# bab iii

# metodologi penelitian

## 3.1. Data Penelitian

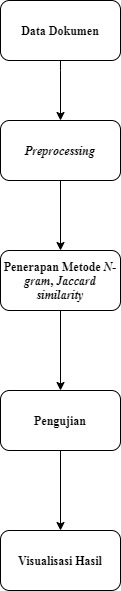
*Dataset* atau sumber data yang digunakan dalam penelitian ini bersumber dari data dokumen abstrak skripsi mahasiswa Budi Luhur. Data tersebut kemudian diolah melalui tahap *preprocessing* sehingga data tersebut bisa digunakan sebagai *dataset*. Berikut adalah contoh *record* dari data dokumen abstrak skripsi mahasiswa Budi Luhur hasil *preprocessing* dapat dilihat pada Tabel 3.1 berikut :

Tabel 1 3.1 Sampel Data Abstrak

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **nim** | **uploaddate** | **file\_size** | **content** |
| 1 | 1511500025 | 29-12-20 | 367036 | implementasi algoritma multinomial naive bayes classifier tf ltf stemming nazief adriani klasifikasi sentimen alam interview twitter rausal valino adjir humaika resources consulting konsultan human resources departement hrd management depan percaya naung….. |
| 2 | 1511500082 | 29-12-20 | 23007 | implementasi algoritma depth first search maze based level generator prototipe game d top down shooter proposterous defiance bas mobile fadel achmad assegaf video game hibur suka dewasa video game hilang jenuh jalan aktivitas hari latih konsentrasi fokus…. |
| 3 | 1511500132 | 29-12-20 | 41297 | implementasi aman file algoritma advanced encryption standard aes bit bas web smk pgri ciledug imbar yuli hartoko dikarnakan kembang komputer aman komputer simpan tukar sifat rahasia smk pgri ciledug instansi gerak tangerang file soal uji file sifat rahas…. |
| 4 | 1511500157 | 29-12-20 | 177537 | implementasi augmented reality metode marker based tracking briefing terima calon karyawan magang sewiwi deddy rivaldy teliti tuju kembang aplikasi augmented reality briefing media kertas sewiwi briefing isi tulis jarang calon karyawan magang bingung baya…. |
| 5 | 1511500199 | 29-12-20 | 99221 | aplikasi aman file algoritma advanced encryption standard aes bas web toko khayangan outdoor yayat khayangan outdoor gerak jual alat daki buku ketik ulang komputer file rentan curi rusak kam guna mesti simpan sadap susup tanggung curi pindah simpan komput…. |

## 3.2. Penerapan Metode

Untuk membangun sistem pendeteksi plagiarisme menggunakan metode *n-gram* dan *Jaccard similarity* terhadap algoritme *winnowing,* terdapat beberapa tahapan yang menjadi rancangan utama, rancangan ini sebagai gambaran proses tahapan awal hingga akhir sistem berjalan, yang terdapat pada Gambar 3.1 berikut.



Gambar 1 3.1 Tahapan Metode

Pada Gambar 3.1, proses *upload* dokumen dilakukan untuk mendapatkan *dataset* dan proses pengujian *similarity,* dalam penelitian ini *dataset* diperoleh dari data dokumen abstrak skripsi mahasiswa Budi Luhur. Kemudian dokumen abstrak dalam bentuk *pdf*  ini di­*upload* ke dalam sistem agar bisa di *convert* menjadi teks, lalu di simpan ke *database* untuk dilakukan proses *preprocessing.* Agar bisa dilakukan proses *case folding,* menghilangkan karakter, *slang word* dan *stop word* sehingga dokumen ini bisa digunakan sebagai *dataset.* Kemudian untuk proses pengecekan *similarity,* pertama dilakukan proses *scan* dokumen abstrak berpentuk *pdf* yang di*scan* melalui sistem, lalu sistem meng*convert* dokumen *pdf* tadi menjadi teks, kemudian masuk ke tahap implementasi metode untuk mengecek tingkat *similarity* dari dokumen yang di*scan* tadi.

