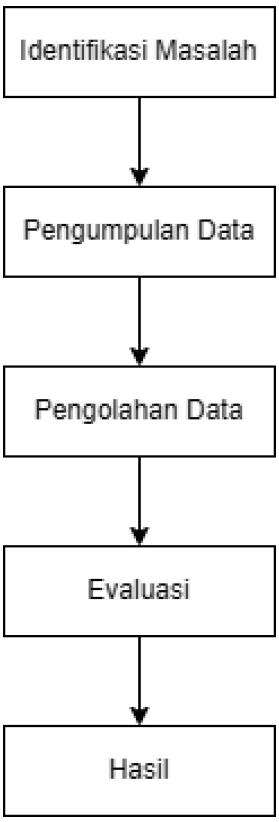
BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Proses penelitian menjadi lebih mudah dengan adanya desain penelitian yang menjelaskan alur kegiatan secara sistematis selama penelitian berlangsung. Berikut adalah tahapan-tahapan desain penelitian yang dirancang agar penelitian dapat berjalan secara terarah dan efektif.



Gambar 3. 1 Desain Penelitian

Sumber: data penelitian 2025

Penelitian dimulai dengan mengidentifikasi masalah utama, yaitu membedakan antara email sah (non-spam) dan spam menggunakan algoritma *naïve bayes*. Tahap berikutnya adalah pengumpulan data melalui studi literatur terkait spam email, machine learning, dan metode *Naive Bayes*, dengan sumber dari buku dan jurnal nasional maupun internasional. Data yang telah dikumpulkan kemudian diproses untuk memastikan bahwa data tersebut dalam format yang dapat digunakan oleh model. Setelah data diproses, model *Naive Bayes* dilatih menggunakan data latih (training data). Pada tahap ini, model belajar untuk membedakan antara spam dan non-spam berdasarkan pola yang ada pada data latih.

Setelah model dilatih, dilakukan pengujian menggunakan data uji (test data) yang tidak terlihat oleh model saat pelatihan. Hasil prediksi dari model kemudian dibandingkan dengan label asli untuk melihat seberapa baik model dalam mengklasifikasikan email sebagai spam atau non-spam. Hasil pengujian dievaluasi dengan menggunakan metrik seperti akurasi, *precision*, *recall*, dan *F1-score* untuk menilai performa model dalam mengklasifikasikan data dengan benar. Berdasarkan evaluasi, hasil model dapat dianalisis untuk menentukan apakah model sudah cukup baik dalam memisahkan spam dan non-spam. Jika hasilnya kurang memuaskan, maka perbaikan dapat dilakukan pada proses pelatihan atau pengolahan data.

3.2 Metode pengumpulan data

Dalam penelitian ini, data yang digunakan merupakan data sekunder yang diambil dari dataset publik. Data ini tidak dihasilkan melalui pengumpulan primer, seperti wawancara atau observasi, tetapi diperoleh dari sumber terpercaya yang menyediakan dataset untuk penelitian. Dataset yang digunakan dalam penelitian ini

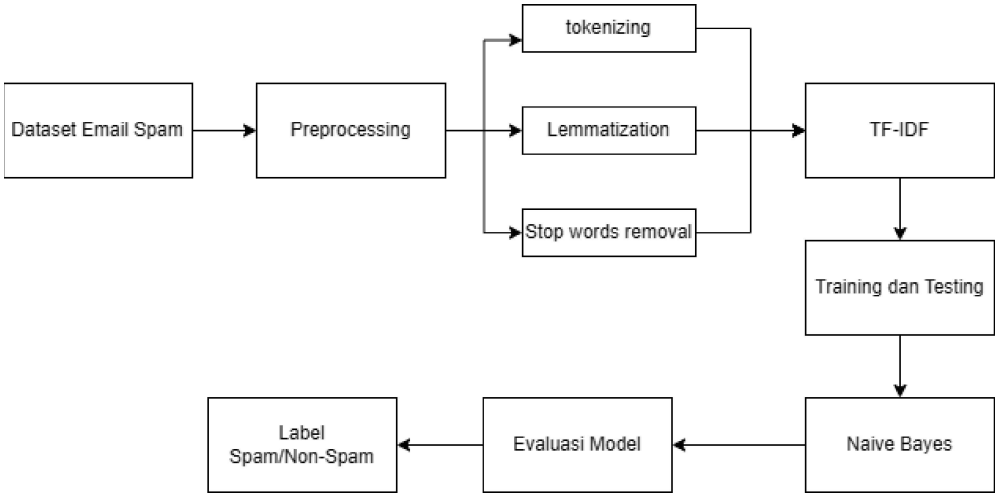
diunduh dari Harvard Dataverse, sebuah platform penyimpanan data terbuka yang menyediakan berbagai dataset berkualitas untuk keperluan analisis dan pengembangan lebih lanjut.

