

Pendeteksian Spoiler Film Secara Real-Time pada Obrolan Live Streaming Twitch Menggunakan Algoritma Boyer-Moore dan Regular Expression

Muhammad Adam Mirza - 18223015

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132, Indonesia

E-mail: adammirza310@gmail.com, 18223015@std.stei.itb.ac.id

Abstrak—Penelitian ini menerapkan algoritma Boyer-Moore dan Regular Expression untuk mendeteksi spoiler film pada obrolan live streaming Twitch. Sistem diimplementasikan sebagai ekstensi peramban yang dapat diaktifkan pada halaman Twitch, kemudian memeriksa pesan baru yang muncul pada elemen live chat. Deteksi dilakukan melalui tiga mode, yaitu pencocokan frasa spoiler, pencocokan entitas dan aksi spoiler, serta mode hybrid yang menggabungkan Boyer-Moore dengan pola regex. Pesan yang terdeteksi sebagai spoiler diburamkan dan diberi penanda agar pengguna dapat memilih apakah pesan akan ditampilkan kembali. Pengujian offline dilakukan pada 100 pesan berlabel untuk membandingkan akurasi, presisi, recall, F1-score, waktu pemrosesan, dan jumlah perbandingan karakter. Hasil pengujian menunjukkan bahwa mode phrase memperoleh akurasi 0,86, mode entity-action memperoleh akurasi 0,93, dan mode hybrid memperoleh akurasi 1,00 pada dataset uji yang digunakan. Penggabungan tiga jenis aturan membuat sistem mampu menangkap lebih banyak bentuk spoiler, sementara rata-rata waktu proses tetap berada di bawah 1 ms per pesan pada dataset uji.

Kata kunci—Boyer-Moore, ekstensi peramban, pencocokan string, Regular Expression, spoiler, Twitch

I. PENDAHULUAN

Obrolan pada *live streaming* Twitch memiliki karakteristik cepat, pendek, dan tidak selalu dapat dikendalikan oleh penonton. Pesan yang muncul pada *live chat* dapat berisi komentar umum, reaksi terhadap tayangan, lelucon, maupun informasi alur cerita. Pada konteks penonton yang belum menyelesaikan suatu film atau serial, informasi mengenai kematian tokoh, identitas tersembunyi, pengkhianatan, atau akhir cerita dapat dipandang sebagai *spoiler*. Masalah tersebut dapat mengganggu pengalaman menonton karena pesan muncul langsung pada area yang sama dengan percakapan utama.

Masalah pendeteksian spoiler pada live chat dapat dimodelkan sebagai pencarian pola, karena setiap pesan perlu diperiksa terhadap frasa atau struktur kalimat tertentu yang mengindikasikan spoiler. Sebuah pesan *chat* perlu diperiksa terhadap sejumlah frasa atau pola yang menunjukkan indikasi spoiler. Karena data yang diproses berbentuk pesan teks pendek, metode pencocokan string sesuai digunakan untuk memeriksa kemunculan frasa spoiler secara langsung. Pada penelitian ini, algoritma *Boyer-Moore* digunakan untuk mencocokkan frasa spoiler yang bersifat eksplisit, sedangkan *Regular Expression*

digunakan untuk mengenali pola kalimat yang lebih bervariasi.

Sistem yang dibuat tidak ditujukan untuk memahami makna cerita secara semantik. Sistem hanya mendeteksi indikasi spoiler berdasarkan aturan yang telah disediakan. Batasan ini dipilih agar penelitian tetap berada pada ruang lingkup pencocokan string dan *regex*. Keberhasilan deteksi pada sistem ini bergantung pada kelengkapan aturan yang disediakan, yaitu frasa eksplisit, pasangan entitas-aksi, dan pola ekspresi reguler.

Penelitian ini bertujuan untuk merancang dan menguji prototipe ekstensi peramban yang dapat diaktifkan pada halaman Twitch. Setelah ekstensi aktif, pesan baru pada *live chat* diproses, kemudian pesan yang terdeteksi sebagai *spoiler* diburamkan dan diberi tombol untuk menampilkan kembali pesan tersebut. Evaluasi dilakukan melalui pengujian *offline* terhadap dataset berlabel dan pengujian live untuk memperlihatkan bahwa sistem dapat bekerja pada tampilan Twitch.

II. DASAR TEORI

A. Pencocokan String

Pencocokan string merupakan proses menemukan posisi suatu pola di dalam teks. Jika teks dinotasikan sebagai T dan pola dinotasikan sebagai P, maka pencocokan dilakukan untuk menentukan apakah P muncul sebagai bagian dari T. Materi pencocokan string dalam IF2211 digunakan sebagai dasar pemilihan algoritma karena pesan *chat* dapat dipandang sebagai teks, sedangkan aturan *spoiler* dapat dipandang sebagai pola [1].

