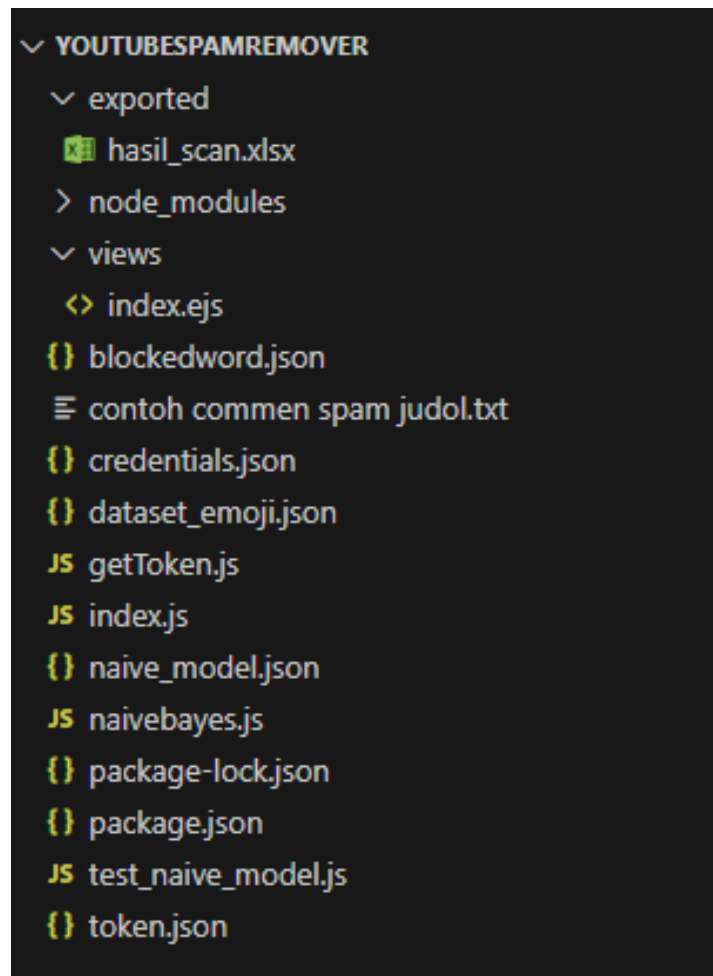


BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Implementasi sistem adalah tahapan untuk menerapkan sistem sehingga dapat dioperasikan. Berdasarkan implementasi sistem, dapat mengkonfirmasi modul-modul perancangan sehingga pengguna sistem dapat memberikan masukan-masukan terhadap pengembangan sistem dengan menggunakan aplikasi Visual Studio Code. Berikut gambar skruktur yang mencakup seluruh isi folder aplikasi yang sudah berjalan dengan lancar:



Gambar 4.1 Struktur Folder

4.1.1 Instalasi dan Konfigurasi Sistem

Terdapat beberapa tahap instalasi dan konfigurasi awal yang harus dilakukan pada lingkungan pengembangan. Proses ini bertujuan untuk menyiapkan dependensi yang diperlukan, termasuk pustaka Node.js dan kredensial akses API dari Google.

A. Instalasi Node.js dan npm

1. Unduh dan instal Node.js dari situs resmi “<https://nodejs.org>”
2. Lakukan instalasi dengan mengikuti petunjuk installer (klik “Next” hingga selesai)
3. Pastikan instalasi berhasil dengan menjalankan perintah berikut di terminal atau CMD: “node -v npm -v”

B. Instalasi Visual Studio Code

1. Aplikasi dapat diunduh dari “<https://code.visualstudio.com>”
2. Setelah instalasi selesai, VS Code digunakan untuk membuat dan mengelola struktur folder proyek.

C. Pembuatan Folder Proyek

1. Buka VS Code dan buat folder baru di penelitian ini di bernama “YouTubeSpamRemover”
2. Di dalam folder tersebut dibuat file utama bernama “index.js” sebagai entry point program.

D. Konfigurasi Google Cloud Console dan YouTube Data API

1. Masuk ke Google Cloud Console
2. Membuat proyek baru (atau menggunakan proyek yang telah ada)
3. Mengaktifkan YouTube Data API v3 klik Enable lalu manage melalui menu API & Services > Library
4. Membuat kredensial OAuth Client ID: Masuk ke API & Services > Credentials
5. Klik Create Credentials > OAuth client ID
6. Pilih Desktop App

7. Unduh file `credentials.json` dan simpan ke dalam folder proyek “YouTubeSpamRemover”

E. Mendapatkan Token kunci akses API YouTube

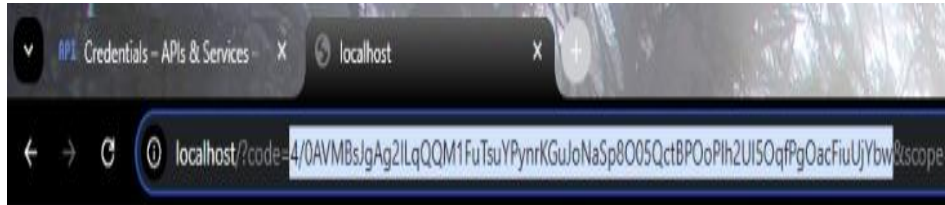
Fungsi Token digunakan untuk mengakses API YouTube atas nama pengguna dengan kodingan yang kita buat bisa berupa pengambilan komentar, penghapusan komentar, pengambilan data subscribe dan lain-lain, berikut cara mendapatkan Token :

1. Buat file di dalam folder “YouTubeSpamRemover” dengan nama (Misal: `GetToken.js`)
2. Isi kodingan sebagai berikut:

```
JS getToken.js > ...
1  const fs = require('fs');
2  const readline = require('readline');
3  const { google } = require('googleapis');
4  const SCOPES = ['https://www.googleapis.com/auth/youtube.force-ssl'];
5  const TOKEN_PATH = 'token.json';
6  // Load client secrets
7  fs.readFile('credentials.json', (err, content) => {
8    if (err) return console.error('Gagal membaca credentials.json', err);
9    authorize(JSON.parse(content));
10 });
11 function authorize(credentials) {
12   const { client_secret, client_id, redirect_uris } = credentials.installed;
13   const oAuth2Client = new google.auth.OAuth2(client_id, client_secret, redirect_uris[0]);
14   // Generate auth URL
15   const authUrl = oAuth2Client.generateAuthUrl({
16     access_type: 'offline',
17     scope: SCOPES,
18   });
19   console.log('\n🔑 Buka URL ini di browser Anda, lalu masukkan kode yang muncul:');
20   console.log(authUrl);
21   const rl = readline.createInterface({
22     input: process.stdin,
23     output: process.stdout,
24   });
25   rl.question('\nMasukkan kode: ', (code) => {
26     rl.close();
27     oAuth2Client.getToken(code, (err, token) => {
28       if (err) return console.error('❌ Gagal mendapatkan token:', err);
29       oAuth2Client.setCredentials(token);
30       fs.writeFileSync(TOKEN_PATH, JSON.stringify(token));
31       console.log('✅ Token disimpan di', TOKEN_PATH);
32     });
33   });
34 }
35
```

