan perhitungan untuk menentukan nilai eigen terlebih dahulu.

$$\lambda_u = \det(A - \lambda I) = 0$$

Menghasilkan dengan urutan dari terbesar:

$$\lambda_1 = 3,94, \lambda_2 = 1,873, \lambda_3 = 0,815, \lambda_4 = 0,531, \lambda_5 = 0,264, \lambda_6 = 0,097$$

Dari nilai eigen yang telah didapatkan tersebut digunakan juga untuk menentukan nilai-nilai singular.

$$\Sigma_n = \sqrt{\lambda_u}$$

Menghasilkan nilai-nilai singular yang diurutkan dari nilai terbesar:

$$\Sigma_1 = 1,954, \Sigma_2 = 1,368, \Sigma_3 = 0,906, \Sigma_4 = 0,797, \Sigma_5 = 0,668, \Sigma_6 = 0,511$$

Langkah terakhir untuk menentukan matriks U adalah dengan melakukan normalisasi pada vektor-vektor singular U, namun sebelumnya harus menentukan terlebih dahulu vektor eigen dari matriks U.

$$(\lambda I - A \times A^T)x = 0$$

Menghasilkan vektor-vektor eigen matriks U:

0,623	0,317	-1,413	0,935	-0,383	-4,099
1,174	0,105	-0,554	0,405	0,277	4,116
1,087	0,029	-0,773	0,457	-0,308	3,578
0,463	0,092	-1,779	1,256	-0,639	7,841
0,492	0,542	-1,796	-0,704	-1,020	-1,716

0,121	0,151	-4,487	-2,534	1,114	-1,150
0,705	0,767	-0,095	0,437	-2,300	-2,014
8,640	-0,122	0,263	-1,059	-0,206	0,511
8,640	-0,122	0,263	-1,059	-0,206	0,511
4,028	-0,046	-0,757	2,480	0,234	15,705
8,640	-0,122	0,263	-1,059	-0,206	0,511
5,911	0,538	0,774	-0,588	0,487	0,927
6,206	-0,076	-0,822	3,000	0,452	-8,074
8,640	-0,122	0,263	-1,059	-0,206	0,511
8,640	-0,122	0,263	-1,059	-0,206	0,511
6,206	-0,076	-0,822	3,000	0,452	-8,074
1,000	1,000	1,000	1,000	1,000	1,000
1,000	1,000	1,000	1,000	1,000	1,000

Kemudian dilakukan normalisasi vektor-vektor pada setiap kolomnya dengan membagi setiap komponen dengan *magnitude* atau panjang dari vektor tersebut.

Akan diberikan contoh kasus pengerjaan sesuai dengan rumus diberikan.

$$\hat{U}_n = \frac{U_n}{\|U_n\|}$$

$$\hat{U}_1 = (0,623, 1,174, 1,087, 0,463, 0,492, 0,121, 0,705, 8,640, 8,640, 4,028, 8,640, 5,911, 6,206, 8,640, 8,640, 6,206, 1, 1)$$

$$/ \sqrt{0,623^2 + 1,174^2 + 1,087^2 + 0,463^2 + 0,492^2 + 0,121^2 + 0,705^2 + 8,640^2 + 8,640^2 + 4,028^2 + 8,640^2 + 5,911^2 + 6,206^2 + 8,640^2 + 8,640^2 + 6,206^2 + 1^2 + 1^2}$$

$$\hat{U}_1 = (0,623, 1,174, 1,087, 0,463, 0,492, 0,121, 0,705, 8,640, 8,640, 4,028, 8,640, 5,911, 6,206, 8,640, 8,640, 6,206, 1, 1)$$

$$/ \sqrt{507,36}$$

$$\hat{U}_1 = (0,623, 1,174, 1,087, 0,463, 0,492, 0,121, 0,705, 8,640, 8,640, 4,028, 8,640, 5,911, 6,206, 8,640, 8,640, 6,206, 1, 1)$$

$$/ 22,525$$

$$0,0264$$

$$0,0512$$

$$0,0471$$

$$0,0162$$

$$0,0233$$

$$0,0054$$

$$0,0336$$

$$0,3911$$

$$0,3911$$

$$0,3911$$

$$0,2691$$

$$0,2744$$

$$0,3911$$

$$0,3911$$

$$0,2744$$

$$0,0478$$

$$0,0478$$

$$0,0026$$

Dari contoh salah satu vektor matriks U, maka matriks U memberikan hasil akhir:

-0,0264	0,1725	-0,2212	-0,0579	0,2138	0,1072
-0,0512	0,0575	-0,1055	0,0663	0,0427	-0,0960
-0,0471	0,0163	-0,1424	0,0610	0,0633	0,0789
-0,0162	0,0521	-0,3224	0,1391	0,1752	0,1610
-0,0233	0,2940	-0,2953	-0,1248	-0,0534	0,3120
-0,0054	0,0823	-0,7412	-0,3475	-0,2981	-0,3153
-0,0336	0,4155	-0,0135	0,0152	0,0981	0,6878
-0,3911	-0,0701	0,0256	-0,0153	-0,1516	0,0639
-0,3911	-0,0701	0,0256	-0,0153	-0,1516	0,0639
-0,3911	-0,0701	0,0256	-0,0153	-0,1516	0,0639
-0,2691	0,2890	0,1199	0,0142	-0,1061	-0,1439
-0,2744	-0,0415	-0,1156	0,0316	0,5711	-0,1648
-0,3911	-0,0701	0,0256	-0,0153	-0,1516	0,0639
-0,3911	-0,0701	0,0256	-0,0153	-0,1516	0,0639
-0,2744	-0,0415	-0,1156	0,0316	0,5711	-0,1648
-0,0478	0,5415	0,1700	0,0385	-0,0215	-0,2986
-0,0478	0,5415	0,1700	0,0385	-0,0215	-0,2986
-0,0026	0,0092	-0,2773	0,9090	-0,1924	-0,0609

Tahap selanjutnya adalah menentukan nilai-nilai singular, nilai singular tersebut berasal dari nilai eigen yang tidak nol dan hasilnya akan disajikan dalam matriks diagonal. Dikarenakan nilai singular telah ditentukan sebelumnya, maka akan disajikan matriks diagonal seperti pada di bawah ini.