Pada penelitian ini, pencocokan string tidak hanya digunakan untuk frasa lengkap, tetapi juga untuk pencarian entitas dan aksi. Entitas merepresentasikan nama tokoh atau objek cerita, sedangkan aksi merepresentasikan peristiwa seperti “mati”, “kalah”, “disegel”, atau “menjadi pengkhianat”. Apabila entitas dan aksi ditemukan dalam pesan yang sama, pesan tersebut diberi indikasi spoiler.

B. Algoritma Boyer-Moore

Boyer-Moore merupakan algoritma pencocokan string yang membandingkan pola dari kanan ke kiri dan menggunakan informasi karakter yang tidak cocok untuk melakukan pergeseran pola [2]. Pendekatan ini berbeda dari pencocokan langsung yang membandingkan pola dari kiri ke kanan pada setiap posisi. Pada implementasi penelitian ini, heuristic *bad character* digunakan untuk menentukan pergeseran ketika terjadi ketidakcocokan karakter.

Boyer-Moore sesuai untuk pencarian frasa spoiler yang eksplisit karena frasa tersebut telah diketahui sebelumnya. Contoh frasa yang digunakan adalah “*batman is bruce wayne*”, “*tony stark mati*”, dan “*iron man dies*”. Jumlah perbandingan karakter dicatat agar perilaku algoritma dapat diamati pada pesan yang berbeda.

C. Regular Expression

Regular Expression adalah pola yang digunakan untuk mencocokkan kombinasi karakter dalam string. Pada *JavaScript*, *regex* digunakan sebagai objek untuk melakukan pencocokan, pencarian, maupun pengujian bentuk string [3]. Dalam penelitian ini, *regex* digunakan untuk pola spoiler yang tidak selalu memiliki frasa tetap, misalnya pola “*ternyata X villain*” atau “*endingnya X meninggal*”.

Penggunaan *regex* diperlukan karena tidak semua spoiler muncul sebagai frasa lengkap. Pesan seperti “*endingnya dia meninggal*” dapat tidak cocok dengan daftar frasa tertentu, tetapi tetap dapat dideteksi melalui pola yang memuat kata “*endingnya*” dan “*meninggal*”. Dengan demikian, *regex* melengkapi pencocokan *Boyer-Moore* pada kasus yang lebih bervariasi.

D. Obrolan Live Streaming Twitch dan Spoiler

Twitch menyediakan layanan *live streaming* dengan fitur obrolan langsung. Pada implementasi berbasis ekstensi, halaman Twitch dipantau melalui *content script* yang dimuat pada halaman dengan pola URL tertentu. Mekanisme *content script* memungkinkan berkas *JavaScript* ekstensi dijalankan pada halaman yang sesuai dengan konfigurasi manifest [4]. Pada penelitian ini, pendekatan tersebut digunakan untuk memantau elemen pesan pada *live chat*.

Spoiler pada penelitian ini didefinisikan sebagai pesan yang memuat informasi peristiwa cerita yang seharusnya belum diketahui penonton. Definisi tersebut diterapkan secara operasional melalui aturan deteksi, bukan melalui pemahaman konteks cerita. Oleh karena itu, pesan yang tidak sesuai dengan aturan dapat tidak terdeteksi meskipun secara manusiawi dapat dianggap sebagai *spoiler*.

III. METODOLOGI

A. Ruang Lingkup Sistem

Sistem yang dibuat berupa ekstensi peramban Chrome berbasis Manifest V3. Ekstensi dapat diaktifkan dan dinonaktifkan melalui *popup*. Ketika aktif, sistem hanya memproses pesan baru yang muncul setelah pengamatan dimulai. Pembatasan ini dilakukan agar pesan lama tidak diproses berulang dan agar beban pemrosesan pada halaman Twitch tetap kecil.

Alur sistem terdiri atas aktivasi ekstensi, pemantauan elemen *chat*, prapemrosesan pesan, deteksi dengan *Boyer-Moore*, deteksi dengan *regex*, pemberian label spoiler, dan penyimpanan *log*. Log digunakan sebagai bahan evaluasi karena memuat pesan, metode deteksi, aturan yang cocok, waktu pemrosesan, dan jumlah perbandingan karakter.

B. Data Aturan Deteksi

Aturan deteksi dibagi menjadi tiga kelompok. Kelompok pertama adalah frasa spoiler eksplisit. Kelompok kedua adalah entitas dan aksi spoiler. Kelompok ketiga adalah pola *Regular Expression*.

