NORMALISASI KOMENTAR MEDIA SOSIAL PASANGAN CALON GUBERNUR 2024 DENGAN STATISTICAL MACHINE TRANSLATION

1st Kahil Akbar Bayu Adityo
Informatika, Direktorat Kampus Surabaya
Universitas Telkom
Surabaya, Indonesia
kahil.akbar@gmail.com

2nd Alqis Rausanfita
Informatika, Direktorat Kampus Surabaya
Universitas Telkom
Surabaya, Indonesia
alqisfita@telkomuniversity.ac.id

3rd Daud Muhajir
Informatika, Direktorat Kampus Surabaya
Universitas Telkom
Surabaya, Indonesia
daudmuhajir@telkomuniversity.ac.id

***Abstrak* — Tingginya aktivitas masyarakat dalam membahas pemilihan Gubernur melalui media sosial menghasilkan data komentar dalam jumlah besar, namun komentar tersebut sering menggunakan bahasa informal, bahasa sehari-hari, singkatan, serta bercampur dengan bahasa daerah dan dialek lokal yang sulit dipahami. Hal ini menghambat pemrosesan data komentar untuk keperluan analisis atau tujuan lainnya. Proses normalisasi manual membutuhkan waktu dan sumber daya yang sangat banyak, terutama jika data yang diolah berjumlah besar. Normalisasi secara manual juga rentan terhadap inkonsistensi dan kesalahan manusia. Jumlah data komentar di media sosial yang terus meningkat membuat normalisasi manual semakin tidak mungkin dan tidak efisien untuk dilakukan, sehingga diperlukan solusi otomatisasi. Sistem normalisasi teks otomatis dikembangkan menggunakan pendekatan *Phrase-Based Statistical Machine Translation* dengan memanfaatkan Moses. Dataset korpus paralel dibangun dari 31.889 pasangan kalimat informal-formal, sedangkan korpus monolingual terdiri dari 1.613.381 kalimat yang diambil dari Wikipedia. Model dievaluasi menggunakan metrik *BLEU* untuk mengukur kualitas hasil normalisasi. Model terbaik mencapai skor *BLEU* 82,16 pada data test dan 81,04 pada data validasi, berhasil mengenali berbagai pola bahasa informal seperti singkatan tidak baku, kata berulang dengan angka, dan bahasa gaul. Namun, sistem memiliki keterbatasan terhadap kemampuan penanganan *Out-Of-Vocabulary*.**

***Kata kunci*— normalisasi teks, *PBSMT*, Moses, media sosial, gubernur**

I. PENDAHULUAN

Pemilihan Gubernur merupakan bagian penting dari proses demokrasi di Indonesia, peran sentral gubernur tentunya akan menarik atensi publik di media sosial. Platform seperti Instagram menjadi ruang masyarakat dalam menyampaikan opini terkait pasangan calon, terutama selama masa kampanye. Namun, komentar yang dihasilkan cenderung menggunakan bahasa informal, singkatan, serta campuran bahasa daerah dan dialek, yang menyulitkan proses analisis apabila data tersebut hendak digunakan untuk keperluan penelitian atau tujuan lainnya. Mengatasi hal tersebut, diperlukan proses normalisasi teks, yaitu transformasi dari bentuk tidak baku ke bentuk baku sesuai Kamus Besar Bahasa Indonesia (KBBI). Proses ini penting untuk memastikan kualitas dan konsistensi data yang digunakan dalam pemrosesan bahasa alami. Pendekatan manual dalam normalisasi sangat tidak efisien pada skala besar, sehingga solusi otomatis menjadi pilihan yang logis. Salah satu pendekatan yang relevan adalah *Statistical Machine Translation* (*SMT*), khususnya *Phrase-Based Statistical Machine Translation* (*PBSMT*), yang mampu mempelajari pola transformasi dari teks informal ke formal menggunakan data paralel [1]. Penelitian ini bertujuan menerapkan metode *PBSMT* untuk melakukan normalisasi komentar media sosial pasangan calon Gubernur Jawa Timur 2024. Dengan menggunakan *framework moses* dan implementasi model berbasis *website*, sistem ini diharapkan dapat memberikan kontribusi secara metodologis dalam pengembangan sistem normalisasi bahasa Indonesia serta menyediakan dataset paralel yang berguna bagi studi sejenis selanjutnya.

II. KAJIAN TEORI

A. Penelitian Terdahulu

Penelitian terdahulu yang dilakukan oleh Eleison et al. [2] menunjukkan bahwa *PBSMT* mampu menghasilkan performa yang baik untuk normalisasi bahasa Indonesia, dengan skor *BLEU* sebesar 64% pada *dataset* berisi 100.000 kalimat yang memiliki panjang rata-rata 7,51 kata. Penelitian tersebut juga menegaskan bahwa *PBSMT* cocok digunakan pada data dengan tingkat *noise* tinggi, seperti komentar media sosial, sehingga relevan dengan konteks penelitian ini. Penelitian yang dilakukan oleh Kurnia dan Yulianti [3] memberikan kontribusi signifikan dalam pemetaan pola umum bahasa informal Indonesia, khususnya dalam konteks komentar media sosial, serta kategorisasi perubahan ke bentuk formal. Mereka tidak hanya menyoroti transformasi kata tidak baku menjadi baku, tetapi juga mengidentifikasi pola-pola yang sering muncul dalam tuturan sehari-hari, seperti penghilangan awalan (contohnya "baca" dari "membaca"), penggunaan singkatan tidak baku ("yg" untuk "yang"), substitusi huruf dengan angka ("b2k" untuk "babak"), serta penggunaan partikel-partikel informal seperti "dong", "sih",

dan "deh". Penerapan KenLM sebagai *language model* merupakan bagian *default* dalam *framework* Moses yang telah menjadi standar dalam implementasi *PBSMT* untuk berbagai keperluan, termasuk normalisasi teks. Dalam penelitian oleh Hossain, Bhuiyan, & Islam [4], komponen-komponen utama dalam Moses seperti *translation model* dan *decoder* berhasil dimanfaatkan untuk translasi teks *Bangla-English*, menunjukkan fleksibilitas sistem ini dalam menangani pasangan bahasa dengan struktur berbeda. Meskipun penelitian ini menggunakan pasangan bahasa *Bangla-English*, pendekatan yang digunakan tetap relevan untuk penelitian ini yang menargetkan pasangan teks informal-formal dalam bahasa Indonesia.

B. *Phrase-Based Statistical Machine Translation* (*PBSMT*)

Phrase-Based Statistical Machine Translation (*PBSMT*) merupakan pendekatan dari *Statistical Machine Translation* (*SMT*) yang menggunakan frasa sebagai unit dasar normalisasi, berbeda dengan *SMT* konvensional yang berbasis kata. Dengan pendekatan ini, *PBSMT* mampu menangkap konteks antar kata dalam satu frasa, sehingga hasil terjemahan menjadi lebih natural dan kontekstual [5]. Selain komponen utama seperti *translation model, language model*, dan *decoder*, *PBSMT* juga mengandalkan komponen tambahan seperti *phrase table* dan *reordering model* [6], [7]. Kedua komponen ini bekerja secara terintegrasi untuk meningkatkan kualitas terjemahan.

