Deteksi Otomatis Komentar Kasar dan Spam dengan *Regular Expression* dan *Fuzzy Matching*

Joel Hotlan Haris Siahaan - 13523025
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13523025@std.stei.itb.ac.id

***Abstract*—Seiring meningkatnya aktivitas pengguna di platform daring, komentar kasar dan pesan spam menjadi tantangan utama dalam menjaga kualitas interaksi. Untuk mengatasi hal tersebut, makalah ini mengusulkan kerangka deteksi otomatis berbasis algoritma yang menggabungkan *Regular Expressions (regex)* dan *fuzzy matching* menggunakan *Levenshtein distance*. *Regex* digunakan untuk mengenali pola eksplisit dari kata-kata kasar dan spam, sementara *fuzzy matching* diterapkan untuk mendeteksi variasi penulisan yang dimodifikasi agar lolos dari penyaringan konvensional. Penelitian ini mengimplementasikan sistem deteksi pada sejumlah komentar uji, dan menunjukan bahwa pendekatan kombinasi ini mampu memetakan pola komentar bermasalah secara efektif. Hasilnya menunjukan bahwa *regex* unggul dalam deteksi langsung dan cepat, sedangkan *fuzzy matching* berperan penting dalam mengidentifikasi komentar dengan manipulasi ejaan. Makalah ini menyimpulkan bahwa integrasi keduanya penting untuk menghasilkan sistem moderasi konten yang adaptif dan dapat diandalkan.**

Kata kunci—fuzzy matching; regex; Levenshtein distance;deteksi komentar; spam; moderasi konten otomatis

I. PENDAHULUAN

Dalam era digital saat ini, platform daring seperti media sosial dan forum diskusi telah menjadi sarana utama bagi Masyarakat untuk berinteraksi dan berbagi pendapat. Namun, akses terbuka ini mendatangkan tantangan serius seperti maraknya komentar kasar, ujaran kebencian, serta pesan-pesan spam yang dapat menggangu kenyamanan pengguna lain. Konten semacam ini berpotensi untuk memicu konflik, menyebarkan disinformasi, dan merusak reputasi platform.

Upaya moderasi konten secara manual seringkali kurang efektif, terutama ketika volume komentar sangat besar dan penyebarannya yang sangat tinggi. Oleh karena itu, dibutuhkan solusi optimal yang mampu mendeteksi dan menyaring komentar bermasalah secara efisien dan akurat.

Makalah ini akan membahas sistem deteksi otomatis komentar kasar dan spam dengan pendekatan gabungan menggunakan *Regular Expressions (regex)* dan *fuzzy matching* berbasis *Levenshtein Distance*. Regex digunakan untuk mengenali pola kata kasar atau frasa spam yang eksplisit, sementara *fuzzy matching* digunakan untuk mengidentifikasi bentuk-bentuk komentar yang sudah dimodifikasi atau diplesetkan untuk menghindari deteksi langsung.

Dengan menggabungkan kedua pendekatan ini, diharapkan sistem mampu menangani berbagai variasi penulisan komentar negative secara lebih fleksibel dan cerdas.

II. DASAR TEORI

A. Komentar Kasar dan Spam

Komentar kasar adalah bentuk ujaran yang mengandung penghinaan, pelecehan, atau kata-kata tidak pantas dalam komunikasi daring. Konten semacam ini berdampak negative terhadap pengalaman pengguna lain dalam berekspresi, menciptakan suasana tidak nyaman dan dapat memicu konflik antar pengguna bahkan kelompok tertentu. Komentar kasar dapat muncul dalam bentuk langsung seperti makian “bodoh”, “anjing”, maupun dalam bentuk implisit yang tetap menyampaikan makna agresif secara tidak langsung.

Salah satu bentuk komentar yang sangat sensitif adalah komentar yang mengandung unsur SARA (Suku, Agama, Ras, dan Antargolongan). Komentar SARA bersifat diskriminatif dan dapat memicu konflik serius. Dalam moderasi konten, komentar SARA sering kali dikategorika sebagai ujaran kebencian karena menyasar identitas seseorang atau kelompok tertentu berdasarkan latar belakang sosial atau budaya mereka.

Selain itu, spam juga menjadi gangguan umum dalam ruang komunikasi daring secara luas. Spam adalah pesan atau informasi yang dikirim secara massal dan tidak diminta melalui media elektronik seperti email, SMS, atau komentar di media sosial dan forum diskusi. Spam seringkali berisi konten promosi, penipuan, atau materi yang tidak diinginkan. Dalam konteks media sosial atau forum, spam sering muncul dalam bentuk komentar berulang berisi ajakan untuk mengunjungi tautan tertentu atau teks yang tidak relevan dengan topik pembahasan.