Pemisahan aturan dilakukan supaya kontribusi tiap mode dapat dibandingkan, terutama antara pencocokan frasa langsung, kombinasi entitas-aksi, dan pola *regex*.

TABEL 1. CONTOH ATURAN DETEKSI

Jenis	Contoh
Frasa	batman is bruce wayne
Entitas	gojo, tony, batman
Aksi	mati, disegel, villain
Regex	endingnya .* meninggal

C. Prapemrosesan Pesan

Pra Pemrosesan dilakukan sebelum pencocokan. Pesan diubah menjadi huruf kecil, karakter berulang disederhanakan, dan beberapa variasi *leetspeak* umum dinormalisasi. Tahap ini digunakan agar pesan seperti “G0J0 nanti di-segel” dapat lebih mudah dicocokkan dengan aturan “gojo” dan “segel”. Pra Pemrosesan juga mengurangi pengaruh tanda baca yang sering muncul pada *chat*.

D. Mekanisme Deteksi Spoiler

Deteksi dilakukan secara bertahap. Tahap pertama mencocokkan pesan terhadap daftar frasa menggunakan *Boyer-Moore*. Apabila frasa ditemukan, pesan langsung diklasifikasikan sebagai spoiler. Tahap kedua mencari keberadaan entitas dan aksi pada pesan yang sama dengan *Boyer-Moore*. Tahap ketiga menggunakan *Regular Expression* apabila dua tahap sebelumnya belum menghasilkan kecocokan.

Frasa eksplisit ditempatkan pada tahap pertama karena aturan ini paling spesifik dan paling kecil kemungkinannya memicu deteksi keliru. Pencocokan entitas dan aksi digunakan untuk memperluas cakupan, sedangkan *regex* digunakan untuk bentuk kalimat yang lebih umum. Pada mode *hybrid*, ketiga tahap tersebut dijalankan secara berurutan sampai salah satu aturan cocok.

E. Kriteria Keberhasilan

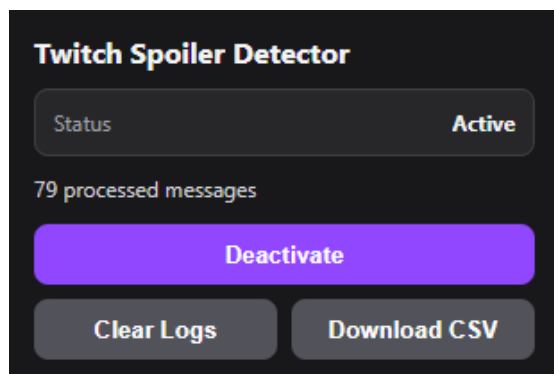
Suatu pesan dianggap berhasil diklasifikasikan apabila hasil prediksi sama dengan label pada dataset uji. Pada pengujian *live*, keberhasilan diamati dari kemampuan ekstensi memberi penanda dan memburamkan pesan yang cocok dengan aturan. Karena pengujian *live* tidak selalu memiliki label manual, data *live* digunakan sebagai bukti fungsi antarmuka dan waktu proses, bukan sebagai dasar utama perhitungan akurasi.

IV. IMPLEMENTASI DAN PENGUJIAN

A. Implementasi Ekstensi

Program diimplementasikan menggunakan *JavaScript* sebagai ekstensi Chrome. Struktur program terdiri atas manifest, *popup*, *content script*, modul *Boyer-Moore*, modul *regex*, data aturan, dan modul evaluasi *offline*. *Popup* digunakan untuk mengaktifkan atau menonaktifkan deteksi, menghapus log, dan mengunduh log dalam format CSV.

Content script mengamati perubahan pada elemen *chat* menggunakan mekanisme pengamatan DOM. Setiap elemen pesan yang baru muncul diberi penanda internal setelah diproses agar pesan yang sama tidak diperiksa berulang. Apabila pesan diklasifikasikan sebagai spoiler, isi pesan diburamkan dan tombol *reveal* ditambahkan.



Gambar 1. Tampilan popup ekstensi ketika deteksi aktif.

B. Skenario Pengujian Offline

Pengujian *offline* dilakukan terhadap 100 pesan berlabel. Dataset terdiri atas pesan spoiler dan non-spoiler yang disusun menyerupai bentuk percakapan singkat pada *chat*. Tiga mode diuji secara terpisah, yaitu *phrase*, *entity-action*, dan *hybrid*. Mode *phrase* hanya menggunakan frasa eksplisit. Mode *entity-action* menambahkan kombinasi entitas dan aksi. Mode *hybrid* menambahkan pola *regex* setelah dua tahap *Boyer-Moore*.

