Jurnal Elektronik Ilmu Komputer Udayana
Volume 14, No 3. February 2026

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Penentuan Entitas Tokoh Pada Satua Bali Menggunakan Algoritma Conditional Random Fields

I Made Widi Arsa Ari Saputra[a1], I Ketut Gede Suhartana[a2], I Gusti Ngurah Anom Cahyadi Putra[b3], I Wayan Supriana[b3]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus Unud, Jimbaran,Bali, 80361, Indonesia
[1]widiarsamade@gmail.com
[2] wayan.supriana@unud.ac.id
[2]anom.cp@unud.ac.id
[2] ikg.suhartana@unud.ac.id

Abstrak

Pengenalan Entitas Bernama (NER) adalah proses untuk mengidentifikasi dan mengklasifikasi Entitas Bernama (NEs) pada suatu teks. Penelitian sebelumnya melakukan pengenalan entitas bernama pada bahasa bali menggunakan metode rule-based yang berfokus pada kelas person memberikan akurasi yang kurang memadai, karena data yang digunakan berfokus kepada fitur semantik dari kata pada kalimat(Kurniadi & ER, 2021). Tujuan dari penelitian ini adalah menerapkan fitur gramatikal dari kata berupa label part-of-speech, dan juga fitur lainnya untuk meningkatkan akurasi pada proses pengenalan entitas bernama yang berfokus pada entitas person. Metode Conditional Random Fields merupakan metode diskriminatif yang hanya berfokus kepada persebaran probabilitas bersyarat pada fitur ke label. Penelitian ini menggunakan 4 fitur, diantaranya: fitur semantik, fitur gramatikal, fitur transisi, serta fitur representasi embedding. Penelitian ini menggunakan data dari penelitian (Bimantara et al ea, 2024) yang telah dilabeli dengan label part-of-speech dan entitas bernama dengan skema labeling beginning-inside-outside (BIO). Penelitian ini menghasilkan accuracy 0.96 dengan precision sebesar 0.96, recall sebesar 0.96, dan total f1-score sebesar 0.96.

Keywords: *conditional random fields, pengenalan entitas bernama, bahasa bali, regularisasi, satua bali*

1. Pendahuluan

Bahasa Bali merupakan salah satu bahasa daerah di negara Indonesia, keberadaan bahasa Bali perlu dilestarikan, baik dalam bidang pendidikan maupun melalui teknologi digital [1]. Satua bali merupakan salah satu kebudayaan kesusastraan bali purwa pada pulau Bali. Satua bali dikenal sebagai sastra lisan yang menyebar dari mulut ke mulut tanpa diketahui pencipta aslinya [2]. Satua bali berperan dalam pembentukan karakter anak-anak di Bali, karena Satua bali sangat erat isinya dengan pendidikan moral karakter. Dalam analisis suatu cerita satua bali terdapat berbagai point analisis salah satunya adalah analisis penokohan.

Untuk melakukan analisis penokohan ini, tentunya perlu untuk mencari semua karakter yang terlibat pada suatu satua bali. Salah satu cara untuk mengekstraksi tokoh dari suatu satua bali adalah dengan melakukan ekstraksi informasi [3]. Untuk melakukan ekstraksi informasi diperlukan beberapa komponen, seperti parsing sintaksis, ekstraksi entitas, ataupun ekstraksi relasi. Ekstraksi informasi dokumen dalam bahasa Bali akan digunakan untuk mengekstraksi dokumen Satua Bali. Selain itu, NER yang dibuat dapat digunakan sebagai fitur preprocessing teks Bali yang memudahkan digitalisasi bahasa Bali yang dapat diakses dan tidak akan hilang.

Pengenalan entitas bernama (NER) adalah salah satu komponen penting dalam pengolahan bahasa alami (NLP) [4]. Istilah "Named Entity" telah banyak digunakan dalam bidang *information extraction* (IE), *question answering* (QA), dan berbagai bidang NLP lainnya [5]. NER merupakan langkah awal dalam ekstraksi informasi yang bertujuan untuk mengidentifikasi entitas yang disebutkan dalam teks dan mengkategorikannya ke dalam kategori yang telah ditentukan sebelumnya.