B. Moderasi Konten Otomatis

Moderasi konten otomatis adalah proses penyaringan komentar yang tidak layak secara sistematis menggunakan teknologi, tanpa campur tangan manusia secara langsung. Sistem ini sangat penting untuk untuk platform daring

berskala besar seperti media sosial, forum diskusi, serta aplikasi pesan.

Moderasi konten dapat dilakukan dengan dua metode dengan pendekatan utama berbasis aturan (rule-based) dan berbasis pembelajaran mesin (machine learning). Pendekatan berbasis aturan menggunakan daftar aturan atau pola tetap, seperti daftar kata terlarang, *regular expressions (regex)* , atau algoritma lain untuk mendeteksi konten bermasalah. Metode ini cepat dan mudah diimplementasikan untuk mendeteksi pola-pola yang jelas, seperti spam promosi atau kata-kata kasar eksplisit. Namun, pendekatan ini terbatas dalam mendeteksi konten yang telah diplesetkan atau ujaran yang kontekstual.

Pendekatan berbasis pembelajaran mesing memanfaatkan algoritma statistic dan model pembelajaran yang dilatih dengan data tertenu untuk mengklasifikasikan suatu komentar tergolong bermaslaah atau tidak. Model ini mampu mengenali pola yang lebih kompleks dan kontekstual, serta belajar dari data seiring waktu. Namun, machine learning memiliki kekurangan karena kebutuhan data pelatihan yang besar, memerlukan sumber daya komputasi yang tinggi. serta memiliki potensi bias yang bergantung pada data yang dipakai

Dalam Makala ini, digunakan pendekatan berbasis aturan karena lebih sederhana dan efisien untuk scenario ringan. Gabungan antara *regular expressions* dan *fuzzy matching* dapat meningkatkan akurasi deteksi dengan tetap mempertahankan peforma yang baik.

C. Regular Expressions (regex)

Regular Expressions (regex) adalah metode pencocokan pola dalam teks yang digunakan untuk menemukan, mengidentifikasi, atau mengganti *string* berdasarkan struktur tertentu. Regex dapat digunakan dalam deteksi kata-kata kasar atau spam dalam sistem moderasi komentar.

Dalam konteks moderasi konten, regex memungkinkan sistem untuk mengenali kata atau frasa bermasalah secara cepat dan efisien. Beberapa pola dasar dalam regex yang umum digunakan antara lain :

- Kurung sikut [ ] : menyatakan disjungsi, yaitu mencocokan salah satu karakter dalam daftar.

RE	Match	Example Patterns
`/[wW]oodchuck/`	Woodchuck or woodchuck	"Woodchuck"
`/[abc]/`	'a', 'b', *or* 'c'	"In uomini, in soldati"
`/[1234567890]/`	any digit	"plenty of 7 to 5"

- Rentang karakter [a – z], [A – Z], [0 – 9]: mencocokan semua huruf kecil, huruf kapital, atau digit dalam rentang yang diberikan.

RE	Match	Example Patterns Matched
`/[A-Z]/`	an uppercase letter	"we should call it 'Drenched Blossoms'"
`/[a-z]/`	a lowercase letter	"my beans were impatient to be hoed!"
`/[0-9]/`	a single digit	"Chapter 1: Down the Rabbit Hole"

- Simbol ^ dalam [ ] : digunakan untuk menyatakan negasi, yaitu mencocokkan karakter selain yang ditentukan.

RE	Match (single characters)	Example Patterns Matched
`[^A-Z]`	not an uppercase letter	"Oyfn pripetchik"
`[^Ss]`	neither 'S' nor 's'	"I have no exquisite reason for't"
`[^\.]`	not a period	"our resident Djinn"
`[e^]`	either 'e' or '^'	"look up ^ now"
`a^b`	the pattern 'a^b'	"look up a^b now"

- Tanda tanya ?: menyatakan bahwa karakter sebelumnya bersifat opsional, artinya bisa ada atau tidak ada.

RE	Match	Example Patterns Matched
`woodchucks?`	woodchuck or woodchucks	"woodchuck"
`colou?r`	color or colour	"colour"

- Titik . : mencocokkan satu karakter apapun (kecuali newline), sehingga bisa digunakan untuk menangkap variasi penulisan.

RE	Match	Example Patterns
`/beg.n/`	any character between *beg* and *n*	begin, beg'n, begun

Selain itu, regex mendukung kombinasi lebih lanjut seperti :

- Tanda kurung ( ): digunakan untuk mengelompokkan pola yang ingin diperlakukan sebagai satu unit.
- Operator percabangan | : digunakan sebagai logika "atau" yang mencocokkan salah satu dari beberapa pola alternatif.
- Simbol pengulangan *, +, dan {n,m}: symbol ini menyatakan berapa kali sebuah karakter atau grup boleh muncul.

