BAB V. PENUTUP

5.1 Kesimpulan

1. Implementasi algoritma *Boyer-Moore* pada sistem filterisasi konten teks dilakukan dengan cara mencocokkan input teks dari pengguna terhadap daftar kata tidak pantas (*blacklist*) yang telah ditentukan. Proses pencocokan dilakukan menggunakan pendekatan pergeseran karakter berdasarkan dua *heuristic* utama, yaitu *Occurrence Heuristic* dan *Match Heuristic*, untuk mempercepat pencarian pola. Algoritma ini diterapkan dalam prototipe *platform* komunikasi personal berbasis *web* pada berbagai titik input pengguna, seperti saat mengetik postingan dan membalas komentar. Fungsi-fungsi utama seperti *buildBadCharacterTable* dan *BoyerMooreSearch* digunakan dalam *backend* maupun *frontend* untuk mendeteksi dan memblokir kata-kata tidak pantas sebelum konten diproses lebih lanjut.
2. Berdasarkan hasil pengujian, algoritma *Boyer-Moore* memiliki waktu pemrosesan yang cepat, yaitu sekitar 0,023 detik per pengecekan, namun tingkat akurasinya secara mandiri hanya mencapai 12,5%. Hal ini menunjukkan bahwa *Boyer-Moore* kurang efektif dalam menyaring kata tidak pantas yang telah dimodifikasi atau ditulis dengan variasi tidak standar. Sebaliknya, algoritma *Fuzzy Regex*, meskipun sedikit lebih lambat dengan waktu proses 0,048 detik, menunjukkan akurasi yang jauh lebih tinggi yaitu 62,5% saat digunakan sendiri. Kombinasi antara algoritma *Boyer-Moore*, *Fuzzy Regex*, teknik normalisasi, dan penanganan overlap mampu meningkatkan akurasi hingga 95,8% dengan waktu proses sekitar 0,095 detik, menjadikannya solusi paling optimal untuk menyaring konten teks pada platform komunikasi berbasis teks.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, peneliti menyarankan beberapa hal terkait sistem yaitu:

1. Pengembangan Deteksi Sarkasme dan Sinonim Kasar
Banyak kata tidak sopan yang disamarkan dalam bentuk sindiran, sarkasme, atau dengan menggunakan sinonim. Sistem ke depan perlu mampu mengenali bentuk ujaran kebencian secara implisit, termasuk ekspresi *sarkastik* dan istilah baru yang bermakna merendahkan.
2. Fitur Pembelajaran Mandiri (*Self-Learning*)
Untuk menjaga efektivitas sistem seiring waktu, dapat dipertimbangkan fitur *self-learning* yang memungkinkan sistem mempelajari pola baru dari laporan pengguna atau data pelanggaran yang dikurasi secara manual.
3. Integrasi Pemahaman Semantik dan Tata Bahasa
Diperlukan pengembangan sistem yang mampu memahami konteks kalimat melalui analisis semantik dan struktur tata bahasa. Hal ini akan mendukung algoritma dalam membedakan penggunaan kata yang sama dalam konteks berbeda, sehingga mengurangi kesalahan deteksi dan meningkatkan presisi penyaringan.