Conditional random fields (CRF) merupakan salah satu model probabilistik untuk melakukan segmentasi dan tagging suatu urutan data [6]. CRF hadir untuk mengatasi permasalahan ketergantungan asumsi yang besar pada model hidden markov (HMM) [7][8]. CRF dapat menentukan sendiri seberapa banyak fitur yang akan digunakan dalam suatu model CRF, berbeda dengan model HMM yang bersifat lokal dimana setiap kata (fitur) hanya bergantung pada label saat ini dan setiap label sebelumnya .CRF juga mengatasi permasalahan adanya label bias pada model maximum entropy markov (MEMM) karena distribusi kondisional label dari urutan data pada model CRF dilakukan secara keseluruhan dibandingkan model MEMM yang memformulasikan distribusi kondisional label hanya untuk setiap elemen data [9]. CRF banyak digunakan pada permasalahan natural language processing, computer vision, dan bioinformatic. salah satu permasalahan dari natural language processing adalah pengenalan entitas bernama (NER).

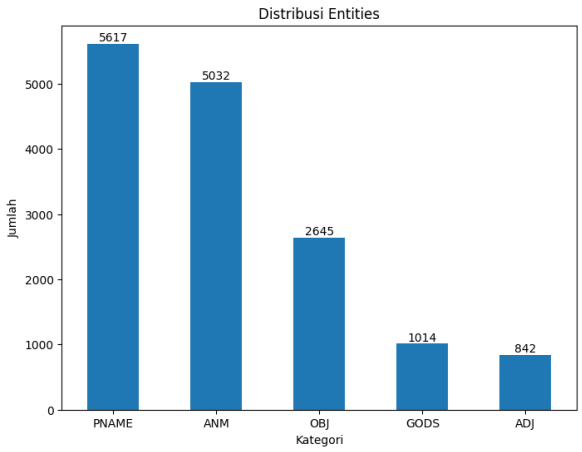
Pada penelitian yang dilakukan oleh Ekbal pada tahun 2007 [10] mengajukan pendekatan pada sistem pengenalan entitas bernama menggunakan algoritma hidden markov model untuk mengenali entitas nama pada teks berbahasa Bengali dan Hindi dan mendapatkan hasil f-score 84.50% dan 78.35%. kemudian beliau mengajukan kembali pendekatan tentang pengenalan entitas bernama menggunakan algoritma conditional random fields [11] dan mendapatkan hasil f-score 81.15%, dan 78.29% Sehingga, dari kedua penelitian tersebut beliau menyimpulkan hidden markov model lebih efektif digunakan dibanding conditional random fields. Kesimpulan ini dipertegas pada penelitian dari Chopra pada tahun 2016 [12], beliau melakukan pengenalan entitas bernama menggunakan algoritma hidden markov model pada bahasa hindi dan mendapatkan hasil f-score 97.14%. sehingga, hidden markov model menunjukan efisiensi lebih pada proses pengenalan entitas nama dibanding algoritma conditional random fields.

Namun, pada penelitian yang dilakukan oleh Jaariyah pada tahun 2017 [9], penelitian ini menggunakan algoritma conditional random fields untuk melakukan pendekatan entitas nama untuk mengenali entitas bernama pada teks berbahasa Indonesia dan mendapatkan hasil f-score 90.06%. Sedangkan pada penelitian yang dilakukan oleh Yusliani pada tahun 2020 [8], beliau mengajukan algoritma hidden markov model dalam pengenalan entitas bernama pada teks berbahasa indonesia dan mendapat f-score 86.14%. sehingga pada bahasa indonesia, algoritma conditional random fields justru lebih efektif untuk melakukan named entity recognition dibandingkan dengan algoritma hidden markov model.

2. Metode Penelitian

Penelitian ini menggunakan metode Conditional Random Fields untuk melakukan tugas pengenalan entitas bernama yang berfokus pada entitas person. Conditional Random Fields terdiri dari 3 proses: Penghitungan nilai potensial, estimasi parameter, dan inferensi.

