

Penyaring Komentar Cyber-Bullying pada Konten Blog

Danar Dono, *Departemen Informatika Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: donodanar35@gmail.com)*, Pembimbing Eka Rahayu Setyaningsih, S.Kom., M.Kom., *Departemen Informatika Institut Sains dan Teknologi Terpadu, Surabaya, Jawa Timur, Indonesia (e-mail: eka@stts.edu)*, Co. Pembimbing C. Pickerling, S.Kom., M.Kom., *Departemen Informatika Institut Sains dan Teknologi Terpadu, Surabaya, Jawa Timur, Indonesia (e-mail: pickerling@gmail.com)*

Abstrak—*Cyberbullying* merupakan ancaman nyata dalam interaksi di antara penulis konten blog dan pembaca blog. Tugas akhir ini bertujuan membahas tentang masalah sebagai berikut ini. Pertama, mengembangkan sistem manajemen konten blog menggunakan *framework* CodeIgniter. Kedua, mengembangkan fitur penyaring *cyberbullying* untuk mengklasifikasikan komentar menjadi *cyberbullying* atau *non-cyberbullying*.

Multinomial Naive Bayes digunakan sebagai metode untuk klasifikasi komentar. Dengan menggunakan *4-fold cross validation*, dataset sejumlah 6.537 komentar dibagi menjadi 4 sub himpunan dataset D₁, D₂, D₃, D₄ yang berisi masing-masing 3 *fold* untuk data latih dan 1 *fold* untuk data uji. *Accuracy and error rate* digunakan untuk mengukur kinerja model dalam mengklasifikasikan komentar. Adapun indikator-indikator *usability*, *reliability*, *functionality*, *efficiency* digunakan untuk mengukur kualitas *website* dengan melibatkan 39 responden untuk menilai kualitas *website* menggunakan skala likert 1-5.

Berdasarkan pengujian dapat disimpulkan bahwa: pengembangan sistem manajemen konten menggunakan *framework* CodeIgniter memberikan pengalaman bagi penggunaannya dengan tingkat *usability* sebesar 3,49 dari skala 5,00; memiliki tingkat *reliability* sebesar 4,28 dari skala 5,00; memiliki tingkat *functionality* sebesar 4,11 dari skala 5,00; dan memiliki tingkat *efficiency* sebesar 3,78 dari skala 5,00. Pengembangan sistem manajemen konten memiliki beberapa fitur seperti penggunaan *pretty URL*; penggunaan *Ion Auth* dan *Facebook API* untuk otentifikasi secara ringkas dan ringan; penggunaan *Chartjs* untuk menyajikan grafik laporan; menyediakan interaksi antara pembaca dan penulis melalui fitur *rating*, komentar dan pelaporan penyalahgunaan komentar; menyediakan fitur pengelompokan artikel berdasarkan topik artikel dan kata kunci. Pengembangan fitur penyaring komentar *cyberbullying* menghasilkan: pertama, model dengan fitur *stemming-unigram*, model tanpa fitur *stemming-unigram*, model dengan fitur *stemming-bigram*, model dengan tanpa fitur *stemming-bigram* masing-masing memiliki akurasi sebesar 73%, 70%, 67%, 66%. Kedua, model *stemming-unigram* memiliki ukuran fitur kata sebesar 21% atau setara 4.910 fitur kata dibandingkan ukuran fitur kata dari model dengan tanpa fitur *stemming-bigram* sebesar 23.787 fitur kata. Ketiga, penggunaan fitur *stemming-unigram* mampu memberikan akurasi yang baik sebesar 73% dan tingkat *error* 27% serta efisien secara ukuran fitur kata.

Kata Kunci—*CMS*, *cyberbullying*, *Multinomial Naive Bayes*.

I. PENDAHULUAN

Dewasa ini ada banyak sistem manajemen konten (CMS) yang digunakan untuk mengelola konten suatu situs web. CMS biasanya dilengkapi dengan fitur komentar bagi penggunaannya, memungkinkan terjadinya interaksi antara pembuat konten dan pembaca artikel. Komentar dapat berisi *cyberbullying* dalam konten blog. *Cyberbullying* adalah perlakuan kejam yang disengaja kepada orang lain dengan mengirimkan atau mengedarkan materi berbahaya atau terlibat dalam bentuk agresi sosial menggunakan internet atau teknologi digital lainnya. *Cyberbullying* berdampak negatif pada korban trauma psikologis, emosional dan sosial. Fitur filter komentar *cyberbullying* diperlukan untuk menyaring berbagai komentar berpotensi *cyberbullying* dalam konten blog. Tugas akhir ini akan membahas bagaimana mengembangkan sistem filter komentar *cyberbullying* pada konten blog yang nantinya dapat digunakan untuk meminimalisir penyalahgunaan komentar, khususnya terkait bullying oleh pengguna melalui fitur komentar pada konten blog. *Cyberbullying* merupakan ancaman nyata dalam interaksi di antara penulis konten blog dan pembaca artikel blog. *Cyberbullying* berdampak negatif terhadap korbannya seperti gangguan mental, tekanan emosional, hingga trauma sosial. Tugas akhir ini membahas bagaimana pengembangan sistem manajemen konten dan pengembangan fitur penyaring komentar *cyberbullying* untuk mengklasifikasi kategori komentar menjadi *cyberbullying* atau *non-cyberbullying*.

II. TINJAUAN PUSTAKA

Pada bab ini dijelaskan tentang teori penunjang yang digunakan dalam pengembangan sistem ini meliputi:

A. Cyberbullying

Menurut Williard (2005), *cyberbullying* adalah perlakuan kejam yang dilakukan dengan sengaja kepada orang lain dengan mengirimkan atau mengedarkan materi berbahaya atau terlibat dalam bentuk agresi sosial menggunakan internet atau teknologi digital lainnya. Adapun aspek-aspek *cyberbullying* meliputi [1]:

1. *Flamming* yakni perilaku pengiriman pesan teks dengan kata-kata kasar dan frontal.
2. *Harassment* yakni perilaku pengiriman pesan tidak sopan kepada seseorang berupa gangguan yang dikirimkan melalui email, sms, atau pesan singkat di jaringan media sosial secara terus menerus.
3. *Denigration* yakni perilaku mengubur keburukan seorang di internet dengan maksud merusak reputasi dan nama baik dari orang yang dituju..
4. *Impersonation* yakni perilaku berpura-pura menjadi orang lain dan mengirimkan pesan yang tidak baik.
5. *Outing and Trickery* yakni perilaku menyebarkan rahasia orang lain atau foto pribadi orang lain.
6. *Exclusion* yakni perilaku yang dengan sengaja dan kejam menghilangkan orang dari grup online
7. *Cyberstalking* yakni perilaku berulang kali mengirimkan ancaman berbahaya atau pesan yang mengintimidasi menggunakan komunikasi elektronik.

Dalam tugas akhir ini lebih difokuskan pada aspek-aspek *cyberbullying* meliputi *flamming*, *harassment*, *denigration*, dan *cyberstalking*.

B. Codeigneter

Codeigneter (CI) merupakan framework berbasis web untuk orang yang ingin membuat aplikasi web menggunakan bahasa PHP. Tujuannya adalah membuat pengembangan proyek lebih cepat daripada menulis kode dari awal. Adapun alasan memakai Codeigneter meliputi [2]:

1. Sumber Terbuka, berlisensi *open source*, pengembang website dapat melakukan berbagai hal dengan CI.
2. Ringan, sistem inti CI membutuhkan sangat sedikit pustaka.
3. Cepat, kinerja CI terbukti cepat dibandingkan dengan kerangka kerja web lainnya.
4. MVC, dengan menggunakan Model-View-Controller, sistem web menjadi terpisah antara logika dan presentasi.
5. *Pretty URL*, URL yang dihasilkan oleh CI bersih dan ramah mesin pencari, karena CI menggunakan pendekatan berbasis segmen daripada *query string*.
6. Dapat diperluas, CI memungkinkan pengembang untuk menambahkan *library* dan *helper*.
7. Dokumentasi yang jelas, hampir semua fitur, *library*, dan *helper* di CI telah didokumentasikan dan diatur dengan baik.
8. Komunitas yang besar dan ramah, dengan bergabung dengan komunitas CI membantu pengembang dari pemula hingga mahir untuk berbagi pengetahuan, membuat pengembangan aplikasi web menjadi lebih mudah.
9. *Pack a Punch*, CI hadir dengan berbagai *library* yang akan membantu tugas pengembangan situs web yang umum digunakan seperti mengakses database, mengirim email, memvalidasi formulir, dll.