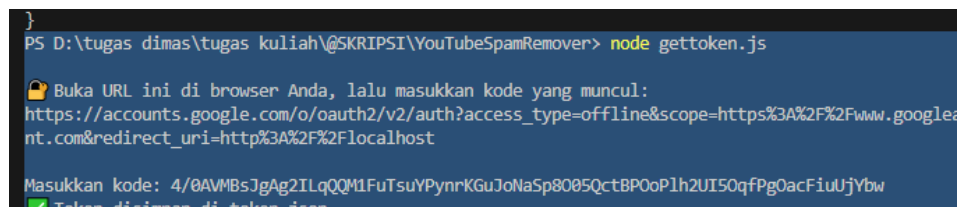
Gambar 4.2 Kodingan File "GetToken.js"

3. Jalankan “Node gettoken.js” di terminal maka akan keluar URL menuju ke browser
4. Pilih Akun Goggle yang digunakan > lanjutkan
5. Token akan keluar di URL atas lalu salin bagian yang merupakan kode akses



Gambar 4.3 Kode Akses Token

6. Tempel kode tersebut ke terminal:



Gambar 4.4 Tampilan Terminal GetToken.js

7. Maka Akan Otomatis file “Token.json” ada di folder “YouTubeSpamRemover”

4.1.2 Lingkungan Pengembangan

Sistem dikembangkan menggunakan kombinasi bahasa pemrograman JavaScript (Node.js) untuk sisi backend dan EJS (Embedded JavaScript) untuk antarmuka pengguna. Selain itu, Python juga digunakan dalam tahap awal pengambilan data komentar di YouTube dan pelabelan komentar dari format Excel ke JSON. Lingkungan pengembangan yang digunakan dirancang untuk mendukung pemrosesan data teks, integrasi dengan API eksternal (YouTube Data API), dan antarmuka pengguna berbasis web. Adapun perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

Tabel 4.1 Spesifikasi Lingkungan Pengembangan

Library	Perintah Instalasi	Fungsi
express	npm install express	Membuat server web
ejs	npm install ejs	Templating engine untuk antarmuka (tampilan web)
googleapis	npm install googleapis	Library resmi Google untuk akses ke YouTube Data API
natural	npm install natural	Untuk klasifikasi Naive Bayes
exceljs	npm install exceljs	Menulis hasil klasifikasi ke Excel
dotenv	npm install dotenv	Membaca konfigurasi .env secara aman
body-parser	npm install body-parser	Parsing data dari request form/JSON

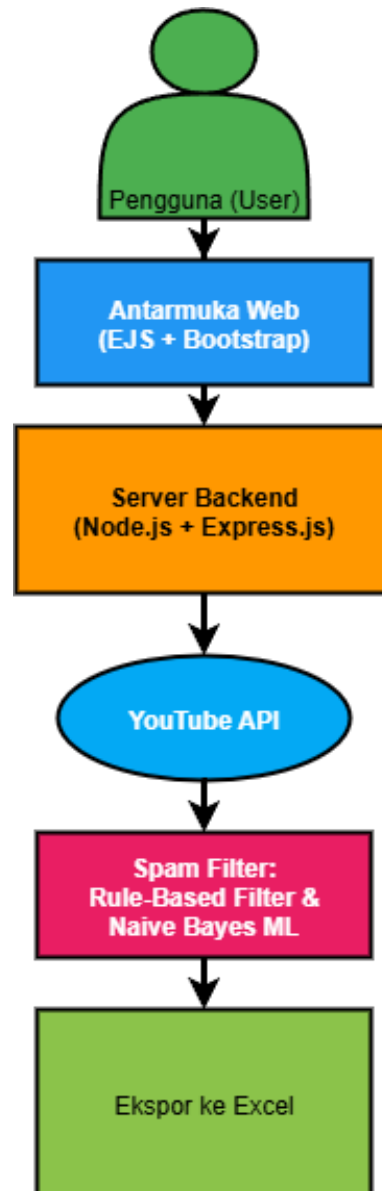
Lingkungan ini dirancang agar mendukung sistem secara optimal dan akan tersimpan di folder “node_modules”, terutama dalam menangani proses autentikasi terhadap API YouTube, klasifikasi teks secara real time, serta visualisasi hasil klasifikasi *spam* melalui antarmuka web.

4.1.3 Arsitektur Sistem

Arsitektur sistem ini dirancang dengan pendekatan konsep yang memisahkan antara backend, frontend, serta pemrosesan data dan klasifikasi spam.

A. Diagram Arsitektur Sistem

Berikut adalah ilustrasi arsitektur sistem secara umum:



Gambar 4.5 Diagram Arsitektur Sistem

Alur Umum Sistem :

1. Pengguna membuka halaman web dan melakukan autentikasi akun YouTube melalui OAuth 2.0.
2. Sistem mengambil komentar dari video YouTube yang dipilih.
3. Komentar dianalisis oleh dua metode: rule-based dan Naive Bayes.
4. Hasil analisis ditampilkan di frontend, disertai checkbox untuk pemilihan manual.

B. Komponen Utama Sistem

1. Frontend Web (User Interface)

Dibangun menggunakan EJS dan Bootstrap. Berfungsi untuk menampilkan komentar dari YouTube, memberikan tanda (checkbox) pada komentar yang dideteksi sebagai spam, dan memberikan kontrol bagi pengguna untuk mengeksport atau menyembunyikan komentar.

2. Backend Server (Node.js + Express)

Mengatur logika aplikasi, memproses komentar, menjalankan filter spam (rule-based dan Naive Bayes), serta menangani permintaan pengguna dari antarmuka web.

3. YouTube Data API v3

Digunakan untuk otorisasi pengguna (OAuth 2.0), pengambilan komentar video secara real-time, dan menyembunyikan komentar spam secara otomatis.

4. Spam Filter Engine

- a. *Rule-Based Filter*: Menyaring komentar menggunakan daftar kata terlarang dan karakter Unicode mencurigakan.
- b. *Naive Bayes Classifier*: Mengklasifikasikan komentar sebagai spam atau bukan spam berdasarkan model pembelajaran mesin.