C. Moses

Moses adalah *framework open-source* untuk membangun sistem penerjemahan berbasis statistik seperti *PBSMT* yang banyak digunakan dalam penelitian berbagai pasangan bahasa. *Framework* ini menyediakan *pipeline* lengkap yang mencakup *preprocessing*, *training*, dan *decoding*, serta fitur *tuning* parameter, evaluasi BLEU, dan penanganan *out-of-vocabulary* (*OOV*) untuk menghasilkan terjemahan yang akurat [8].

D. Phrase Table

Phrase table adalah komponen *PBSMT* yang digunakan untuk menyimpan pasangan frasa dari bahasa sumber dan bahasa target. Pasangan ini tidak hanya mencakup satu kata, tetapi juga rangkaian kata (n-gram) atau frasa. *Phrase table* didasarkan pada pola statistik, sehingga mencerminkan probabilitas terjemahan antara frasa tertentu. *Phrase table* dibuat selama proses *training* sistem, dimana setiap pasangan frasa diberikan nilai probabilitas berdasarkan frekuensi kemunculan dalam korpus pelatihan. Dengan kata lain, *phrase table* memungkinkan sistem untuk memilih pasangan terjemahan yang paling tepat berdasarkan probabilitas yang didapatkan [8].

E. N-gram

N-gram merupakan model statistik yang digunakan dalam pemrosesan bahasa alami untuk memprediksi kata berdasarkan n kata sebelumnya. Dalam *PBSMT*, n-gram berperan dalam *language model* untuk mengevaluasi kemungkinan susunan kata yang alami dalam bahasa target. Model ini menghitung probabilitas kemunculan suatu rangkaian kata berdasarkan frekuensi dalam korpus. Secara matematis, salah satu bentuk dari n-gram model yaitu *bigram* (n = 2) dirumuskan sebagai berikut.

$$P(e_n|e_1, e_2, \ldots, e_{n-1}) \approx P(e_n|e_{n-1}) = \frac{count(e_{n-1}, e_n)}{count(e_{n-1})} \quad (1)$$

dimana $P(e_n|e_{n-1})$ merepresentasikan probabilitas kemuncuklan kata e_n jika diberikan kata target sebelumnya e_{n-1}. Nilai $count(e_{n-1}, e_n)$ adalah jumlah kemunculan pasangan kata e_{n-1} dan e_n secara bersamaan dalam korpus, sedangkan $count(e_{n-1})$ adalah jumlah kemunculan kata w_{n-1} secara keseluruhan dalam korpus [9].

F. KenLM

KenLM merupakan *library* yang dirancang untuk membangun *language model* berbasis n-gram secara efisien, baik dari segi kecepatan pemrosesan maupun penggunaan memori. Dalam *PBSMT*, KenLM digunakan untuk menyusun *language model* yang dapat mengevaluasi seberapa alami suatu susunan kata dalam bahasa target. Model ini sangat membantu dalam menentukan pilihan frasa yang paling mungkin muncul berdasarkan konteks sebelumnya, sehingga meningkatkan kualitas terjemahan. Secara matematis, probabilitas kata berikutnya dalam model KenLM dirumuskan sebagai berikut.

$$P(e_n|e_1^{n-1}) \approx P(e_n|e_f^{n-1}) \prod_{i=1}^{f-1} b(e_i^{n-1}) \quad (2)$$

dimana e_n adalah kata yang ingin diprediksi, e_1^{n-1} adalah semua kata sebelum kata yang ingin diprediksi, f mewakili jumlah kata sebelum kata yang ingin diprediksi yang masih dapat ditemukan dalam data pelatihan, dan $b(e_i^{n-1})$ adalah penalti *backoff* yang diberikan ketika model harus mundur ke n-gram dengan panjang yang lebih pendek [10].

G. Tuning

Tuning adalah proses penting dalam *PBSMT* yang bertujuan untuk mengoptimalkan parameter dari berbagai model agar dapat menghasilkan terjemahan yang lebih akurat. Dalam pendekatan ini, digunakan model log-linear yang menggabungkan sejumlah fitur atau sub-model (seperti *phrase table*, *reordering model*, dan *language model*) yang masing-masing diberikan bobot tertentu. Proses *tuning* dilakukan dengan menyesuaikan bobot-bobot tersebut agar memberikan hasil terjemahan terbaik berdasarkan metrik evaluasi tertentu. Salah satu algoritma yang sering digunakan untuk tujuan ini adalah *Minimum Error Rate Training* (*MERT*), yang berfungsi untuk meminimalkan tingkat kesalahan sistem terhadap data pengembangan. *MERT* mengiterasi proses pencocokan antara prediksi model dan referensi hingga tercapai hasil optimal, atau hingga tidak terjadi perubahan signifikan pada parameter model. Secara matematis, probabilitas terjemahan dapat dirumuskan sebagai berikut.

$$p(x) = exp\left(\sum \lambda_i h_i(x)\right) \quad (3)$$

dimana $p(x)$ merupakan probabilitas terjemahan yang dihasilkan oleh model, λ_i adalah bobot atau parameter yang terkait dengan setiap fitur yang akan dipelajari dan dioptimalkan selama proses *tuning*, dan $h_i(x)$ merepresentasikan fitur ke-i yang digunakan dalam proses terjemahan [11].

H. *Out-Of-Vocabulary* (*OOV*)

Out-of-Vocabulary (*OOV*) mengacu pada kata-kata yang tidak dikenali oleh sistem terjemahan karena tidak ada dalam

korpus ketika *training* [12]. Sistem biasanya akan mempertahankan kata OOV dalam bentuk aslinya pada output hasil terjemahan. Dalam *PBSMT*, *OOV* menjadi tantangan karena kata-kata yang tidak ada dalam tabel frasa tidak dapat diproses langsung, sehingga mengurangi kualitas terjemahan.