C. Text Preprocessing

Text preprocessing merupakan memproses data uji sebelum digunakan dalam program bertujuan untuk

mengurangi jumlah kosa kata, menyeragamkan kata dan menghilangkan *noise*. Adapun tahapan-tahapan *preprocessing* dalam meliputi [3]:

1. *Case folding* adalah proses mengubah huruf dalam dokumen menjadi huruf kecil. Dokumen mengandung beragam variasi bentuk huruf sampai tanda baca. Variasi huruf ini harus diseragamkan dan tanda bacanya harus dihilangkan. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter lainnya dihilangkan dan dianggap sebagai delimiter.
2. *Cleaning* yakni proses menghilangkan dokumen dari mention, hastag, link, emoticon, dan karakter lainnya yang tak berguna.
3. Normalisasi bahasa gaul adalah proses mengubah kata-kata tak lazim berupa kata-kata gaul menjadi kata-kata formal berbahasa Indonesia. Umumnya, tidak semua komentar pada suatu artikel menggunakan bahasa formal. Pembaca biasanya juga memakai bahasa gaul, seperti nggak, gue, loe, dll. Kata-kata yang tak lazim ini perlu diseragamkan melalui normalisasi bahasa menjadi bahasa formal berbahasa Indonesia.
4. *Stemming* adalah proses mengubah kata menjadi kata dasar. Pada umumnya kata dalam dokumen memiliki variasi kombinasi imbuhan kata yang beragam, seperti imbuhan awalan, akhiran, sisipan, dan kombinasi. Kata-kata tersebut perlu diseragamkan menjadi kata dasar supaya seragam dan mengurangi kompleksitas kata. Adapun algoritma stemming yang terkenal ialah algoritma Nazief dan Adriani. Algoritma ini dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan menjadi awalan (prefiks), sisipan (infiks), dan akhiran (suffiks) dan gabungan awalan-akhiran (confiks). Algoritma ini menggunakan kamus kata dasar dan mendukung *recording*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih.
5. *Stopword* yakni proses menghapus kata-kata yang tidak perlu dari dokumen. Pada tahap *stopword-list*, kata-kata yang tidak penting dibuang dari daftar kata, misalnya kata "yang", "dimana", "mengapa", "yaitu", "yakni", dan sebagainya.
6. *Tokenizing* adalah proses pemotongan dokumen menjadi kata-kata setelah menjadi proses *filtering*. Hasil pemotongan kata-kata tersebut dijadikan kumpulan kata dan membentuk daftar kata. Potongan tersebut dikenal dengan istilah token.
7. *N-gram* merupakan proses yang secara luas digunakan dalam pengolahan teks dan bahasa. N-gram merupakan kumpulan kata yang diberikan dalam sebuah paragraf dan ketika menghitung n-gram biasanya dilakukan dengan menggerakkan satu kata maju ke depan. N-gram memberikan probabilitas atau kemungkinan kata-kata berikutnya yang mungkin bisa digunakan dalam penggabungan pada keseluruhan kalimat. Adapun tipe n-gram yang biasanya digunakan seperti unigram, bigram, trigram. Jika diasumsikan X adalah jumlah kata dalam suatu kalimat K, maka jumlah n-gram dari kalimat K adalah $Ngram-K = X - (N-1)$ [4]. Misalnya, kalimat K "korupsi menyengsarakan masyarakat luas" memiliki 3

bigram yang terdiri atas: (1) korupsi menyengsarakan, (2) menyengsarakan masyarakat, (3) masyarakat luas.

D. Algoritma Stemming Nazief dan Adriani

Salah satu algoritma *stemming* bahasa Indonesia yang cukup terkenal ialah algoritma *stemming* yang dibuat oleh Bobby Nazief dan Mirna Adriani. Algoritma ini dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan menjadi *prefix* (awalan), *infix* (sisipan), dan *suffix* (akhiran) dan *prefix-suffix* (gabungan awalan-akhiran). Algoritma ini menggunakan kamus kata dasar dan mendukung *recording*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih. Adapun urutan langkah-langkah algoritma *stemming* Nazief dan Adriani dijelaskan sebagai berikut[5]:

1. Algoritma mengecek kata yang belum di-*stemming* di kamus kata dasar. Jika kata tersebut berhasil ditemukan, maka algoritma berhenti dan kata itu dianggap sebagai kata dasar. Tetapi jika algoritma tidak menemukan kata tersebut di kamus kata dasar, maka lanjut ke langkah selanjutnya.
 2. Algoritma menghapus *inflection suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, “-nya”). Jika penghapusan nya berhasil dan *suffixes* adalah partikel (“-lah” atau “-kah”), algoritma menghapus akhiran *possessive pronoun* (“-ku”, “-mu” atau “-nya”).
 3. *Derivation suffixes* (“-i” atau “-an”) dihapus. Jika penghapusan berhasil, algoritma berlanjut ke langkah keempat. Jika gagal, maka lakukan:
 - a. Jika *derivation suffixes* adalah “-an” dan karakter terakhir kata adalah “-k”, maka algoritma menghapus karakter “-k”. Kemudian lanjut ke langkah keempat. Tetapi jika gagal, maka lanjut ke langkah 3b.
 - b. Algoritma mengembalikan *suffixes* (“-i”, “-an”, “-kan”) yang terhapus, lalu lanjut ke langkah keempat.
 4. Algoritma menghapus *derivation prefix*, yang terdiri dari beberapa langkah meliputi:
 - a. Jika penghapusan *suffix* berhasil di langkah ketiga, algoritma memeriksa *prefix-suffix* yang tidak diizinkan. Jika algoritma menemukannya, algoritma selesai. Jika algoritma tidak menemukannya, maka lanjut ke langkah 4b.
 - b. Jika *prefix* mirip dengan *prefix* sebelumnya, algoritma selesai.
 - c. Jika penghapusan *derivation prefix* telah terjadi selama 3 kali, maka algoritma selesai.
 - d. Jika algoritma mengecek jenis *derivation prefix* dan menghapus *prefix*.
 - e. Jika kata dasar ditemukan, algoritma selesai. Sebaliknya, langkah keempat diulangi kembali untuk menghapus *prefix* kedua.
 - f. Algoritma melakukan proses *recording*, bergantung pada jenis *prefix*.
 5. Jika semua langkah di atas gagal, algoritma berhenti dan mengembalikan kata awalnya dan proses selesai.
- Algoritma *stemming* Nazief dan Adriani dikembangkan kembali oleh Asian Jelita dijelaskan sebagai berikut [6]:
1. Memakai kamus kata dasar yang lebih lengkap.
 2. Menambah aturan untuk penanganan kata jamak, misalnya kata “buku-buku”, maka *stemming* dikembalikan menjadi “buku”. Namun kehati-hatian harus diibekikan pada beberapa kata dengan tanda hubung “-” seperti “bolak-balik”, “berbalas-balasan”, “seolah-olah”. Sebagai contohnya, kata “berbalas-balasan” memiliki kata dasar “balas”, sehingga *stemming* bisa dilakukan dengan mengembalikan kata awal sebelum tanda hubung atau kata akhir setelah tanda hubung. Sebaliknya, kata “bolak-balik” tidak memiliki kata dasar yang sama, sehingga *stemming* dikembalikan sebagai kata yang sama sebagai kata “bolak-balik”.
 3. Menambah *prefix* dan *suffix* dan penambahan aturan sebagai berikut:
 - a. Penambahan partikel (*inflection suffix*) “pun”, seperti untuk kata “siapa pun”.
 - b. Untuk tipe *prefix* “ter-” dimodifikasi kondisinya, sehingga baris 4 dalam tabel 2.2 diatur tipe ke “ter-”, alih-alih “none”. Aturan ini mendukung kasus seperti kata “terpercaya” yang memiliki kata dasar “percaya”.
 - c. Untuk *prefix* tipe “pe-” dimodifikasi kondisinya sehingga kata “pekerja” dan “peserta” memiliki *prefix* bertipe “pe-”.
 - d. Untuk *prefix* bertipe “mem-” dimodifikasi kondisinya sehingga kata dengan permulaan *prefix* “memp-” adalah bertipe “mem-”.
 - e. Untuk *prefix* bertipe “meng-” dimodifikasi kondisinya sehingga kata dengan permulaan *prefix* “meng-” adalah “meng-”.
 4. Menyesuaikan prioritas aturan:
 - a. Jika kata memiliki *prefix* dengan “ber-” dan *suffix* dengan *inflection suffix* “-lah”, diusahakan untuk dihapus *prefix*-nya sebelum *suffix*, seperti untuk kata “bermasalah” dimana kata dasarnya adalah “masalah”.
 - b. Jika kata memiliki *prefix* dengan “ber-” dan *suffix* dengan *derivation suffix* “-an”, diusahakan untuk dihapus *prefix* sebelum *suffix*, seperti untuk kata “berbadan” dimana kata dasarnya adalah “badan”.
 - c. Jika kata memiliki *prefix* dengan “men-” dan *suffix* dengan *derivation suffix* “-i”, diusahakan untuk dihapus *prefix* sebelum *suffix*, seperti untuk kata “menilai” dimana kata dasarnya adalah “nilai”.
 - d. Jika kata memiliki *prefix* dengan “di-” dan *suffix* dengan *derivation suffix* “-i”, diusahakan untuk dihapus *prefix* sebelum *suffix*, seperti untuk kata “dimulai” dimana kata dasarnya adalah “mulai”.
 - e. Jika kata memiliki *prefix* dengan “pe-” dan *suffix* dengan *derivation suffix* “-i”, diusahakan untuk dihapus *prefix* sebelum *suffix*, seperti untuk kata “petani” dimana kata dasarnya adalah “tani”.
 - f. Jika kata memiliki *prefix* dengan “ter-” dan *suffix* dengan “-i”, diusahakan untuk dihapus *prefix* sebelum *suffix*, seperti untuk kata “terkendali” dimana kata dasarnya adalah “kendali”.