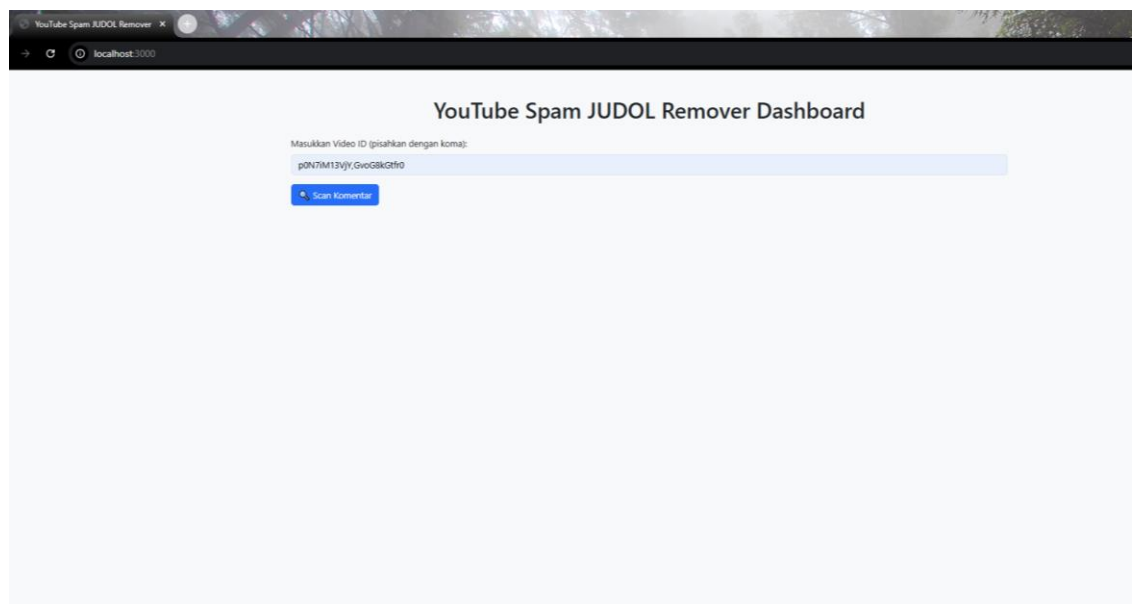
5. Ekspor Data ke Excel (ExcelJS)

Hasil deteksi spam dapat diekspor dalam format .xlsx untuk dokumentasi atau laporan. Pengguna dapat memilih komentar yang ingin disembunyikan dan mengklik tombol "Sembunyikan Komentar". Hasil deteksi dapat diekspor ke file Excel untuk dokumentasi atau evaluasi.

4.1.4 Implementasi Modul Frontend Web

Antarmuka pengguna (user interface) dari aplikasi YouTube Spam JUDOL Remover dirancang menggunakan teknologi HTML5, CSS3, dan framework Bootstrap 5 untuk mempercepat proses styling dan layout responsif. Templating engine yang digunakan adalah EJS (Embedded JavaScript Templates) agar data hasil pemindaian komentar dapat dirender secara dinamis dari backend ke tampilan web berjalan di <http://localhost:3000/>.

A. Tampilan Form Input

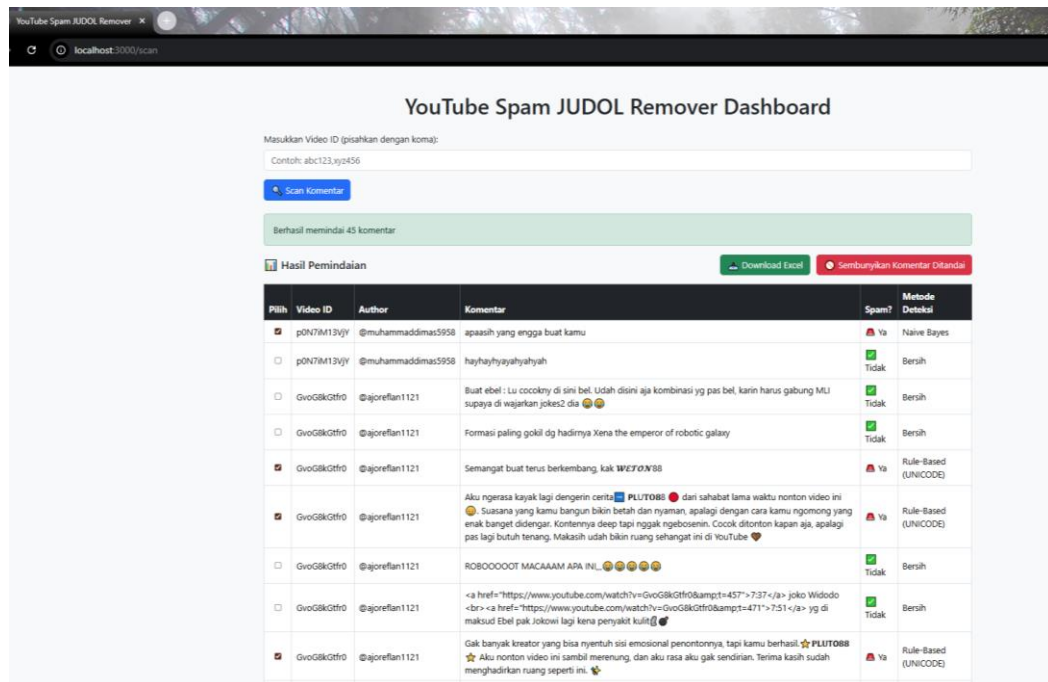


Gambar 4.6 Tampilan Form Input

Pengguna dapat memasukkan satu atau beberapa ID video YouTube (dipisah dengan “,”) yang ingin dipindai melalui form input di halaman utama.

B. Tampilan Hasil Pemindai Komentar

Setelah proses pemindaian selesai, hasilnya akan ditampilkan dalam bentuk tabel. Setiap komentar yang terdeteksi sebagai spam akan ditandai dengan checkbox secara otomatis tercentang.