I. *Bilingual Evaluation Understudy* (*BLEU*)

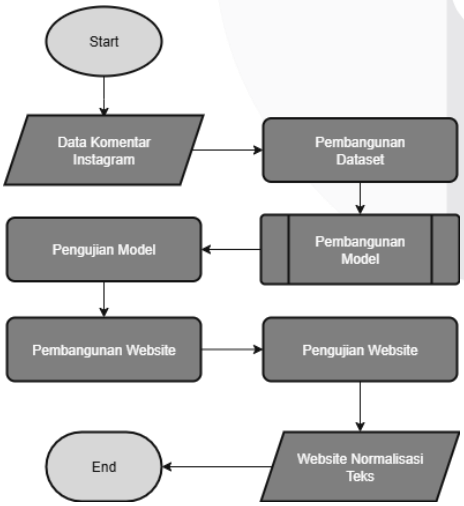
Bilingual Evaluation Understudy (*BLEU*) adalah metrik evaluasi yang digunakan untuk mengukur kualitas terjemahan dalam *PBSMT*. Metrik ini membandingkan hasil terjemahan dengan referensi (data bahasa target) menggunakan pendekatan berbasis n-gram. Nilai BLEU memiliki rentang antara 0 hingga 1, dimana semakin tinggi nilai BLEU menunjukkan hasil terjemahan yang semakin mendekati terjemahan referensi yang dibuat oleh manusia [2]. Secara matematis, skor *BLEU* dapat dihitung dengan persamaan sebagai berikut.

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n \log p_n \quad (4)$$

dimana BP adalah *brevity penalty* yang memberikan *penalty* pada hasil terjemahan yang lebih pendek dari kalimat referensi, w_n adalah bobot untuk setiap n-gram ($w_n = 1/N$), dan p_n adalah *precision* dari n-gram yang dihitung sebagai rasio antara jumlah n-gram yang cocok dengan referensi terhadap total n-gram dalam hasil terjemahan. Kedua, persamaan untuk menghitung *brevity penalty* berdasarkan rasio panjang terjemahan (c) terhadap panjang referensi (r).

III. METODE

Penelitian dimulai dari tahap pengambilan data, dilanjut dengan pembersihan data serta menyiapkan data berupa teks bahasa Indonesia formal. Selanjutnya, dilakukan pembangunan sistem normalisasi teks dengan menggunakan *moses*. Lalu, akan dilakukan beberapa pengujian pada model. Terakhir, pembuatan dan pengujian *website* sebagai antarmuka sistem normalisasi teks seperti pada Gambar 1.



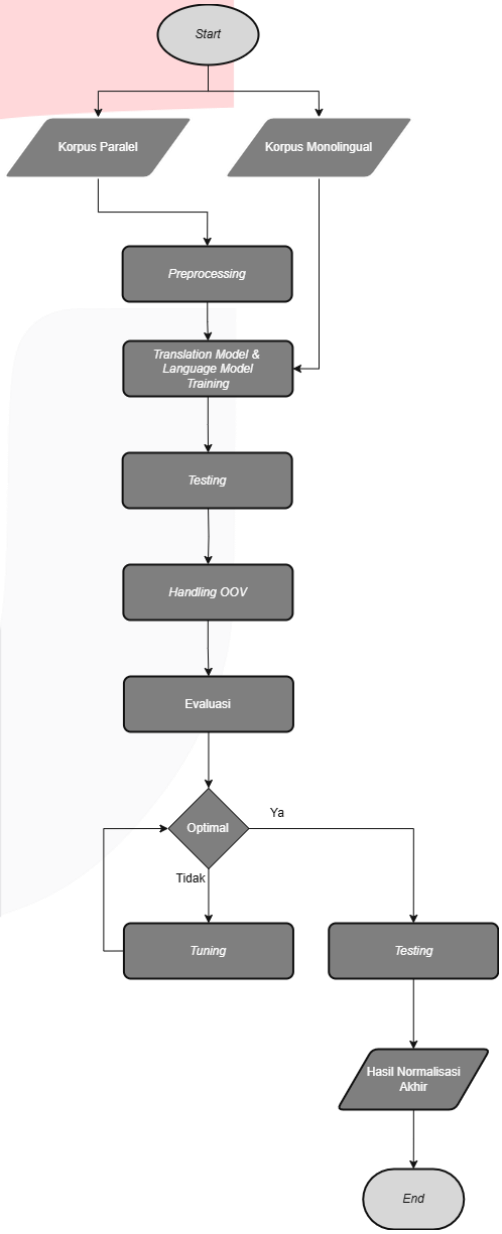
Gambar 1
(Alur Penelitian)

A. Pembangunan *Dataset*

Data yang diperlukan untuk pembangunan sistem terdapat dua macam, yaitu data korpus paralel yang berisi pasangan bahasa informal-formal dan data korpus monolingual yang berisi teks bahasa target (formal). Korpus paralel berbentuk 2 *file* yang tiap barisnya merupakan pasangan translasi bahasa sumber (informal) dan bahasa target (formal). Sedangkan, korpus monolingual berbentuk 1 *file* yang hanya berisi bahasa target. Data korpus paralel informal diperoleh dari *crawling* data akun instagram pasangan calon Gubenur Jawa Timur 2024. Sedangkan korpus paralel formal, dibangun dengan menerjemahkan manual dari korpus paralel informal. Terakhir, korpus monolingual diperoleh dari *dataset* yang dibangun dari kalimat wikipedia.

B. Pembangunan Model

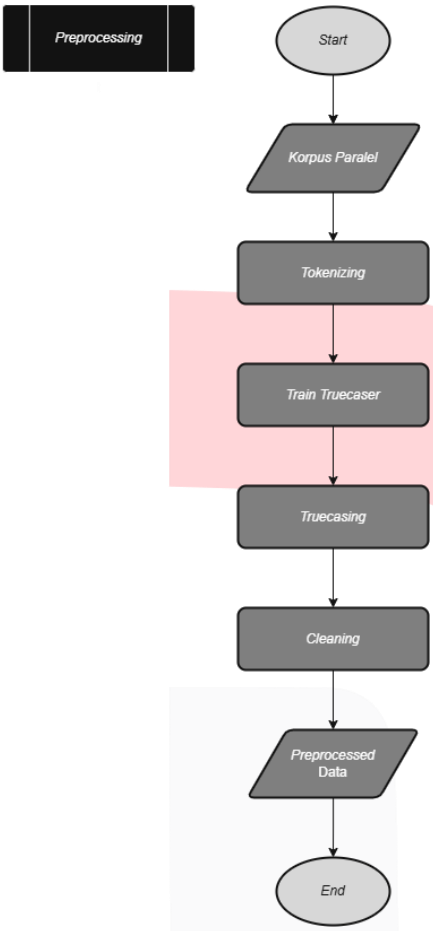
Gambar 2 menunjukkan rancangan sistem normalisasi teks berbasis *PBSMT*. Proses dimulai dari preprocessing korpus paralel, dilanjutkan dengan training model translasi dan model bahasa. Setelah dilakukan pengujian dan penanganan kata *OOV*, tahap evaluasi dan tuning dilakukan secara iteratif hingga model mencapai performa optimal.



Gambar 2
(Rancangan Sistem *PBSMT*)

a.) *Preprocessing*

Terdapat tiga tahap *preprocessing* yang dilakukan pada dataset korpus paralel yaitu *tokenization*, *truecasing*, dan *cleaning*. Sebelum melakukan *truecasing*, diperlukan model *truecaser* seperti yang ditunjukkan pada Gambar 3.



