

BAB III METODOLOGI PENELITIAN

3.1 Subyek dan Obyek Penelitian

Subyek penelitian ini merupakan data dari abstraksi tugas akhir mahasiswa Fakultas Informatika pada tahun 2022 yang diambil dari repositori Institut Teknologi Telkom Purwokerto (ITTP). Sedangkan obyek penelitian ini adalah topik modeling dari abstraksi tugas akhir mahasiswa di fakultas informatika pada tahun 2022.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Penelitian ini menggunakan alat penelitian berupa perangkat keras dan perangkat lunak sebagai alat untuk menunjang penelitian, sebagai berikut :

1. Perangkat Keras

- a. Laptop Acer E5475G
- b. *Mouse*

2. Perangkat Lunak

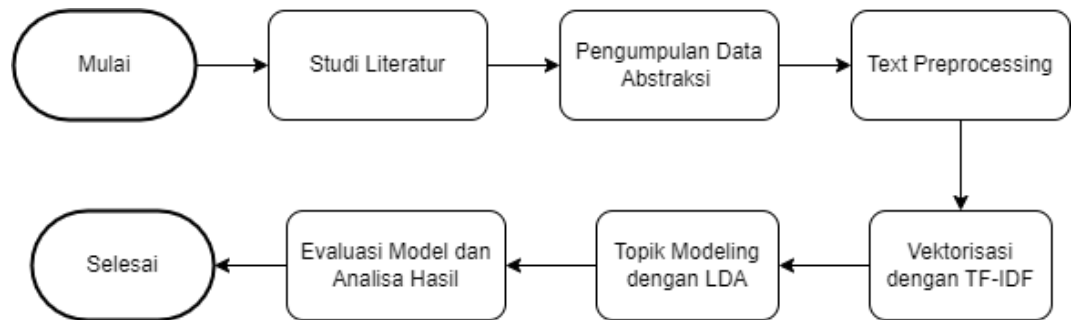
- a. *Windows 10 Home Edition*
- b. *Microsoft Office 2019*
- c. *Jupyter Notebook*
- d. *Python 3.9.1*
- e. *Google Chrome*

3.2.2 Bahan Penelitian

Bahan penelitian yang digunakan didapat dari abstraksi di tugas akhir mahasiswa fakultas informatika pada tahun 2022 yang diambil dari *website* repositori ITTP.

3.3 Diagram Alir Penelitian

Penelitian ini dilakukan sesuai dengan tahapan pada Gambar 3.1.



Gambar 3.1. Tahapan Penelitian

3.3.1 Studi Literatur

Studi literatur dilakukan untuk mendapatkan informasi terkait penelitian yang dilakukan. Informasi yang didapatkan oleh berupa referensi-referensi tentang algoritma yang dipakai, referensi tentang penelitian yang telah dilakukan sebelumnya oleh peneliti lain dengan metode yang sama. Hal ini dilakukan untuk mendapatkan informasi yang menunjang penelitian yang dilakukan.

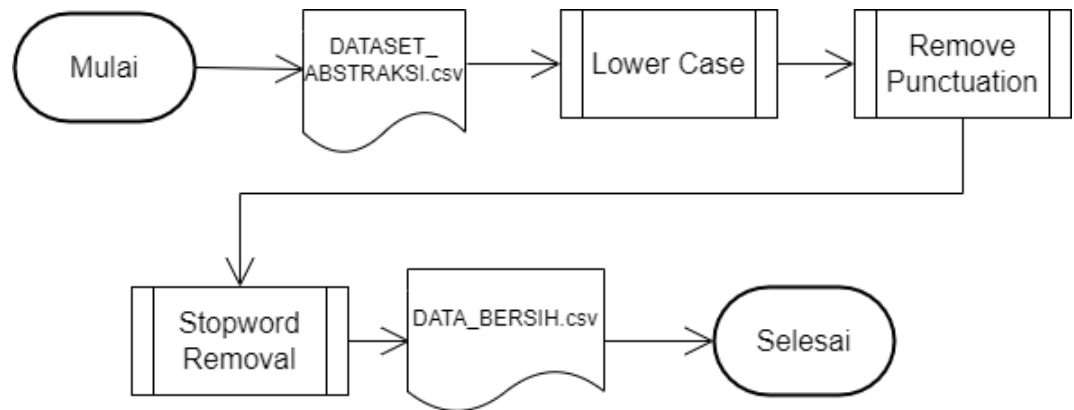
3.3.2 Pengumpulan Data

Pengumpulan data dilakukan untuk mencari data yang nantinya dijadikan obyek penelitian ini. Data diambil secara manual dari *website* repositori ITTP yang berjumlah 311 abstraksi. Abstraksi pada Tugas Akhir merupakan rangkuman dari inti topik pembicaraan pada penelitian Tugas Akhir tersebut.