E. Multinomial Naive Bayes

Multinomial Naive Bayes (MNB) merupakan variasi dari metode pembelajaran *Naive Bayes* yang digunakan untuk mengklasifikasikan sekumpulan dokumen. MNB merupakan

metode *supervised learning*. Sehingga setiap data perlu diberikan label sebelum dilakukan pelatihan data. Metode pembelajaran yang membangkitkan fungsi untuk memetakan input ke output yang diinginkan. Dengan kata lain, kualitas hasil pembelajaran bergantung pada kesesuaian antara input dan output yang diberikan user. Dengan demikian, user sangat berperan dalam memvalidasi input dan output tersebut.

Metode MNB menempuh dua tahap yakni pelatihan data dan pengujian (klasifikasi) data. Pada tahap pelatihan data dilakukan proses analisis terhadap sampel dokumen berupa pemilihan *vocabulary* yakni kata yang mungkin muncul dalam koleksi dokumen sampel yang sedapat mungkin merepresentasikan dokumen. Selanjutnya penentuan probabilitas *prior* bagi tiap sampel dokumen. Pada tahap pengujian ditentukan nilai kategori dari suatu dokumen berdasarkan *term* yang muncul dalam dokumen yang diklasifikasikan. Klasifikasi dokumen bertujuan untuk menentukan kelas terbaik untuk suatu dokumen. Kelas terbaik dalam MNB ditentukan dengan mencari maximum aposteriori (C_{map}) kelas C melalui persamaan:

$$C_{\text{map}} = \operatorname{argmax}_C P(C) \prod_k P(t_k | C) \quad (1)$$

Dimana $P(C)$ merupakan probabilitas *prior* dari kelas C. Sementara $P(t_k | C)$ merupakan probabilitas fitur-fitur kata dari kelas C. Pada persamaan di atas, terdapat banyak probabilitas *posterior* atau probabilitas bersyarat yang dikalikan. Hal ini dapat menyebabkan *floating point underflow*. Masalah ini timbul pada dokumen yang memiliki kata yang sangat besar. Hasil perkalian probabilitas *posterior* dari seluruh fitur kata membuat nilai C_{map} sangat kecil sekali, sehingga menimbulkan kesalahan saat dilakukan proses perbandingan. Karena itu, proses perhitungan akan lebih baik jika dilakukan penjumlahan pada logaritma dari probabilitas. Kelas dengan logaritma tertinggi merupakan kelas dengan probabilitas terbaik untuk dokumen $\log(xy) = \log(x) + \log(y)$. Persamaan yang menggunakan logaritma dari probabilitas dapat dinyatakan dalam persamaan:

$$C_{\text{map}} = \operatorname{argmax}_C [\log P(C) + \sum_{1 \leq k \leq n} \log P(t_k | C)] \quad (2)$$

$P(t_k | C)$ merupakan probabilitas *posterior* dari kelas C. Sementara $P(C)$ merupakan probabilitas *prior* dari kelas C. Probabilitas *prior* $P(C)$ dapat dihitung dengan persamaan:

$$P(C) = \frac{N_C}{N} \quad (3)$$

Berikut penjelasan dari rumus di atas:

1. $P(C)$ merupakan probabilitas *prior* suatu dokumen yang berada di kelas c.
2. N_C merupakan jumlah dokumen dengan kelas c.
3. N merupakan jumlah seluruh dokumen.

Sementara $P(t_k | C)$ merupakan probabilitas *posterior* pada fitur kata t yang berada pada kelas c. $P(t_k | C)$ bisa didapatkan dengan menghitung melalui persamaan:

$$P(t_k | C) = \frac{n_k}{n} \quad (4)$$

Berikut penjelasan dari rumus di atas:

1. $P(t_k | C)$ merupakan probabilitas *posterior* pada fitur kata t dari kelas c.
2. n_k merupakan jumlah kemunculan fitur kata t pada semua dokumen yang memiliki kelas c.
 n merupakan jumlah fitur kata pada kelas c.
Perhitungan *maximum likelihood* memiliki kelemahan

yakni suatu kata dalam kelas yang tidak terlihat pada data pelatihan akan memiliki pembagian dengan nilai 0 (*division by zero*). Untuk mengatasi masalah ini diterapkan teknik *laplace smoothing*, sehingga persamaannya menjadi:

$$P(t_k | C) = \frac{n_k + 1}{n + | \text{kata} |} \quad (5)$$

Dimana, $| \text{kata} |$ merupakan jumlah kata yang unik yang muncul pada seluruh dokumen. Dengan kata lain, $| \text{kata} |$ merupakan total fitur kata [7].

F. Teknik K-Fold Cross-Validation

K-fold cross-validation merupakan teknik validasi model yang berguna untuk mengevaluasi kinerja model dimana *dataset* D dibagi dipisahkan menjadi sub himpunan data (*fold*) sejumlah k sub himpunan. Sehingga masing-masing *fold* berisi $1/k$ bagian *dataset*. Selanjutnya *dataset* D_1, D_2, \dots, D_n berisi masing-masing (k-1) *fold* untuk data latih dan 1 *fold* untuk data uji. Dengan demikian, setiap data akan dijadikan sebagai data latih dan data uji sebanyak 1 kali.

Misalnya, dengan $k=4$, maka didapatkan himpunan data D_1 berisi 3 *fold* yakni f_2, f_3, f_4 untuk data latih dan f_1 untuk data uji. Himpunan data D_2 berisi 3 *fold* yakni f_1, f_3, f_4 untuk data latih dan f_2 untuk data uji. Demikian seterusnya untuk himpunan data D_3 dan D_4 . Sehingga setiap *fold* pernah menjadi data uji sebanyak satu kali.