Gambar 3
(Preprocessing Data Korpus)

b.) *Language Model Training*

Language Model Training dilakukan menggunakan KenLM. Proses ini dimulai dengan mengubah bentuk korpus monolingual menjadi bentuk *binary*. Selain itu, n-gram yang digunakan dalam *training* ditentukan berdasarkan hasil dari skenario pengujian, untuk memastikan konfigurasi yang optimal.

c.) *Translation Model Training*

Training Translation Model dilakukan melalui beberapa langkah penting. Proses dimulai dengan phrase alignment, lalu phrase extraction and scoring, dan terakhir pembuatan lexicalised reordering tables, proses training ini sudah tersedia di pipeline moses. Semua hasil dari proses training dikonfigurasi ke dalam file konfigurasi moses untuk menjalankan pipeline PBSMT selanjutnya.

d.) *Tuning*

Tuning dilakukan menggunakan *Minimum Error Rate Training* (*MERT*) untuk mengoptimalkan parameter model *PBSMT*. Proses ini membutuhkan data paralel yang berbeda dari data yang digunakan untuk training. Data paralel yang digunakan untuk tuning harus terlebih dahulu melalui tahap *preprocessing*, seperti *tokenization*, *truecasing*, dan *cleaning*. *Tuning* dilakukan hingga tidak adanya perubahan weight atau *n-best list* (daftar n kandidat terjemahan terbaik) yang dihasilkan saat *testing*.

e.) *Handling OOV*

OOV Handling dalam penelitian ini mencakup dua pendekatan untuk menangani kata-kata yang tidak ditemukan dalam korpus pelatihan. Pendekatan pertama adalah membiarkan kata *OOV* tetap ada dalam hasil terjemahan, yang merupakan konfigurasi default pada *moses*. Pendekatan kedua adalah *drop unknown* (menghapus kata *OOV* dari hasil terjemahan). Kedua pendekatan ini dianalisis untuk menentukan metode yang paling sesuai dengan data penelitian.

f.) Evaluasi

Evaluasi dilakukan dengan menggunakan metrik evaluasi *BLEU untuk* mengukur kesesuaian hasil translasi dengan referensi yang terdapat dalam korpus paralel. Pada proses ini, hasil translasi dibandingkan dengan korpus paralel bahasa target yang telah melalui proses *truecasing* untuk memastikan format teks konsisten.

g.) *Testing*

Testing dilakukan dengan langkah-langkah berikut. Pertama, mengubah format *phrase table* dan *lexicalised reordering model* ke dalam bentuk *binary* untuk mempercepat proses. Setelah itu, dilakukan proses *decoding* untuk menghasilkan teks formal dari input teks informal menggunakan model *PBSMT* yang telah dilatih

C. Pengujian

Pengujian dilakukan menggunakan metode *black box testing* untuk memastikan fungsi *website* berjalan dengan baik. Selain itu, terdapat dua skenario pengujian utama untuk menemukan konfigurasi sistem normalisasi teks yang optimal, yaitu pengaruh jumlah pasangan kata pada model N-gram dan pengaruh jumlah data dalam korpus monolingual. Skenario pertama menguji pengaruh jumlah pasangan kata pada N-gram (n=2,3,4,5) terhadap performa sistem *PBSMT*. Karena frasa umumnya terdiri dari lebih dari satu kata, nilai minimal dimulai dari n=2. Setiap konfigurasi diuji menggunakan metrik *BLEU* untuk mengevaluasi akurasi dan kesesuaian hasil terjemahan, dengan tujuan menentukan konfigurasi N-gram yang menghasilkan skor *BLEU* tertinggi. Skenario kedua menguji pengaruh jumlah data pada korpus monolingual, dibagi menjadi empat bagian: 25%, 50%, 75%, dan 100% dari total data. Uji dilakukan dengan konfigurasi N-gram *default* bernilai 3. Hasil terjemahan dari tiap skenario dianalisis menggunakan *BLEU* untuk melihat pengaruh peningkatan jumlah data terhadap kualitas terjemahan.

D. Pembangunan Website

Website yang dibangun bertujuan untuk mengimplementasikan sistem normalisasi teks dengan *PBSMT*. Dari segi arsitektur, model *PBSMT* yang telah dilatih akan dijadikan *API* menggunakan *framework Flask* pada *backend*. *API* ini akan menjadi jembatan antara *frontend* dan *backend* untuk menerima input dan mengirimkan output hasil normalisasi teks berdasarkan model yang dipilih. Pada

bagian *frontend*, digunakan *JavaScript* untuk mengakses *API* dan mengatur interaksi antarmuka.

IV. HASIL DAN PEMBAHASAN

A. Pembangunan *Dataset*

Dataset terdiri dari dua jenis korpus, yaitu korpus paralel dan korpus monolingual. Korpus paralel berisi pasangan kalimat informal-formal yang disusun dalam dua file sejajar (jumlah data sama), sedangkan korpus monolingual berisi kalimat dalam bahasa formal dari Wikipedia. Korpus paralel dibentuk dengan mengumpulkan komentar informal dari Instagram dan kemudian dinormalisasi secara manual menjadi kalimat formal. Komentar informal dikumpulkan dari akun-akun Instagram milik pasangan calon gubernur dan wakil gubernur Jawa Timur 2024 selama masa kampanye. Setelah proses pengumpulan data, diperoleh 40.523 data untuk korpus paralel dan 1.613.381 data untuk korpus monolingual. Sebelum digunakan, korpus paralel distandarisasi dengan mengganti elemen non-leksikal seperti mention, hashtag, dan emotikon menjadi token standar agar tidak dikenali sebagai *OOV* seperti pada Tabel 1. Simbol penting seperti titik, koma, dan tanda baca lain dipertahankan karena berkontribusi pada makna. Setelah proses standarisasi dan deduplikasi, jumlah data yang siap digunakan adalah 31.889 kalimat.