Dataset ini terdiri dari 5.728 baris data, di mana setiap baris merepresentasikan sebuah email yang terbagi menjadi dua kategori utama, yaitu spam dan non-spam (ham). Secara spesifik, dataset ini terdiri dari 1.369 data spam dan 4.329 data non-spam, yang disajikan dalam format tabular dengan dua kolom utama: kolom "text" yang berisi teks email, dan kolom "spam" yang berisi label kategori, berupa "spam" atau "ham".

Dataset ini digunakan untuk melatih dan menguji model deteksi spam berbasis algoritma *Naive Bayes*. Sebelum digunakan, data mentah ini melalui tahapan preprocessing, seperti penghapusan stopwords, lemmatisasi, dan konversi teks menjadi representasi numerik menggunakan metode *TF-IDF*, agar dapat digunakan dalam proses analisis lebih lanjut.

3.3 Metode Perancangan

Agar alur kerja algoritma *Naive Bayes* dapat dipahami dengan baik, penelitian ini mengacu pada kerangka analisis berikut.



Gambar 3. 2 Metode Perancangan

Sumber: data penelitian 2025

Mengacu pada ilustrasi tersebut, proses analisis klasifikasi spam email dengan algoritma *Naive Bayes* membutuhkan beberapa langkah penyelesaian seperti berikut ini.

3.3.1 Preprocessing data

Dataset yang akan digunakan diambil dari Harvard Dataverse, dengan jumlah 1369 pada data kategori spam, serta data non spam yang berjumlah 4329. Sebelum data digunakan untuk membangun model, dilakukan proses *preprocessing* untuk memastikan data dalam format yang sesuai. data terlebih dahulu dihapus karakter khusus, kata-kata yang tidak penting, serta diubah menjadi huruf kecil. Setelah itu dilakukan langkah-langkah seperti dibawah ini.

3.3.1.1 Tokenization

Tokenisasi merupakan proses awal dalam *preprocessing* teks yang bertujuan untuk memecah teks atau kalimat menjadi bagian-bagian yang lebih kecil, yaitu token. Token ini umumnya berupa kata-kata atau simbol yang dapat digunakan

dalam analisis lebih lanjut. Proses tokenisasi sangat penting karena memungkinkan teks dipecah menjadi unit-unit yang dapat dianalisis dengan lebih mudah, terutama dalam algoritma klasifikasi seperti *Naive Bayes*. Langkah-langkah Tokenisasi adalah sebagai berikut:

1. Mengambil Teks Mentah: Data yang digunakan dalam penelitian ini terdiri dari teks email yang dapat berisi kalimat, simbol, dan tanda baca. Sebelum proses tokenisasi, teks ini berupa kalimat utuh tanpa pemisah kata yang jelas.
2. Memecah Kalimat menjadi Token: Tokenisasi dilakukan dengan memecah teks kalimat menjadi kata-kata atau token. Misalnya, kalimat "*Congratulations! You have won a free vacation*." akan diubah menjadi token-token seperti [*"Congratulations", "You", "have", "won", "a", "free", "vacation"*].
3. Alat yang Digunakan: Tokenisasi pada penelitian ini dilakukan menggunakan *library Natural Language Toolkit (NLTK)* pada *Python*, yang menyediakan berbagai fungsi untuk tokenisasi teks. Fungsi *word_tokenize* pada *NLTK* digunakan untuk memecah kalimat menjadi kata-kata individu.