1,9544	0	0	0	0	0
0	1,3680	0	0	0	0
0	0	0,9055	0	0	0
0	0	0	0,7969	0	0
0	0	0	0	0,6684	0
0	0	0	0	0	0,5104

Langkah terakhir sebelum menentukan hasil SVD adalah menentukan terlebih dahulu nilai-nilai matriks V^T , kemudian barulah nilai-nilai yang telah ditentukan akan dikalikan dan dapat memperoleh hasil perhitungan SVD. Dikarenakan proses untuk menentukan nilai-nilai matriks V sama dengan menentukan nilai-nilai matriks U maka di bawah ini akan disajikan berupa hasil akhir nilai matriks singular V.

$$V =$$

0,770694	0,158905	0,012539	0,127897	0,074556	0,043547
0,158905	0,38655	0,00627	0,349272	0,068286	0,037278
0,012539	0,00627	3,813689	0,233914	0,558448	0,103158
0,127897	0,349272	0,233914	1,794225	0,037278	0,00627
0,074556	0,068286	0,558448	0,037278	0,620464	0,134167
0,043547	0,037278	0,103158	0,00627	0,134167	0,134167

Namun yang diminta disini adalah nilai matriks singular V^T , sehingga dari hasil akhir matriks V akan dilakukan transpose dari matriks tersebut dan mendapatkan hasil di bawah ini.

$V^T =$

-0,0135	-0,0175	-0,9823	-0,1200	-0,1418	-0,0065
0,1447	0,2395	-0,1232	0,9520	0,0042	0,0162
-0,8625	-0,2234	0,0298	0,1978	-0,2492	-0,3226
-0,3559	-0,0141	-0,0157	0,0394	0,0684	0,9310
-0,2561	0,1560	-0,1302	-0,0185	0,9303	-0,1653
-0,2068	0,9316	0,0419	-0,1958	-0,2182	-0,0399

Setelah semua nilai singular telah ditemukan, maka perhitungan SVD dapat dilakukan dengan cara mengkalikan seluruh nilai-nilai singular tersebut. Di bawah ini merupakan hasil dari perhitungan SVD.

$A = U \Sigma V^T$

0,1761	0,1761	2,3679	0,1761	0,1761	0,0950
0,0792	0,0555	0,0792	0,0792	0,0792	0,0792
0,0792	0,0792	0,0792	0,3816	0,0792	0,0792
-0,1761	0,1761	1,1015	0,1457	0,1761	0,1761
0,3010	0,3010	0,0029	0,3010	0,0069	0,0073
0,7782	0,2498	0,2246	0,1943	0,1735	1,6046
1,8041	0,4771	2,3072	0,4771	0,0694	1,8735
1,4225	1,8735	0,7782	0,3678	0,3036	0,0648
1,4225	1,8735	0,7782	0,3678	0,3036	0,0648
1,4225	1,8735	0,7782	0,3678	0,3036	0,0648
0,0919	1,5266	0,4771	0,4771	0,0049	0,4077
0,0867	0,0555	0,4771	2,7756	0,4771	0,1197
1,4225	1,8735	0,7782	0,3678	0,3036	0,0648
1,4225	1,8735	0,7782	0,3678	0,3036	0,0648
0,0867	0,0555	0,4771	2,7756	0,4771	0,1197
0,2637	3,0531	2,8710	0,7782	0,2082	0,0590
0,2637	3,0531	2,8710	0,7782	0,2082	0,0590
0,0121	0,4198	0,0900	0,0781	1,2663	0,7782

7. Cosine Similarity

Pada tahap ini dimana *cosine similarity* bertujuan untuk mengukur tingkat kemiripan antara kata kunci pencarian dengan setiap dokumen yang diteliti. Pada

tahap ini perhitungan dilakukan dengan menggunakan data hasil perhitungan TF-IDF. Perhitungan *cosine similarity* memiliki rumus yang digunakan untuk mengukur tingkat kemiripan, rumus yang digunakan berdasarkan rumus (5).

Dimana dengan contoh kemiripan kata kunci pencarian (Q) dengan dokumen 1 (d1):

$$\begin{aligned} \cos Sim_{(Q,d1)} &= (0,1761 \times 0,095 + 0,0792 \times 0,0792 + 0,0792 \times 0,0792 + 0,1761 \times 0,1761 \\ &\quad + 0,301 \times 0,0073 + 0,7782 \times 1,6046 + 1,8041 \times 1,8735 \\ &\quad + 1,4225 \times 0,0648 + 1,4225 \times 0,0648 + 1,4225 \times 0,0648 \\ &\quad + 0,0919 \times 0,4077 + 0,0867 \times 0,1197 + 1,4225 \times 0,648 + 1,4225 \times 0,648 \\ &\quad + 0,0867 \times 0,1197 + 0,2637 \times 0,059 + 0,2637 \times 0,059 \\ &\quad + 0,0121 \times 0,7782) / (\sqrt{(0,1761^2 + 0,0792^2 + 0,0792^2 + 0,1761^2 + 0,301^2 \\ &\quad + 0,7782^2 + 1,8041^2 + 1,4225^2 + 1,4225^2 + 1,4225^2 + 0,0919^2 + 0,0867^2 \\ &\quad + 1,4225^2 + 1,4225^2 + 0,0867^2 + 0,2637^2 + 0,2637^2 \\ &\quad + 0,0121^2)} \times \sqrt{(0,095^2 + 0,0792^2 + 0,0792^2 + 0,1761^2 + 0,0073^2 \\ &\quad + 1,6046^2 + 1,8735^2 + 0,0648^2 + 0,0648^2 + 0,0648^2 + 0,4077^2 + 0,1197^2 \\ &\quad + 0,648^2 + 0,648^2 + 0,1197^2 + 0,059^2 + 0,059^2 + 0,7782^2)}) \\ \cos Sim_{(q,d1)} &= 5,251 / (6,9658 \times 14,3054) = 5,251 / 9,9823 = 0,5026 \text{ atau } 50\% \end{aligned}$$

Sehingga tingkat nilai kemiripan yang didapat antara kata kunci pencarian (Q) dengan dokumen 1 (d1) melalui perhitungan *cosine similarity* menghasilkan kemiripan sebesar 50%. Dengan rumus dan cara perhitungan yang sesuai maka hal tersebut dapat diterapkan untuk d2, d3, d4, d5 dan menghasilkan kemiripan sebagai berikut.

$$\cos Sim_{(q,d2)} = 0,4512 \text{ atau } 45\%$$

$$\cos Sim_{(q,d3)} = 0,4512 \text{ atau } 45\%$$

$$\cos Sim_{(q,d4)} = 0,5044 \text{ atau } 50\%$$

$$\cos Sim_{(q,d5)} = 0,5832 \text{ atau } 58\%$$

Pada tahap ini ditentukan juga nilai ambang batas bawah atau *threshold* dengan perhitungan sebagai berikut.

$$T = \frac{(58 + 45)}{2} = \frac{103}{2} = 51,5$$

Sehingga didapat bahwa *threshold* ada di angka 51,5% dan berita yang memiliki nilai kemiripan di bawah angka *threshold* akan tereliminasi. Berita yang lolos untuk dilanjutkan pada tahap berikutnya adalah berita 5.