Tabel 1
(Hasil Standarisasi Korpus Paralel)

Sebelum	Sesudah
Kemenangan keknyaa 70% ini mahhh😍	Kemenangan keknyaa 70 xemoticonx ini mahhh xemoticonx
PASTI UNTUK Ibu @khofifah.ip dan Pak @emildardak DOOOOONG	PASTI UNTUK Ibu xmentionx dan Pak xmentionx DOOOOONG
Kita sefrekwensi bu Arumi, sama" maniak kerupuk... 😂😂❤️❤️❤️	Kita sefrekwensi bu Arumi, sama" maniak kerupuk... xemoticonx xemoticonx xemoticonx xemoticonx xemoticonx xemoticonx xemoticonx xemoticonx

Proses pengubahan kalimat informal menjadi formal mengikuti beberapa aturan tertentu. Pertama, setiap kata yang digunakan harus terdaftar dalam KBBI dan tidak termasuk dalam kategori cakapan atau ragam tidak baku. Jika sebuah kata tergolong tidak baku meskipun tercantum dalam KBBI, maka kata tersebut harus diubah sesuai konteks. Misalnya, kata "biar" harus diubah menjadi "agar" atau "supaya". Namun, aturan ini tidak berlaku untuk kata tambahan seperti "sih", "dong", "eh", atau "yah" karena tidak memiliki padanan formal yang tepat, sehingga tetap dibiarkan. Aturan kedua adalah menghilangkan karakter dan tanda baca yang berlebihan, seperti mengubah "yangggg" menjadi "yang" atau "coblos!!" menjadi "coblos!". Sedangkan aturan ketiga adalah bentuk standarisasi seperti "xmentionx", "xhashtagx", dan "xemoticonx" tetap dipertahankan dalam korpus formal sesuai dengan bentuk pada Tabel 1. Selama proses pembuatan dataset, peneliti menemukan pola-pola khas dalam komentar akun pasangan calon gubernur Jawa Timur 2024. Salah satu yang paling sering muncul adalah penggunaan kata tambahan seperti "sih", "dong", dan "mah" sebagai penegas dalam komunikasi informal. Ditemukan pula bahasa gaul berupa singkatan seperti "salfok" (salah fokus), "galfok" (gagal fokus), dan "baper" (bawa perasaan), serta kata yang kehilangan imbuhan seperti "ngayomi" (mengayomi) dan "ketemu" (bertemu). Singkatan tidak baku seperti "yg" (yang), "dgn" (dengan), dan "cm" (cuma), serta penggunaan angka sebagai pengganti huruf seperti "se7" (setuju), juga banyak dijumpai. Penekanan makna ditemukan pada kata berbahasa Jawa seperti "uenak" (enak) dan "ruame" (ramai), yang menyiratkan arti sangat. Pola lain yang muncul meliputi bentuk kata ulang yang digantikan dengan angka 2 atau tanda petik dua, seperti "bertahun2" (bertahun-tahun) dan "senyum"" (senyum-senyum), serta penggunaan akhiran "-e" sebagai pengganti "-nya" dalam bahasa Indonesia, contohnya "mulut e" (mulutnya) dan "gubernur e" (gubernurnya).

B. Pembangunan Model

a.) Preprocessing

Tahapan *preprocessing* terdiri dari empat tahap, yaitu *tokenizing*, *train truecaser* model, *truecasing*, dan *cleaning*. Keempat tahap ini diterapkan pada seluruh bagian *dataset*, yaitu *train*, *test*, dan *validation*. Hasil dari masing-masing tahap dapat dilihat pada tabel terpisah, yaitu hasil *tokenizing* pada Tabel 2, *truecaser model* pada Tabel 3, *truecasing* pada Tabel 4, dan *cleaning* pada Tabel W. Namun, tahap *truecasing* pada penelitian ini hanya digunakan untuk menyesuaikan dengan standar input yang dibutuhkan Moses dalam pembuatan model dan bukan sebagai aspek evaluasi keakuratan model. Oleh karena itu, kebenaran kapitalisasi kalimat pada hasil normalisasi tidak akan menjadi fokus dalam penelitian ini, baik hasilnya benar maupun salah.

Tabel 2
(Hasil *Tokenizing*)

Sebelum *Tokenizing*	Setelah *Tokenizing*
Bapak cocok banget jd kepala.daerah ggak.jaim tp wibawa ..mantap.pak	Bapak cocok banget jd kepala.daerah ggak.jaim tp wibawa .. mantap.pak
Aku sih ya, menunggu perubahan baru	Aku sih ya , menunggu perubahan baru
Salfok kesedotan saya..goyang² sndri xemoticonx xemoticonx	Salfok kesedotan saya .. goyang ² sndri xemoticonx xemoticonx

Tabel 3
(Hasil Model *Truecaser*)

Hasil Model *Truecaser*
wujudkan (18/20) WUJUDKAN (1) Wujudkan (1) membangun (80/82) Membangun (2) cita-citanya (2/2) Jombang (16/18) JOMBANG (1) jombang (1) pengemis (3/4) Pengemis (1)

Tabel 4
(Hasil *Truecasing)*

Sebelum *Truecasing*	Setelah *Truecasing*
Nelayan kita senang dengan Bu Khofifah	nelayan kita senang dengan Bu Khofifah
jawa timur maju bareng khofifah	Jawa timur maju bareng khofifah

Aku sih ya, menunggu perubahan baru	aku sih ya , menunggu perubahan baru

Tabel 5
(Jumlah Data Setelah *Cleaning*)

Split	Jumlah Data Sebelum Cleaning	Jumlah Data Setelah Cleaning
train	22322	22288
test	6377	6365
validation	3190	3180

b.) Language Model Training

Tahapan *Language Model Training* dilakukan dengan menggunakan data dari Wikipedia dengan total 1613373 data. Proses ini dilakukan menggunakan KenLM sebagai *language model*. *Language Model* akan berbentuk sebuah *file .arpa* yang terdiri dari tiga komponen, kecuali pada tingkat n-gram tertinggi karena tidak memiliki nilai *backoff* seperti pada Tabel 6.

Tabel 6
(Hasil *Language Model*)

Hasil *Language Model*
\1-grams: -7.220481 Rhytiphora -0.08980977 -5.927479 cinerascens -1.2423062 -1.9087471 adalah -0.9057203 -4.0583544 spesies -0.51501954 ..... \2-grams: -2.046699 kumbang </s> 0 -1.4920924 diapresiasikan oleh -0.06256088 -1.8540441 kesesuaiannya oleh -0.06256088 -1.0240985 digumpalkan oleh -0.06256088 \3-grams: ..... -2.7928908 yang adalah </s> -0.7510083 dengan Altstadt dari -0.7559657 besar. Referensi untuk -1.3569641 dalam tidurnya. Ia .....

c.) Translation Model Training

Tahapan *translation model training* membutuhkan dua input yaitu, korpus paralel hasil *cleaning* pada *split train* dan *language model.* Proses ini juga memerlukan pengaturan metode *alignment* dan *reordering model*. Terdapat enam pilihan metode *alignment* yaitu *intersection*, *grow*, *grow-diag*, *union*, *srctotgt*, dan *tgttosrc*. Sedangkan, terdapat tiga pilihan *reordering model* yaitu *msd-bidirectional-fe*, *phrase-msd-bidirectional-fe*, dan *hier-mslr-bidirectional-fe*. Kombinasi keduanya menghasilkan 18 model seperti pada Tabel 7, yang kemudian akan melalui proses *tuning* dan evaluasi untuk menentukan *model* terbaik.