3.3.1.2 Lemmatization

Lemmatization ialah teknik yang digunakan untuk mengubah kata-kata menjadi bentuk dasarnya. *lemmatization* melakukan pemrosesan dengan mempertimbangkan konteks dan tata bahasa dari kata tersebut. dalam implementasinya pada python, proses ini menggunakan pustaka *NLTK*, khususnya menggunakan fungsi *WordNetLemmatizer*. Hal ini penting untuk mengurangi variasi kata yang tidak perlu, sehingga analisis teks lebih terfokus pada makna kata-kata yang relevan. Langkah-langkah *Lemmatization*:

1. Tokenisasi Teks: Sebelum melakukan *lemmatization*, teks yang telah ditokenisasi pada langkah sebelumnya akan digunakan. Tokenisasi memastikan bahwa teks sudah terpisah menjadi unit-unit kecil (kata atau token).
2. Penerapan *Lemmatization*: Proses *lemmatization* diterapkan pada setiap token menggunakan tool atau library tertentu. Pada penelitian ini, digunakan *WordNetLemmatizer* dari *library NLTK* untuk melakukan *lemmatization*. Contoh hasil pada tahap ini yaitu, misalnya kata *running* diubah menjadi *run*, kata *better* diubah menjadi *good*, serta kata *cats* diubah menjadi *cat*.

3.3.1.3 Stopwords removal

Stopwords removal merupakan proses penghilangan kata-kata yang dianggap tidak penting dalam analisis teks, karena kata-kata ini umumnya tidak memberikan informasi yang relevan untuk tujuan klasifikasi. Kata-kata tersebut sering kali berupa kata penghubung, kata depan, atau kata kerja bantu yang sering muncul dalam kalimat, tetapi tidak membantu dalam membedakan antara kategori-kategori teks yang berbeda. Sebagai contoh, kata seperti *"and", "or", "the", "for", "is",* dan lainnya termasuk dalam kategori *stopwords*.

Stopwords removal dilakukan menggunakan daftar *stopwords* yang sudah ada, seperti yang disediakan oleh pustaka *NLTK* (*Natural Language Toolkit*) pada *Python*. Daftar ini sudah mencakup kata-kata umum yang ada dalam bahasa Inggris (dan bahasa lainnya) yang umumnya dianggap tidak relevan dalam analisis teks.

3.3.2 TF-IDF

TF-IDF merupakan metode untuk memberikan bobot pada setiap kata dalam dokumen berdasarkan seberapa sering kata tersebut muncul di dokumen atau

dikenal sebagai *TF* (*Term Frequency*) dan seberapa jarang kata tersebut muncul pada dokumen *IDF* (*Inverse Document Frequency*). Contoh implementasinya adalah sebagai berikut.

Tabel 3. 1 Contoh Teks Email

Teks email	Kategori
Save your money by getting an oem software	Spam
any software backups for lowest pricest	Spam

Sumber: data penelitian 2025

Dari teks tersebut secara anomali dapat dikategorikan sebagai spam. Karena mengajak pemilik email untuk mengklik tautan yang disediakan. Oleh karena itu, perhitungannya adalah sebagai berikut:

hitung jumlah masing-masing kata pada setiap teks pesan:

- Teks 1: *save, your, money, getting, oem, software* = 6 kata
- Teks 2: *any, software, backup, for, low, price* = 6 kata

Langkah – langkah:

1. Menghitung *TF*

$$TF(t) = \frac{jumlah\ kata\ t\ dalam\ dokumen}{jumlah\ kata\ dalam\ dokumen} \quad \textbf{Rumus 1}$$