### 3.2.1. Data Dokumen

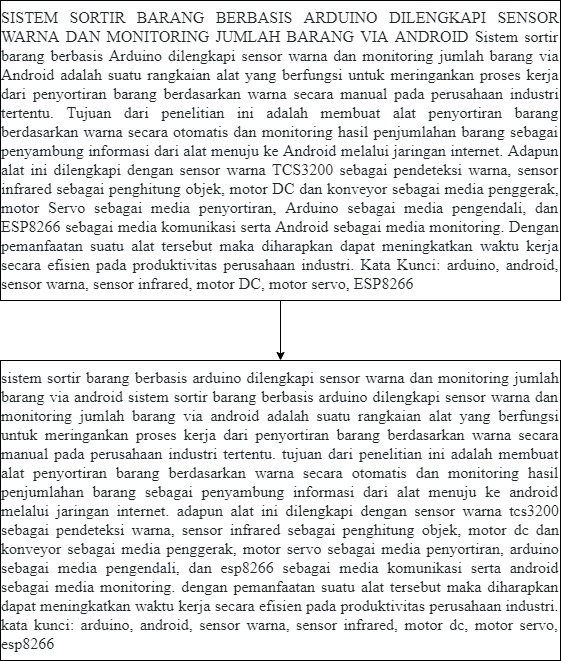
Pada tahap ini dilakukan proses pengumpulan data dokumen abstrak dari skripsi mahasiswa Budi Luhur. Dimana data dokumen ini ada yang dijadikan sebagai *dataset* dan ada juga yang menjadi data tes. *Dataset* ini berasal dari dokumen abstrak skripsi mahasiswa yang sudah di publish di perpustakaan Budi Luhur. Sedangkan data tes berasal dari dokumen abstrak mahasiswa yang akan menjalankan skripsi, sehingga mahasiswa bisa melakukan pengecekan *similarity* di dokumen abstraknya.

### 3.2.2. *Preprocessing*

Pada tahapan *preprocessing* ini, dilakukan beberapa proses untuk mendapatkan *dataset* yang bersih, sehingga proses pengujian tingkat *similarity* lebih tepat dan akurat. Proses *preprocessing* yang digunakan pada penelitian ini diantaranya yaitu :

1. *Casefolding*

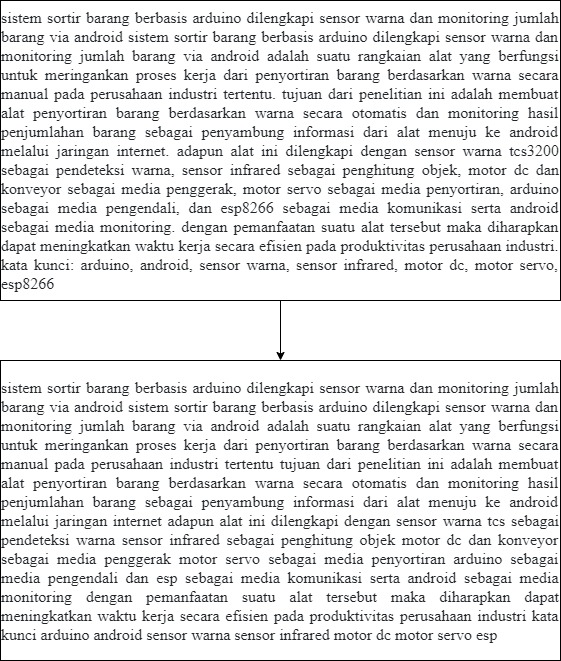
Pada Gambar 3.2 dilakukan penyetaraan kata yang mengandung huruf besar untuk diubah menjadi huruf kecil, misalnya Tujuan menjadi tujuan, Adapun menjadi adapun, dan seterusnya.



Gambar 2 3.2 *Casefolding*

1. Menghapus karakter selain a-z

Pada Gambar 3.3 menjelaskan bahwa karakter selain a sampai z atau karakter selain huruf, karakter tersebut dihilangkan, seperti contoh 8266, / (garis miring), dan seterusnya sehingga hanya menyisakan karakter huruf saja.



Gambar 3 3.3 Menghapus karakter selain a-z

1. Mengganti *Slang Word*

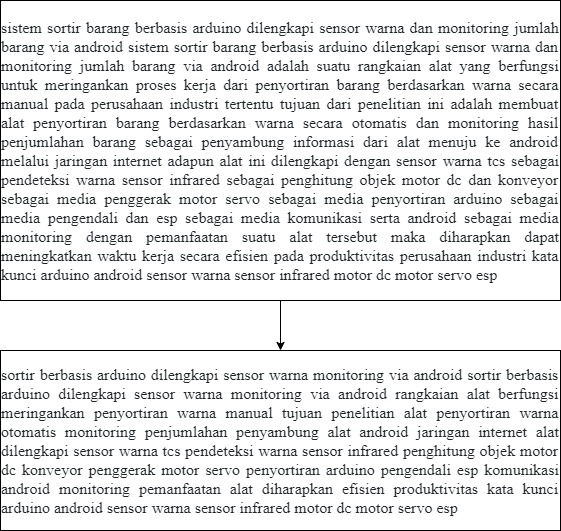
Pada Gambar 3.4 mengganti kata tidak baku seperti “analisa” dan “clien”, kata yang tidak baku tersebut biasanya berupa singkatan atau bahasa yang kekinian, untuk itu supaya kata-kata dalam teks setara dengan EYD, maka kata tersebut digantikan dengan kata baku yang seharusnya yaitu menjadi “analisis” dan “klien”, pergantian kata-kata ini berdasarkan kamus yang terdapat dalam library slangword.