8. Topic Encoded Data

Tahapan terakhir pada penelitian ini adalah menentukan topik dari setiap atau masing-masing dokumen yang diteliti. Untuk menentukan topik tersebut dapat mengacu pada hasil akhir SVD dan berdasar pada nilai atau angka tertinggi tanpa memperhatikan apakah nilai tersebut bernilai positif atau negatif. Sehingga di bawah ini dapat dilihat mengenai topik-topik yang dapat diekstrak dari hasil akhir perhitungan SVD.

$d_5 =$ spbu, pertamina, daftar, baru, bbm

Kelima tahapan perhitungan metode LSA menghasilkan, bahwa dari 5 dokumen berita dengan ambang batas bawah kemiripan sebesar 51,5% telah mengeliminasi sebanyak 4 dokumen berita yang berguna untuk memangkas waktu perhitungan pada tahap berikutnya. Sehingga menghasilkan hanya 1 berita dengan penjelasan pada Tabel 4.7.

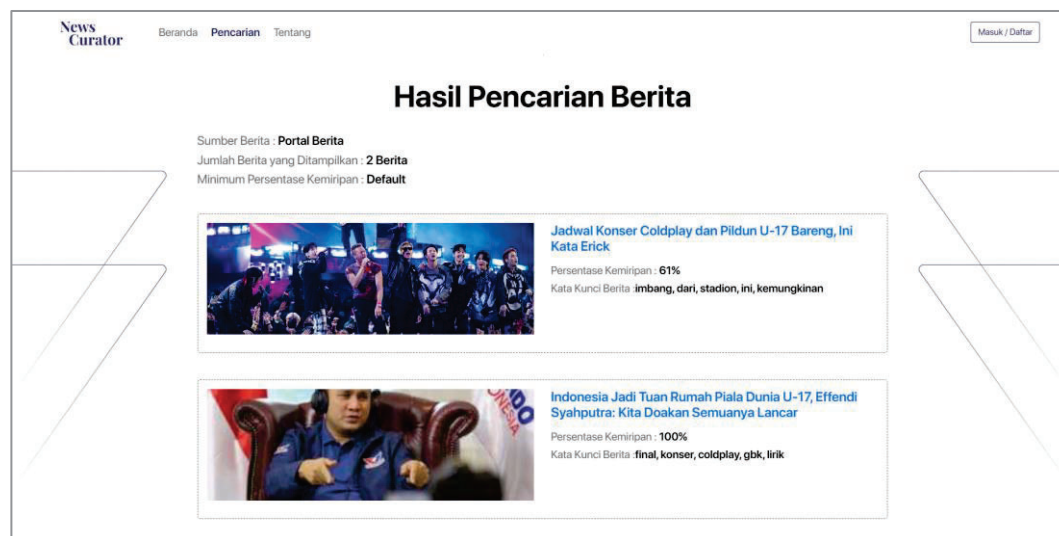
Tabel 4.7 Hasil Akhir Pencarian

Kata Kunci pencarian		Kenaikan Harga BBM Terbaru
Nomor Berita	Persentase Kemiripan	Topik Berita
Berita 5	58%	spbu, pertamina, daftar, baru, bbm

Perbandingan antara implementasi LSA yang dihitung secara manual dan implementasi LSA pada sistem baik proses satu persatu maupun hasil akhirnya dapat dilihat pada Lampiran 7 pada Gambar L7.1 hingga sampai Gambar L7.14.

4.1.4 Hasil Pencarian

Tampilan pada Gambar 4.2 merupakan tampilan hasil pencarian dimana daftar berita yang telah melalui seluruh proses LSA dan yang memenuhi *threshold* akan ditampilkan pada tampilan hasil pencarian. Hasil pencarian meliputi sumber berita, jumlah berita yang ditampilkan, nilai minimum *similarity*, kemudian untuk daftar berita meliputi judul, nilai kemiripan dan, kata kunci,



Gambar 4.2 Tampilan Halaman Hasil Pencarian

Sementara untuk hasil implementasi halaman pengguna yang telah dibentuk pada sistem dapat dilihat pada Lampiran 8 Gambar L8.1 hingga Gambar L8.7. Halaman admin yang telah dibentuk pada sistem juga dapat dilihat pada Lampiran 9 pada Gambar L9.1 hingga Gambar L9.6.

4.2 Testing

4.2.1 Black Box Testing

Uji coba fungsionalitas *website* menggunakan BBT demi menghindari kekeliruan dalam pengkodean. Melalui BBT menghasilkan bahwa *website* 100% berjalan sesuai dengan 26 skenario uji coba. Tabel 4.8 di bawah ini merupakan hasil dari BBT. Hasil selengkapnya mengenai uji fungsionalitas dapat dilihat pada Lampiran 4 pada Tabel L4.1.

Tabel 4.8 Hasil Uji Coba Fungsionalitas dengan *Black Box Testing*

No	Skenario	Status
S1	Memilih kategori portal berita, memasukkan kata kunci, jumlah berita, dan nilai minimum kemiripan dengan benar.	Sesuai
S2	Memilih kategori portal berita. Kata kunci, jumlah berita dan nilai minimum tidak diisi.	Sesuai
S3	Memilih kategori portal berita. Kata kunci dan nilai minimum kemiripan diisi dengan benar, dan jumlah berita melebihi ketentuan.	Sesuai
S4	Memilih kategori dokumen. Kata kunci, nilai kemiripan pencarian diisi dengan benar. Dokumen berita diunggah.	Sesuai
S5	Memilih kategori dokumen. Kata kunci, dan nilai minimum kemiripan diisi dengan benar. Dokumen berita tidak diunggah	Sesuai

No.	Kata kunci pencarian “Piala Dunia U-17”			
	Threshold 0,1	Threshold 0,2	Threshold 0,3	Threshold Sistem
...
16	Gibran Ajukan Kota Solo Tuan Rumah Piala Dunia U-17			