C. Skenario Pengujian Live

Pengujian *live* dilakukan dengan memasang ekstensi pada Chrome, membuka halaman *channel* Twitch, mengaktifkan deteksi melalui *popup*, dan mengamati pesan yang muncul pada *chat*. Pesan yang cocok dengan aturan dibuat buram dan ditandai sebagai *spoiler*. Pengujian *live* difokuskan untuk memastikan bahwa content script dapat membaca pesan baru, memberi label spoiler, memburamkan pesan, dan tetap berjalan pada antarmuka Twitch.

D. Metrik Pengujian

Metrik yang digunakan pada pengujian *offline* adalah TP, TN, FP, FN, akurasi, presisi, *recall*, *F1-score*, rata-rata waktu pemrosesan, dan rata-rata jumlah perbandingan karakter. Akurasi dihitung sebagai $(TP + TN) / \text{total}$. Presisi dihitung sebagai $TP / (TP + FP)$, sedangkan *recall* dihitung sebagai $TP / (TP + FN)$. *F1-score* digunakan karena nilai ini memperlihatkan apakah sistem hanya tepat saat mendeteksi *spoiler*, atau juga mampu menangkap sebagian besar spoiler yang ada.

V. HASIL DAN ANALISIS

A. Hasil Pengujian Offline

Hasil pengujian *offline* ditampilkan pada Tabel II. Dataset uji berjumlah 100 pesan. Dari Tabel II terlihat bahwa cakupan deteksi meningkat setelah sistem tidak hanya bergantung pada frasa eksplisit. Mode *entity-action* menambah jumlah *true positive*, sedangkan *regex* menutup beberapa kasus yang masih lolos dari dua mode sebelumnya. Mode *hybrid* memperoleh nilai tertinggi pada dataset uji karena seluruh pesan berlabel spoiler dan non-spoiler berhasil diklasifikasikan sesuai label.

TABEL II. RINGKASAN HASIL PENGUJIAN OFFLINE

Mode	Acc.	Prec.	Recall	F1	Waktu
Phrase	0,860	1,000	0,774	0,873	0,450
Ent.-Act.	0,930	1,000	0,887	0,940	0,621
Hybrid	1,000	1,000	1,000	1,000	0,725

Pada mode *phrase*, 48 pesan spoiler berhasil dideteksi dan 14 pesan spoiler tidak terdeteksi. Tidak terdapat *false positive* pada mode ini, sehingga presisi bernilai 1,00.

Namun, *recall* hanya mencapai 0,774 karena beberapa spoiler tidak menggunakan frasa yang sama persis dengan daftar frasa. Hal ini menunjukkan bahwa pencocokan frasa memiliki ketepatan tinggi, tetapi cakupannya terbatas.

Mode *entity-action* meningkatkan jumlah *true positive* menjadi 55 dan menurunkan *false negative* menjadi 7. Peningkatan tersebut terjadi karena pesan yang tidak cocok dengan frasa lengkap masih dapat dideteksi apabila mengandung entitas dan aksi spoiler. Contoh kasusnya adalah pesan “gojo nanti disegel” yang tidak cocok pada mode *phrase*, tetapi berhasil dideteksi melalui aturan “gojo + disegel”.

Mode *hybrid* memperoleh akurasi, presisi, *recall*, dan *F1-score* sebesar 1,00 pada dataset uji. Nilai ini menunjukkan bahwa seluruh pesan pada dataset uji berhasil diklasifikasikan sesuai label. Namun, hasil tersebut tidak dapat digeneralisasi ke seluruh percakapan Twitch karena dataset masih terbatas dan disusun berdasarkan aturan yang tersedia pada penelitian ini.

B. Contoh Detail Deteksi

Beberapa contoh hasil klasifikasi ditampilkan pada Tabel III. Tabel ini tidak memuat seluruh baris pengujian, tetapi hanya contoh yang mewakili kondisi penting, yaitu pesan aman, frasa eksplisit, kombinasi entitas-aksi, pola *regex*, dan pesan identitas tokoh.