Dengan metode *K-Fold Cross-Validation* bisa mengukur kualitas semua model klasifikasi yang dibangun. Dengan metode ini dapat juga membandingkan sejumlah model klasifikasi. Selain itu metode ini dapat digunakan untuk menyeleksi dan memilih model yang terbaik di antara semua model yang dibangun [8]. Dalam tugas akhir ini, *4-Fold Cross-Validation* digunakan dalam validasi model sehingga model akan dilatih dan diuji sebanyak 4 kali.

G. Ukuran Evaluasi Model Klasifikasi

Untuk mengevaluasi suatu model klasifikasi diperlukan ukuran tertentu. Ukuran tersebut meliputi *accuracy*, *error rate*, *recall*, *specificity*, *precision*, *f-score*. Parameter *accuracy* and *error rate* sebaiknya digunakan untuk *dataset* dengan proporsi jumlah kelas yang berimbang. Adapun parameter *recall*, *specificity*, *precision*, *f-score* sebaiknya digunakan untuk *dataset* dengan proporsi jumlah kelas yang tidak berimbang. Ukuran-ukuran tersebut dapat dihitung dengan *confusion matrix* yang diilustrasikan pada tabel 2.1:

TABEL I
CONFUSION MATRIX

	Prediksi Salah	Prediksi Benar
Aktual Salah	True negative (TN)	False positive (FP)
Aktual Benar	False negative (FN)	True positive (TP)

Berikut keterangan penjelasan dari *confusion matrix*:

- TP atau *true positive*, yakni jumlah tuple positif yang dilabeli dengan benar oleh model klasifikasi. Yang dimaksud tuple positif ialah tuple aktual yang berlabel positif, seperti kategori komentar *cyberbullying*.
- TN = *true negative*, yakni jumlah tuple negatif yang dilabeli dengan benar oleh model klasifikasi. Yang dimaksud tuple negatif ialah tuple aktual yang berlabel negatif, seperti kategori komentar *non-cyberbullying*.

- FP = *false positive*, yakni jumlah tuple negatif yang salah dilabeli oleh model klasifikasi. Misalnya, kategori komentar *non-cyberbullying*, tetapi oleh model klasifikasi dilabeli *cyberbullying*.
- FN = *false negative*, yakni jumlah tuple positif yang salah dilabeli oleh model klasifikasi. Misalnya, kategori komentar *cyberbullying*, tetapi oleh model klasifikasi dilabeli *non-cyberbullying*.

Berikut ini dijelaskan masing-masing ukuran evaluasi klasifikasi meliputi [9]:

1. *Accuracy* atau tingkat pengenalan menyatakan persentase dari jumlah tuple dalam data uji yang diklasifikasikan dengan benar oleh model klasifikasi. *Accuracy* dihitung dengan rumus: $(TP+TN)/(TP+TN+FP+FN)$.
2. *Error rate* menyatakan tingkat persentase kesalahan atau kekeliruan dari jumlah tuple dalam data uji yang diklasifikasikan oleh model klasifikasi. *Error rate* dihitung dengan rumus: $(FP+FN)/(TP+TN+FP+FN)$.
3. *Recall* adalah ukuran kelengkapan yakni berapa persentase tuple positif yang dilabeli sebagai positif. *Recall* dihitung dengan rumus: $TP/(TP+FN)$.
4. *Specificity* atau *true negative rate* dihitung dengan rumus: $TN/(FP+TN)$.
5. *Precision* atau ukuran kepastian, yakni berapa persentase tuple yang dilabeli sebagai positif adalah benar pada kenyataannya, yang dirumuskan sebagai: $TP/(TP+FP)$.
6. *F-Score* atau rata-rata harmonik dari *precision* dan *recall* dihitung dengan rumus: $(2*precision*recall)/(precision+recall)$.

H. Ukuran Evaluasi Situs Web

Mengevaluasi situs web berguna untuk mengukur kualitas suatu situs web. Salah satu dalam mengevaluasi situs web ialah *Web Quality Evaluation Method* (WebQEM). Olsina dan Rossin (2002) mengidentifikasi terdapat empat aspek dalam mengevaluasi suatu *website* meliputi *functionality* (fungsionalitas), *reliability* (kehandalan), *usability* (kegunaan), dan *efficiency* (efisiensi). Berikut dijelaskan aspek-aspek dari WebQEM sebagai berikut [10]:

1. Aspek kegunaan adalah tingkat kualitas dari sistem yang mudah dipelajari, mudah digunakan dan mendorong pengguna untuk menggunakan sistem sebagai alat bantu positif dalam menyelesaikan tugas. Aspek kegunaan dapat diukur dengan mengukur kemudahan pengguna dalam mempelajari tampilan antar muka (*easy to use*). Berikutnya, aspek kegunaan dapat diukur dengan menilai seberapa menarik tampilan antar muka bagi pengguna (*user friendly*). Terakhir, aspek kegunaan dapat diukur dengan menilai seberapa responsif tampilan antar muka baik di layar PC maupun mobile (*responsible*).
2. Aspek kehandalan adalah peluang sebuah komponen, sub-sistem atau sistem melakukan fungsinya dengan baik, seperti yang dipersyaratkan, dalam kurun waktu tertentu dan dalam kondisi operasi tertentu pula. Kehandalan dapat diukur dengan menguji seberapa baik sistem memberikan hasil yang akurat, tanpa kegagalan. Selain itu, kehandalan juga dapat diukur dari penanganan terjadinya kesalahan dan pemulihan dari kegagalan sistem. Kehandalan juga dapat diukur dengan mengukur tingkat ketersediaan konten yang dibutuhkan pengguna (*availability*).
3. Aspek fungsionalitas adalah kemampuan sistem dalam menjalankan tugas-tugas atau fungsi utamanya. Suatu fitur

dapat dikatakan berfungsi dengan baik tatkala masukan dan keluarannya sesuai. Dengan kata lain, fungsionalitas merupakan kesesuaian antara masukan dan keluaran yang direncanakan. Oleh karena itu, aspek fungsionalitas dapat diukur dengan menilai seberapa baik fitur-fitur dalam suatu situs dapat bekerja secara baik.

4. Aspek efisiensi adalah kemampuan menjalankan tugas dengan baik dan tepat tanpa membuang waktu, tenaga maupun biaya. Efisiensi berhubungan dengan respon, waktu pemrosesan, dan pemanfaatan sumber daya yang digunakan sistem. Jika suatu perangkat lunak memerlukan respon, waktu pemrosesan dan sumber daya yang banyak maka dapat dikatakan perangkat lunak itu tidak efisien, berlaku sebaliknya. Adapun aspek efisiensi dapat diukur dengan mengukur seberapa cepat suatu situs dalam memproses konten-kontennya (*time behaviour*). Selain itu, efisiensi juga dapat diukur dengan menilai kemudahan dalam mengakses fitur-fitur atau layanan-layanan dalam suatu situs web (*accessibility*).

I. Tool dan Library

Tool dan *Library* yang dipakai dalam pengembangan CMS pada tugas akhir ini meliputi:

1. MySQL adalah layanan database yang dikelola sepenuhnya untuk menyebarkan aplikasi *cloud-native* menggunakan database *open source* terpopuler di dunia [11].
2. JQuery adalah pustaka JavaScript yang dirancang untuk menyederhanakan traversal dan manipulasi pohon DOM HTML, serta penanganan *event*, animasi CSS, dan Ajax [12].
3. Bootstrap adalah kerangka kerja CSS sumber terbuka yang diarahkan pada pengembangan web front-end yang responsif dan mengutamakan seluler [13].
4. DataTables adalah *plug-in* jQuery open-source untuk membuat tabel HTML dinamis [14].
5. Chartjs adalah pustaka JavaScript sumber terbuka untuk visualisasi data, yang mendukung 8 jenis bagan: batang, garis, area, pai, gelembung, radar, kutub, dan pencar [15].
6. TinyMCE adalah editor teks kaya online yang dirilis sebagai perangkat lunak sumber terbuka di bawah LGPL yang memiliki kemampuan untuk mengubah bidang *textarea* HTML atau elemen HTML lainnya menjadi contoh editor [16].
7. Ion Auth adalah pustaka CodeIgniter yang dikembangkan oleh Ben Edmunds untuk otentikasi sederhana dan ringan [17].