2.1 Pengumpulan Data

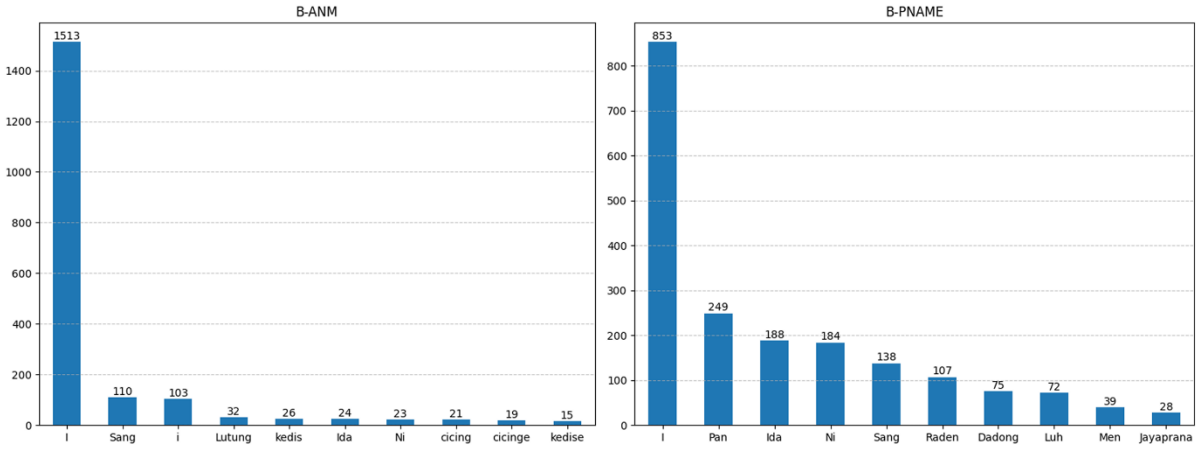


Gambar 1. Distribusi Label Dataset

Pada proses pengumpulan data, data yang akan digunakan berupa data teks berbahasa Bali seperti satua bali. Data satua bali akan diperoleh dari penelitian yang pernah dilakukan oleh (Bimantara et al ea, 2024). Terdapat 21 kelas kata yang digunakan berdasarkan penelitian sebelumnya [13] dan 11 label entitas tokoh dengan pola penamaan BIO [14]. Data ini berjumlah 89.917 fitur kata lengkap

p-ISSN: 2301-5373
e-ISSN: 2654-5101

dengan kelas kata dan label entitas bernama dengan jumlah label tokoh sebanyak 15.150 kata. Dari 15.150 kata ini, terlihat label PNAME dan ANM memberikan dominasi besar pada satua bali sebagai entitas tokoh, dapat dilihat pada **gambar 1**. Label PNAME pada dataset memiliki karakteristik penamaan daerah bali, seperti awalan I, Pan, Ida, Ni dan Sang. seperti pada **gambar 2**, atau Sang sehingga hal ini akan menjadi karakteristik dataset saat ini, dan model yang dilatih kemungkinan akan bergantung dengan keadaan ini. Label ANM juga memberikan distribusi yang besar pada dataset, mengindikasikan banyaknya tokoh binatang pada dataset saat ini, sehingga akan menjadi tantangan pada pelatihan model untuk membedakan binatang sebagai tokoh dalam cerita atau sebagai tokoh figuran.



Gambar 2. Distribusi Awalan Kata dari Tokoh pada Dataset

2.2 Perancangan Sistem

Sistem dari penelitian ini akan dimulai dengan melakukan pra-pemrosesan data untuk membersihkan data dan mengganti label sesuai keperluan penelitian. Kemudian dilakukan rekayasa fitur agar bentuk data dapat diproses kedalam model. Setelah bentuk data sudah sesuai, barulah dilakukan training menggunakan algoritma CRF dengan metode evaluasi grid-search cross validation.

1. Pra-pemrosesan data
   Data yang telah didapatkan kemudian dilakukan pra-pemrosesan berupa mengganti label B-OBJ dan I-OBJ menjadi O, dan entitas lain diruubah menjadi B-PER. Kemudian melakukan rekayasa fitur dengan merubah bentuk setiap observasi data kedalam bentuk vektor fitur, rekayasa ini merupakan hiperparameter sehingga bentuk dan jumlahnya dapat menyesuaikan dengan domain, tugas, dan keadaan dataset.

2. Rekayasa Fitur
   Fitur yang digunakan untuk melakukan training pada model CRF disini sejumlah 48 fitur. Tabel 1 memetakan vektor fitur yang digunakan pada penelitian ini sejumlah 4 tipe vektor, yaitu: vektor emisi sebagai identitas emisi dari setiap observasi kata, vektor transisi sebagai identitas transisi dari kata dan jendelanya, vektor pos sebagai identitas kelas kata dari observasi kata, dan vektor embedding sebagai identitas embedding dari kata. Ke 48 fitur ini akan diterapkan untuk setiap observasi kata sebagai representasi vektor dari data sebelum masuk ke model. Fitur transisi mendapatkan porsi jumlah yang paling banyak, yaitu dengan 20 fitur, untuk menangkap dengan jelas karakteristik urutan kata pada kalimat dalam satua bali. Hal ini penting agar model dapat mempelajari pola kemunculan dari suatu entitas tokoh diberikan suatu kalimat dalam satua bali. Karena ambiguitas tokoh dalam satua bali terutama pada tokoh binatang yang terkadang mejadi tokoh figuran, sehingga pemodelan fitur transisi dirasa penting untuk dilakukan agar model dapat mengenali tokoh utama dan tokoh figuran.