TABEL III. CONTOH DETAIL HASIL DETEKSI

Pesan	Label	Metode	Aturan
film ini keren	0	None	-
tony stark mati	1	BM Phrase	tony stark mati
gojo disegel	1	BM Ent-Act	gojo + disegel
endingnya meninggal	1	Regex	endingnya.*
batman is bruce	1	BM Phrase	batman is bruce

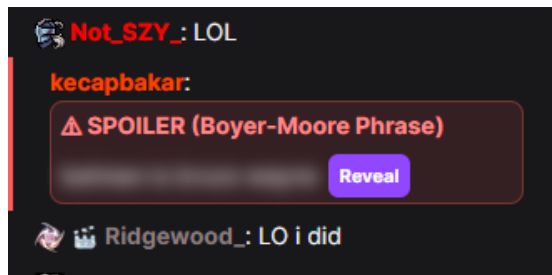
C. Analisis Waktu Pemrosesan

Rata-rata waktu pemrosesan pada mode *phrase* adalah 0,450 ms per pesan. Mode *entity-action* membutuhkan 0,621 ms per pesan, sedangkan mode *hybrid* membutuhkan 0,725 ms per pesan. Kenaikan waktu terjadi karena jumlah pemeriksaan bertambah, tetapi seluruh nilai rata-rata masih berada di bawah 1 ms pada lingkungan pengujian yang digunakan.

Jumlah perbandingan karakter juga meningkat ketika mode *entity-action* dan *hybrid* digunakan. Mode *phrase* memiliki rata-rata 266,05 perbandingan, sedangkan dua mode lainnya memiliki rata-rata 653,62 perbandingan. Nilai tersebut wajar karena pencarian tidak lagi berhenti pada daftar frasa, tetapi juga memeriksa daftar entitas dan aksi.

D. Hasil Pengujian Live pada Twitch

Pengujian *live* menunjukkan bahwa ekstensi dapat diaktifkan pada halaman Twitch dan memproses pesan yang muncul pada *live chat*. Pada saat pengujian, *popup* menampilkan status aktif dan jumlah pesan yang telah diproses. Pesan yang cocok dengan aturan diberi label spoiler dan diburamkan, sedangkan tombol *reveal* digunakan untuk membuka pesan apabila pengguna menghendaki.



Gambar 2. Tampilan pesan spoiler yang diburamkan pada live chat Twitch.

Gambar 2 menunjukkan bahwa pesan “batman is bruce wayne” diklasifikasikan sebagai spoiler oleh mode *Boyer-Moore Phrase*. Pesan tersebut tidak dihapus dari halaman, tetapi disamarkan. Pilihan ini digunakan agar informasi tetap dapat diakses melalui tombol *reveal* apabila pengguna memang ingin melihat isi pesan.

E. Keterbatasan

Penelitian ini memiliki beberapa keterbatasan. Pertama, sistem masih berbasis aturan sehingga sangat bergantung pada daftar frasa, entitas, aksi, dan *regex* yang disediakan. Kedua, dataset uji masih terbatas dan belum mencakup semua variasi bahasa pada *chat Twitch*. Ketiga, sistem belum dapat membedakan pesan yang benar-benar membocorkan cerita dengan pesan yang hanya berisi dugaan atau permintaan untuk tidak memberi spoiler.

Keterbatasan lain terletak pada struktur halaman *Twitch* yang dapat berubah sewaktu-waktu. Perubahan pada elemen *DOM* dapat menyebabkan *content script* perlu disesuaikan kembali. Dengan keterbatasan tersebut, implementasi ini lebih tepat diposisikan sebagai prototipe akademik untuk menguji penerapan *Boyer-Moore* dan *regex*, bukan sebagai sistem moderasi spoiler yang siap digunakan pada seluruh kondisi *Twitch*.

Pada penerapan dengan volume chat yang lebih tinggi, jumlah aturan dan kompleksitas *regex* tetap perlu dibatasi. *Regex* yang terlalu umum atau terlalu banyak dapat meningkatkan biaya pemrosesan. Oleh karena itu, urutan deteksi dibuat bertahap, yaitu frasa eksplisit terlebih dahulu, kemudian *entity-action*, dan *regex* sebagai lapisan terakhir. Urutan ini membuat sebagian pesan dapat selesai diproses sebelum seluruh aturan dijalankan.

Rata-rata waktu pemrosesan di bawah 1 ms per pesan menunjukkan bahwa sistem masih dapat digunakan untuk prototipe pendeteksian real-time pada skala pengujian ini. Hal tersebut disebabkan oleh pembatasan pemrosesan hanya pada pesan baru yang muncul setelah ekstensi diaktifkan. Pesan yang telah diproses diberi penanda internal sehingga tidak diperiksa ulang.

F. Analisis Kesalahan pada Mode Phrase

Mode *phrase* memiliki nilai presisi 1,00 karena seluruh pesan yang terdeteksi sebagai spoiler memang berlabel spoiler pada dataset uji. Akan tetapi, mode ini menghasilkan 14 false negative. Kondisi tersebut terjadi karena sebagian pesan tidak memakai frasa yang sama persis dengan daftar frasa. Pesan “gojo nanti disegel”, “tony stark tewas di ending”, dan “endingnya dia meninggal” merupakan contoh pesan yang tidak selalu dapat dicakup oleh frasa tetap.