Tabel 7
(Kombinasi *Alignment Method* dan *Reordering Model*)

Kombinasi	*Alignment Method*	*Reordering Model*
1	intersection	msd-bidirectional-fe
2	intersection	phrase-msd-bidirectional-fe
3	intersection	hier-mslr-bidirectional-fe
4	grow	msd-bidirectional-fe
5	grow	phrase-msd-bidirectional-fe
6	grow	hier-mslr-bidirectional-fe
7	grow-diag	msd-bidirectional-fe
8	grow-diag	phrase-msd-bidirectional-fe
9	grow-diag	hier-mslr-bidirectional-fe
10	union	msd-bidirectional-fe
11	union	phrase-msd-bidirectional-fe
12	union	hier-mslr-bidirectional-fe
13	srctotgt	msd-bidirectional-fe
14	srctotgt	phrase-msd-bidirectional-fe
15	srctotgt	hier-mslr-bidirectional-fe
16	tgttosrc	msd-bidirectional-fe
17	tgttosrc	phrase-msd-bidirectional-fe
18	tgttosrc	hier-mslr-bidirectional-fe

Selain itu, tahapan ini menghasilkan tiga komponen utama yang ditampilkan pada tabel terpisah. *Phrase table* (Tabel 8) berisi pasangan frasa dari korpus paralel beserta nilai probabilitas dan informasi *alignment*. Reordering model (Tabel 9) memuat pola penyusunan kata dari bahasa sumber ke bahasa target. Sementara itu, file konfigurasi *moses.ini* (Tabel 10) berfungsi sebagai parameter dalam proses *decoding* model normalisasi.

Tabel 8
(Hasil *Phrase Table*)

Hasil *phrase table*
menang wes ... sampah ada solusi nya ||| menang sudah , sampah ada solusinya ||| 0.5 9.54118e-05 1 0.0251494 ||| 0-0 1-1 2-2 3-3 4-4 5-5 ||| ||| |||

Tabel 9
(Hasil *Reordering Model*)

Hasil *reordering model*
di daerahhh situ ya pak Luman Semangattt ||| di daerah situ ya Pak LUMAN semangat ||| 0.6 0.2 0.2 0.6 0.2 0.2

Tabel 10
(Konfigurasi *moses.ini*)

Konfigurasi *moses.ini*
[feature] UnknownWordPenalty WordPenalty PhrasePenalty

Konfigurasi *moses.ini*
PhraseDictionaryMemory name=TranslationModel0 num-features=4 path=/mnt/d/skripsi/Code/src/working/train/model/phrase-table.gz input-factor=0 output-factor=0 LexicalReordering name=LexicalReordering0 num-features=6 type=wbe-msd-bidirectional-fe-allff input-factor=0 output-factor=0 path=/mnt/d/skripsi/Code/src/working/train/model/reordering-table.wbe-msd-bidirectional-fe.gz Distortion KENLM name=LM0 factor=0 path=/mnt/d/skripsi/Code/src/lm/lm/lm.for.blm order=3 [weight] UnknownWordPenalty0= 1 WordPenalty0= -1 PhrasePenalty0= 0.2 TranslationModel0= 0.2 0.2 0.2 0.2 LexicalReordering0= 0.3 0.3 0.3 0.3 0.3 0.3 Distortion0= 0.3 LM0= 0.5

d.) Tuning

Tahapan *Tuning* dilakukan pada setiap kombinasi model hasil *translation model training*. *Tuning* menghasilkan beberapa *file* penting yang dibuat dari hasil model pada setiap iterasi. Misalkan pada iterasi pertama akan dihasilkan *file* ***run1.best100.out.gz*** yang berisikan 100 kandidat normalisasi terbaik seperti pada Tabel 11. Kandidat dengan nilai negatif yang paling kecil akan dipilih sebagai kalimat dengan hasil terbaik pada iterasi *tuning* yang sedang berjalan dan akan disimpan pada *file* ***run1.out***. Selanjutnya *file* ***run1.moses.ini*** menyimpan konfigurasi model setelah proses *tuning,* termasuk *weight* optimal yang diperoleh pada iterasi tersebut seperti pada Tabel 12. Sementar itu, ***run1.weight.txt*** mencatat nilai *weight* yang digunakan pada iterasi yang sedang berlangsung. Proses *tuning* akan terus berjalan hingga nilai *weight* pada iterasi sebelumnya sama dengan *weight* pada iterasi terakhir.

Tabel 11
(Kandidat Hasil Normalisasi)

Kandidat Hasil Normalisasi
0 ||| putrinya Mbak arumibahsin cantiknya masyaallah ||| LexicalReordering0= -0.313451 0 0 0 -0.308041 0 0 0 Distortion0= 0 LM0= -79.5308 0 ||| putrinya mbak arumibahsin cantiknya masyaallah ||| LexicalReordering0= -0.31922 0 0 0 -0.313809 0 0 0 Distortion0= 0 LM0= -81.0098 WordPenalty0= -5 PhrasePenalty0= 5 TranslationModel0= -3.84176 -4.46129 -0.865759 -0.865759 ||| -107.058 0 ||| putrinya Mbak cantiknya arumibahsin masyaallah ||| LexicalReordering0= -0.262158 0 0 -4.09434 -0.256747 -4.09434 0 0 Distortion0= -4 LM0= -79.5308 WordPenalty0= -5 PhrasePenalty0= 5 TranslationModel0= -4.15281 -4.75841 -0.54599 -

Kandidat Hasil Normalisasi
0.54599 ||| -107.611 ….

Tabel 12
(Konfigurasi *moses.ini* pada Iterasi Pertama)

Konfigurasi moses.ini setelah *tuning*
[weight] LexicalReordering0= 0.131538 0.0224085 0.00651416 0.0187969 0.266012 0.0199146 0.00706209 0.00895909 Distortion0= 0.165089 LM0= -0.00103748 WordPenalty0= 0.0911557 PhrasePenalty0= -0.0226529 TranslationModel0= 0.0118176 0.0113901 0.208211 0.00744037 UnknownWordPenalty0= 1

e.) Handling OOV

Moses menyediakan tiga cara untuk menangani terjadinya *OOV* pada korpus yaitu membiarkan kata atau frasa *OOV* apa adanya, menghapus kata atau frasa *OOV* dengan menambahkan argumen "-drop-unknown" pada proses evaluasi, dan menggunakan modul transliterasi. Sayangnya, modul transliterasi hanya tersedia pada beberapa bahasa dan bahasa Indonesia formal tidak termasuk di dalamnya, sehingga cara penanagan *OOV* pada penelitian ini akan menggunakan cara pertama dan kedua. Dari kedua pendekatan tersebut, cara yang dipilih untuk digunakan dalam *base model* adalah pendekatan dengan skor evaluasi tertinggi berdasarkan nilai *BLEU*.

f.) Evaluasi

Tahapan evaluasi dilakukan pada 18 kombinasi yang ada pada Tabel 7. Hasil evaluasi dari 18 kombinasi yang ada dapat dilihat pada Tabel 13. Berdasarkan hasil tersebut, model dengan skor *BLEU* tertinggi dipilih sebagai *base model* yang akan digunakan pada tahap implementasi sistem yaitu kombinasi 16, model dengan *alignment method* ***tgttosrc*** dan *reordering model* ***msd-bidirectional-fe***.