Construct	Arti
`X?`	X muncul **satu** atau tidak sama sekali
`X*`	X muncul **nol** atau banyak
`X+`	X muncul **satu** atau banyak
`x{n}`	X muncul tepat n kali
`x{n,}`	X muncul setidaknya n kali
`x{n,m}`	X muncul antara n sampai m kali

D. Fuzzy Matching

Fuzzy matching adalah teknik pencocokan *string* yang memperbolehkan kesalahan ketik, variasi karakter, atau bentuk ejaan alternatif. Ini sangat berguna dalam mendeteksi komentar yang sengaja diplesetkan untuk menghindari penyaringan secara exact match.

Salah satu algoritma utama yang digunakan dalam *fuzzy matching* adalah *Levenshtein Distance*, yang didefinisikan sebagai jumlah minimum operasi edit(penyisipan, penghapusan, atau penggantian katakter) yang diperlukan untuk mengubah satu *string* menjadi *string* lainnya. Misal, jarak antara kata kasar "bodoh" dan "bod0h" adalah 1, karena hanya memerlukan satu subtitusi karakter untuk membuat kedua kata menjadi sama persis.

Dalam bahasa pemrograman python, *fuzzy matching* biasanya digunakan bersama dengan fungsi pembobotan seperti partial_ratio dan token_sort_ratio yang ada di pustaka RapidFuzz. Pendekatan ini tidak hanya menghitung jarak karakter, tetapi juga mempertimbangkan Tingkat kemiripan antar *string* secara lebih kontekstual

III. IMPLEMENTASI

A. *Lingkungan Implementasi*

Sistem deteksi komentar ini diiplementasikan menggunakan bahasa pemrograman Python karena beberapa alasan. Pertama, Python memiliki sintaks yang sederhana dan miudah dibaca, sehingga cocok untuk pengembangan sistem berbasis teks dan logika pencocokan pola. Kedua, Python memiliki ekosistem Pustaka yang sangat kaya untuk pemrosesas *string*, pencocokan teks, dan analisis data, yang sangat mendukung kebutuhan dari sistem moderasi komentar otomatis. Sleain itu, Python juga bersifat lintas platform dan mudah diintegrasikan dengan sistem backend maupun antarmuka aplikasi.

Untuk mendukung implementasi sistem, digunakan beberapa Pustaka Python seperti Pustaka re, rapidfuzz, dan typing. Pustaka re digunakan untuk melakukan pencocokan pola teks berbasis *regular expressions*. Dalam program ini, re digunakan untuk normalisasi teks dan mendeteksi pola mencurigakan seperti tautan, nomor telepon, dan karakter berulang. Pustaka rapidfuzz adalah Pustaka yang berfungsi untuk melakukan *fuzzy matching* dengan algoritma *Levenshtein Distance* dan fungsi pembobotan partial_ratio. Dalam program ini, rapidfuzz digunakan untuk mendeteksi kata kasar dan spam yang ditulis dengan plesetan. Modul typing digunakan untuk memberikan anotasi tipe data yang membantu dokumentasi internal kode dan meningkatkan keterbacaan serta ketepatan selama pengembangan.

B. *Struktur Program*

Program utama dibungkus dalam sebuah kelas Bernama AdvancedfCommentDetector. Kelas ini memiliki fungsi-fungsi utama untuk melakukan deteksi kata kasar dan spam berdasarkan data yang disiapkan :

- Kumpulan kata kasar dalam Bahasa Indonesia dan Inggris
- Kata-kata spam umum seperti promosi, hadiah, atau perintah klik tautan tertentu
- Kata-kata terkait judi *online* seperti “slot *online*” dan “poker”
- Pola mencurigakan seperti nomor telepon, link situs, karakter berulang, dan pesan ajakan