Hasil tersebut menunjukkan bahwa pencocokan frasa murni sesuai untuk spoiler yang eksplisit, tetapi kurang

memadai untuk percakapan yang menggunakan parafrasa. Pada chat *Twitch*, pengguna sering memendekkan kata, menggunakan variasi bahasa Indonesia dan Inggris, serta menambahkan tanda baca tidak teratur. Variasi ini membuat daftar frasa harus diperluas secara terus-menerus apabila sistem hanya mengandalkan mode *phrase*.

Kegagalan pada mode *phrase* tidak menunjukkan bahwa *Boyer-Moore* tidak sesuai, melainkan menunjukkan bahwa bentuk pola yang diberikan kepada algoritma terlalu sempit. *Boyer-Moore* tetap bekerja sesuai fungsi pencocokan string, tetapi cakupan pendeteksian ditentukan oleh daftar pola yang tersedia. Oleh karena itu, perlu ditambahkan bentuk aturan lain agar sistem tidak hanya bergantung pada frasa lengkap.

G. Pengaruh Entity-Action terhadap Cakupan

Mode *entity-action* ditambahkan untuk mengurangi ketergantungan terhadap frasa lengkap. Pada mode ini, sebuah pesan dianggap memiliki indikasi spoiler apabila ditemukan nama entitas dan kata aksi dalam pesan yang sama. Dengan aturan tersebut, pesan yang tidak identik dengan frasa tertentu masih dapat terdeteksi.

Peningkatan *recall* dari 0,774 menjadi 0,887 menunjukkan bahwa pendekatan *entity-action* memperluas cakupan deteksi. Namun, pendekatan ini tetap memiliki risiko apabila diterapkan pada data yang lebih bebas. Pesan yang berisi dugaan, pertanyaan, atau larangan memberi spoiler dapat saja memuat entitas dan aksi, tetapi belum tentu benar-benar membocorkan alur cerita. Risiko tersebut belum muncul sebagai false positive pada dataset uji, tetapi perlu diperhatikan pada penerapan yang lebih luas.

H. Pengaruh Regular Expression terhadap Pola Kalimat

Regular Expression berperan pada pesan yang memiliki struktur kalimat spoiler, tetapi tidak memiliki entitas tertentu. Pola seperti “endingnya .* meninggal” atau “ternyata .* villain” dapat menangkap bentuk kalimat yang lebih umum. Dengan demikian, *regex* digunakan sebagai lapisan terakhir setelah *Boyer-Moore* tidak menemukan frasa maupun kombinasi entitas dan aksi.

Pada dataset uji, penambahan *regex* membuat mode *hybrid* dapat mendeteksi seluruh pesan spoiler. Contohnya, pesan “endingnya dia meninggal” tidak terdeteksi oleh mode *phrase* maupun *entity-action*, tetapi dapat terdeteksi oleh pola *regex*. Hasil ini menunjukkan bahwa *regex* berguna untuk pola kalimat yang lebih terbuka, meskipun daftar pola tetap perlu disusun dengan hati-hati agar tidak terlalu longgar.

I. Validitas dan Replikasi Pengujian

Pengujian offline dilakukan dengan dataset berlabel agar metrik seperti akurasi, presisi, *recall*, dan *F1-score* dapat dihitung. Dataset tersebut digunakan sebagai pengujian terkontrol karena setiap pesan telah diberi label spoiler atau non-spoiler. Label manual digunakan sebagai acuan sehingga setiap prediksi dapat dihitung sebagai TP, TN, FP, atau FN.

Walaupun demikian, dataset uji masih terbatas pada 100 pesan dan sebagian pola pesan disusun berdasarkan aturan yang tersedia dalam sistem. Oleh karena itu, hasil mode *hybrid* yang mencapai 1,00 perlu ditafsirkan sebagai hasil pada dataset tersebut, bukan sebagai jaminan performa pada seluruh percakapan *Twitch*. Pengujian

lanjutan perlu menggunakan pesan yang lebih banyak, lebih acak, dan mencakup lebih banyak variasi bahasa.

Agar pengujian dapat direplikasi, data uji dan hasil evaluasi disimpan dalam berkas JSON dan CSV. Berkas tersebut memuat mode pengujian, label, prediksi, metode yang cocok, aturan yang cocok, waktu pemrosesan, dan jumlah perbandingan. Format ini memudahkan pemeriksaan ulang karena setiap prediksi dapat ditelusuri sampai ke pesan asalnya.