Gambar 4.7 Tampilan Hasil Pemindai Komentar

C. Tombol Aksi Lanjutan

Tersedia dua tombol lanjutan dijelaskan sebagai berikut:

1. Download Exel

Tombol berwarna hijau ,berfungsi hanya untuk mendownload seluruh isi tabel hasil pemindaian tanpa menghilangkanan isi komentar vidio di YouTube, tersimpan dalam bentuk file Exel bernama “hasil_scan”

2. Sembunyikan Komentar Ditandai

Tombol berwarna merah ,berfungsi untuk menyembunyikan isi komentar-komentar yang di checklist pada Tabel kolom “Pilih”

3. Checkbox

Terdapat di dalam tabel pada kolom “Pilih”, berfungsi untuk filter terakhir secara manual dikarenakan mungkin ada beberapa kesalahan komentar yang di tandai tetapi bukan *spam*

4.1.5 Implementasi Modul Backend

Backend dimulai dengan mengimpor modul-modul yang diperlukan serta menginisialisasi server Express. Potongan kode berikut memperlihatkan bagian awal dari `index.js`:

```
require("dotenv").config();
const fs = require("fs");
const path = require("path");
const express = require("express");
const bodyParser = require("body-parser");
const { google } = require("googleapis");
const ExcelJS = require("exceljs");
const natural = require("natural");
const unidecode = require("unidecode");

const app = express();
const PORT = process.env.PORT || 3000;
const exportDir = path.join(__dirname, "exported");
if (!fs.existsSync(exportDir)) fs.mkdirSync(exportDir);

let lastScanComments = [];
let spamClassifier = null;
```

Gambar 4. 8 Inisialisasi Server dan Pustaka

File utama backend disimpan dalam `index.js`, yang berisi konfigurasi server, rute HTTP, dan pemanggilan fungsi dari pustaka-pustaka eksternal. Secara garis besar, alur backend mencakup:

1. Inisialisasi pustaka dan server
2. Konfigurasi autentikasi OAuth 2.0
3. Pemanggilan API YouTube untuk mengambil komentar
4. Proses filtering dengan dua pendekatan: rule-based dan Naive Bayes
5. Penyembunyian komentar yang terindikasi sebagai spam
6. Ekspor data hasil klasifikasi ke file Excel

4.1.6 Implementasi YouTube Data API v3

Sebelum dapat memanggil layanan YouTube API, pengguna harus melakukan proses autentikasi OAuth 2.0 agar sistem memperoleh akses yang sah dan aman terhadap akun YouTube pengguna. Proses ini diawali dengan

menghasilkan URL login, yang kemudian mengarahkan pengguna untuk memberi izin akses terhadap akun Google-nya.

```

async function authorize() {
  const credentials = JSON.parse(fs.readFileSync("credentials.json"));
  const { client_secret, client_id, redirect_uris } =
credentials.installed;
  const oAuth2Client = new google.auth.OAuth2(client_id, client_secret,
redirect_uris[0]);
  const TOKEN_PATH = "token.json";
  if (fs.existsSync(TOKEN_PATH)) {
    oAuth2Client.setCredentials(JSON.parse(fs.readFileSync(TOKEN_PATH))
);
  }
  return oAuth2Client; }
  throw new Error("✗ Token belum tersedia. Jalankan proses autentikasi
terlebih dahulu.");
}

```

Gambar 4.9 Proses Autentikasi OAuth 2.0

Setelah pengguna memberikan otorisasi, token akses akan digunakan untuk bisa menggunakan metode perintah pada YouTube Data API v3.

A. Pengambilan Komentar dengan *commentThreads.list*

Untuk mengambil komentar dari sebuah video, aplikasi menggunakan metode *commentThreads.list*. Metode ini memungkinkan sistem untuk membaca komentar utama dari video berdasarkan videoId yang diberikan pengguna.

```

async function getComments(auth, videoId) {
  const youtube = google.youtube({ version: "v3", auth });
  const comments = [];
  let nextPageToken = "";
  do {
    const res = await youtube.commentThreads.list({
      part: "snippet",
      videoId,
      maxResults: 100,
      pageToken: nextPageToken
    });
    res.data.items.forEach(item => {
      const snippet = item.snippet.topLevelComment.snippet;
      comments.push({
        videoId,
        author: snippet.authorDisplayName,
        text: snippet.textDisplay,
        commentId: item.snippet.topLevelComment.id
      });
    });
    nextPageToken = res.data.nextPageToken;
  } while (nextPageToken);
  return comments; }

```

Gambar 4.10 Pengambilan Komentar dengan *commentThreads.list*

B. Penyembunyian Komentar dengan *comments.setModerationStatus*

Jika sebuah komentar diklasifikasikan sebagai spam, sistem memberikan opsi kepada pengguna untuk menyembunyikan komentar secara otomatis. Penyembunyian dilakukan dengan metode *comments.setModerationStatus*, mengubah status komentar menjadi “rejected” (ditolak), sehingga komentar tidak lagi muncul di publik.

```
async function hideSpamComments(auth, spamIds) {
  const youtube = google.youtube({ version: "v3", auth });
  for (const id of spamIds) {
    try {
      await youtube.comments.setModerationStatus({ id,
moderationStatus: "rejected" });
      console.log(`✔ Disembunyikan: ${id}`);
    } catch (err) {
      console.error(`✖ Gagal sembunyikan ${id}:`, err.message);
    }
  }
}
```

Gambar 4.11 Penyembunyian Komentar dengan *comments.setModerationStatus*

Fungsi ini hanya dapat dijalankan jika token autentikasi yang dimiliki memiliki izin penuh terhadap video dan komentar yang bersangkutan. Oleh karena itu, sistem hanya bekerja pada video milik pengguna yang sedang login.

C. Pembatasan dan Biaya Kuota YouTube API

Tabel berikut menunjukkan beberapa biaya kuota dari API yang digunakan:

Tabel 4.2 Biaya Kuota YouTube API

Metode API	Biaya Kuota
<i>commentThreads.list</i>	1
<i>comments.setModerationStatus</i>	50
<i>videos.list</i> (untuk verifikasi)	1

Sistem dibatasi untuk memproses komentar dalam jumlah tertentu agar tidak melebihi batas kuota harian, Pengelolaan kuota ini penting untuk memastikan sistem tetap berjalan tanpa terganggu batasan harian dari Google.

4.1.7 Implementasi Rule-Based Filter

A. Deteksi Kata Terlarang (Blocked Words)

Jika sebuah komentar mengandung salah satu dari kata-kata yang sudah dimasukkan kedalam file “*blockedword.json*” maka komentar tersebut langsung diklasifikasikan sebagai spam, berikut contoh isi file:

```
{ } blockedword.json > ...
1  [
2      "gacor",
3      "maxwin",
4      "jp",
5      "SG88", "Nagamas", "🍷", "🍷",
6      "jepe",
7      "ALEXIS",
8      "O J O L 7 7"
9  ]
```

Gambar 4.12 Isi File *blockedword.json*

Kata-kata yang bisa di masukkan ke file tidak terbatas semua kata bisa dimasukkan otomatis menjadi *spam*.