Berikut adalah kode python lengkap dari program deteksi otomatis komentar kasar dan spam:

```python
import re
from rapidfuzz import fuzz
from typing import Dict, List, Tuple

class AdvancedCommentDetector:
    def __init__(self):
        self.profanity_id = [
            "anjing", "bangsat", "goblok", "tolol", "bodoh", "bego", "dungu",
            "idiot", "stupid", "kampret", "bajingan", "brengsek", "sialan",
            "keparat", "tai", "shit", "damn", "fuck", "bitch", "asshole",
            "kontol", "memek", "ngentot", "jancok", "kimak", "pantek",
            "peler", "itil", "perek", "pelacur", "lonte", "jablay",
            "bangke", "asu", "monyet", "babi", "anjrit", "anying"
        ]

        self.profanity_en = [
            "fuck", "shit", "damn", "bitch", "asshole", "bastard", "crap",
            "hell", "piss", "cock", "dick", "pussy", "whore", "slut",
            "motherfucker", "dickhead", "shithead", "dumbass", "retard",
            "faggot", "nigger", "cunt", "twat", "prick", "wanker"
        ]

        self.spam_general = [
            "klik disini", "click here", "gratis", "free money", "dapat uang",
            "transfer gratis", "bonus", "hadiah", "prize", "winner",
            "congratulations", "selamat", "menang", "jackpot", "promo",
            "diskon", "cashback", "rebate", "komisi", "affiliate"
        ]

        self.gambling_spam = [
            "slot online", "poker online", "casino online", "judi online",
            "taruhan", "betting", "sportsbook", "togel", "lottery",
            "roulette", "blackjack", "baccarat", "domino", "capsa",
            "bandar", "agen", "daftar sekarang", "register now",
            "deposit", "withdraw", "minimal bet", "maxwin", "gacor",
            "bocoran", "prediksi", "angka jitu", "rumus", "trik menang",
            "situs judi", "gambling site", "bet now", "live casino",
            "scatter", "wild", "freespin", "bonus deposit", "welcome bonus"
        ]

        self.regex_patterns = [
            r"\b[a-z]*[0-9@#$%^&*]+[a-z]*\b",
            r"\b\w*([a-z])\1{2,}\w*\b",
            r"\b\w+(\s+\w+)*\b",
            r"(http[s]?://|www\.)[^\s]+",
            r"\b[a-z]+\.(com|net|org|xyz|click|tk)\b",
            r"(\+62|0)[0-9\-\s]{8,15}",
            r"\b(wa|whatsapp|telegram|line)\s*[:=]?\s*[0-9\-\s]+",
            r"\b(pin|bbm)\s*[:=]?\s*[a-f0-9]{8}\b",
            r"\bdm\s+(me|saya|gue)\b"
        ]

        self.char_replacements = {
            '4': 'a', '3': 'e', '1': 'i', '0': 'o', '5': 's',
            '@': 'a', '$': 's', '!': 'i', '7': 't', '8': 'b',
            'x': 'ks', 'z': 's', 'c': 'k', 'q': 'k',
            'ph': 'f', 'ck': 'k'
        }

    def normalize_text(self, text: str) -> str:
        text = text.lower().strip()
        text = re.sub(r'[^\w\s]', ' ', text)

        for old, new in self.char_replacements.items():
            text = text.replace(old, new)

        text = re.sub(r'(.)\1{2,}', r'\1\1', text)
        text = re.sub(r'\s+', ' ', text)

        return text

    def check_regex_patterns(self, text: str) -> List[str]:
        matches = []
        for pattern in self.regex_patterns:
            if re.search(pattern, text, re.IGNORECASE):
                matches.append(pattern)
        return matches
```

```python
def fuzzy_match_words(self, text: str, word_list: List[str], threshold: int = 75) -> List[Tuple[str, int]]:
    matches = []
    normalized_text = self.normalize_text(text)

    for word in word_list:
        if word in normalized_text:
            matches.append((word, 100))
            continue

        score = fuzz.partial_ratio(word, normalized_text)
        if score >= threshold:
            matches.append((word, score))

    return matches

def check_spam_patterns(self, text: str) -> Dict[str, List]:
    results = {
        "url_links": [],
        "phone_numbers": [],
        "suspicious_contacts": [],
        "repeated_chars": []
    }

    url_pattern = r"(http[s]?://[^\s]+|www\.[^\s]+|[a-z]+\.(com|net|org|xyz|click|tk|me))"
    urls = re.findall(url_pattern, text, re.IGNORECASE)
    if urls:
        results["url_links"] = [url[0] if isinstance(url, tuple) else url for url in urls]

    phone_pattern = r"(\+62|0)[0-9\-\s]{8,15}"
    phones = re.findall(phone_pattern, text)
    if phones:
        results["phone_numbers"] = phones

    contact_pattern = r"\b(wa|whatsapp|telegram|line|pin|bbm)\s*[:=]?\s*[a-f0-9\-\s]+"
    contacts = re.findall(contact_pattern, text, re.IGNORECASE)
    if contacts:
        results["suspicious_contacts"] = contacts

    repeat_pattern = r"\b\w*([a-z])\1{3,}\w*\b"
    repeats = re.findall(repeat_pattern, text, re.IGNORECASE)
    if repeats:
        results["repeated_chars"] = repeats

    return results

def calculate_spam_score(self, results: Dict) -> int:
    score = 0

    if results["profanity_id"]:
        score += len(results["profanity_id"]) * 25
    if results["profanity_en"]:
        score += len(results["profanity_en"]) * 25

    if results["spam_general"]:
        score += len(results["spam_general"]) * 20
    if results["gambling_spam"]:
        score += len(results["gambling_spam"]) * 30

    if results["regex_patterns"]:
        score += len(results["regex_patterns"]) * 10

    spam_patterns = results["spam_patterns"]
    if spam_patterns["url_links"]:
        score += len(spam_patterns["url_links"]) * 15
    if spam_patterns["phone_numbers"]:
        score += len(spam_patterns["phone_numbers"]) * 20
    if spam_patterns["suspicious_contacts"]:
        score += len(spam_patterns["suspicious_contacts"]) * 25
    if spam_patterns["repeated_chars"]:
        score += len(spam_patterns["repeated_chars"]) * 5

    return min(score, 100)
```

```python
def detect_comment(self, text: str, profanity_threshold: int = 75, spam_threshold: int = 70) -> Dict:
    results = {
        "original_text": text,
        "normalized_text": self.normalize_text(text),
        "profanity_id": [],
        "profanity_en": [],
        "spam_general": [],
        "gambling_spam": [],
        "regex_patterns": [],
        "spam_patterns": {},
        "spam_score": 0,
        "is_spam": False,
        "is_profanity": False,
        "risk_level": "LOW"
    }

    results["profanity_id"] = self.fuzzy_match_words(text, self.profanity_id, profanity_threshold)
    results["profanity_en"] = self.fuzzy_match_words(text, self.profanity_en, profanity_threshold)

    results["spam_general"] = self.fuzzy_match_words(text, self.spam_general, spam_threshold)
    results["gambling_spam"] = self.fuzzy_match_words(text, self.gambling_spam, spam_threshold)

    results["regex_patterns"] = self.check_regex_patterns(text)

    results["spam_patterns"] = self.check_spam_patterns(text)

    results["spam_score"] = self.calculate_spam_score(results)

    results["is_profanity"] = bool(results["profanity_id"] or results["profanity_en"])
    results["is_spam"] = results["spam_score"] >= 30

    if results["spam_score"] >= 70:
        results["risk_level"] = "HIGH"
    elif results["spam_score"] >= 40:
        results["risk_level"] = "MEDIUM"
    else:
        results["risk_level"] = "LOW"

    return results
```

```python
def print_detection_result(self, result: Dict):
    print(f"\nText: \"{result['original_text']}\"")
    print(f"Spam Score: {result['spam_score']}/100")
    print(f"Risk Level: {result['risk_level']}")

    if result['is_profanity']:
        print("PROFANITY DETECTED!")
        if result['profanity_id']:
            print(f"   - Indonesian: {[f'{word}({score})' for word, score in result['profanity_id']]}")
        if result['profanity_en']:
            print(f"   - English: {[f'{word}({score})' for word, score in result['profanity_en']]}")

    if result['is_spam']:
        print("SPAM DETECTED!")
        if result['spam_general']:
            print(f"   - General Spam: {[f'{word}({score})' for word, score in result['spam_general']]}")
        if result['gambling_spam']:
            print(f"   - Gambling Spam: {[f'{word}({score})' for word, score in result['gambling_spam']]}")

        spam_patterns = result['spam_patterns']
        if any(spam_patterns.values()):
            print("   - Suspicious Patterns:")
            if spam_patterns['url_links']:
                print(f"     • URLs: {spam_patterns['url_links']}")
            if spam_patterns['phone_numbers']:
                print(f"     • Phone Numbers: {spam_patterns['phone_numbers']}")
            if spam_patterns['suspicious_contacts']:
                print(f"     • Contact Info: {spam_patterns['suspicious_contacts']}")

    if not result['is_spam'] and not result['is_profanity']:
        print("CLEAN COMMENT")

    print("-" * 60)
```