J. Implikasi terhadap Penerapan Real-Time

Rata-rata waktu pemrosesan di bawah 1 ms per pesan menunjukkan bahwa sistem masih dapat digunakan untuk prototipe pendeteksian real-time pada skala pengujian ini. Hal tersebut disebabkan oleh pembatasan pemrosesan hanya pada pesan baru yang muncul setelah ekstensi diaktifkan. Pesan yang telah diproses diberi penanda internal sehingga tidak diperiksa ulang.

Pada penerapan dengan volume chat yang lebih tinggi, jumlah aturan dan kompleksitas regex tetap perlu dibatasi. Regex yang terlalu umum atau terlalu banyak dapat meningkatkan biaya pemrosesan. Oleh karena itu, urutan deteksi dibuat bertahap, yaitu frasa eksplisit terlebih dahulu, kemudian entity-action, dan regex sebagai lapisan terakhir. Urutan ini membuat sebagian pesan dapat selesai diproses sebelum seluruh aturan dijalankan.

K. Perbandingan dengan Mode Tunggal

Perbandingan antarmode menunjukkan bahwa penggunaan satu jenis aturan saja belum cukup untuk menangkap variasi spoiler. Mode phrase dapat bekerja baik ketika pesan memuat frasa yang telah tersedia, tetapi gagal ketika pesan menggunakan padanan kata atau susunan kalimat yang berbeda. Mode entity-action memperbaiki kondisi tersebut, namun masih belum dapat menangkap kalimat spoiler yang tidak memuat entitas tertentu.

Mode hybrid menjadi pilihan akhir karena setiap lapisan aturan memiliki fungsi yang berbeda. Frasa digunakan untuk kasus eksplisit, entitas-aksi digunakan untuk kasus yang masih memuat tokoh dan kejadian, sedangkan regex digunakan untuk pola kalimat umum. Susunan bertahap ini mengurangi kebutuhan untuk menulis semua kemungkinan spoiler sebagai frasa lengkap. Daftar frasa tetap dapat ditambah, tetapi cakupan awal sudah dibantu oleh dua lapisan aturan lain.

L. Pertimbangan Antarmuka Pengguna

Pada pengujian live, pesan spoiler tidak dihapus dari halaman. Pesan hanya diburamkan dan diberi label. Pesan tidak langsung dihapus agar pengguna masih dapat memeriksa isi pesan apabila deteksi dianggap keliru. Apabila pesan dihapus sepenuhnya, pengguna tidak dapat memeriksa apakah deteksi benar atau salah. Dengan tombol reveal, kesalahan deteksi masih dapat ditangani oleh pengguna tanpa perlu mematikan ekstensi.

Pemberian label juga digunakan agar pengguna mengetahui alasan pesan berubah tampilan. Label memuat metode yang mendeteksi pesan, misalnya Boyer-Moore Phrase. Label metode juga membantu proses evaluasi karena penyebab deteksi dapat dilacak dari tampilan maupun log CSV.

M. Saran Pengembangan

Pengembangan berikutnya dapat dilakukan dengan memperluas daftar aturan deteksi, baik dalam bentuk frasa *spoiler*, entitas tokoh, aksi *spoiler*, maupun pola *regular expression*. Perluasan daftar aturan diperlukan karena sistem berbasis pencocokan *string* sangat bergantung pada kelengkapan pola yang disediakan. Apabila suatu *spoiler* disampaikan dengan frasa yang belum tercakup dalam daftar aturan, pesan tersebut berpotensi tidak terdeteksi. Namun, perluasan aturan tetap perlu dilakukan secara terkontrol karena aturan yang terlalu umum dapat meningkatkan kemungkinan *false positive*.

Daftar aturan juga dapat dipisahkan berdasarkan judul film atau serial. Pemisahan tersebut memungkinkan pengguna memilih daftar *spoiler* yang relevan dengan konten yang sedang dihindari. Jika pengguna tidak menonton suatu serial tertentu, aturan untuk serial tersebut dapat dinonaktifkan agar *false positive* berkurang. Pendekatan ini masih tetap berada pada pencocokan *string* karena aturan hanya dipilih dan disusun berdasarkan kategori.

Pengembangan berikutnya juga dapat dilakukan dengan menambahkan aturan negasi dan konteks kalimat. Contohnya, pesan “jangan *spoiler* tony mati sih” dapat memuat entitas “tony” dan aksi “mati”, tetapi konteksnya belum tentu dimaksudkan sebagai *spoiler* faktual. Aturan tambahan dapat dibuat untuk mendeteksi kata seperti “jangan”, “belum”, “kalau”, atau “mungkin” agar pesan dugaan dan larangan tidak langsung diperlakukan sama dengan *spoiler* eksplisit.