III. METODE

Perancangan sistem ini menggunakan metodologi *iterative waterfall* yang terdiri atas empat tahapan meliputi:

1. Analisis sistem yakni tahapan analisis terhadap kebutuhan sistem. Untuk itu, diperlukan sejumlah literatur terkait berupa jurnal dan buku-buku yang relevan terkait pengembangan sistem manajemen konten dan analisis sentimen *cyberbullying* guna

mendapat informasi terkait kebutuhan, fitur dan batasan dalam pengembangan sistem.

2. Desain sistem yakni tahapan desain sistem berupa perancangan arsitektur, database, interface, dan desain prosedural sesuai dengan kebutuhan dan fitur-fitur yang akan dikembangkan.
3. Implementasi yakni tahapan implementasi sistem dalam terhadap modul-modul yang dikembangkan dengan cara pemograman. Pengembangan CMS dalam tugas akhir ini menggunakan bahasa PHP menggunakan *framework* CodeIgniter.
4. Pengujian yakni tahapan pengujian terhadap sistem baik berupa pengujian fungsional maupun non fungsional, guna mengetahui bahwa sistem yang dikembangkan dapat berjalan secara baik. Adapun pengujian fungsional sistem menggunakan *black box testing*. Sedangkan pengujian non fungsionalitas meliputi *usability*, *functionality*, *realibility*, dan *efficiency* dilakukan melalui kuesioner dimana sejumlah 39 responden diminta menjawab kuesioner dengan jawaban likert skala 1 – 5 untuk mengetahui respon pengguna terhadap kualitas sistem yang dikembangkan. Sedangkan untuk mengetahui tingkat keakuratan sistem dalam mengklasifikasikan kategori komentar digunakan teknik *validasi 4-fold cross-validation* dengan $k=4$, maka didapatkan himpunan data D_1 berisi 3 *fold* yakni f_2 , f_3 , f_4 untuk data latih dan f_1 untuk data uji. Himpunan data D_2 berisi 3 *fold* yakni f_1 , f_3 , f_4 untuk data latih dan f_2 untuk data uji. Demikian seterusnya untuk himpunan data D_3 dan D_4 . Sehingga setiap *fold* pernah menjadi data uji sebanyak satu kali. Adapun model akan dievaluasi menggunakan parameter *accuracy and error rate*.

IV. HASIL EKSPERIMEN DAN PENELITIAN

Pada Bab IV dijelaskan tentang hasil eksperimen dan penelitian. Adapun penjelasannya meliputi:

A. Fitur-Fitur Sistem

Sistem CMS dalam tugas akhir ini memiliki beberapa fitur yang terbagi atas fitur *front end* dan fitur *back end*. Bagian *front end* berinteraksi langsung dengan pengunjung di depan *website*. Sistem ini memiliki 3 level hak akses meliputi pembaca, member, dan admin dimana masing-masing user memiliki hak akses berbeda. Hal ini ditunjukkan tabel I:

TABEL II
FITUR-FITUR FRONT END

Fitur	Pembaca	Member	Admin
Front End			
1. Registrasi	V	X	X
2. Login	X	V	V
3. Artikel	V	V	V
4. Komentar artikel	X	V	V
5. Registrasi	V	X	X
6. Login	X	V	V
7. Artikel	V	V	V
8. Komentar artikel	X	V	V
9. Rating artikel	X	V	V
10. Statistik artikel	X	V	V
11. Kategori artikel	V	V	V

TABEL II
(LANJUTAN)

Fitur	Pembaca	Member	Admin
12. Following	X	V	V
13. Peringatan <i>cyberbullying</i>	X	V	V
14. Pelaporan Komentar <i>cyberbullying</i>	X	V	V
15. Tag Terpopuler	V	V	V
16. Artikel Terpopuler	V	V	V
17. Berbagi ke sosial media	V	V	V
18. Tentang web	V	V	V
19. Kebijakan web	V	V	V
20. Evaluasi web	X	V	V
21. Profil	V	V	V

Selanjutnya CMS juga memiliki fitur *backend*. Pada bagian *back end* terdapat 2 level hak akses yakni member dan admin. Bagian *back end* bekerja di belakang *website* untuk mengelola konten *blog*. Pada bagian *backend*, pembaca tidak bisa mengaksesnya. Selengkapnya ditunjukkan pada tabel II di bawah ini:

TABEL III
FITUR-FITUR BACK END

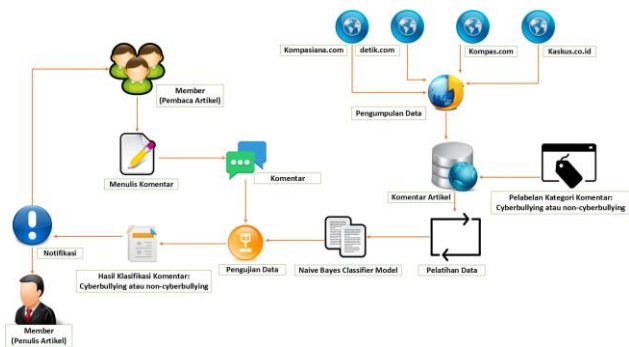
Fitur	Pembaca	Member	Admin
Back End			
1. Konfigurasi profil	X	V	V
2. Manajemen user	X	X	V
3. Notifikasi	X	V	V
4. Mengelola artikel	X	V	V
5. Mengelola follower-following	X	V	V
6. Mengelola rating	X	V	V
7. Mengelola komentar	X	V	V
8. Konfigurasi profil	X	V	V
9. Manajemen user	X	X	V
10. Notifikasi	X	V	V
11. Mengelola artikel	X	V	V
12. Mengelola follower-following	X	V	V
13. Mengelola rating	X	V	V
14. Mengelola komentar	X	V	V
15. Manajemen penyaring komentar <i>cyberbullying</i>	X	V	V
16. Mengelola kategori	X	X	V
17. Manajemen galeri	X	V	V
18. Manajemen tag	X	X	V
19. Manajemen tentang web	X	X	V
20. Manajemen kebijakan	X	X	V
21. Manajemen evaluasi web	X	X	V

B. Arsitektur Penyaring Komentar Cyberbullying

Berikut dijelaskan arsitektur penyaring komentar *cyberbullying* beserta cara kerjanya. Pertama, *dataset* komentar diambil secara manual dari berbagai sumber data seperti detik.com, Kompas.com, dll. Kedua, *dataset* komentar disimpan ke database. Ketiga, *dataset* komentar secara manual diberi label sentimen. Keempat, *dataset* komentar dibagi menjadi *dataset* training dan *dataset* test. Kelima, *dataset* pelatihan dilatih untuk menghasilkan *Multinomial Naive Bayes Model*. Model tersebut digunakan untuk menyaring komentar dari pembaca yang mengirimkan komentar terkait konten *blog*. Kenam, model akan mendeteksi komentar yang mengandung *cyberbullying*, kemudian sistem mengirimkan notifikasi kepada pembuat artikel dan pembaca artikel tentang peringatan *cyberbullying*.