3.3.3 Text Preprocessing

Tahapan *text preprocessing* ditunjukkan pada Gambar 3.2. yaitu terdiri memanggil file data dalam format csv, mengubah menjadi huruf kecil,

menghapus tanda baca, menghilangkan *stopword*, selanjutnya hasil data akan disimpan dalam bentuk csv.



Gambar 3.2. Text Preprocessing

Seperti yang dijabarkan pada Gambar 3.2. tahapan *preprocessing* yang digunakan pada penelitian ini adalah *lower case*, *remove punctuation*, dan *stopword removal*. *Preprocessing* bertujuan sebagai langkah awal yang digunakan sebagai pembersihan data yang nantinya akan digunakan dalam penelitian.

1. Lower Case

Lower case adalah proses untuk merubah huruf kapital yang terdapat di suatu kata/kalimat menjadi huruf kecil. Hasil dari proses *lower case* dapat dilihat di Tabel 3.1

Tabel 3 1. Hasil Lower Case

Sebelum	Sesudah
Program ini dibuat untuk DINPORA	program ini dibuat untuk dinpora
Lower case digunakan untuk mengecilkan tulisan	lower case digunakan untuk mengecilkan tulisan

2. Remove Punctuation

Remove punctuation adalah proses untuk membersihkan data dari tanda baca yang tidak dipakai dalam penelitian. Terdapat beberapa karakter yang tidak

terpakai didalam penelitian ini seperti “, -, _, dan sebagainya. Hasil dari *remove punctuation* dapat dilihat di Tabel 3.2.

Tabel 3 2. Hasil Remove Punctuation.

Sebelum	Sesudah
program ini dibuat untuk dinpora.	program ini dibuat untuk dinpora
lower case digunakan untuk mengecilkan tulisan.	lower case digunakan untuk mengecilkan tulisan

3. Stopword Removal

Stopword removal adalah proses untuk menghilangkan kata-kata yang tidak memiliki makna. Terdapat beberapa kata yang tidak memiliki makna yang tidak digunakan dalam penelitian ini seperti “yang”, “dari”, dan sebagainya. Hasil dari *stopword removal* dapat dilihat di Tabel 3.3.

Tabel 3 3. Hasil Stopword Removal.

Sebelum	Sesudah
program ini dibuat untuk dinpora.	program dinpora.
lower case digunakan untuk mengecilkan tulisan.	lower case mengecilkan tulisan.

3.3.4 Vektorisasi dengan TF-IDF

Sebelum membangun model LDA, diperlukan adanya tahap vektorisasi. Vektorisasi atau pembobotan kata pada penelitian ini menggunakan cara *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF dihitung dengan menggunakan persamaan seperti pada persamaan (2.3). Misalkan, kita akan mencari kata bobot TF-IDF kumpulan dokumen berikut:

Dokumen 1 (D1) = Figma sering digunakan pada proses perancangan UI/UX aplikasi website maupun aplikasi lainnya Praktik Kerja Lapangan (PKL) bermanfaat sebagai tolak ukur pengalaman terjun.

Dokumen 2 (D2) = Figma adalah tools yang dapat menjawab semua permasalahan dalam mendesain maupun meredesain sebuah website.

Dokumen 3 (D3) = Perancangan redesign website memerlukan proses yang didukung oleh tools yang dapat mempermudah dalam proses pembuatannya.

Ketiga dokumen tersebut, setelah dibersihkan maka akan menjadi sebagai berikut.

Tabel 3 4. Contoh Hasil Text Processing

Nomor Dokumen	Hasil Pembersihan (Remove Punctuation dan Stopword Removal)
D1	['figma', 'proses', 'perancangan', 'uiux', 'aplikasi', 'website', 'aplikasi', 'praktik', 'kerja', 'lapangan', 'pkl', 'bermanfaat', 'tolak', 'ukur', 'pengalaman', 'terjun']
D2	['figma', 'tools', 'permasalahan', 'mendesain', 'meredesain', 'website']
D3	['perancangan', 'redesign', 'website', 'proses', 'didukung', 'tools', 'mempermudah', 'proses', 'pembuatannya']

Langkah pertama adalah mencari *Term Frequency* (TF) dari setiap kata atau *word* dalam dokumen D1, D2, D3. Hasil TF sebagai berikut.

Tabel 3 5. Contoh Hasil TF.