Tabel 1. Fitur Vektor

tipe	fitur	jumlah	deskripsi
emisi	bias	1	bobot dari bias yang digunakan adalah 1.0 untuk setiap fitur
	word	1	representasi karakter dari kata saat ini
	is_first	1	boolean untuk menandakan emisi awal dari suatu kalimat
	is_last	1	boolean untuk menandakan emisi akhir dari suatu kalimat
	islower	1	boolean untuk menandakan kata saat ini merupakan berhuruf kecil
	prefix	2	karakter awal dari kata saat ini dengan jendela 2
	suffix	2	karakter terakhir dari kata saat ini dengan jendela 2
	isupper	1	boolean untuk melihat kapitalisasi dari kata saat ini
	is_title	1	boolean untuk melihat kapitalisasi dari huruf pertama kata saat ini
transisi	prev_word	2	representasi kata sebelumnya dari kata saat ini dengan jendela 2
	next_word	2	representasi kata setelahnya dari kata saat ini dengan jendela 2
	prev_word_prefix	4	prefix dari kata sebelumnya yang masing masing dengan jendela 2
	prev_word_suffix	4	suffix dari kata sebelumnya yang masing masing dengan jendela 2
	next_word_prefix	4	prefix dari kata setelahnya yang masing masing dengan jendela 2
	next_word_suffix	4	suffix dari kata setelahnya yang masing masing dengan jendela 2
pos	pos	1	label kelas kata dari kata saat ini
	prev_pos	2	label kelas kata dari kata sebelumnya dengan jendela 2
	next_pos	2	label kelas kata dari kata setelahnya dengan jendela 2
embedding	emb_word	1	embedding vektor dari kata saat ini
	emb_prev_word	2	embedding vektor dari kata sebelumnya dengan jendela 2
	emb_next_word	2	embedding vektor dari kata setelahnya dengan jendela 2
	emb_prev_word_prefix	2	embedding vektor dari prefix kata sebelumya dengan jendela 2
	emb_prev_word_suffix	2	embedding vektor dari suffix kata sebelumya dengan jendela 2
	emb_next_word_prefix	2	embedding vektor dari prefix kata setelahnya dengan jendela 2
	emb_next_word_suffix	2	embedding vektor dari suffix kata setelahnya dengan jendela 2

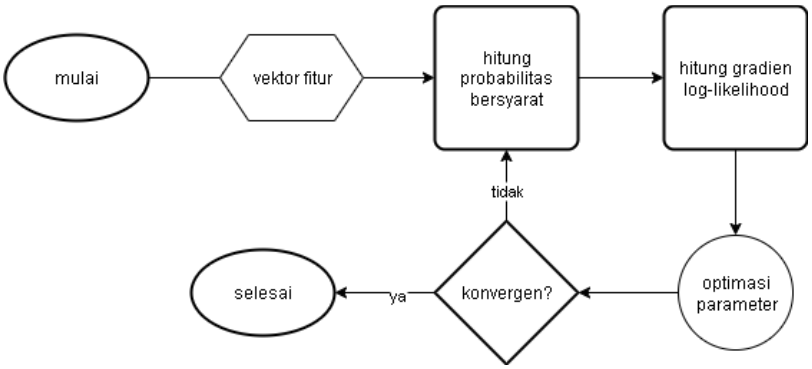
3. Pelatihan

Seperti yang terlihat pada **gambar 3**, proses pelatihan CRF diawali dengan parameter vektor fitur hasil dari rekayasa fitur. Kemudian dilakukan penghitungan probabilitas bersyarat untuk seluruh data pada training dataset.

$$p_\theta(\mathbf{y} \mid \mathbf{x}) = \frac{\prod_{i=1}^{n+1} M_i(\mathbf{y_{i-1}}, \mathbf{y_i} \mid \mathbf{x})}{(\prod_{i=1}^{n+1} M_i(\mathbf{x}))_{\text{start, stop}}} \quad (1)$$

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Numerator dari fungsi merupakan nilai potensial tiap-tiap observasi fitur (fungsi fitur setiap kata) dalam ruang lingkup kalimat saat ini. Sedangkan denominator merupakan fungsi partisi yang bertugas untuk menormalisasi probabilitas agar berjumlah 1. Z(x) atau fungsi partisi, adalah nilai dot produk untuk seluruh urutan label yang mungkin terhadap data observasi (x1, x2, x3,...,xn) yang diberikan. apabila jumlah data observasi merupakan n, sedangkan jumlah label merupakan Y, maka jumlah kombinasi urutan label yang mungkin terjadi adalah nY urutan label.