B. Deteksi Simbol *Unicode* dan *Fancy Text*

Selain kata-kata eksplisit, komentar spam juga sering menggunakan simbol-simbol aneh, huruf *Unicode* yang dimodifikasi, atau fancy text. Tujuannya adalah untuk menghindari deteksi otomatis oleh sistem moderasi YouTube, atau

untuk menarik perhatian pembaca, sistem menggunakan proses normalisasi karakter, yaitu mengubah teks menjadi bentuk standarnya menggunakan metode *Unicode* normalization (NFKD) dan menghapus karakter non-standar:

```
function isSymbolSpam(text) {
  const normalized = text.normalize("NFKD");
  const simplified = unidecode(normalized);

  const emojiRegex = /\p{Emoji}/gu;
  const allowedSymbols = /[\\/',";:\}\{\[\+=_\-\]\(\*&^\%$#@!\~`>^<"]/g;

  const cleanedOriginal = normalized.replace(emojiRegex,
  "").replace(allowedSymbols, "");
  const cleanedSimplified = simplified.replace(emojiRegex,
  "").replace(allowedSymbols, "");

  return cleanedOriginal !== cleanedSimplified ||
  hasFancyUnicode(text);
}
```

Gambar 4.13 Deteksi *Unicode* dan Fancy Text

Untuk simbol-simbol yang biasa digunakan (`“\/,“;:\}\{\[\+=_\-\]\(\*&^\%$#@!\~`>^<”`) merupakan termasuk *Unicode symbol* tetapi dikecualikan dan dianggap teks tidak spam.

4.1.8 Implementasi Klasifikasi Komentar dengan Naive Bayes

Naive Bayes Classifier ini digunakan untuk mengenali pola spam berdasarkan data pelatihan yang telah ditandai sebelumnya sebagai “spam” dan “bukan” (ham) di file (dataset_emoji.json) dan dilatih oleh library *Natural*.

```
{
  "text": "Hah gmn 2 universee",
  "label": "bukan"
},
{
  "text": "Waktu istirahat mereka membicarakan ✨ALEXIS17✨",
  "label": "spam"
},
```

Gambar 4.14 Contoh Pelabelan Data Untuk Pelatihan

Setelah proses pelatihan, model disimpan dalam file eksternal (naïve_model.json) agar dapat digunakan kembali tanpa perlu melatih ulang setiap kali aplikasi dijalankan.

```
{ "_events": {}, "_eventsCount": 0, "classifier": { "classFeatures": { "bukan": { "0": 57, "1": 25, "2": 24, "3": 14, "4": 26, "5": 31, "6": 27, "7": 18, "8": 35, "9": 10, "10": 43, "11": 28, "12": 16, "13": 15, "14": 17, "15": 15, "16": 19, "17": 16, "18": 18, "19": 17, "20": 36, "21": 12, "22": 15, "23": 16, "24": 7, "25": 18, "26": 16, "27": 6, "28": 15, "29": 105, "30": 25, "31": 8, "32": 10, "33": 9, "34": 15, "35": 9, "36": 10, "37": 10, "38": 7, "39": 9, "40": 20, "41": 10, "42": 19, "43": 9, "44": 10, "45": 8, "46": 8, "47": 9, "48": 7, "49": 9, "50": 3, "51": 2, "52": 3, "53": 2, "54": 2, "55": 5, "57": 2, "59": 8, "60": 9, "61": 2, "62": 2,
```

Gambar 4.15 Isi data yang sudah dilatih *Naive Bayes*

Sebelum diklasifikasikan, komentar akan diproses terlebih dahulu melalui tahap preprocessing, termasuk:

1. Konversi huruf kapital menjadi huruf kecil
2. Penghapusan simbol tidak penting
3. Normalisasi *Unicode* (untuk menangani fancy text)
4. Penghapusan angka atau tanda baca berlebihan

4.1.9 Implementasi Ekspor Hasil Scan ke Excel

Implementasi ekspor dilakukan menggunakan pustaka *exceljs* yang berjalan di sisi backend (Node.js). Pustaka ini digunakan untuk membuat file Excel (.xlsx) secara dinamis, mengatur header kolom, mengisi data per komentar, dan menyimpan file di server atau menyediakannya untuk diunduh oleh pengguna.

A. Struktur data yang diekspor

Data yang dikumpulkan dari hasil proses klasifikasi mencakup:

1. Vidio ID: Asal vidio yang komentarnya diambil

2. Author: Akun penulis komentar di vidio
3. Comment: Isi dari komentar
4. Spam: Jika *spam* maka "YA" jika bukan *spam* maka "TIDAK"
5. Method: Metode yang digunakan seperti "Naïve bayes", "Unicode Symbol" dan "Hybrid" jika komentar tidak spam maka ditandai dengan "Clean"

Proses pembuatan file Excel diawali dengan membuat workbook dan worksheet menggunakan exceljs. Selanjutnya, kolom dan header ditentukan secara eksplisit:

```
async function exportToExcel(comments) {
  const workbook = new ExcelJS.Workbook();
  const sheet = workbook.addWorksheet("Spam Comments");

  sheet.columns = [
    { header: "Video ID", key: "videoId", width: 20 },
    { header: "Author", key: "author", width: 25 },
    { header: "Comment", key: "text", width: 50 },
    { header: "Spam", key: "spam", width: 10 },
    { header: "Method", key: "method", width: 20 }
  ];
}
```

Gambar 4.16 Pembuatan struktur file ekspor

B. Tujuan ekspor excel

Ekspor hasil scan ke Excel memberikan kemudahan bagi pengguna untuk:

1. Menyimpan catatan hasil moderasi komentar
2. Membuktikan adanya aktivitas spam secara legal/formal
3. Melakukan evaluasi terhadap performa sistem klasifikasi
4. Menganalisis kata-kata spam yang sering muncul

Fitur ini sangat relevan bagi pemilik channel YouTube atau peneliti keamanan siber yang membutuhkan dokumentasi terstruktur dari hasil filter komentar secara otomatis

4.2 Pengujian Sistem

Tabel berikut menyajikan beberapa skenario pengujian sistem beserta input, proses, dan output yang diharapkan:

No	Skenario Pengujian	Input	Proses	Output yang Diharapkan
1	Ambil komentar dari video YouTube	ID video YouTube valid	Mengakses YouTube Data API	Komentar berhasil diambil dan ditampilkan
2	Deteksi spam dengan blocked words	Komentar mengandung “slot”, “judi”	Rule-based filtering	Komentar ditandai sebagai spam
3	Deteksi fancy text / simbol aneh	Komentar mengandung teks <i>Unicode</i> unik	<i>Unicode</i> normalization	Komentar ditandai sebagai spam
4	Klasifikasi <i>Naive Bayes</i>	Komentar tidak mengandung kata spam	Model <i>Naive Bayes</i>	Komentar diklasifikasi “ham” atau “spam”
5	Hide komentar spam	Checklist komentar spam oleh pengguna	API <code>setModerationStatus</code>	Komentar disembunyikan di YouTube
6	Ekspor ke Excel	Hasil klasifikasi	Pembuatan file menggunakan ExcelJS	File Excel dengan data komentar tersedia untuk diunduh

4.2.1 Pengujian Ambil Komentar dari Vidio Youtube

Pengambil komentar berhasil mengambil seluruh komentar tanpa error dengan catatan tidak jika ada banyak vidio maka dipisahkan dengan “,”.