C. *Tahapan Deteksi*

Sistem deteksi komentar ini dibangun melalui beberapa tahap utama. Tahap pertama adalah normalisasi teks, yang dilakukan untuk menyederhanakan bentuk komentar sehingga dapat dianalisis secara konsisten. Proses normalisasi meliputi konversi huruf menjad huruf kecil, penghapusan tanda baca berlebih, penghapusan spasi berulang, serta penggantian karakter-karakter yang biasa diplesetkan, seperti karakter 0 menjadi o. Normalisasi ini bertujuan untuk menyamakan bentuk ejaan dari komentar sebelum dilakukan pencocokan *string*.

Tahap selanjutnya adalah deteksi kata kasar dan spam dengan *fuzzy matching*. Sistem akan mencocokan teks komentar dengan daftar kata-kata kasar dan frasa spam baik dalam Bahasa Indonesia maupun Bahasa Inggris. Teknik *fuzzy matching* ini mampu mendeteksi kata yang ditulis dengan variasi, kesalahan ketik, atau modifikasi ejaan, dengan menghitung skor kemiripan berdasarkan *Levenshtein distance*. Kata-kata dengan tingkat kemiripan di atas ambang batas tertentu akan dianggap sebagai pelanggaran.

Setelah itu, dilakukan pendeteksian pola mencurigakan menggunakan *regular expressions (regex)*. Pada tahap ini, sistem akan mencari pola teks tertentu seperti tautan, nomor telepon, serta karakter atau huruf yang diulang secara tak wajar.

Tahap berikutnya adalah perhitungan skor spam yang dilakukan dengan mempertimbangkan jumlah dan jenis pelanggaran yang ditemukan. Setiap elemen pelanggaran memiliki bobot penilaian tersendiri, misalnya kata kasar memberikan skor pelanggaran lebih tinggi daripada pengulangan karakter secara tak wajar. Total skor kemudian digunakan untuk mengklasifikasikan tingkat resiko komentar ke dalam kategori *low*, *medium*, atau *high* dalam konteks pelanggaran komentar.