Berdasarkan kajian terhadap sampel berita yang dapat dilihat pada Lampiran 5 Tabel L5.1 dengan hasil rekomendasi pencarian yang diberikan berdasarkan kata kunci pencarian “Piala Dunia U-17” pada Tabel 4.9, menghasilkan bahwa:

1. Berdasarkan 16 berita yang direkomendasikan, dimana ke-15 berita tersebut adalah sesuai dengan pencarian. Sebagai contoh adalah pada seluruh hasil rekomendasi terkecuali pada berita nomor 11.
2. Berdasarkan 16 berita yang direkomendasikan, terdapat 1 berita yaitu pada berita nomor 11 yang tidak memiliki kesesuaian dengan pencarian namun tetap direkomendasikan oleh sistem.
3. Berdasarkan 22 sampel berita dimana terdapat 16 berita yang direkomendasikan dan 6 berita yang tidak direkomendasikan. Setelah dikaji kembali menghasilkan bahwa 6 berita tersebut sesuai dengan pencarian namun tidak direkomendasikan oleh sistem.
4. Berdasarkan keseluruhan sampel dan hasil rekomendasi pencarian, tidak terdapat satupun berita yang tidak sesuai dan tidak direkomendasikan sistem.

A. Pengujian Pada *Threshold* 0,1

Pada pengujian *confusion matrix* pertama terhadap ambang batas bawah sebesar 0,1 atau dalam persen yaitu 10% menghasilkan 16 data berita. Hasil rekomendasi berita yang ditampilkan kemudian akan dibandingkan pada sampel data yang dapat dilihat pada Lampiran 5 Tabel L5.1. Hasil perbandingan menunjukkan sebagai berikut:

1. Hasil rekomendasi berita yang memiliki kecocokan dengan piala dunia u-17 terdapat 15 data berita. Sehingga dapat dinyatakan bahwa terdapat 15 data berita yang sesuai dan direkomendasikan oleh sistem atau yang disebut dengan *True Positive*.

2. Hasil rekomendasi berita yang tidak memiliki kecocokan dengan piala dunia u-17 terdapat 1 data berita. Sehingga dapat dinyatakan bahwa terdapat 1 data berita yang tidak sesuai namun tetap direkomendasikan oleh sistem atau yang disebut dengan *False Positive*.
3. Data berita yang memiliki kecocokan dengan piala dunia u-17 namun tidak termasuk pada hasil rekomendasi terdapat 6 data berita. Sehingga dapat dinyatakan bahwa terdapat 6 data berita yang sesuai namun tidak direkomendasikan oleh sistem atau yang disebut dengan *False Negative*.
4. Data berita yang tidak memiliki kecocokan dengan piala dunia u-17 dan tidak termasuk pada hasil rekomendasi terdapat 0 data berita. Sehingga tidak terdapat satupun berita yang tidak sesuai dan tidak direkomendasikan oleh sistem atau yang disebut dengan *True Negative*.

Hasil perbandingan di atas kemudian diolah pada Tabel 4.10 *confusion matrix* untuk kebutuhan perhitungan *accuracy*, *precision*, dan *recall*.

Tabel 4.10 *Confusion Matrix* dengan *Threshold* 0,1

		True Values	
		True	False
Prediction	True	15	1
	False	6	0

Berdasarkan tabel perhitungan *confusion matrix* di atas kemudian dilakukan perhitungan terhadap *accuracy*, *precision*, dan *recall* pada *threshold* 0,1.

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) = \left(\frac{15+0}{15+0+6+1} \right) = 0,68 \times 100\% = 68\%$$

$$Precision = \left(\frac{TP}{TP+FP} \right) = \left(\frac{15}{15+1} \right) = 0,71 \times 100\% = 94\%$$

$$Recall = \left(\frac{TP}{TP+FN} \right) = \left(\frac{15}{15+6} \right) = 0,94 \times 100\% = 71\%$$

B. Pengujian Pada *Threshold* 0,2

Pada pengujian *confusion matrix* kedua terhadap ambang batas bawah sebesar 0,2 atau dalam persen yaitu 20% menghasilkan 15 data berita. Hasil rekomendasi berita yang ditampilkan kemudian akan dibandingkan pada sampel data yang dapat dilihat pada Lampiran 5 Tabel L5.1. Hasil perbandingan menunjukkan sebagai berikut:

1. Hasil rekomendasi berita yang memiliki kecocokan dengan piala dunia u-17 terdapat 14 data berita. Sehingga dapat dinyatakan bahwa terdapat 14 data berita yang sesuai dan direkomendasikan oleh sistem atau yang disebut dengan *True Positive*.
2. Hasil rekomendasi berita yang tidak memiliki kecocokan dengan piala dunia u-17 terdapat 1 data berita. Sehingga dapat dinyatakan bahwa terdapat 1 data berita yang tidak sesuai namun tetap direkomendasikan oleh sistem atau yang disebut dengan *False Positive*.
3. Data berita yang memiliki kecocokan dengan piala dunia u-17 namun tidak termasuk pada hasil rekomendasi terdapat 7 data berita. Sehingga dapat dinyatakan bahwa terdapat 7 data berita yang sesuai namun tidak direkomendasikan oleh sistem atau yang disebut dengan *False Negative*.
4. Data berita yang tidak memiliki kecocokan dengan piala dunia u-17 dan tidak termasuk pada hasil rekomendasi terdapat 0 data berita. Sehingga tidak terdapat satupun berita yang tidak sesuai dan tidak direkomendasikan oleh sistem atau yang disebut dengan *True Negative*.

Hasil perbandingan di atas kemudian diolah pada Tabel 4.11 *confusion matrix* untuk kebutuhan perhitungan *accuracy*, *precision*, dan *recall*.

Tabel 4.11 *Confusion Matrix* dengan *Threshold* 0,2

		<i>True Values</i>	
		<i>True</i>	<i>False</i>
<i>Prediction</i>	<i>True</i>	14	1
	<i>False</i>	7	0

Berdasarkan tabel perhitungan *confusion matrix* di atas kemudian dilakukan perhitungan terhadap *accuracy*, *precision*, dan *recall* pada *threshold* 0,2.