Berhasil memindai 45 komentar

Hasil Pemindaian Download Excel Sembunyikan Komentar Ditandai

Pilih	Video ID	Author	Komentar	Spam?	Metode Deteksi
☐	GvoG8kGtfr0	@ajoreflan1121	Buat ebel : Lu cocokny di sini bel. Udah disini aja kombinasi yg pas bel, karin harus gabung MLI supaya di wajarkan jokes2 dia 😂😂	☒ Tidak	Bersih
☐	GvoG8kGtfr0	@ajoreflan1121	Formasi paling gokil dg hadirnya Xena the emperor of robotic galaxy	☒ Tidak	Bersih
☒	GvoG8kGtfr0	@ajoreflan1121	Semangat buat terus berkembang, kak WETON88	☐ Ya	Rule-Based (UNICODE)
☒	GvoG8kGtfr0	@ajoreflan1121	Aku ngerasa kayak lagi dengerin cerita PLUTO88 dari sahabat lama waktu nonton video ini 😊. Suasana yang kamu bangun bikin betah dan nyaman, apalagi dengan cara kamu ngomong yang enak banget didengar. Kontennya deep tapi nggak ngebosenin. Cocok ditonton kapan aja, apalagi pas lagi butuh tenang. Makasih udah bikin ruang sehangat ini di YouTube ❤️	☐ Ya	Rule-Based (UNICODE)
☐	GvoG8kGtfr0	@ajoreflan1121	ROBOOOOOT MACAAAM APA INI,, 😂😂😂😂😂	☒ Tidak	Bersih
☐	GvoG8kGtfr0	@ajoreflan1121	7:37 joko Widodo 7:51 yg di maksud Ebel pak Jokowi lagi kena penyakit kulit🤢	☒ Tidak	Bersih
☒	GvoG8kGtfr0	@ajoreflan1121	Gak banyak kreator yang bisa nyentuh sisi emosional penontonnya, tapi kamu berhasil.🌟 PLUTO88 🌟 Aku nonton video ini sambil merenung, dan aku rasa aku gak sendirian. Terima kasih sudah menghadirkan ruang seperti ini. 🌟	☐ Ya	Rule-Based (UNICODE)
☒	GvoG8kGtfr0	@ajoreflan1121	❤️ MONA4D ❤️, kamu emang beda! Setiap sesi main bikin excited, apalagi pas dapat kemenangan besar. Ceritanya bikin relate, dan visualnya top banget. Sukses selalu! 🎮	☐ Ya	Rule-Based (UNICODE)
☒	GvoG8kGtfr0	@ajoreflan1121	Aku nonton tanpa skip karena emang sebgus itu!👏 PLUTO88 Mulai dari visual, suara, sampai pesan yang disampaikan semuanya pas banget. Makasih udah bikin hari ini lebih bermakna!	☐ Ya	Rule-Based (UNICODE)

Gambar 4.17 Hasil Pengujian Ambil Komentar dari Vidio Youtube

4.2.2 Pengujian Deteksi Spam dengan Blocked Words

Modul deteksi berbasis *blocked words* terbukti mampu menandai seluruh komentar yang mengandung kata atau frasa yang telah didefinisikan sebagai indikator spam. Seluruh komentar uji yang mengandung salah satu dari kata kunci yang dimasukkan ke dalam daftar terblokir berhasil diklasifikasikan sebagai spam tanpa ada kesalahan deteksi (false negative), sehingga mencapai akurasi 100%.

4.2.3 Pengujian Deteksi fancy text/symbol aneh

Komentar dengan karakter *Unicode* atau *fancy text*, seperti “**slot**” dan “**CUAN**”, berhasil dideteksi melalui proses normalisasi karakter yang mengubah simbol menjadi bentuk teks standar. Setelah dinormalisasi, komentar tersebut dikenali sebagai spam oleh sistem rule-based, menunjukkan bahwa mekanisme deteksi simbol aneh berjalan dengan baik

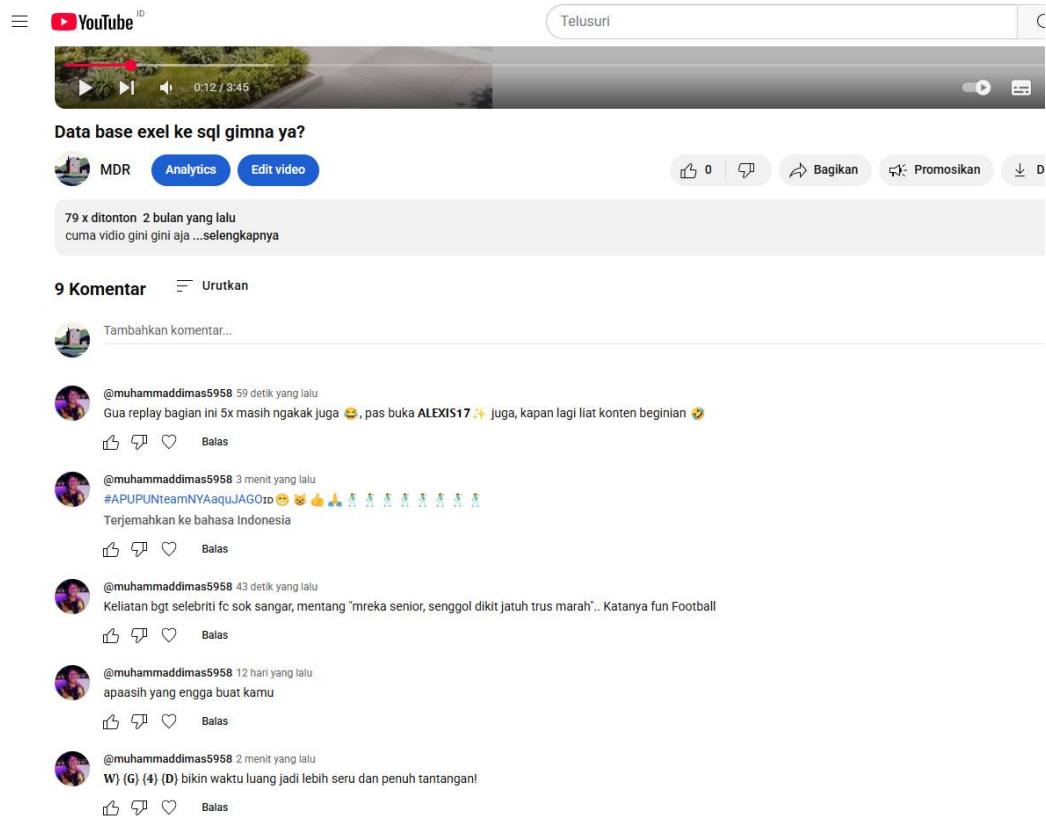
4.2.4 Pengujian Klasifikasi Naïve Bayes

Model *Naive Bayes* dilatih menggunakan dataset komentar yang telah diberi label, dan diuji menggunakan metode validasi silang sederhana. Hasil pengujian memperlihatkan bahwa model memiliki kemampuan yang cukup baik dalam mengidentifikasi pola linguistik spam, dengan akurasi rata-rata sebesar 85%.