Gambar 3. Alur Pelatihan

Setelah itu, dihitung *log-likelihood* dari probabilitas bersyarat yang akan digunakan untuk menghitung nilai gradien log-likelihood yang diperlukan untuk mengetahui arah perubahan nilai dari parameter λ.

$$\mathcal{L}(\theta) = -\left(\sum_{i=1}^{n+1} \log M_i\,(y_{i-1}, y_i \mid \boldsymbol{x}) - \log Z\,(\boldsymbol{x})\right) + \alpha|\theta|_1 + \frac{\beta}{2}|\theta|_2^2 \qquad (2)$$

Setelah gradien didapatkan barulah optimasi parameter menggunakan algoritma L-BFGS.

Algoritma Limited-memory BFGS (L-BFGS) merupakan metode optimisasi numerik berbasis gradien yang digunakan untuk memaksimalkan log-likelihood dalam pelatihan model Conditional Random Fields (CRF), L-BFGS menggunakan pendekatan terbatas memori dengan menyimpan sejumlah vektor gradien dan parameter dari iterasi sebelumnya.
Alur proses:

a. Inisialisasi parameter bobot λ_0.
b. Hitung gradien awal, dari log-likelihood sebelumnya
c. Hitung perubahan nilai lambda dan gradien
d. Perbarui Hessian matriks
e. Tentukan arah pencarian d_k menggunakan estimasi inverse Hessian
f. Lakukan line search untuk menemukan panjang langkah optimal α_k yang meminimalkan fungsi objektif di sepanjang arah d_k .
g. Perbarui parameter: $\lambda_{k+1} = \lambda_k + \alpha_k\, d_k$
h. Perbarui memori terbatas: Simpan perubahan gradien Δg_k dan parameter $\Delta\lambda_k$.
i. Ulangi langkah 2–6 hingga konvergen

4. Metode Pengujian
Pada penelitian ini uji coba parameter akan dilakukan dengan metode grid-search cross validation. Tujuannya adalah mencari kombinasi parameter terbaik untuk model, serta menghindari bias pada model dengan menggunakan cross-validation. Alur dari uji coba parameter sebagai berikut:

a. Menentukan nilai hiperparameter yang akan diuji
Hiperparameter yang akan diuji disini adalah regularisasi yang digunaka pada algoritma Conditional Random Fields yaitu L1 dan L2. Nilai L1 akan dieksplorasi pada

rentang 10^{-5} hingga 10^{-1} dengan skala logaritmik basis 10 dengan kenaikan 1 pada eksponen, ini juga berlaku pada L2.

b. Membentuk Grid Kombinasi
Pada tahap ini dilakukan penyusunan kombinasi 2 parameter yang diuji yaitu L1 dan L2. Dengan 5 nilai untuk setiap parameter, berarti akan ada 25 kombinasi yang akan diujikan.

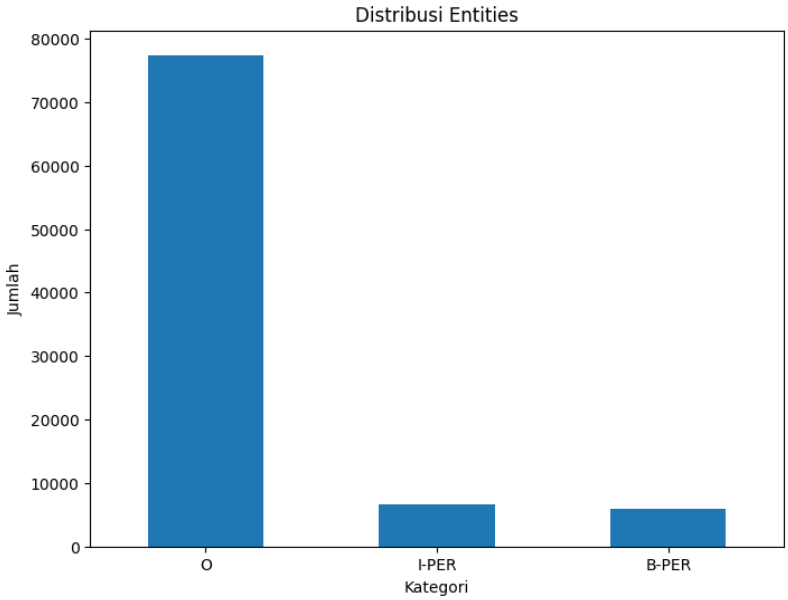
c. Menerapkan Cross-Validation
Dataset akan dibagi dengan 5 lipatan, sehingga pembagian pelatihan dan pengujian adalah 4:1. Untuk setiap kombinasi parameter akan dilatih dengan k-1 lipatan, sedangkan pengujian akan dilakukan pada lipatan yang tersisa. Proses ini akan diulangi hingga semua lipatan digunakan sebagai data uji. Kemudian setelah semua lipatan telah digunakan sebagai data uji, akan dilakukan perhitungan rata-rata akurasi dari semua lipatan yang merepresentasikan akurasi optimal pada pengujian kombinasi saat ini.

d. Memilih kombinasi terbaik
Kombinasi terbaik dipilih berdasarkan akurasi tertinggi. Kemudian model akan disimpan untuk dilakukan deployment.