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) = \left(\frac{14+0}{14+0+7+1} \right) = 0,63 \times 100\% = 63\%$$

$$Precision = \left(\frac{TP}{TP+FP} \right) = \left(\frac{14}{14+1} \right) = 0,66 \times 100\% = 93\%$$

$$Recall = \left(\frac{TP}{TP+FN} \right) = \left(\frac{14}{14+7} \right) = 0,93 \times 100\% = 66\%$$

C. Pengujian Pada *Threshold* 0,3

Pada pengujian *confusion matrix* ketiga terhadap ambang batas bawah sebesar 0,3 atau dalam persen yaitu 30% menghasilkan 13 data berita. Hasil

rekomendasi berita yang ditampilkan kemudian akan dibandingkan pada sampel data yang dapat dilihat pada Lampiran 5 Tabel L5.1. Hasil perbandingan menunjukkan sebagai berikut:

1. Hasil rekomendasi berita yang memiliki kecocokan dengan piala dunia u-17 terdapat 12 data berita. Sehingga dapat dinyatakan bahwa terdapat 12 data berita yang sesuai dan direkomendasikan oleh sistem atau yang disebut dengan *True Positive*.
2. Hasil rekomendasi berita yang tidak memiliki kecocokan dengan piala dunia u-17 terdapat 1 data berita. Sehingga dapat dinyatakan bahwa terdapat 1 data berita yang tidak sesuai namun tetap direkomendasikan oleh sistem atau yang disebut dengan *False Positive*.
3. Data berita yang memiliki kecocokan dengan piala dunia u-17 namun tidak termasuk pada hasil rekomendasi terdapat 9 data berita. Sehingga dapat dinyatakan bahwa terdapat 9 data berita yang sesuai namun tidak direkomendasikan oleh sistem atau yang disebut dengan *False Negative*.
4. Data berita yang tidak memiliki kecocokan dengan piala dunia u-17 dan tidak termasuk pada hasil rekomendasi terdapat 0 data berita. Sehingga tidak terdapat satupun berita yang tidak sesuai dan tidak direkomendasikan oleh sistem atau yang disebut dengan *True Negative*.

Hasil perbandingan di atas kemudian diolah pada Tabel 4.12 *confusion matrix* untuk kebutuhan perhitungan *accuracy*, *precision*, dan *recall*.

Tabel 4.12 *Confusion Matrix* dengan *Threshold* 0,3

		<i>True Values</i>	
		<i>True</i>	<i>False</i>
<i>Prediction</i>	<i>True</i>	12	1
	<i>False</i>	9	0

Berdasarkan tabel perhitungan *confusion matrix* di atas kemudian dilakukan perhitungan terhadap *accuracy*, *precision*, dan *recall* pada *threshold* 0,3.

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) = \left(\frac{12+0}{12+0+9+1} \right) = 0,55 \times 100\% = 55\%$$

$$Precision = \left(\frac{TP}{TP+FP} \right) = \left(\frac{12}{12+1} \right) = 0,57 \times 100\% = 92\%$$

$$Recall = \left(\frac{TP}{TP+FN} \right) = \left(\frac{12}{12+9} \right) = 0,92 \times 100\% = 57\%$$

D. Pengujian Pada *Threshold* Sistem

Pada pengujian *confusion matrix* keempat terhadap ambang batas bawah yang ditentukan berdasarkan nilai kemiripan tertinggi ditambah terendah kemudian dibagi dua, menghasilkan 13 data berita. Hasil rekomendasi berita yang ditampilkan kemudian akan dibandingkan pada sampel data yang dapat dilihat pada Lampiran 5 Tabel L5.1. Hasil perbandingan menunjukkan sebagai berikut:

1. Hasil rekomendasi berita yang memiliki kecocokan dengan piala dunia u-17 terdapat 12 data berita. Sehingga dapat dinyatakan bahwa terdapat 12 data berita yang sesuai dan direkomendasikan oleh sistem atau yang disebut dengan *True Positive*.
2. Hasil rekomendasi berita yang tidak memiliki kecocokan dengan piala dunia u-17 terdapat 1 data berita. Sehingga dapat dinyatakan bahwa terdapat 1 data berita yang tidak sesuai namun tetap direkomendasikan oleh sistem atau yang disebut dengan *False Positive*.
3. Data berita yang memiliki kecocokan dengan piala dunia u-17 namun tidak termasuk pada hasil rekomendasi terdapat 9 data berita. Sehingga dapat dinyatakan bahwa terdapat 9 data berita yang sesuai namun tidak direkomendasikan oleh sistem atau yang disebut dengan *False Negative*.
4. Data berita yang tidak memiliki kecocokan dengan piala dunia u-17 dan tidak termasuk pada hasil rekomendasi terdapat 0 data berita. Sehingga tidak terdapat satupun berita yang tidak sesuai dan tidak direkomendasikan oleh sistem atau yang disebut dengan *True Negative*.

Hasil perbandingan di atas kemudian diolah pada Tabel 4.13 *confusion matrix* untuk kebutuhan perhitungan *accuracy*, *precision*, dan *recall*.

Tabel 4.13 *Confusion Matrix* dengan *Threshold* Sistem

		<i>True Values</i>	
		<i>True</i>	<i>False</i>
<i>Prediction</i>	<i>True</i>	12	1
	<i>False</i>	9	0

Berdasarkan tabel perhitungan *confusion matrix* di atas kemudian dilakukan perhitungan terhadap *accuracy*, *precision*, dan *recall* pada *threshold* sistem.

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN} \right) = \left(\frac{12+0}{12+0+9+1} \right) = 0,55 \times 100\% = 55\%$$

$$Precision = \left(\frac{TP}{TP+FP} \right) = \left(\frac{12}{12+1} \right) = 0,57 \times 100\% = 92\%$$

$$Recall = \left(\frac{TP}{TP+FN} \right) = \left(\frac{12}{12+9} \right) = 0,92 \times 100\% = 57\%$$

E. Analisis Pengujian *Confusion Matrix*

Berdasarkan hasil pengujian yang telah dilakukan terhadap keempat *threshold* yang telah ditentukan, maka hasil pengujian akan di rata-rata untuk kesimpulan bagaimana *accuracy*, *precision*, dan *recall* pada *website*. Hasil perhitungan rata-rata dapat dilihat pada Tabel 4.14.