4.2.5 Pengujian Hide Komentar spam

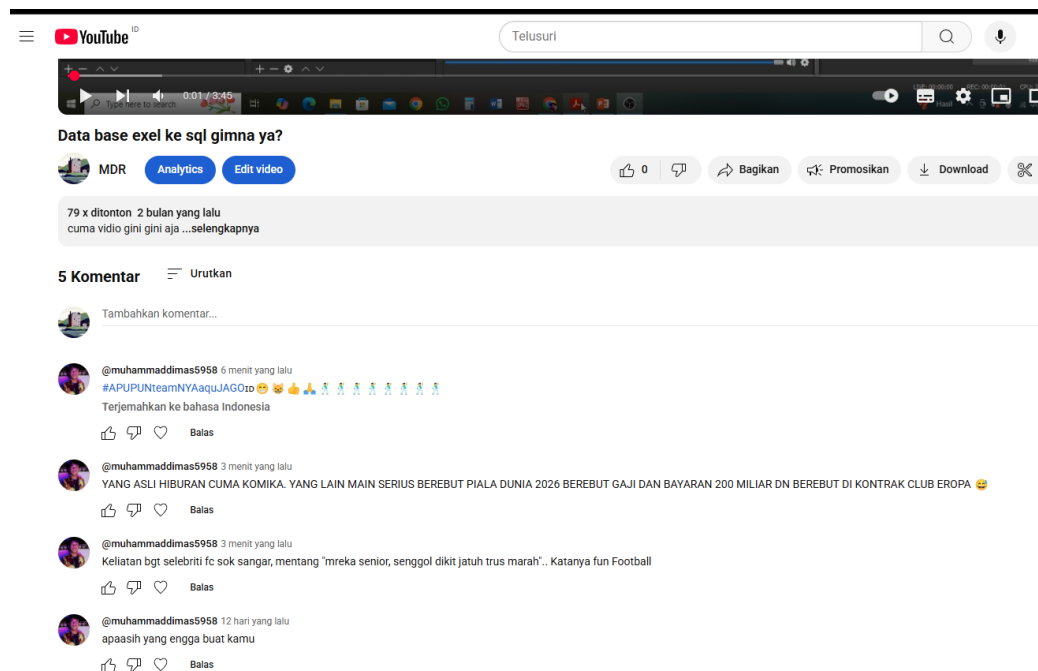
Pengujian terhadap fitur penyembunyian komentar menunjukkan bahwa penggunaan metode *setModerationStatus* dari YouTube Data API v3 berhasil menandai komentar yang terdeteksi sebagai spam dengan status 'rejected'. Hal ini menyebabkan komentar tersebut tidak lagi ditampilkan secara publik di platform YouTube, sesuai dengan tujuan fungsionalitas sistem.

Gambar sebelum komentar disembunyikan:



Gambar 4.18 Sebelum komentar spam disembunyikan

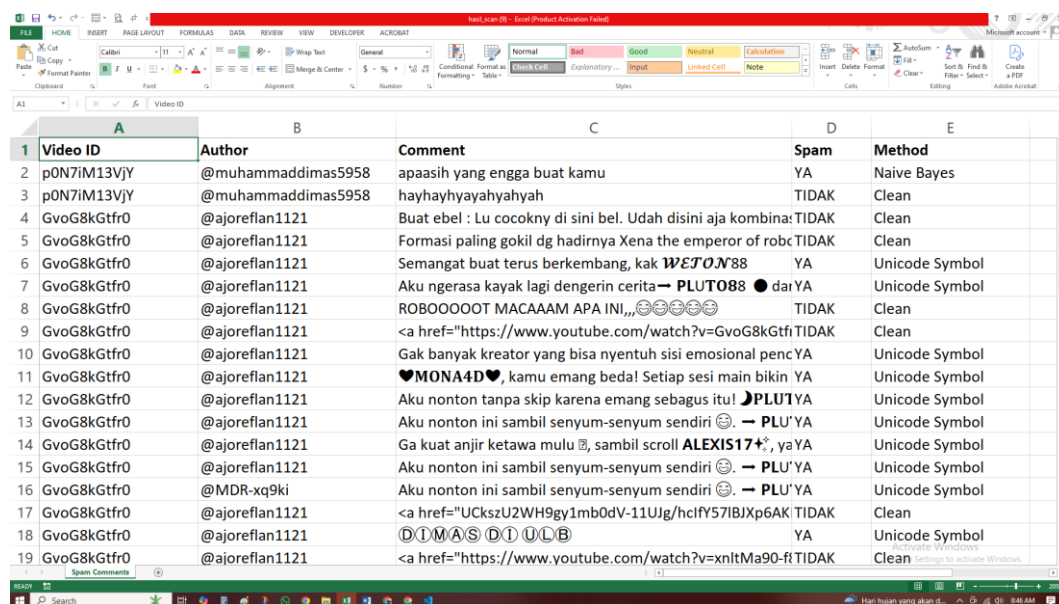
Gambar sesudah komentar disembunyikan:



Gambar 4.19 Sesudah komentar spam disembunyikan

4.2.6 Pengujian Ekpor Hasil Ke Excel

File Excel dibuat secara otomatis menggunakan pustaka ExcelJS dan berhasil menyimpan data hasil deteksi komentar secara terstruktur. Pengguna dapat mengakses dan mengunduh file ini untuk keperluan dokumentasi lebih lanjut atau evaluasi manual