3. Hasil dan Pembahasan

3.1 Data Latih

Data ini saya saring dengan mengganti label B-OBJ dan I-OBJ menjadi O, dan entitas lain saya ubah labelnya menjadi B-PER dan I-PER. Penggantian ini dilakukan untuk berfokus hanya pada entitas tokoh, sehingga label lain seperti “ANIMAL” dan “GOD” dianggap juga sebagai tokoh, label “OBJECT” tidak dibutuhkan karena ini merepresentasikan kata ganti. Data latih yang telah melalui proses pra-pemrosesan data dengan 3 label yaitu B-PNAME, I-PNAME, dan O. Rekayasa fitur dilakukan untuk setiap kata dan disimpan dalam array 2D, yang dimensi pertama merupakan kalimat dari setiap dataset. Dataset ini terdiri dari 6.6315 kalimat yang disimpan dalam array dengan kelas kata dan label entitasnya. Pada **gambar 4** ditunjukkan distribusi dari label B-PER, I-PER dan juga O.



Gambar 4. Distribusi Label Data Latih

p-ISSN: 2301-5373
e-ISSN: 2654-5101

3.2 Hasil

Implementasi CRF menggunakan library sklearn-crfsuite yang sudah secara default mengimplementasikan algoritma L-BFGS. Pelatihan model CRF menggunakan regularisasi L1 beserta L2. L1 regularization mendorong model untuk menghasilkan parameter yang lebih jarang (sparse), artinya beberapa fitur akan memiliki bobot nol sehingga diabaikan dalam model. L2 regularization mencegah overfitting dengan mendorong parameter model untuk tidak menjadi terlalu besar. Ini mengurangi sensitivitas model terhadap data pelatihan yang bervariasi. L2 regularization lebih fokus pada distribusi parameter daripada memaksakan sparsity.

Table 2. Hasil Grid-Search Cross Validation

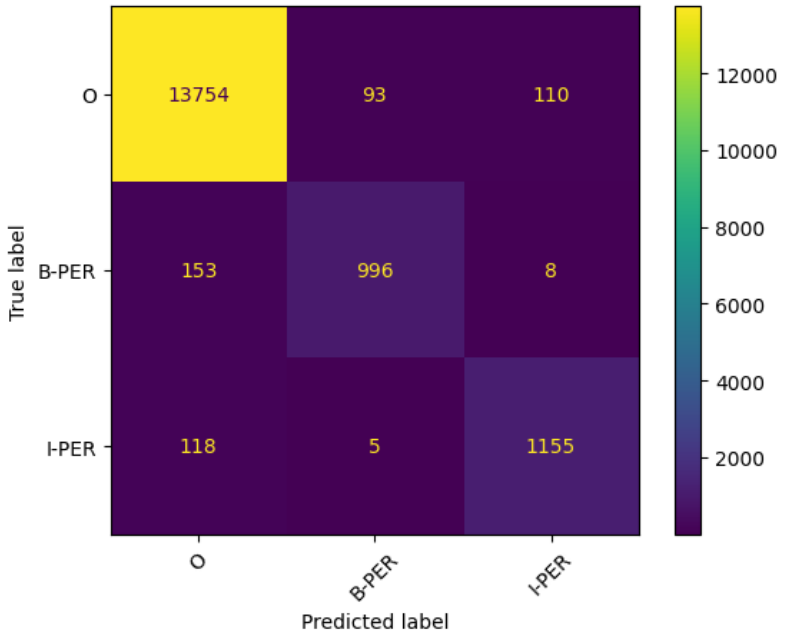
L1	L2	F1-score	accuracy	precision	recall
0.0001	0.0001	0.965360165	0.965606426	0.965258024	0.965606426
	0.001	0.966242487	0.966510067	0.96612566	0.966510067
	0.01	0.967347341	0.967619312	0.967255172	0.967619312
	0.1	0.967313309	0.96763744	0.967216799	0.96763744
0.001	0.0001	0.967051205	0.967236454	0.966973224	0.967236454
	0.001	0.966903442	0.967136791	0.966805971	0.967136791
	0.01	0.967234176	0.967497685	0.967135744	0.967497685
	0.1	0.967541741	0.967853787	0.967444056	0.967853787
0.01	0.0001	0.968308431	0.968503797	0.968219608	0.968503797
	0.001	0.968772897	0.968942969	0.968703175	0.968942969
	0.01	0.967893083	0.968106581	0.967816634	0.968106581
	0.1	0.967696585	0.968000707	0.967604502	0.968000707
0.1	0.0001	0.968594696	0.968783144	0.968510977	0.968783144
	0.001	0.968104334	0.968282621	0.96802272	0.968282621
	0.01	0.968594696	0.968783144	0.968510977	0.968783144
	0.1	0.968411335	0.968611665	0.968321444	0.968611665