Gambar 4 3.5 Mengganti *Slang Word*

1. Menghapus *Stop Word*

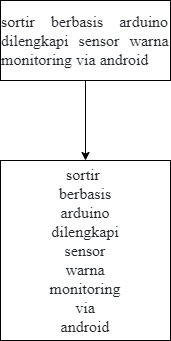
Dalam penelitian Tala, (2003), terdapat kumpulan kata-kata yang termasuk ke dalam *stoplist* yaitu kata umum yang dianggap tidak terlalu memiliki makna yang penting dan kemunculan kata ini sangat tinggi frekuensinya, seperti pada contoh teks di bawah ini yaitu pada kata “untuk”, “dan”, “bagi”, kata tersebut dihilangkan karena masuk ke dalam daftar *stop word*.



Gambar 5 3.6 Menghapus *Stop Word*

1. *Tokenization*

Pada Gambar 3.7, proses tokenisasi dilakukan pemisahan sebuah kalimat ke dalam bentuk pencahan per satu kata. Kata-kata yang telah dipisahkan kemudian dimasukan ke dalam sebuah index array untuk digunakan pada tahap berikutnya yaitu proses penerapan metode *n-gram.*



Gambar 6 3.7 *Tokenization*

### 3.2.3. *N-Gram*

Dalam menentukan hasil deteksi plagiarisme dengan menggunakan metode *n-gram*,dimana dokumen atau sekumpulan kata akan diproses dan akan dibentuk sebanyak *n-gram* atau memisahkan string sepanjang n yang akan dihitung pergeserannya secara terus menurus ke depan sejumlah nilai n sampai akhir dokumen. Sebagai contoh *n-gram* dari kalimat **N = Sortir berbasis Arduino**, dengan nilai N=3 maka (sor ort rti tir irb rbe ber erb rba bas asi sis isa sar ard rdu dui uin ino).

### 3.2.4. *Rolling Hash*

Pada tahapan ini dilakukan perhitungan *rolling hash* untuk mencari nilai *hash* dari setiap *string* yang sudah di potong pada tahapan *n-gram*. Setiap *string* diubah menjadi ASCII lalu dihitung dengan rumus *hash* dibawah:

…(3.1)

c1 = nilai ascii dari huruf pertama dari satu buah kgram

b = bilangan prima

k = nilai *k-gram*

Setelah hasil dari *hash* pertama didapatkan selanjutnya digunakan rumus *rolling hash* untuk menghitung nilai *hash* ke dua dan seterus nya tanpa menghitung ASCII yang sudah di hitung sebelumnya.

..(3.2)

Hasil dari perhitungan sebelum nya di kurangi hasil *hash* dari huruf pertama dari *gram* sebelumnya lalu dikali dengan hasil *hash* dari perhitungan ASCII terakhir dari *gram* sekarang.

### 3.2.5. Pembentukan *Window*

Setelah nilai *hash* ditemukan semua maka langkah selanjutnya adalah pembentukan *window*. Nilai dari *rolling hash* akan dikelompokan berdasarkan nilai *wgram*. Pada kasus ini nilai *window* yang dipergunakan adalah w=3.

### 3.2.6. Nilai *Fingerprint*

Proses selanjutnya setelah pembentukan *window* adalah mencari nilai *fingerprint.* Nilai *fingerprint* ditentukan berdasarkan nilai terkecil dari setiap *window* yang ada.

### 3.2.7. *Jaccard Similarity*

Pada tahapan ini untuk mencari nilai *similarity* pada sebuah dokumen dengan rumus *Jaccard similarity* dengan melihat *irisan* dan *union* dari *fingerprint* antar dua dokumen. Berikut adalah rumus *jaccard similarity* :

*Similarity* (X,Y) =x 100% ……………………(3.3)

X = Dokumen 1

Y = Dokumen 2

Dimana *irisan* dari dokumen X dan Y dibagi dengan *union* dari dokumen X dan Y kemudian di kali 100% maka menghasilkan persentase *similarity* dari dokumen X dan Y.

## 3.3. Rancangan Pengujian

Pengujian dilakukan untuk mengetahui tingkat *similarity* dari sebuah dokumen. Pada penelitian ini digunakan metode pengujian *Black box Testing* dalam melakukan proses pengujian program. *Black Box Testing* merupakan pengujian yang dapat dilakukan dengan melakukan pengamatan, pada hasil eksekusi melalui beberapa data tes dan memeriksa fungsional yang terdapat pada perangkat lunak. Pada langkah pengujian ini akan dijelaskan mengenai proses yang berjalan pada sistem yang dibangun, mulai dari *upload* dokumen, menjadikan dokumen sebagai *dataset* dan data tes sampai menampilkan hasil perhitungan tingkat *similarity.*