Video ID	Author	Comment	Spam	Method
p0N7iM13VjY	@muhammaddimas5958	apaasih yang engga buat kamu	YA	Naive Bayes
p0N7iM13VjY	@muhammaddimas5958	hayhayhyayhyahyah	TIDAK	Clean
GvoG8kGtfr0	@ajoreflan1121	Buat ebel : Lu cocokny di sini bel. Udah disini aja kombina:	TIDAK	Clean
GvoG8kGtfr0	@ajoreflan1121	Formasi paling goki dg hadirnya Xena the emperor of robc	TIDAK	Clean
GvoG8kGtfr0	@ajoreflan1121	Semangat buat terus berkembang, kak WETON 88	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121	Aku ngerasa kayak lagi dengerin cerita PLUTO88 ● dai	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121	ROBOOOOOT MACAAAM APA INI,,,🤖🤖🤖🤖	TIDAK	Clean
GvoG8kGtfr0	@ajoreflan1121		TIDAK	Clean
GvoG8kGtfr0	@ajoreflan1121	Gak banyak kreator yang bisa nyentuh sisi emosional penc	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121	♥MONA4D♥, kamu emang beda! Setiap sesi main bikin	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121	Aku nonton tanpa skip karena emang sebgas itu! PLU 1YA	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121	Aku nonton ini sambil senyum-senyum sendiri 😊. PLU YA	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121	Ga kuat anjir ketawa mulu 🤔, sambil scroll ALEXIS17 *, ya	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121	Aku nonton ini sambil senyum-senyum sendiri 😊. PLU YA	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121	Aku nonton ini sambil senyum-senyum sendiri 😊. PLU YA	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121		TIDAK	Clean
GvoG8kGtfr0	@ajoreflan1121	DIMAS D I U L B	YA	Unicode Symbol
GvoG8kGtfr0	@ajoreflan1121		TIDAK	Clean

Gambar 4.20 Hasil ekspor ke excel

4.3 Pembahasan Hasil Pengujian

Pembahasan berikut menguraikan efektivitas masing-masing komponen sistem, serta menyoroti kelebihan, kekurangan, dan implikasi dari pendekatan yang digunakan.

4.3.1 Efektivitas Rule-Based Filter

Pengujian terhadap modul rule-based filter, yang mencakup deteksi kata terlarang dan normalisasi simbol *Unicode* (fancy text), menunjukkan hasil yang sangat baik. Komentar yang mengandung kata kunci promosi judi seperti “slot”,

“gacor”, “cuan”, dan variasi lainnya berhasil dideteksi secara akurat dengan tingkat keberhasilan 100%, selama kata kunci tersebut tercantum dalam daftar blokir (blocked words).

Sistem juga mampu mengidentifikasi komentar yang menggunakan simbol *Unicode* atau bentuk tulisan menyerupai teks asli (fancy text), seperti “**slot**” atau “**CUAN**”. Normalisasi dilakukan dengan pendekatan *Unicode* NFKD atau menggunakan pustaka *unidecode*, sehingga komentar tersebut dapat dikenali sebagai bentuk spam meskipun menggunakan karakter manipulatif. Hal ini menunjukkan bahwa rule-based filter dapat mengantisipasi taktik kamuflase teks yang umum digunakan dalam komentar spam.

4.3.2 Akurasi Klasifikasi Naïve Bayes

Model klasifikasi *Naïve Bayes* yang diterapkan menunjukkan kemampuan klasifikasi yang cukup baik dalam membedakan antara komentar spam dan non-spam. Berdasarkan uji terbatas terhadap data berlabel, model mencapai akurasi rata-rata sebesar 85%. Komentar bernada positif dan netral seperti “makasih infonya”, “videonya keren”, atau “izin save ya bang” diklasifikasikan sebagai non-spam, sementara komentar promosi seperti “link slot gacor” atau “join grup cuan 18+” berhasil terdeteksi sebagai spam.

Namun demikian, masih ditemukan sejumlah kasus *false positive*, di mana komentar sah diklasifikasikan sebagai spam, terutama jika mengandung kata ambigu yang juga digunakan dalam komentar promosi. Oleh karena itu, peran rule-based filter sebagai lapisan awal sangat penting untuk mengurangi beban pada klasifikasi berbasis machine learning.

4.3.3 Keunggulan Pendekatan Hybrid

Dengan adanya dua lapisan deteksi, sistem mampu menjangkau lebih banyak bentuk spam yang tidak dapat ditangani oleh salah satu metode saja. Hal ini membuktikan bahwa pendekatan hybrid memberikan hasil yang lebih andal dibandingkan pendekatan tunggal.

4.3.4 Efektivitas Fitur Penyembunyian Komentar

Fitur penyembunyian komentar yang menggunakan metode *setModerationStatus* dari YouTube Data API berhasil menyembunyikan komentar yang ditandai sebagai spam. Proses ini dilakukan secara otomatis setelah pengguna melakukan konfirmasi melalui antarmuka web. Tidak ditemukan kendala atau error dalam pemanggilan API tersebut, dan komentar yang diberi status “rejected” tidak lagi ditampilkan secara publik pada video terkait.

4.3.5 Evaluasi Fitur Ekspor ke Excel

Fitur ekspor hasil scan ke file Excel juga berjalan dengan baik. File yang dihasilkan berisi data yang lengkap dalam format terstruktur, dengan isi kolom “Asal Vidio”, “Penulis Komen”, “Isi Komentar” , “Status”, dan “Metode Digunakan”. Hal ini mempermudah pengguna dalam melakukan pelacakan, dokumentasi, dan pelaporan hasil pendeteksian spam. Ekspor ini sangat berguna sebagai bahan audit, terutama jika pengguna ingin memverifikasi hasil klasifikasi.

4.4 Simpulan Implementasi dan Pengujian Sistem

Metode rule-based filter berhasil mendeteksi komentar spam secara pasti berdasarkan kata kunci terlarang dan simbol *Unicode* yang dimanipulasi. Di sisi

lain, metode *Naive Bayes* mampu menangani klasifikasi terhadap komentar yang bersifat lebih kontekstual. Penggabungan kedua metode ini terbukti meningkatkan akurasi deteksi spam dan meminimalisasi kesalahan klasifikasi.

Fitur penyembunyian komentar menggunakan API YouTube juga berhasil dijalankan, memungkinkan pengguna menyembunyikan komentar spam secara selektif melalui antarmuka web. Selain itu, fitur ekspor ke Excel memberikan kemudahan dalam dokumentasi dan evaluasi hasil pendeteksian.

Secara keseluruhan, sistem yang telah dibangun menunjukkan kinerja yang stabil, akurat, dan relevan dalam konteks deteksi dan penanganan komentar spam promosi judi online di platform YouTube. Dengan demikian, aplikasi ini layak digunakan sebagai alat bantu moderasi komentar oleh pengguna YouTube, khususnya untuk menjaga kualitas ruang interaksi digital dari konten spam yang mengganggu