C. Arsitektur Pendeteksi Cyberbullying

Berikut dijelaskan arsitektur pendeteksi *cyberbullying* beserta cara kerjanya. Pertama, *dataset* pelatihan dilakukan dengan *preprocessing* meliputi *case folding*, *cleaning*, normalisasi bahasa gaul, *stemming*, *stopword*, dan *tokenizing*. Kedua, *dataset* dilatih dalam tiga tahap yaitu membuat fitur kata, menghitung probabilitas prior, dan menghitung probabilitas fitur kata. Ketiga, membuat fitur kata melibatkan pemisahan kata-kata unik dari *dataset* komentar untuk menghasilkan daftar kata unik dalam model kata. Keempat, menghitung probabilitas prior melibatkan pembentukan probabilitas prior yaitu proporsi antara jumlah *cyberbullying* dengan jumlah *non-cyberbullying*. Kelima, menghitung probabilitas fitur kata melibatkan pembentukan probabilitas setiap fitur kata untuk menghasilkan daftar kata unik dengan probabilitas *cyberbullying* dan probabilitas *non-cyberbullying*. Keenam, dari hasil data latih akan dihasilkan *Multinomial Naive Bayes Model*. Ketujuh, member menulis



Gambar. 1. Arsitektur Penyaring Komentar Cyberbullying

komentar dan kemudian komentar tersebut dikirim ke sistem. Kedelapan, sistem akan melakukan *preprocessing* meliputi *case folding*, *cleaning*, normalisasi bahasa gaul, *stemming*, *stopword*, dan *tokenizing* pada komentar dari member. Kesembilan, komentar akan diuji sentimennya oleh sistem dengan menghitung Vmap dan menentukan sentimen komentar tersebut. Kesepuluh, dalam penghitungan Vmap, semua kata komentar akan diberi bobot untuk menghasilkan sentimen *cyberbullying* dan *non-cyberbullying* sentimen. Kesebelas, jika bobot sentimen *cyberbullying* lebih besar dari bobot sentimen *non cyberbullying*, maka dapat disimpulkan bahwa komentar tersebut termasuk dalam *cyberbullying*. Jika bobot sentimen *cyberbullying* tidak lebih besar dari bobot sentimen *non-cyberbullying* maka dapat disimpulkan bahwa

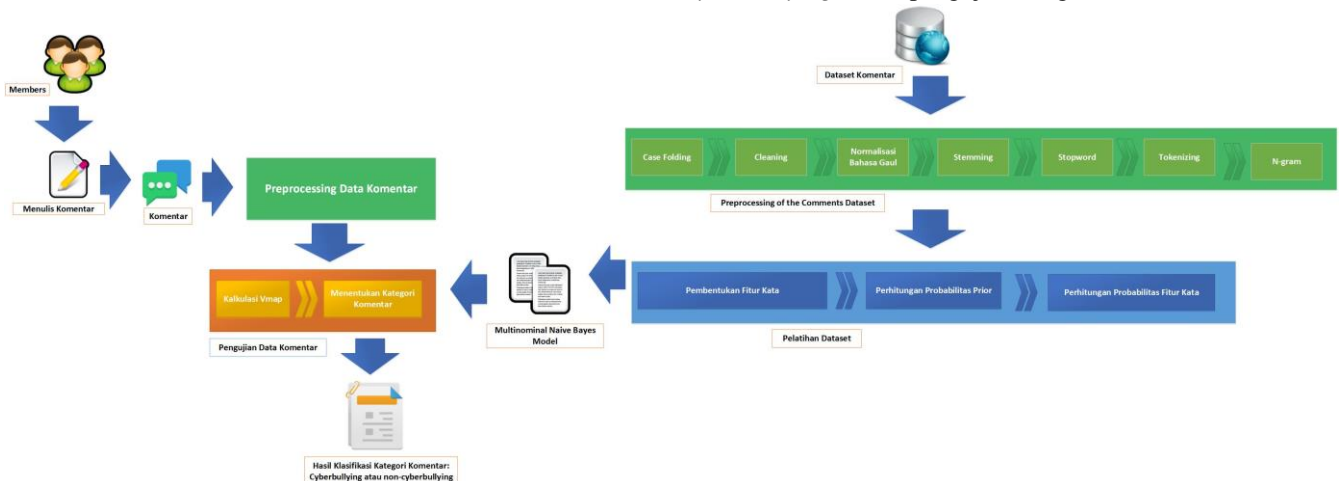
komentar tersebut adalah *non-cyberbullying*. Terakhir, hasilnya adalah kategori komentar *cyberbullying* atau kategori komentar *non-cyberbullying*.

D. Spesifikasi dan Vertifikasi Dataset

Berikut dijelaskan spesifikasi *dataset* yang digunakan dalam pengembangan sistem penyaring komentar *cyberbullying*. Pertama, topik artikel terdiri dari sosial budaya, politik, dan olahraga. Ketiga topik tersebut dipilih karena biasanya merupakan diskusi sensitif dan rawan *cyberbullying* di kalangan pembacanya. Kedua, artikel terdiri dari 508 artikel dari berbagai sumber. Ketiga, komentar terdiri dari 6.537 komentar. Komentar diberi label secara manual. Keempat, kategori sentimen terdiri dari dua kategori komentar yaitu kategori *cyberbullying* dan kategori *non-cyberbullying*. Kelima, kategori komentar *cyberbullying* terdiri dari 3.942 komentar dan kategori komentar *non-cyberbullying* terdiri dari 2.595 komentar. Keenam, sumber datanya antara lain detik.com, kompas.com, kaskus.com, kompasiana.com, akun instagram berita indonesia seperti vivanews, tempo, dan liputan6. Ketujuh, verifikasi *dataset* komentar melibatkan verifikasi internal dan verifikasi eksternal. Validasi internal merupakan pemeriksaan kembali yang dilakukan penulis terhadap komentar-komentar yang telah dilabeli terkait kandungan kategori *cyberbullying* beserta jenis *cyberbullying*-nya. Sementara validasi eksternal adalah pemeriksaan kategori komentar yang dilakukan oleh pihak-pihak eksternal yang bertugas sebagai penguji atau validator terhadap komentar-komentar yang telah dilabeli kategorinya. Validasi eksternal melibatkan 3 orang penguji.

E. Pengujian

Pengujian fungsionalitas menggunakan *black box* terhadap fitur-fitur sistem. Pengujian ini dilakukan dengan kesesuaian antara reaksi sistem terhadap inputan. Jika sistem merespon sesuai dengan inputan maka pengujian berhasil, berlaku sebaliknya. Adapun fitur-fitur yang diuji meliputi: fitur login dan registrasi, fitur konfigurasi akun, fitur manajemen galeri, fitur manajemen artikel, fitur manajemen komentar, fitur manajemen *following-follower*, fitur notifikasi, fitur konfigurasi *website*, fitur tentang *website*, fitur manajemen kebijakan *website*, fitur manajemen FAQ, fitur manajemen tagar, fitur manajemen user, fitur manajemen kategori, fitur evaluasi *website*, fitur *dashbord*, dan fitur penyaring *cyberbullying*. Hasil pengujian fungsionalitas ialah fitur-fitur



Gambar. 2. Arsitektur Pendeteksi Cyberbullying

dapat berjalan dengan baik sesuai dengan inputan yang direncanakan.

Pengujian terhadap model penyaring komentar *cyberbullying* dilakukan dengan *4-k fold cross validation* dengan memberikan 4 perlakuan yang berbeda pada setiap model yang dikembangkan. Adapun perlakuan terhadap model tersebut meliputi: model dilatih dengan *dataset* komentar yang telah mengalami *preprocessing stemming* dan menggunakan fitur *bigram*, model dilatih dengan *dataset* komentar yang telah mengalami *preprocessing* tanpa melalui *stemming* dan menggunakan fitur *bigram*, model dilatih dengan *dataset* komentar yang mengalami *preprocessing stemming* dan menggunakan fitur *unigram*, model dilatih dengan *dataset* komentar yang mengalami *preprocessing* tanpa melalui *stemming* dan menggunakan fitur *unigram*. Setelah dilakukan pengujian terhadap model, maka diketahui hasil pengujian model sebagai berikut:

TABEL IV
HASIL EVALUASI MODEL DENGAN FITUR STEMMING-BIGRAM

Aspek	Hasil
Total <i>dataset</i>	6537
TP	3436
TN	916
FP	1679
FN	506
Akurasi	67%
Error	33%
Recall	87%
Precision	67%
F-score	76%

Pada tabel IV merupakan hasil evaluasi model menggunakan fitur *stemming* dan fitur *bigram*. Berdasarkan tabel tersebut, diketahui bahwa penggunaan fitur *stemming* dan fitur *bigram* mampu memberikan tingkat akurasi sebesar 67% dan tingkat *error* sebesar 33%. Adapun tingkat *recall*, *precision*, dan *f-score* masing-masing sebesar 87%, 67%, 76%.