2. Menghitung *IDF*

$$IDF(t) = Log(\frac{Total\ Dokumen}{Dokumen\ yang\ mengandung\ kata\ t}) \quad \textbf{Rumus 2}$$

3. Menghitung TF-IDF

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \quad \textbf{Rumus 3}$$

Hasil perhitungannya dapat dilihat pada tabel berikut ini:

Tabel 3. 2 Perhitungan TF-IDF

Kata	TF Kalimat 1	TF Kalimat 2	DF	IDF	TF-IDF teks 1	TF-IDF teks 2
save	$^1/_6 = 0.16$	0	1	$Log\left(\frac{2}{1}\right) = 0.301$	$0.16 \times 0.301 = 0.048$	0
Your	$^1/_6 = 0.16$	0	1	$Log\left(\frac{2}{1}\right) = 0.301$	$0.16 \times 0.301 = 0.048$	0
money	$^1/_6 = 0.16$	0	1	$Log\left(\frac{2}{1}\right) = 0.301$	$0.16 \times 0.301 = 0.048$	0
getting	$^1/_6 = 0.16$	0	1	$Log\left(\frac{2}{1}\right) = 0.301$	$0.16 \times 0.301 = 0.048$	0
oem	$^1/_6 = 0.16$	0	1	$Log\left(\frac{2}{1}\right) = 0.301$	$0.16 \times 0.301 = 0.048$	0
Software	$^1/_6 = 0.16$	$^1/_6 = 0.16$	2	$Log\left(\frac{2}{2}\right) = 0$	0	0
Any	0	$^1/_6 = 0.16$	1	$Log\left(\frac{2}{1}\right) = 0.301$	0	$0.16 \times 0.301 = 0.048$
Backup	0	$^1/_6 = 0.16$	1	$Log\left(\frac{2}{1}\right) = 0.301$	0	$0.16 \times 0.301 = 0.048$
For	0	$^1/_6 = 0.16$	1	$Log\left(\frac{2}{1}\right) = 0.301$	0	$0.16 \times 0.301 = 0.048$
Low	0	$^1/_6 = 0.16$	1	$Log\left(\frac{2}{1}\right) = 0.301$	0	$0.16 \times 0.301 = 0.048$
Price	0	$^1/_6 = 0.16$	1	$Log\left(\frac{2}{1}\right) = 0.301$	0	$0.16 \times 0.301 = 0.048$