Tabel 13
(Hasil Evaluasi pada Seluruh Kombinasi Model)

Kombinasi	Skor *BLEU*
1	81.51
2	81.38
3	81.20
4	81.82
5	81.95
6	81.83
7	82.02
8	82.00
9	82.08
10	81.84
11	81.87
12	81.78
13	81.65
14	81.66

Kombinasi	Skor *BLEU*
15	81.29
16	82.16
17	82.00
18	82.02

g.) *Testing*

Pada penelitian ini dilakukan dua cara untuk *testing* hasil normalisasi. Pertama, dengan menguji *decoder* pada data validasi dan kedua menguji *decoder* dengan data baru. Hasil *testing* pada data validasi dapat dilihat pada Tabel 14, sedangkan hasil *testing* dengan data baru dapat dilihat pada Tabel 15. Dari kedua hasil *testing* diperoleh bahwa proses ini memakan waktu selama 15-20 detik tiap melakukan normalisasi satu kalimat dan panjang kalimat tidak memengaruhi lamanya waktu normalisasi. Proses ini dilakukan pada perangkat lokal sehingga sangat bergantung dengan performa CPU yang digunakan.

Tabel 14
(*Testing* dengan Data Validasi)

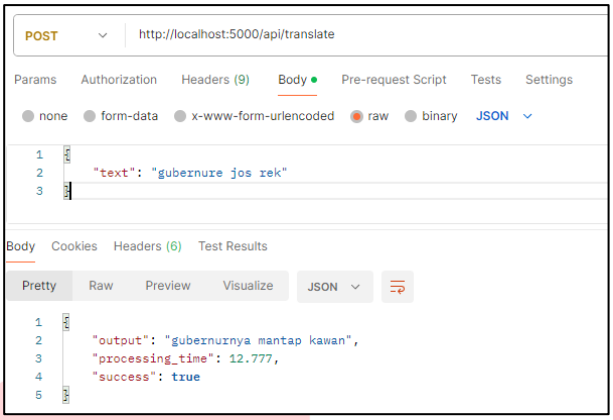
Input	Output	*Ground Truth*	Waktu *Test* (detik)
seneng banget liat pasangan ini xemoticonx xemoticonx	senang sekali melihat pasangan ini xemoticonx xemoticonx	senang sekali melihat pasangan ini xemoticonx xemoticonx	20,18
Manyala mas wagub xemoticonx xemoticonx xemoticonx	menyala Mas wakil gubernur xemoticonx xemoticonx xemoticonx	menyala Mas Wakil Gubernur xemoticonx xemoticonx xemoticonx	17,68

Tabel 15
(*Testing* dengan Data Baru)

Input	Output	Waktu *Test* (detik)
syg ga ada debat gubernur jatim di tv, pengen tau visi misinya	sayang tidak ada debat gubernur Jawa Timur di tv, ingin tahu visi misinya	14,99
yg penting gubernur baru bisa handle kemacetan surabaya deh	yang penting gubernur baru bisa handle kemacetan Surabaya deh	18,56

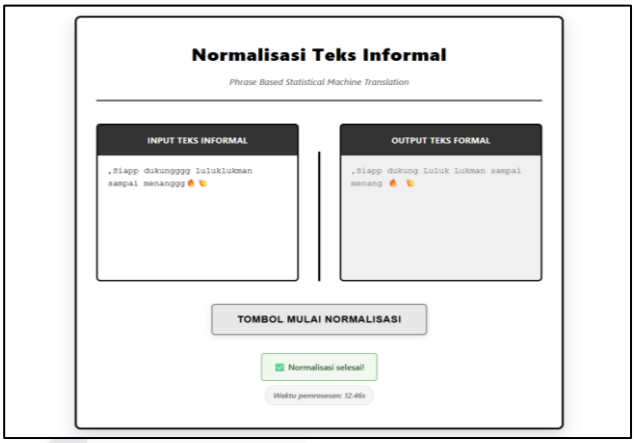
C. Pembangunan *Website*

Integrasi model dengan *website* dilakukan menggunakan *Flask* sebagai *backend* sistem yang menghubungkan model normalisasi. Proses *decoding* memanfaatkan *base model* yang telah dibangun serta skrip *decoding* dari Moses, serupa dengan tahap pengujian menggunakan data baru, namun diimplementasikan dalam bentuk *endpoint /translate* seperti ditunjukkan pada Gambar 4. *Backend* dijalankan tanpa menggunakan *WSL*, karena hanya membutuhkan file model hasil training seperti *moses.ini*, *phrase-table*, dan *reordering-model*.



Gambar 4
(Hasil Respon *Endpoint*)

Setelah model berhasil diintegrasikan, tahap selanjutnya adalah pembuatan antarmuka. Antarmuka dibuat menggunakan JavaScript untuk pemanggilan *API* dan HTML untuk tampilan antarmukanya. Tampilan antarmuka dapat dilihat pada Gambar 5.



Gambar 5
(Tampilan Antarmuka)

D. Pengujian Model

Hasil pengujian pertama difokuskan pada pengaruh jumlah pasangan kata dalam model n-gram terhadap performa *base model*. Berdasarkan Tabel 16, model dengan n-gram 3 memberikan performa terbaik dengan skor *BLEU* sebesar 82,16, disusul oleh n-gram 2 (82,03), n-gram 5 (81,99), dan n-gram 4 (81,86). Hasil ini menunjukkan bahwa peningkatan nilai *n* tidak selalu berbanding lurus dengan peningkatan kualitas hasil normalisasi.

Tabel 16
(Hasil Pengujian Pengaruh Jumlah Pasang Kata pada N-gram Model)

Jumlah Pasang Kata	Skor *BLEU*
2	82,03
3	82,16
4	81,86
5	81,99

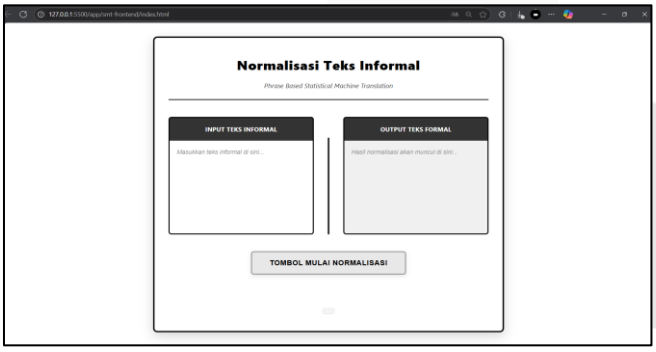
Pengujian kedua difokuskan pada pengaruh jumlah data dalam korpus monolingual terhadap performa *base model*. Tabel 17 menunjukkan hasil pengujian pengaruh jumlah data pada korpus monolingual terhadap skor *BLEU*. Skor *BLEU* mengalami peningkatan seiring bertambahnya jumlah data yang digunakan, dari 81,94 pada 25% data menjadi 82,16 pada 100% data. Hal ini menunjukkan bahwa semakin banyak data yang digunakan dalam *language model training*, maka kemampuan model untuk mengenali pola bahasa formal juga meningkat. Namun, peningkatan tersebut tidak signifikan. Salah satu penyebabnya adalah karena jumlah data pada 25% sudah cukup besar, yaitu sekitar 400 ribu kalimat, sehingga sudah mampu merepresentasikan struktur kebahasaan. Dengan kata lain, ketika data sudah mencapai jumlah tertentu, penambahan data tambahan tidak selalu menghasilkan peningkatan performa yang signifikan.