Tabel 4.14 Perhitungan Rata-Rata *Accuracy*, *Precision*, dan *Recall*

No. Pengujian	Threshold	Accuracy	Precision	Recall
C1	0,1 atau 10%	68%	94%	71%
C2	0,2 atau 20%	63%	93%	66%
C3	0,3 atau 30%	55%	92%	57%
C4	Threshold Sistem	55%	92%	57%
	Rata-rata	60,25%	92,75%	62,75%

Berdasarkan hasil pengujian menunjukkan bahwa *website* ini memiliki rata-rata akurasi sebesar 60,25%, rata-rata presisi sebesar 92,75%, dan rata-rata *recall* sebesar 62,75%. Nilai presisi cukup tinggi karena rekomendasi yang diberikan sesuai, nilai akurasi yang rendah disebabkan oleh kesamaan sampel berita dengan kata kunci pencarian, yang menyebabkan rekomendasi terpotong pada *threshold* yang lebih tinggi. Namun, nilai *recall* yang kurang tinggi menandakan ada ruang untuk meningkatkan kemampuan sistem dalam menemukan informasi yang sesuai dengan pencarian pengguna.

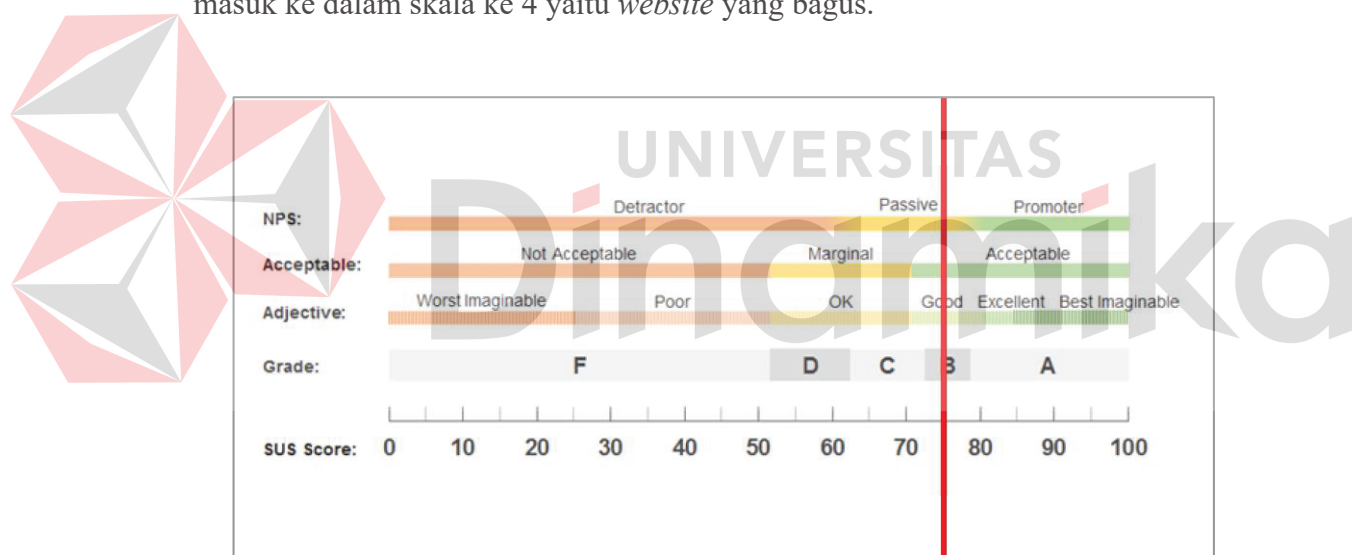
4.2.3 *System Usability Scale*

Pengujian SUS dilakukan dengan menyebarkan kuesioner mengenai 10 pernyataan SUS dengan jawaban berupa skala likert 5 poin kepada masyarakat umum tanpa adanya batasan demografi responden. Sampel data sebanyak 96 responden diolah sesuai dengan perhitungan SUS dapat dilihat pada Tabel 4.15 di bawah, dan hasil selengkapnya dapat dilihat pada Lampiran 6 pada Tabel L6.1.

Tabel 4.15 Hasil *Testing System Usability Scale*

Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Jumlah	Skor SUS
R1	4	1	5	2	4	2	5	1	5	1	36	90
R2	4	1	5	1	3	1	5	1	5	1	37	92,5
R3	5	1	3	1	3	3	5	3	5	2	31	77,5
R4	3	1	4	1	4	3	5	2	5	3	31	77,5
R5	3	2	4	2	5	1	5	3	4	3	30	75
...												
R96	3	2	5	2	4	1	5	2	4	3	31	77,5
Rata-rata Skor SUS											75,28	

Berdasarkan pada Tabel 4.15 di atas, rata-rata skor SUS yang didapat sebesar 75,28 dimana menurut Sauro (2018) nilai 75,28 termasuk kedalam hasil yang diterima oleh pengguna dengan skala nilai B. Tingkat kepuasan pengguna termasuk ke dalam 'passive' sehingga dapat dikatakan kesediaan pengguna dalam merekomendasikan ke pengguna lain bersifat pasif. Pada skala *adjective*, *website* masuk ke dalam skala ke 4 yaitu *website* yang bagus.



Gambar 4.3 Hasil Akhir Pengukuran Nilai SUS

Sumber: (Sauro, 2018)

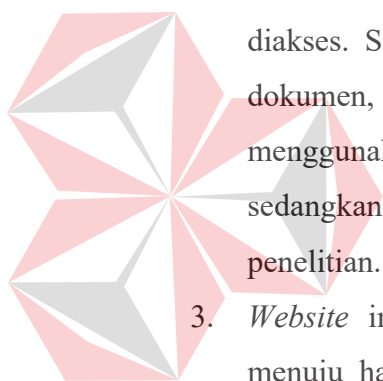
4.3 Evaluasi

Evaluasi mengenai *website* pencarian kemiripan berita berdasarkan relevansi berita dengan kata kunci dengan menggunakan metode *Latent Semantic Analysis*, dapat memperoleh hasil seperti pada di bawah ini.

1. Pembentukan *website* melalui dua fase, yaitu *prototype development* dan *iterative development*. Fase *prototype* melibatkan analisis kebutuhan

pengguna, kebutuhan fungsional dan non-fungsional, serta analisis kebutuhan perangkat. Terdapat dua iterasi dalam fase ini untuk mendapatkan masukan dari pengguna tentang desain dan alur kerja desain melalui perancangan desain, pembuatan *prototype*, penyebaran kuesioner desain, dan penyaringan masukan. Pada fase *iterative development*, perancangan *website* dimulai dengan pembuatan diagram IPO, *system flow*, *context diagram*, *data flow diagram*, dan struktur dokumen *database*. Setelah itu, dilakukan implementasi *website* dan uji coba untuk memastikan setiap kebutuhan fungsional terpenuhi.