Pada **tabel 2,** hasil evaluasi menunjukkan bahwa kombinasi L1 = 0.01 dan L2 = 0.001 menghasilkan kinerja terbaik, ditunjukkan oleh skor F1 tertinggi sebesar 0.968772897. Kombinasi ini juga menghasilkan nilai akurasi, presisi, dan recall yang konsisten tinggi, menandakan bahwa model tidak hanya mampu mengenali entitas secara tepat, tetapi juga konsisten dalam meminimalkan kesalahan deteksi.
Secara umum, perunurnan nilai L1 justru menurunkan kinerja model. Hal ini terlihat dari penurunan bertahap pada skor F1 ketika nilai L1 diturunkan hingga 0.0001, yang mengindikasikan bahwa regularisasi yang terlalu rendah kurang mampu memberikan penalti untuk model dalam mempelajari fitur yang tidak relevan.

Tabel 3. Matriks Evaluasi Model Terbaik L1=0.01 & L2=0.001

	precission	recall	F1-score
O	0.98	0.99	0.98
B-PER	0.91	0.86	0.88
I-PER	0.91	0.90	0.91

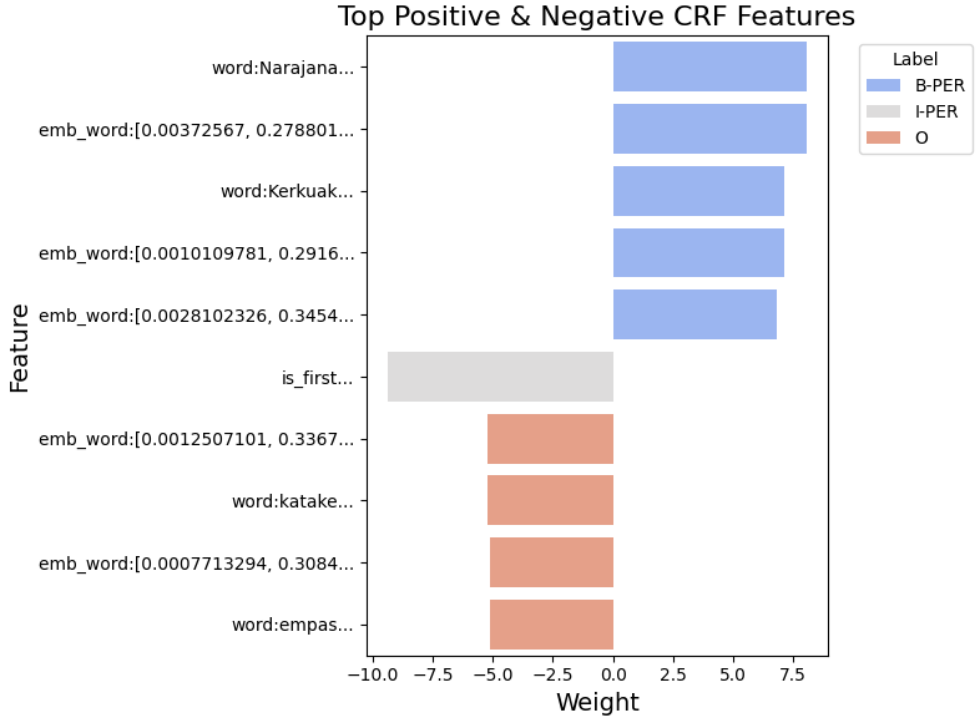
Pada **tabel 3** terlihat jelas model sangat percaya terhadap pengenalan entitas tidak bernama (O) dengan angka precission yang sangat tinggi. Sedangkan untuk label entitas bernama terbilang memiliki nilai yang cukup rendah, nilai 0.86 pada recall dari label B-PER menandakan model hanya mampu mengenali 86% dari keseluruhan entitas bernama yang ada pada data test. Begitu pula pada label I-PER yang mendapatkan persentase 90% untuk recall pada model.