Akhirnya, sistem akan menggabungkan seluruh hasil dalam evaluasi yang menyeluruh. Komentar diklasifikasikan sebagai mengandung spam, kata kasar, atau bersih. Informasi tersebut kemudian akan ditampilkan bersama skor spam, pola yang terdeteksi, serta level resiko. Pendekatan ini memungkinkan

sistem mendeteksi bebagai jenis komentar bermasalah dengan penyamaran kata yang sulit ditangani dengan pencocokan langsung saja.

IV. Hasil dan Pembahasan

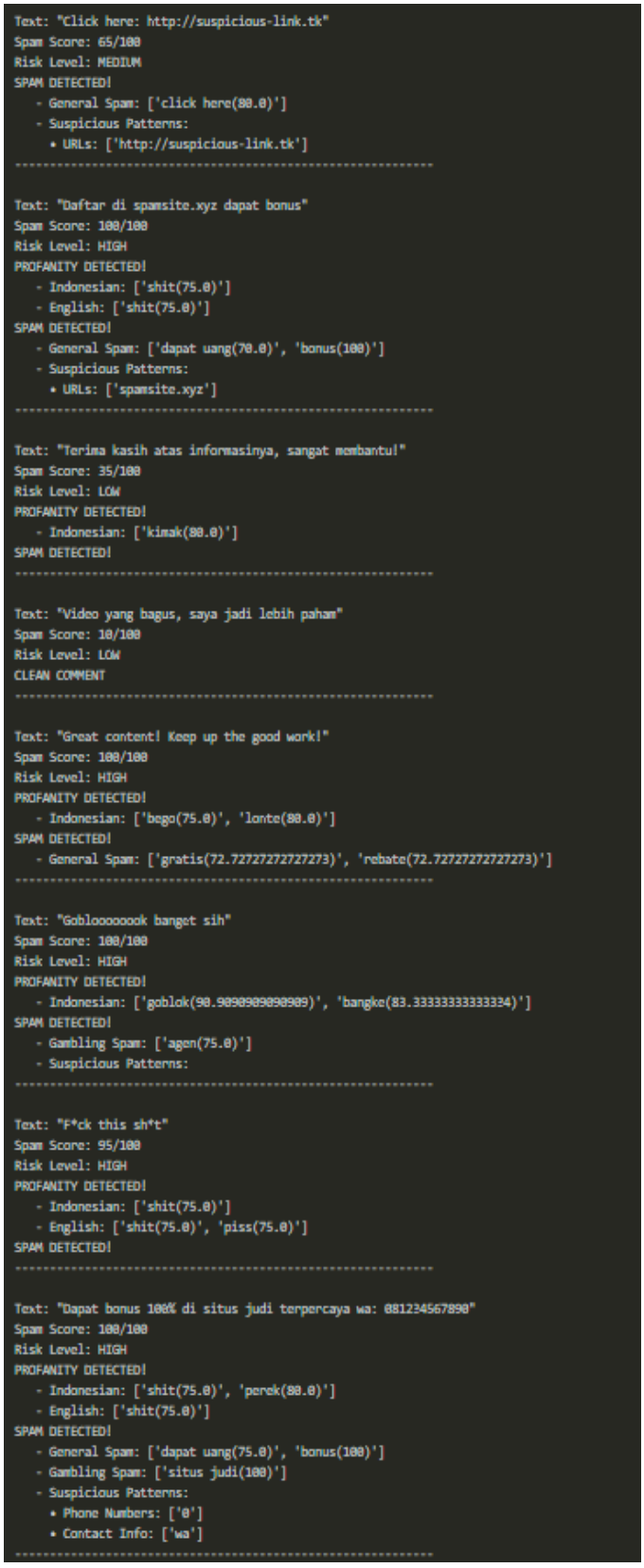
Sistem deteksi komentar yang telah diimplementasikan diuji menggunakan 25 komentar yang terdiri dari berbagai kategori, yaitu komentar yang mengandung kata kasar, komentar spam (umum dan terkait judi), serta komentar bersih. Setiap komentar dianalisis untuk mendeteksi indikasi profanity maupun spam, serta diberi penilaian dalam bentuk *spam score* dan *risk level*.