2. Awalnya, pembahasan tentang *website* pencarian kemiripan berita menggunakan metode LSA terbagi menjadi tiga sumber berita, masing-masing dengan variabel penelitian tertentu. Namun, selama pengembangan *website*, penggunaan *link* sebagai sumber berita terhambat karena beberapa *link* memiliki sisi dan tingkat keamanan yang membuat kontennya tidak dapat diakses. Sebagai solusi, sumber berita diperoleh melalui portal berita dan dokumen, dengan dua variabel penelitian yang berbeda. Portal berita menggunakan judul berita dan sinopsis berita sebagai data penelitian, sedangkan dokumen menyertakan seluruh isi dokumen sebagai variabel penelitian.
3. *Website* ini memiliki beberapa tahapan penggunaan. Pertama, pengguna menuju halaman pencarian. Selanjutnya, pengguna dapat memilih sumber berita yang diinginkan, yaitu dari portal berita atau dokumen. Jika memilih sumber berita dari portal berita, pengguna dapat mengisi kata kunci pencarian, nilai minimum kemiripan, dan jumlah berita hasil perhitungan yang ingin ditampilkan. Jika sumber berita berasal dari dokumen, pengguna dapat mengunggah dokumen sesuai dengan kolom yang ada, memasukkan kata kunci pencarian, dan minimum nilai kemiripan. Setelah itu, sistem akan menghitung dan menampilkan hasil pencarian dengan menerapkan metode LSA sesuai dengan kriteria pencarian pengguna.



BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan penjelasan yang telah disampaikan mengenai bagaimana penggunaan metode LSA sebagai dasar penelitian yang diterapkan pada *website* ini, dimulai dari perancangan, implementasi, dan uji coba dapat ditarik menjadi beberapa kesimpulan seperti pada di bawah ini:

1. *Website* ini mampu untuk membantu dalam mengukur nilai kemiripan sekaligus keakuratan berdasarkan kata kunci pada berita yang ingin dicari. Melalui hal itu juga *website* mampu untuk meminimalisir kekeliruan pengguna dalam perolehan berita yang dicarinya, sehingga tentunya dapat menghemat waktu pengguna dalam memperoleh suatu informasi berita dengan bantuan ekstraksi topik berita yang mampu dilakukan pada *website* ini.
2. Dalam mengukur kebergunaan *website* ini pada masyarakat umum, dilakukan pengujian menggunakan *System Usability Scale*. Pengujian dilakukan kepada 96 responden tanpa adanya batasan demografi menghasilkan nilai sebesar 75,28 atau diterima oleh masyarakat.
3. Uji coba fungsionalitas dilakukan menggunakan metode *Black Box Testing* yang menghasilkan bahwa *website* telah 100% berjalan sesuai dengan 26 skenario yang telah dibentuk.
4. Hasil uji coba kesesuaian logika perhitungan dengan *Confusion Matrix* juga dilakukan, dengan membandingkan hasil perhitungan yang telah dikodekan pada sistem berdasarkan penerapan LSA yang telah berjalan pada *website* dengan perbandingan 4 macam threshold menghasilkan bahwa *website* memiliki *accuracy* sebesar 60,25%, *precision* sebesar 92,75%, dan *recall* sebesar 62,75%

5.2 Saran

Dapat bekerjasama dengan perusahaan-perusahaan media berita agar variabel data berita dapat lebih luas dan lengkap cakupannya, sehingga perhitungan mengenai kesesuaian berita dapat lebih dalam dan terperinci. Mencoba menerapkan

metode yang memiliki kemiripan dengan LSA seperti *Latent Semantic Indexing* atau bahkan metode yang digunakan mesin pencari dengan kecerdasan buatan sekarang ini yaitu *Transformer Based Model* untuk dapat menentukan metode terbaik yang diterapkan dalam hal ini yaitu pada media berita.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Amala, I. A., Richasdy, D., & Purbolaksono, M. D. (2022). *Telkom University News Topic Modeling Using Latent Semantic Analysis (LSA) Method on Online News Portal*. 4(1), 110–115. <https://doi.org/10.47065/bits.v4i1.1584>
- Caelen, O. (2017). A Bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, 81(3–4), 429–450. <https://doi.org/10.1007/s10472-017-9564-8>
- Cook, J. (2019). *Latent Semantic Analysis, Part 1 - Databricks*. <https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bfc/2909405083353266/2622605937013888/6410281937148337/latest.html>
- Denny, M. J., & Spirling, A. (2018). Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do about It. *Political Analysis*, 26(2), 168–189. <https://doi.org/10.1017/pan.2017.44>
- Dwityas, N. A., Mulyana, A., Hesti, S., Briandana, R., & Kurniasari, P. M. (2020). Digital marketing communication strategies: The case of Indonesian news' portals. *International Journal of Economics and Business Administration*, 8(3), 307–316. <https://doi.org/10.35808/ijeba/517>
- Fletcher, R., & Park, S. (2017). The Impact of Trust in the News Media on Online News Consumption and Participation. *Digital Journalism*, 5(10), 1281–1299. <https://doi.org/10.1080/21670811.2017.1279979>
- Gronier, G., & Baudet, A. (2021). Psychometric Evaluation of the F-SUS: Creation and Validation of the French Version of the System Usability Scale. *International Journal of Human-Computer Interaction*, 37(16), 1571–1582. <https://doi.org/10.1080/10447318.2021.1898828>
- Hasugian, P. M., Manurung, J., Logaraz, L., & Ram, U. (2021). Implementation of Tf-Idf and Cosine Similarity Algorithms for Classification of Documents Based on Abstract Scientific Journals. *Infokum*, 9(2), 518–526. <http://infor.seaninstitute.org/index.php/infokum/article/view/201%0Ahttp://infor.seaninstitute.org/index.php/infokum/article/download/201/144>
- Heryanto, I. W. A., Artama, Kurniawan, M. W. S., & Gunadi, G. A. (2020). Segmentasi Warna dengan Metode Thresholding. *Wahana Matematika Dan Sains*, 14(1), 54–64.
- Hickman, L., Thapa, S., Tay, L., Cao, M., & Srinivasan, P. (2022). Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations. *Organizational Research Methods*, 25(1), 114–146. <https://doi.org/10.1177/1094428120971683>