TABEL V
HASIL EVALUASI MODEL DENGAN FITUR BIGRAM TANPA STEMMING

Aspek	Hasil
Total <i>dataset</i>	6537
TP	3489
TN	837
FP	1758
FN	453
Akurasi	66%
Error	34%
Recall	89%
Precision	66%
F-score	76%

Pada tabel V merupakan hasil evaluasi model tanpa fitur *stemming* dan menggunakan fitur *bigram*. Berdasarkan tabel tersebut, diketahui bahwa penggunaan fitur *bigram* tanpa melalui *stemming* hanya mampu memberikan tingkat akurasi sebesar 66% dan tingkat *error* sebesar 34%. Adapun tingkat *recall*, *precision*, dan *f-score* masing-masing sebesar 89%, 66%, 76%.

TABEL VI
HASIL EVALUASI MODEL DENGAN FITUR STEMMING-UNIGRAM

Aspek	Hasil
Total <i>dataset</i>	6537
TP	3464
TN	1324

TABEL VI
(LANJUTAN)

Aspek	Hasil
FP	1271
FN	478
Akurasi	73%
Error	27%
Recall	88%
Precision	73%
F-score	80%

Pada tabel VI merupakan hasil evaluasi model menggunakan fitur *stemming* dan menggunakan fitur *unigram*. Berdasarkan tabel tersebut, diketahui bahwa penggunaan fitur *stemming* dan *unigram* mampu memberikan tingkat akurasi sebesar 73% dan tingkat *error* sebesar 27%. Adapun tingkat *recall*, *precision*, dan *f-score* masing-masing sebesar 88%, 73%, 80%.

TABEL VII
HASIL EVALUASI MODEL DENGAN TANPA FITUR STEMMING DAN UNIGRAM

Aspek	Hasil
Total <i>dataset</i>	6537
TP	2991
TN	1245
FP	1350
FN	449
Akurasi	70%
Error	30%
Recall	87%
Precision	69%
F-score	77%

Pada tabel VII merupakan hasil evaluasi model menggunakan fitur *unigram* tanpa melalui fitur *stemming*. Berdasarkan tabel tersebut, diketahui bahwa tanpa penggunaan fitur *stemming* dan hanya menggunakan fitur *unigram*, model hanya mampu memberikan tingkat akurasi sebesar 70% dan tingkat *error* sebesar 30%. Adapun tingkat *recall*, *precision*, dan *f-score* masing-masing sebesar 87%, 69%, 77%.

TABEL VIII
PERBANDINGAN UKURAN FITUR KATA ANTAR MODEL

Aspek	Hasil
Model01- <i>Stemming-Unigram</i>	4.910
Model02- <i>TanpaStemming-Unigram</i>	7.295
Model03- <i>Stemming-Bigram</i>	22.613
Model04- <i>TanpaStemming-Bigram</i>	23.787

Selanjutnya dijelaskan perbandingan antar model berkaitan dengan ukuran fitur kata dari masing-masing model yang dikembangkan untuk mencari tahu model dengan ukuran fitur kata terkecil atau efisien secara ukurannya. Berdasarkan tabel VIII dapat diketahui bahwa penggunaan fitur *stemming* dan fitur *unigram* dalam pembuatan model klasifikasi dapat memperkecil ukuran fitur kata. Sebaliknya penggunaan fitur *bigram* tanpa menggunakan fitur *stemming* dapat memperbesar ukuran fitur kata. Model dengan fitur *stemming* dan penggunaan *unigram* dalam pembuatan model hanya memberikan ukuran fitur kata sebesar 4.910 kata. Sedangkan penggunaan fitur *bigram* tanpa menggunakan fitur *stemming* dalam pembuatan model memberikan ukuran fitur kata sebesar 23.787 kata. Hal ini berarti *model01-stemming-unigram* hanya sebesar 21% dari ukuran fitur kata *model04-TanpaStemming-Bigram*. Dengan kata lain, model

paling efisien secara ukuran fitur kata ialah model dengan penggunaan fitur *stemming-unigram*.

Berikutnya dijelaskan hasil pengujian non-fungsional terhadap respon dari sejumlah 39 responden terkait kualitas-kualitas *website*. pengujian dilakukan dengan memberikan kuesioner dengan jumlah pertanyaan sebanyak 25 pertanyaan yang masing-masing mewakili aspek-aspek kualitas *website* meliputi *usability*, *reliability*, *functionality*, dan *efficiency* dengan menggunakan skala likert 1 – 5. Adapun rincian demografi responden berdasarkan pekerjaan terdiri atas 44% mahasiswa (17 orang), 18% pegawai swasta (7 orang), 15% freelance (6 orang), 3% PNS (1 orang), 5% wiraswasta (2 orang), dan 16% lain-lain (6 orang) dari total 39 responden. Sementara demografi responden berdasarkan gender terdiri atas 59% wanita (23 orang) dan 41% pria (16 orang) dari total 39 responden. Sementara untuk mengukur predikat kualitas *website* dari masing-masing aspeknya digunakan persamaan:

$$\text{Rata - Rata Aspek Website} = \frac{\text{Total Nilai Jawaban}}{\text{Jumlah Jawaban}} \quad (7)$$

Misalnya, diketahui aspek *usability* memiliki total nilai jawaban sebesar 545 dari seluruh pertanyaan terkait *usability* yang telah dijawab oleh 39 responden, dimana jumlah jawabannya sebesar 156 jawaban. Sehingga rata-rata nilai aspek *usability* dari *website* sebesar: $545 / 156 = 3,49$. Selanjutnya hasil rata-rata nilai tersebut digolongkan sesuai dengan skala predikat ukuran kualitas *website*, sehingga dapat diketahui bahwa *usability*-nya berpredikat cukup. Berikut ini merupakan skala predikat ukuran kualitas *website*:

TABEL XI
SKALA PREDIKAT UKURAN KUALITAS WEBSITE

Predikat	Skala Likert 1 - 5
Sangat baik	5
Baik	4 – 4,99
Cukup	3 – 3,99
Buruk	2 – 2,99
Sangat buruk	<2

Berdasarkan hasil survei kepada 39 responden didapat hasil bahwa: pertama, *website* memiliki kualitas tingkat *usability* berpredikat cukup baik dengan rata-rata skor 3,49 dari skala 5. Kedua, *website* memiliki tingkat *reliability* berpredikat baik dengan rata-rata skor 4,28 dari skala 5. Ketiga, *website* memiliki tingkat *functionality* berpredikat baik dengan rata-rata skor 4,11 dari skala 5. Keempat, *website* memiliki tingkat *efficiency* berpredikat cukup dengan rata-rata skor 3,78 dari skala 5. Berdasarkan hasil ini dapat disimpulkan bahwa *website* memiliki kekurangan pada aspek *usability* dan *efficiency*. Sementara aspek lainnya meliputi *functionality* dan *reliability* sudah baik. Berikut statistik kualitas *website* berdasarkan pengalaman penggunaanya ditunjukkan pada tabel 8.29 sebagai berikut:

TABEL IX
HASIL PENGUJIAN KUALITAS WEBSITE

Aspek	Total Nilai	Jumlah Item	Rata-Rata	Predikat
<i>Usability</i>	545	156	3,49	Cukup
<i>Reliability</i>	501	117	4,28	Baik
<i>Functionality</i>	1.603	390	4,11	Baik
<i>Efficiency</i>	1.180	312	3,78	Cukup

V. KESIMPULAN

Berdasarkan pembahasan di atas, dapat disimpulkan:

- Dari hasil pengembangan sistem manajemen konten untuk mengelola suatu situs *blog* dapat disimpulkan bahwa:
 - Pengembangan sistem manajemen konten menggunakan *framework* CodeIgniter dapat memberikan pengalaman pengguna dengan masing-masing tingkat *usability* yang cukup baik (nilai 3,49 dari skala 5,00), tingkat *reliability* yang baik (nilai 4,28 dari skala 5,00), tingkat *functionality* yang baik (nilai 4,11 dari skala 5,00), dan tingkat *efficiency* yang cukup baik (nilai 3,78 dari skala 5,00) dari hasil penilaian 39 responden.
 - Penggunaan *pretty URL* dari *framework* CodeIgniter dalam pengembangan sistem manajemen konten membuat URL situs menjadi rapi dan bersih.
 - Penggunaan *library* Ion Auth dan Facebook API dalam pengembangan sistem manajemen konten blog berguna untuk pembuatan otentifikasi dan registrasi secara ringkas dan ringan.
 - Penggunaan *library* Chartjs dalam pengembangan sistem manajemen konten blog dapat menyajikan laporan dengan grafik yang menarik.
 - Sistem manajemen konten dikembangkan dengan menyediakan interaksi antara pembaca dan penulis artikel seperti fitur rating yang berguna bagi pembaca untuk penilaian artikel, fitur komentar yang memungkinkan diskusi antar pembaca artikel, dan fitur pelaporan *cyberbullying* yang berguna bagi pembaca untuk melaporkan penyalahgunaan komentar kepada penulis, serta pesan peringatan *cyberbullying* yang berguna bagi pembaca untuk memberikan pesan peringatan terkait bahaya *cyberbullying* terkait komentar yang dikirim pembaca kepada penulis artikel.
 - Sistem manajemen konten dikembangkan dengan menyediakan fitur pengelompokan artikel berdasarkan topik artikel dan kata kunci.
- Berdasarkan uji coba dari 6.537 dataset komentar dengan *4-fold cross validation* dalam pembuatan model klasifikasi kategori komentar *cyberbullying* menggunakan *Multinomial Naive Bayes* dapat disimpulkan bahwa:
 - Model dengan fitur *stemming-unigram* memiliki akurasi 6% (73% - 67%) lebih baik dibandingkan model dengan fitur *stemming-bigram* (67%).
 - Model dengan fitur tanpa *stemming* dan menggunakan fitur *unigram* memiliki akurasi 4% (70% - 66%) lebih baik dibandingkan model dengan fitur tanpa *stemming* dan menggunakan fitur *bigram*.
 - Model dengan fitur *stemming-unigram* memiliki akurasi 3% (73% - 70%) lebih baik dibandingkan model dengan fitur tanpa *stemming* dan menggunakan fitur *unigram*.
 - Model dengan fitur *stemming-bigram* memiliki akurasi 1% (67% - 66%) lebih baik dibandingkan

- model dengan fitur tanpa *stemming* dan menggunakan fitur *bigram*.
- e. Dalam tugas akhir ini, model yang terbaik ialah model dengan fitur *stemming-unigram* yang memiliki sebesar 73% dan tingkat *error* sebesar 27% di antara semua model yang dikembangkan. Sedangkan model yang terburuk ialah model dengan fitur tanpa *stemming* dan menggunakan fitur *bigram* yang akurasi hanya 66%.
 - f. Model *stemming-unigram* memiliki ukuran fitur kata hanya sebesar 21% atau 4.910 fitur kata dari ukuran fitur kata dari model dengan tanpa fitur *stemming* dan menggunakan fitur *bigram* yang memiliki ukuran fitur kata sebesar 23.787 fitur kata. Model *stemming-unigram* memiliki ukuran fitur kata terkecil. Sedangkan model dengan tanpa fitur *stemming* dan menggunakan fitur *bigram* memiliki ukuran fitur yang terbesar.
 - g. Penggunaan fitur *stemming-unigram* dalam pembuatan model klasifikasi *cyberbullying* mampu memberikan akurasi yang terbaik dan efisien secara ukuran fitur kata. Sebaliknya penggunaan tanpa fitur *stemming* dan menggunakan fitur *bigram* dalam pembuatan model klasifikasi *cyberbullying* memberikan akurasi terburuk dan ukuran fitur kata yang besar.

UCAPAN TERIMA KASIH

Terima kasih kepada Daris Afif Muafa, S.Sos., Fiktor Setiawan, S.Sos., dan Febri Sawaluddin, S.Sos. yang telah membantu memvalidasi *dataset* komentar dalam penyusunan tugas akhir ini.

DAFTAR PUSTAKA

- [1] Ranny Rastati, "Bentuk Perundungan Siber di Media Sosial dan Pencegahannya Bagi Korban dan Pelaku," Jurnal Sosioteknologi, vol. 15, No. 2, pp. 176–178, Agustus, 2016.
- [2] Anton Subagia, "Kolaborasi Codeigneter dan Ajax dalam Perancangan CMS," Jakarta: PT Elex Media Komputindo, 2018, pp.1.
- [3] Fachrian Anugerah, "Perbaikan Kinerja Praproses Karakter Berulang dalam Mengenali Kata Pada Klasifikasi Sentimen Berbahasa Indonesia," M.Kom, thesis, Dept. Teknologi Informasi, Institut Teknologi Sepuluh November, Surabaya, Indonesia, 2017, pp. 26-31.
- [4] Wahyu Candra Indhiarta, "Penggunaan N-Gram Pada Analisis Sentimen Pemilihan Kepala Daerah Jakarta Menggunakan Algoritma Naive Bayes," Dept. Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta, Surakarta, 2017, pp. 4.
- [5] PM Prihatini and IKG D. Putra and M. Sudarma, "Stemming Algorithm for Indonesian Digital News Text Precossing," International Journal of Engineering and Emerging Technology, Vol.2, No.2, pp.3–4, Juli-Desember 2017.
- [6] J. Asian and H.E. Williams and S.M.M Tahaghoghl, "Stemming Indonesian," presented at the 28th Australasian Computer Science Conference (ACSC 2005), Melbourne, Australia, 2005.
- [7] Alif Sabrani and I Gede Puru Wirarama Wedashwara W. and Fitri Bimantoro, "Metode Multinomial Naive Bayes Untuk Klasifikasi Artikel Online Tentang Gempa di Indonesia," Jtika, Vo.2, No.1, pp. 91, Maret 2020.
- [8] Suyanto, "Machine Learning: Tingkat Dasar dan Lanjut," Bandung: Informatika Bandung, 2018, pp. 333.
- [9] Suyanto, "Machine Learning: Tingkat Dasar dan Lanjut," Bandung: Informatika Bandung, 2018, pp. 332.
- [10] Fajriyan Nurli, "Analisis Pengaruh Kualitas Website PT. Tiki Jalur Nugraha Ekakurir (JNE) Terhadap Kepuasan Pengguna Dengan Metode WebQEM," Dept. Teknologi dan Informatika, Stikom Surabaya, Surabaya, 2018, pp.14-16.
- [11] MySQL. [Online]. Available: <https://www.mysql.com>
- [12] Anton Subagia, "Kolaborasi Codeigneter dan Ajax dalam Perancangan CMS," Jakarta: PT Elex Media Komputindo, 2018, pp.25.
- [13] Anton Subagia, "Kolaborasi Codeigneter dan Ajax dalam Perancangan CMS," Jakarta: PT Elex Media Komputindo, 2018, pp.43.
- [14] DataTables. [Online]. Available: <https://datatables.net>
- [15] Chartjs. [Online]. Available: <https://www.chartjs.org>
- [16] TinyMCE. [Online]. Available: <https://www.tinymce.com>
- [17] Ion Auth. [Online]. Available: <https://github.com/benedmunds/CodeIgniter-Ion-Auth>

Danar Dono lahir di Klaten, Jawa Tengah, Indonesia, pada tahun 1996. Dia menyelesaikan studi S1 di program Informatika Profesional iSTTS pada tahun 2020. Minat studi di bidang pengembangan web.