Berikut adalah hasil keluaran sistem:

```
ADVANCED COMMENT DETECTION SYSTEM
============================================================

Text: "Kamu itu g0bl0k banget anjing!"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['anjing(100)', 'goblok(100)', 'bangke(83.33333333333334)', 'anying(83.3333333333
SPAM DETECTED!
   - Gambling Spam: ['agen(75.0)']
------------------------------------------------------------

Text: "Bangsat lu tolol bener"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['bangsat(100)', 'tolol(100)', 'bangke(80.0)']
   - English: ['slut(75.0)']
SPAM DETECTED!
------------------------------------------------------------

Text: "Sialan kau bego"
Spam Score: 60/100
Risk Level: MEDIUM
PROFANITY DETECTED!
   - Indonesian: ['bego(100)', 'sialan(100)']
SPAM DETECTED!
------------------------------------------------------------

Text: "You're such a fucking idiot!"
Spam Score: 85/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['idiot(100)', 'fuck(75.0)']
   - English: ['fuck(75.0)']
SPAM DETECTED!
------------------------------------------------------------

Text: "Damn bitch, shut up!"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['shit(75.0)', 'damn(100)', 'bitch(80.0)']
   - English: ['shit(75.0)', 'damn(100)', 'bitch(80.0)', 'slut(75.0)']
SPAM DETECTED!
------------------------------------------------------------

Text: "What the hell is wrong with you?"
Spam Score: 60/100
Risk Level: MEDIUM
PROFANITY DETECTED!
   - English: ['hell(100)', 'twat(75.0)']
SPAM DETECTED!
------------------------------------------------------------

Text: "Daftar sekarang di situs slot gacor! Bonus deposit 100%"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['shit(75.0)', 'itil(75.0)']
   - English: ['shit(75.0)', 'slut(75.0)']
SPAM DETECTED!
   - General Spam: ['bonus(100)']
   - Gambling Spam: ['daftar sekarang(100)', 'deposit(100)', 'gacor(80.0)', 'bonus deposit(100)']
------------------------------------------------------------
```

```
Text: "Agen togel terpercaya, prediksi angka jitu hari ini!"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['perek(80.0)']
   - English: ['dick(75.0)']
SPAM DETECTED!
   - Gambling Spam: ['togel(100)', 'agen(100)', 'prediksi(100)', 'angka jitu(100)']
------------------------------------------------------------

Text: "Casino online terbaik, live roulette 24 jam"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['babi(75.0)']
SPAM DETECTED!
   - Gambling Spam: ['slot online(81.81818181818181)', 'casino online(92.307692307692
------------------------------------------------------------

Text: "Poker online uang asli, minimal bet 10rb"
Spam Score: 100/100
Risk Level: HIGH
SPAM DETECTED!
   - Gambling Spam: ['slot online(72.72727272727273)', 'poker online(100)', 'minimal
------------------------------------------------------------

Text: "Kl1k d1s1n1 unt0k b0nus gr4t1s!!!"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - English: ['cunt(75.0)']
SPAM DETECTED!
   - General Spam: ['klik disini(100)', 'gratis(100)', 'bonus(100)']
------------------------------------------------------------

Text: "DAAAPAT UUUANG GRAAATIS!!!"
Spam Score: 60/100
Risk Level: MEDIUM
SPAM DETECTED!
   - General Spam: ['gratis(83.33333333333334)', 'dapat uang(90.0)']
------------------------------------------------------------

Text: "Slot gac0r m4xw1n x500"
Spam Score: 75/100
Risk Level: HIGH
PROFANITY DETECTED!
   - English: ['slut(75.0)']
SPAM DETECTED!
   - Gambling Spam: ['gacor(80.0)']
------------------------------------------------------------

Text: "Hub WA 08123456789 untuk info lebih lanjut"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - English: ['cunt(75.0)']
SPAM DETECTED!
   - Suspicious Patterns:
     • Phone Numbers: ['0']
     • Contact Info: ['WA']
------------------------------------------------------------
```

```
Text: "DM me for free money telegram @spammer"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['peler(80.0)']
SPAM DETECTED!
   - General Spam: ['free money(100)']
   - Suspicious Patterns:
     • Contact Info: ['telegram']
------------------------------------------------------------

Text: "PIN BBM: 12345678 bonus melimpah"
Spam Score: 75/100
Risk Level: HIGH
SPAM DETECTED!
   - General Spam: ['bonus(100)']
   - Suspicious Patterns:
     • Contact Info: ['PIN']
------------------------------------------------------------

Text: "Kunjungi www.scam-site.com sekarang juga!"
Spam Score: 100/100
Risk Level: HIGH
PROFANITY DETECTED!
   - Indonesian: ['shit(75.0)']
   - English: ['shit(75.0)']
SPAM DETECTED!
   - Suspicious Patterns:
     • URLs: ['www.scam-site.com']
------------------------------------------------------------
```



Pada kategori komentar kasar, sistem mampu mengenali kata-kata tidak pantas baik dalam Bahasa Indonesia seperti “goblok”, “anjing”, “bego”, maupun dalam Bahasa Inggris seperti “fuck”, “bitch”, dan “idiot”. *Fuzzy matching* sangat membantu dalam mendeteksi variasi penulisan yang diplesetkan pengguna, seperti “g0bl0k” dan “f*ck”.