- Hoque, S. (2020). Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js. In *Packt Publishing Ltd* (2nd ed.). Packt Publishing Ltd. www.packt.com
- Hutchison, P. D., Daigle, R. J., & George, B. (2018). Application of latent semantic analysis in AIS academic research. *International Journal of Accounting Information Systems*, 31(June), 83–96. <https://doi.org/10.1016/j.accinf.2018.09.003>
- Imam Riadi, Sunardi, & Widiandana, P. (2020). Investigating Cyberbullying on WhatsApp Using Digital Forensics Research Workshop. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(4), 730–735. <https://doi.org/10.29207/resti.v4i4.2161>
- Kadhim, A. I. (2018). An Evaluation of Preprocessing Techniques for Text Classification. *International Journal of Computer Science and Information Security*, 16(6), 22–32. <https://sites.google.com/site/ijcsis/>
- Kaya, A., Ozturk, R., & Altin Gumussoy, C. (2019). Usability Measurement of Mobile Applications with System Usability Scale (SUS). 389–400. https://doi.org/10.1007/978-3-030-03317-0_32
- Kim, S. W., & Gil, J. M. (2019). Research paper classification systems based on TF-IDF and LDA schemes. *Human-Centric Computing and Information Sciences*, 9(1). <https://doi.org/10.1186/s13673-019-0192-7>
- Kinaswara, T. A., Hidayati, N. R., & Nugrahanti, F. (2019). Rancang Bangun Aplikasi Inventaris Berbasis Website pada Kelurahan Bantengan. *Prosiding Seminar Nasional Teknologi Informasi Dan Komunikasi (SENATIK)*, 2(1), 71–75. <http://prosiding.unipma.ac.id/index.php/SENATIK/article/view/1073>
- Knuth, I., & Masuhr, J. (2021). Decision Drivers for Search Engine Usage – The Mediating and Moderating Role of Lock-in Effects. *International Journal of Business and Management*, 16(10), 86. <https://doi.org/10.5539/ijbm.v16n10p86>
- Li, C., & Liu, K. (2021). *Smart Search Engine : A Design and Test of Intelligent Search of News with Classification*. <https://urn.kb.se/resolve?urn=urn:nbn:se:du-37601>
- Liu, C., Sheng, Y., Wei, Z., & Yang, Y.-Q. (2018). Research of Text Classification Based on Improved TF-IDF Algorithm. *International Conference of Intelligent Robotic and Control Engineering*, 2171(1), 218–222. <https://doi.org/10.1088/1742-6596/2171/1/012021>
- McDonnell, I., & MacKinnon, T. (2022). *Studi Kasus: Misinformasi di Indonesia - GeoPoll*. <https://www.geopoll.com/misinformation-indonesia/>
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A Practitioner's Approach* (Ninth). McGraw-Hill Education.

- Rahmad, F., Suryanto, Y., & Ramli, K. (2020). Performance Comparison of Anti-Spam Technology Using Confusion Matrix Classification. *IOP Conference Series: Materials Science and Engineering*, 879(1). <https://doi.org/10.1088/1757-899X/879/1/012076>
- Rauf, B. W., Raharjo, S., & Sismoro, H. (2020). Deteksi Clickbait dengan Sentence Scoring Based On Frequency di Detik.Com. *Jurnal Teknologi Informasi*, 4(2), 247–252. <https://doi.org/10.36294/jurti.v4i2.1381>
- Rusbandi, M., Rozi, I. F., & Batubulan, K. S. (2021). Otomatisasi Peringkasan Teks Pada Dokumen Hukum Menggunakan Metode Latent Semantic Analysis. *Jurnal Informatika Polinema*, 7(3), 9–16. <https://doi.org/10.33795/jip.v7i3.515>
- Saptono, I., Baktiati, B., Iswandono, D., & Ratih, B. T. S. (2021). *Media Berkelanjutan di Masa Pandemi Laporan Tahunan Dewan Pers 2021*.
- Saravanan, T., Jha, S., Sabharwal, G., & Narayan, S. (2020). Comparative Analysis of Software Life Cycle Models. *Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020*, 906–909. <https://doi.org/10.1109/ICACCCN51052.2020.9362931>
- Sauro, J. (2018). *5 Ways to Interpret a SUS Score*. MeasuringU. <https://measuringu.com/interpret-sus-score/>
- Sholeh, M., Gisfas, I., Cahiman, & Fauzi, M. A. (2021). Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods. *Journal of Physics: Conference Series*, 1823(1). <https://doi.org/10.1088/1742-6596/1823/1/012029>
- Sihombing, W. L., Pratama, R. N. A. C. P., Gaol, Y. H. L., Istiqomah, Sasti, A. F., & Haolia. (2018). Singular Value Decomposition untuk Matriks 5x5 Willi. *Journal Contribution*, 1. <https://doi.org/10.6084/m9.figshare.5946976>
- Softscients. (2021). *Membuat Document Term Matrix - Softscients*. <https://softscients.com/2021/02/16/membuat-document-term-matrix/>
- Sukmayadi, V. (2019). The Dynamics of Media Landscape and Media Policy in Indonesia. *Asia Pacific Media Educator*, 29(1), 58–67. <https://doi.org/10.1177/1326365X19844853>
- Surianto, D. F., Angriani, R., Kadir, P., Syafaat, F., & Fakhri, M. M. (2022). Implementasi Metode Latent Semantic Analysis Pada Peringkasan Artikel Bahasa Indonesia Menggunakan Pendekatan Steinberger Jezeck. 9(4), 894–901. <https://doi.org/10.30865/jurikom.v9i4.4620>
- Tran, T. T. H. (2022). *the Development of an E-Commerce Web Application Using Mern Stack*.

- Trunkett, O. (2020, August 27). *SDLC Methodologies: From Waterfall to Agile*. Virtasant. <https://www.virtasant.com/blog/sdlc-methodologies>
- Verma, A., Khatana, A., & Chaudhary, S. (2017). A Comparative Study of Black Box Testing and White Box Testing. *International Journal of Computer Sciences and Engineering*, 5(12), 301–304. <https://doi.org/10.26438/ijcse/v5i12.301304>
- Wagire, A. A., Rathore, A. P. S., & Jain, R. (2019). Analysis and synthesis of Industry 4.0 research landscape Using latent semantic analysis approach. *Journal of Manufacturing Technology Management*, 34(1), 1–5.
- William, A., & Sari, Y. (2020). CLICK-ID: A novel dataset for Indonesian clickbait headlines. *Data in Brief*, 32, 106231. <https://doi.org/10.1016/j.dib.2020.106231>
- Yuhadisi, S. (2021). Penerapan Metode Modifikasi Hosmer-Lemeshow Test pada Model Regresi Logistik Data Penderita Penyakit Hipertensi. *Prosiding Statistika*, 7(1), 50–55.



UNIVERSITAS
Dinamika