Tabel 17
(Hasil Pengujian Pengaruh Jumlah Data pada Korpus Monolingual)

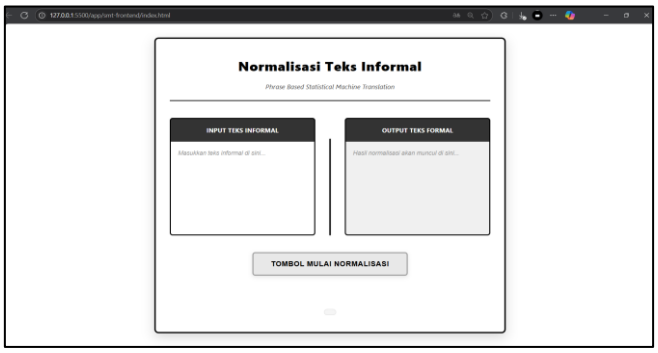
Jumlah Persentase Data	Skor *BLEU*
25	81,94
50	81,97
75	81,98
100	82,16

E. Pengujian *Website*

Pengujian website dilakukan dalam dua skenario. Skenario pertama menguji tampilan antarmuka dengan membuka halaman melalui web browser. Hasilnya, seluruh komponen seperti kotak input, output, dan tombol normalisasi berhasil dimuat dengan baik tanpa error tampilan seperti pada Gambar 6. Skenario kedua menguji fungsi utama, yaitu proses normalisasi teks. Setelah pengguna memasukkan teks informal dan menekan tombol, sistem berhasil mengembalikan hasil normalisasi sesuai output decoder Moses tanpa kendala pengiriman atau penerimaan data melalui API seperti pada Gambar 7.



Gambar 6
(Hasil Pengujian Load Website)



Gambar 7
(Hasil Pengujian Output Normalisasi)

V. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem normalisasi komentar media sosial pasangan calon Gubernur Jawa Timur 2024 menggunakan pendekatan *PBSMT* dengan *framework moses*. Sistem dibangun melalui tahapan lengkap mulai dari *preprocessing*, *language model training*, *translation model training*, *tuning*, evaluasi, hingga implementasi berbasis *website*. Hasil pengujian menunjukkan bahwa model mampu menormalisasi berbagai bentuk bahasa informal, termasuk singkatan, kata tidak baku, dan bentuk kreatif khas media sosial. Evaluasi performa menggunakan metrik BLEU menghasilkan skor 82,16 pada data uji dan 81,04 pada data validasi, menunjukkan kemampuan model dalam melakukan generalisasi dan menghasilkan teks yang sesuai dengan kaidah bahasa Indonesia. Meskipun performa sistem cukup tinggi, terdapat beberapa keterbatasan seperti waktu pemrosesan yang masih lambat (15-20 detik per kalimat) dan ketergantungan terhadap cakupan korpus pelatihan yang tinggi. Namun demikian, hasil penelitian ini membuktikan bahwa pendekatan *PBSMT* berhasil menormalisasi teks pada media sosial.

REFERENSI

[1] H. A. Wibowo *et al.*, “Semi-Supervised Low-Resource Style Transfer of Indonesian Informal to Formal Language with Iterative Forward-Translation,” *2020 International Conference on Asian Language Processing, IALP 2020*, pp. 310–315, 2020, doi: 10.1109/IALP51396.2020.9310459.

[2] K. C. Eleison, S. U. I. Hutahaean, S. C. Tampubolon, T. M. Panggabean, and I. Fitriyaningsih, “An empirical evaluation of phrase-based statistical machine translation for Indonesia slang-word translator,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 3, pp. 1803–1813, 2022, doi: 10.11591/ijeecs.v25.i3.pp1803-1813.

[3] A. Kurnia and E. Yulianti, “Statistical Machine Translation Approach for Lexical Normalization on Indonesian Text,” pp. 288–293, 2020.

[4] A. A. T. Hossain, F. A. Bhuiyan, and Md. A. Islam, “Creating an Efficient Parallel Corpus for Bangla-English Statistical Machine Translation,” *EPRA International Journal of Multidisciplinary Research*

(IJMR)-Peer Reviewed Journal, no. 2, pp. 198–210, 2020, doi: 10.36713/epra2013.

[5] Z. Z. Linn, Y. K. Thu, and P. B. Patil, "Statistical machine translation between myanmar and myeik," *WCSE 2020: 2020 10th International Workshop on Computer Science and Engineering*, vol. 4, no. April, pp. 547–556, 2020, doi: 10.18178/wcse.2020.02.007.

[6] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," *EMNLP 2008 - 2008 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference: A Meeting of SIGDAT, a Special Interest Group of the ACL*, no. October, pp. 848–856, 2008, doi: 10.3115/1613715.1613824.

[7] Y. Liu, K. Wang, C. Zong, and K. Y. Su, "A unified framework and models for integrating translation memory into phrase-based statistical machine translation," *Comput Speech Lang*, vol. 54, pp. 176–206, 2019, doi: 10.1016/j.csl.2018.09.006.

[8] P. Koehn *et al.*, "Moses: Open source toolkit for statistical machine translation," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, no. May 2014, pp. 177–180, 2007.

[9] C. P. Chai, "Comparison of text preprocessing methods," *Nat Lang Eng*, vol. 29, no. 3, pp. 509–553, 2023, doi: 10.1017/S1351324922000213.

[10] G. E. Pibiri and R. Venturini, "Handling massive n-gram datasets efficiently," *ACM Trans Inf Syst*, vol. 37, no. 2, 2019, doi: 10.1145/3302913.

[11] M. Zhang, "On Applying Natural Language Processing Technology to Optimize Accuracy of English Interactive Translation Platform," *Applied Mathematics and Nonlinear Sciences*, vol. 9, no. 1, pp. 1–23, 2024, doi: 10.2478/amns-2024-1524.

[12] A. Araabi and V. Niculae, "How Effective is Byte Pair Encoding for Out-Of-Vocabulary Words in Neural Machine Translation," 2022.