Pada komentar *spam*, sistem mendeteksi berbagai pola yang mencakup kata-kata promosi seperti “bonus”, “gratis”, serta istilah khas spam judi daring seperti “slot gacor”, “poker online”, dan “minimal bet”. Selain itu, pola mencurigakan seperti tautan URL, nomor telepon, dan kontak seperti “WA” atau “telegram” juga berhasil diidentifikasi dengan *regex*.

Pada kategori komentar kombinasi (kasar dan spam), sistem memberikan skor tinggi dengan klasifikasi “HIGH”. Komentar semacam ini mengandung kata kasar sekaligus unsur promosi atau tauta yang mencurigakan, contohnya: “Kamu itu g0bl0k banget anjing!” atau “Dapat bonus 100% di situs judi terpercaya wa:081234567890”.

Hanya ada satu komentar yang benar-benar terdeteksi sebagai komentar bersih, yaitu “Video yang bagus, saya jadi lebih paham”, yang tidak mengandung unsur kasar maupun spam, dan memiliki skor rendah serta resiko “LOW”.

Secara keseluruhan, sistem menunjukan efektivitas tinggi dalam mengidentifikasi komentar bermasalah juga memberikan detail yang informatif. Sistem dapat menjadi alat bantu ringan yang andal untuk kebutuhan moderasi otomatis di *platform* daring.

Di akhir proses pengujian, sistem juga menghasilkan statistik keseluruhan sebagai berikut:

```
DETECTION STATISTICS
Total Comments: 25
Spam Detected: 24
Profanity Detected: 20
Clean Comments: 1
```

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Implementasi sistem deteksi otomatis komentar kasar dan spam ini menunjukan bahwa kombinasi antara *Regular Expressions (regex)* dan *fuzzy matching* berbasis *Levenshtein distance* mampu memberikan hasil yang efektif dan efisien dalam menyaring komentar bermasalah. Regex berguna dalam mendeteksi pola-pola eksplisit kata-kasar maupun spam. Sementara itu, *fuzzy matching* memungkinkan sistem mendeteksi kata-kata yang telah dimodifikasi atau diplesetkan oleh pengguna untuk menghindari filter.

Sistem ini mampu mendeteksi berbaga jenis spam termasuk spam umum, promodi judi daring, serta komentar kasar dalam Bahasa Indonesia maupun Bahasa Inggris. Secara keseluruhan, sistem ini dapat menjadi modul moderai otomatis ringan yang efektif untuk *platform* daring.

B. *Saran*

Untuk pengembangan lebih lanjut, sistem ini dapat ditingkatkan dengan menambahkan model berbasis konteks seperti machine learning agar lebih akurat dalam memahami komentar. Selain itu, pembaruan daftar kata kasar dan spam secara berkala penting untuk dilakukan agar tetap relevan

dengan tren komunikasi daring. Pengujian sistem perlu dilakukan pada data nyata dari berbagai platform untuk mengevaluasi performa sistem dalam situasi sebenarnya.

SOURCE CODE

Github repository: https://github.com/jhotlann/Makalah-stima

UCAPAN TERIMA KASIH

Puji dan Syukur kepada Allah Yang Maha Kuasa karena atas Rahmat dan karunia-Nya penulis dapat menyelesaikan makalah dengan judul "Deteksi Otomatis Komentar Kasar dan Spam dengan *Regular Expressions* dan *Fuzzy Matching*" dengan tepat waktu. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada ibu Dr. Nur Ulfa Maulidevi, S. T, M.Sc. selaku dosen pengampu mata kuliah Strategi Algoritma atas bimbingan dan ilmunya. Penulis juga mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam penulisan makalah ini. Semoga makalah ini dapat diterima dengan baik.

REFERENSI

[1] A. Schmidt dan M. Wiegand, "A survey on hate speech detection using natural language processing," in Proc. 5th In.t Workshop Natural Lang. Process. Social Media, 2017, pp. 1-10.

[2] M. Bachmann, RapidFuzz Documentation, 2024. Tersedia di: https://docs.python.org/3/library/re.html. [Diakses pada: 23 Juni 2025].

[3] R. Munir, "*String* Matching dengan Regex,". Tersedia di: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/24-*String*-Matching-dengan-Regex-(2025).pdf. [Diakses pada: 22 Juni 2025].

[4] Python Software Foundation, "re – *Regular expressions* operations,". Tersedia di: https://docs.python.org/3/library/re.html. [Diakses pada: 23 Juni 2025].

[5] T. J. Holt, A. M. Bossler, dan K. C. Seigfried-Spellar, Cybercrime and Digital Forensics: An Introduction, 2nd ed. New York: Routledge, 2015.

[6] Y. Wibisono dan M. L. Khodra, Modul Praktikum NLP – Regex. Tersedia di: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Modul-Praktikum-NLP-Regex.pdf. [Diakses pada: 24 Juni 2025].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Juni 2025

Joel Hotlan Haris Siahaan
13523025