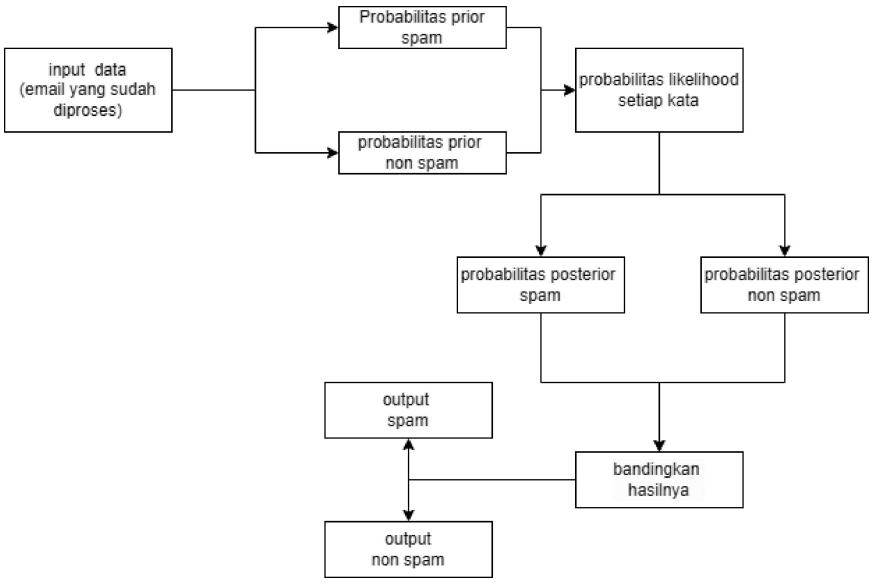
Sumber: data penelitian 2025

3.3.3 Training dan Testing

Dalam penelitian ini, proses *training* dan *testing* dilakukan untuk membangun dan menguji model klasifikasi email spam. Dataset yang digunakan dibagi menjadi dua bagian utama, yaitu *training set* dan *testing set*, untuk memastikan model dapat belajar dari data yang ada dan diuji dengan data yang belum pernah dilihat sebelumnya. Pembagian dataset dilakukan dengan metode *Hold-out validation*. pada proses ini data dibagi menjadi 2, yaitu *training* dan *testing* dengan rasio *training set* sebanyak 80% dan untuk data *testing* sebanyak 20%. Rasio ini dipilih agar model mendapatkan cukup banyak data untuk belajar tetapi tetap memiliki banyak data untuk menguji keakuratannya.

3.3.4 Naïve Bayes

Algoritma Naïve Bayes pada penelitian ini digunakan untuk mengklasifikasikan email berdasarkan 2 kategori, yaitu email spam dan non-spam. proses klasifikasi ini mengikuti langkah-langkah seperti yang dijelaskan pada flowchart berikut.



Gambar 3. 3 Cara Kerja Algoritma Naïve Bayes

Sumber: data penelitian 2025

Flowchart diatas menggambarkan tahapan utama algoritma naïve bayes terhadap klasifikasi email seperti sebagai berikut ini.

1. Input data: dataset email yang sudah dilakukan preprocessing dimasukkan kedalam sistem. Data ini berupa kumpulan teks email yang sudah dikonversi menjadi numerik menggunakan *TF-IDF.*
2. Perhitungan probabilitas prior: sistem menghitung probabilitas awal antara spam dan non spam berdasarkan distibusi data yang tersedia.
3. Perhitungan probabilitas likelihood: untuk setiap kata yang terdapat dalam email, sistem kemudian menghitung probabilitas likelihood, yaitu seberapa besar peluang kata tertentu muncul dalam masing masing kelas spam dan non spam.

4. Perhitungan probabilitas posterior: sistem mengalikan hasil probabilitas likelihood untuk setiap kata dengan probabilitas prior pada masing masing kelas spam dan non spam. setelah itu, sistem membandingkan hasil perhitungan antara kelas spam dan non spam, serta menentukan kelas berdasarkan hasil perhitungan yang lebih besar, hasilnya akan ditampilkan dalam bentuk output.

3.3.5 Evaluasi Model

Dataset akan dibagi menjadi data *training* sebesar 80% dan data *testing* sebesar 20%. Setelah itu model akan dilatih menggunakan data *training* dan diuji menggunakan data *testing*. Setelah model dilatih, confusion matrix akan dihitung berdasarkan prediksi model pada data testing. Nantinya *confusion matrix* ini digunakan untuk menghitung metrik-metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-Score*. *Confusion matrix* akan dihitung menggunakan *library Scikit-Learn* dengan fungsi *confusion_matrix()*. Sedangkan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-Score* akan dihitung dengan menggunakan fungsi *accuracy_score(), precision_score(), recall_score()* dan *f1-score()* dari *scikit learn.*

3.4 Lokasi dan jadwal

Penelitian ini berlokasi di universitas putera batam kampus tembesi yang berlokasi di jalan R.Soeprapto, Kota Batam. Penelitian ini dilakukan selama satu semester, dimulai dari bulan september 2024 hingga bulan januari 2025. Jadwal penelitian ini dapat dilihat pada table berikut.

Tabel 3. 3 Lokasi dan Jadwal Penelitian

No	Aktivitas	Tahun 2024 - 2025																			
		September				Oktober				November				Desember				Januari			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Membuat BAB 1																				
2	Membuat BAB 2																				
3	Membuat BAB 3																				
4	Membuat BAB 4																				

Sumber: data penelitian 2025