Evaluasi juga dapat diperluas dengan mengambil data *chat* yang lebih banyak dari beberapa *channel*. Data tersebut perlu diberi label manual agar akurasi dapat dihitung secara lebih kuat. Pengujian *live* tetap diperlukan, tetapi pengukuran ketepatan sebaiknya dilakukan pada *dataset* yang telah dilabeli karena obrolan langsung tidak selalu menyediakan kebenaran label yang dapat diperiksa.

VI. KESIMPULAN

Berdasarkan implementasi dan pengujian yang dilakukan, *Boyer-Moore* dan *Regular Expression* dapat digunakan sebagai dasar sistem pendeteksi spoiler sederhana pada *live chat* Twitch. *Boyer-Moore* digunakan untuk mencocokkan frasa *spoiler* eksplisit serta pasangan entitas dan aksi, sedangkan *Regular Expression* digunakan untuk mencocokkan pola kalimat yang lebih bervariasi. Dengan pembagian tersebut, sistem tidak hanya bergantung pada satu bentuk pencocokan, tetapi dapat menangani beberapa bentuk pesan, mulai dari frasa tetap seperti “tony stark mati” sampai pola yang lebih umum seperti “endingnya dia meninggal”.

Hasil pengujian *offline* pada 100 pesan menunjukkan bahwa mode *phrase* memperoleh akurasi 0,86, mode *entity-action* memperoleh akurasi 0,93, dan mode *hybrid* memperoleh akurasi 1,00 pada *dataset* uji. Peningkatan performa tersebut terjadi karena mode *hybrid* menggabungkan pencocokan frasa eksplisit, pasangan entitas-aksi, dan pola *regex*. Mode *phrase* cenderung lebih terbatas karena hanya dapat mendeteksi pesan yang sesuai dengan daftar frasa tertentu. Mode *entity-action* memperluas cakupan deteksi dengan mencari keberadaan tokoh dan peristiwa dalam satu pesan. Mode *hybrid* memberikan hasil paling baik pada *dataset* uji karena dapat menutup sebagian kelemahan dari dua mode sebelumnya.

Pengujian *live* menunjukkan bahwa ekstensi dapat diaktifkan pada halaman Twitch, memproses pesan baru yang muncul pada obrolan, memburamkan pesan yang terdeteksi sebagai *spoiler*, serta mencatat hasil deteksi ke dalam log. Hasil *live test* memperlihatkan bahwa pendekatan berbasis

pencocokan string cukup ringan untuk dijalankan di sisi pengguna, selama jumlah aturan dan kompleksitas regex tetap dibatasi. Meskipun demikian, cakupan sistem masih bergantung pada kelengkapan daftar aturan yang digunakan. Pesan yang menggunakan frasa di luar aturan dapat tidak terdeteksi, sedangkan aturan yang terlalu umum dapat meningkatkan risiko *false positive*.

Penelitian lanjutan dapat dilakukan dengan memperluas *dataset* uji, menambahkan lebih banyak frasa *spoiler*, entitas, aksi, dan pola *regex*, serta menyusun aturan berdasarkan kategori film atau serial. Selain itu, aturan negasi dan konteks kalimat juga dapat ditambahkan agar pesan berupa larangan, dugaan, atau pertanyaan tidak langsung diperlakukan sama dengan *spoiler* eksplisit. Dengan perbaikan tersebut, sistem diharapkan dapat memiliki cakupan deteksi yang lebih luas tanpa mengabaikan ketepatan klasifikasi pesan.

TAUTAN REPOSITORI

<https://github.com/AdmMRZ/twitch-spoiler-detection-stima>

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada dosen pengampu IF2211 Strategi Algoritma atas materi pembelajaran pencocokan string dan regex. Terima kasih juga kepada keluarga atas doa dan dukungannya, serta teman-teman yang selalu membantu dan memberikan motivasi.

REFERENSI

- [1] R. Munir, *Pencocokan String (String/Pattern Matching)*. Slide Kuliah IF2211 Strategi Algoritma. Bandung: Program Studi Teknik Informatika, Institut Teknologi Bandung, 2026.
- [2] R. S. Boyer and J. S. Moore, "A fast string searching algorithm," *Communications of the ACM*, vol. 20, no. 10, pp. 762-772, 1977.
- [3] MDN Web Docs, "Regular expressions - JavaScript." [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_expressions.
- [4] Chrome Developers, "Manifest - content scripts." [Online]. Available: <https://developer.chrome.com/docs/extensions/reference/manifest/content-scripts>.

PERNYATAAN

Dengan ini dinyatakan bahwa makalah ini merupakan tulisan sendiri, bukan saduran atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 17 Juni 2026



Muhammad Adam Mirza 18223015