Feature (kata)	D1	D2	D3
aplikasi	2	0	0
bermanfaat	1	0	0
didukung	0	0	1
digunakan	1	0	0
figma	1	1	0
kerja	1	0	0
lapangan	1	0	0
memerlukan	0	0	1
mempermudah	0	0	1
mendesain	0	1	0
menjawab	0	1	0

Feature (kata)	D1	D2	D3
meredesain	0	1	0
pembuatannya	0	0	1
pengalaman	1	0	0
perancangan	1	0	1
permasalahan	0	1	0
pkl	1	0	0
praktik	1	0	0
proses	1	0	2
redesign	0	0	1
terjun	1	0	0
tolak	1	0	0
tools	0	1	1
uiux	1	0	0
ukur	1	0	0
website	1	1	1

Langkah berikutnya adalah mencari bobot IDF setiap feature kata dalam dokumen. Rumus IDF dapat dilihat di persamaan 3.1.

$$IDF = \log_{10}\left(\frac{Term\ Frequency}{Document\ Frequency}\right) \quad (3.1)$$

Contoh IDF kata “aplikasi” muncul dua kali pada satu dokumen maka perhitungan bobot IDF kata “aplikasi” menjadi seperti berikut.

$$IDF(\text{aplikasi},j) = \log(2/1) = 0.3$$

Setelah didapatkan IDF, kemudian nilai TF-IDF dihitung dengan persamaan 3.2.

$$TF-IDF = Term\ Frequency \times Inverse\ Document\ Frequency \quad (3.2)$$

Hasil perhitungan IDF dan TF-IDF pada dokumen D1,D2,D3 adalah sebagai berikut.

Tabel 3 6. Contoh Hasil TF-IDF

Term	D1	D2	D3	TF	DF	IDF	TF-IDF
aplikasi	2	0	0	2	1	0,30103	0,60206
bermanfaat	1	0	0	1	1	0	0

<i>Term</i>	D1	D2	D3	TF	DF	IDF	TF-IDF
didukung	0	0	1	1	1	0	0
digunakan	1	0	0	1	1	0	0
figma	1	1	0	2	2	0	0
kerja	1	0	0	1	1	0	0
lapangan	1	0	0	1	1	0	0
memerlukan	0	0	1	1	1	0	0
mempermudah	0	0	1	1	1	0	0
mendesain	0	1	0	1	1	0	0
menjawab	0	1	0	1	1	0	0
meredesain	0	1	0	1	1	0	0
pembuatannya	0	0	1	1	1	0	0
pengalaman	1	0	0	1	1	0	0
perancangan	1	0	1	2	2	0	0
permasalahan	0	1	0	1	1	0	0
pkl	1	0	0	1	1	0	0
praktik	1	0	0	1	1	0	0
proses	1	0	2	3	2	0,176091	0,528274
redesign	0	0	1	1	1	0	0
terjun	1	0	0	1	1	0	0
tolak	1	0	0	1	1	0	0
tools	0	1	1	2	2	0	0
uiux	1	0	0	1	1	0	0
ukur	1	0	0	1	1	0	0
website	1	1	1	3	3	0	0

3.3.5 Pembangunan Model LDA

Model LDA dibangun menggunakan persamaan 2.1. Langkah-langkah penerapan LDA antara lain:

1. Mencari *coherence value* sebagai acuan untuk jumlah topik yang digunakan dalam pemodelan LDA.
2. Setelah dapat *coherence value*, maka masukkan jumlah topik yang didapat dari hasil perhitungan nilai *coherence value*, jumlah iterasi, parameter alpha dan beta yang ditentukan diawal.
3. Melakukan iterasi sesuai dengan jumlah yang ditentukan. Jumlah iterasi ditentukan dari awal 1000. Pada tiap iterasi kita menghitung distribusi probabilitas topik pada suatu dokumen dan distribusi probabilitas pada suatu

topik.

4. Setelah memiliki dua nilai distribusi, kita bisa memperbarui topik pada tiap kata dengan cara melakukan *sample* dengan bobot. Dimana bobot didapatkan dari perkalian nilai distribusi probabilitas topik pada dokumen dengan distribusi probabilitas kata pada topik.

3.3.6 Evaluasi Model dan Analisis Hasil

Analisis topik dilakukan dengan menganalisis topik-topik tiap *cluster* dalam model LDA yang dihasilkan. Untuk mendapatkan jumlah *cluster*, penulis menggunakan *coherence value* sebagai acuan. Jumlah *cluster* akan ditentukan dengan cara seberapa banyak *coherence value* di tiap *cluster* yang dapat melebihi atau sama dengan *coherence value* tertinggi. Setelah dapat jumlah clusternya, peneliti akan menganalisis sejumlah topik teratas sesuai nilai *coherence*.

Analisis hasil *term* di setiap topik (*cluster*) menggunakan PyLDAvis sebagai alat visualisasi. PyLDAvis dibangun berdasarkan LDAvis, yang merupakan alat untuk visualisasi interaktif untuk LDA. PyLDAvis secara otomatis akan menampilkan 30 *term* teratas dalam setiap topiknya. Berdasarkan penelitian [2], [4], analisis *term* teratas menggunakan contoh 10 kata teratas sebagai acuan analisis.