Gambar 5. Confusion Matrix Model Terbaik

Pada **gambar 5** dapat dilihat, sebanyak 271 kali model gagal mengenali entitas bernama dan cenderung memberikan label O. Kecilnya angka kesalahan prediksi pada label B-PER antar I-PER menunjukkan model cukup mampu membedakan 2 entitas tokoh yang berbeda pada suatu kalimat. Walaupun seringkali model cenderung memberikan label I-PER pada observasi yang seharusnya bukan tokoh (O) sebanyak 110 kali, artinya model masih kurang mampu memprediksi akhir dari nama tokoh pada suatu kalimat.

Model CRF yang dilatih menghasilkan 24.452 bobot secara total pada model. Jumlah ini merupakan jumlah dari seluruh nilai yang mungkin pada fitur vektor dikombinasikan dengan setiap label yang ada. Pada **gambar 6** terlihat bahwa bobot tertinggi dan terendah pada model CRF yang dilatih sebagian besar berasal dari fitur word_embedding dan word itu sendiri, sedangkan fitur lain seperti transisi, kelas kata (POS), suffix, prefix, emisi, dan bias tidak muncul dalam daftar bobot ekstrem.



Gambar 6. Lima Bobot Tertinggi dan Terendah Model

p-ISSN: 2301-5373
e-ISSN: 2654-5101

3.3 Pembahasan

Dari hasil yang didapatkan menunjukkan beberapa hal penting:

1. Dominasi fitur leksikal dan embedding:
    a. Fitur word (kata itu sendiri) dan word_embedding memberikan sinyal yang paling kuat bagi model untuk memprediksi entitas, khususnya label B-PER dan O.
    b. Bobot positif tinggi pada kata-kata spesifik atau embedding tertentu menunjukkan fitur ini sangat menentukan prediksi label tertentu.
    c. Bobot negatif tinggi pada fitur embedding atau kata lain menandakan fitur itu menekan prediksi label tertentu, sehingga berperan sebagai penyeimbang.
2. Fitur transisi, kelas kata dan bias kurang berpengaruh pada bobot ekstrem:
    Walaupun CRF menggunakan transisi antar label dan bias, bobotnya tidak terlalu ekstrem, sehingga tidak muncul di 5 bobot teratas positif atau negatif. Fitur transisi sendiri bersifat penyesuaian global antar label, bukan determinan utama per token sehingga wajar tidak memberikan beban kuat. Baik transisi antar kata ataupun transisi antar kelas kata.
3. Fitur kelas kata, suffix, prefix, dan emisi:
    a. Fitur-fitur ini tampaknya memiliki bobot moderat, dekat dengan rata-rata, sehingga tidak terlalu memengaruhi prediksi secara ekstrem.
    b. Bisa berarti fitur-fitur ini berguna untuk penyesuaian halus atau menangkap pola umum, tetapi tidak sekuat kata spesifik atau embedding dalam membedakan label.

4. Kesimpulan

Berdasarkan hasil eksperimen dan evaluasi terhadap model Conditional Random Fields (CRF) dalam mendeteksi entity "person" pada teks berbahasa Bali (Satua Bali), diperoleh beberapa temuan penting. Akurasi terbaik yang berhasil dicapai oleh model adalah sebesar 96%, dengan skor F1 yang juga tinggi yaitu 96%. Hasil ini menunjukkan bahwa algoritma CRF mampu secara efektif mempelajari pola-pola linguistik yang relevan dalam teks untuk mengidentifikasi entitas nama orang.

Melalui proses tuning parameter regularisasi, kombinasi terbaik untuk parameter L1 dan L2 adalah pada nilai L1 = 0.01 dan L2 = 0.001. Kombinasi ini memberikan keseimbangan yang optimal antara sparsity dan generalisasi model, sehingga model mampu mempertahankan akurasi tinggi sekaligus menghindari overfitting. Temuan ini mengindikasikan bahwa pemilihan parameter regularisasi yang tepat memiliki pengaruh signifikan terhadap performa akhir model CRF.

Untuk penelitian selanjutnya, terdapat beberapa hal yang dapat dipertimbangkan. Pertama, penggunaan teknik preprocessing lanjutan seperti normalisasi kata dan pemanfaatan fitur linguistik tambahan (contohnya: fitur morfologi atau struktur kalimat) dapat ditelusuri lebih lanjut untuk meningkatkan akurasi model. Kedua, karena CRF bersifat supervised, pengumpulan dataset beranotasi yang lebih besar dan lebih bervariasi dapat membantu memperkuat kemampuan generalisasi model pada berbagai jenis teks berbahasa Bali.

Selain itu, akan sangat menarik untuk membandingkan performa CRF dengan pendekatan berbasis neural sequence labeling, seperti BiLSTM-CRF atau transformer-based models (misalnya BERT fine-tuned untuk NER), guna mengevaluasi keunggulan relatif dari pendekatan tradisional dan modern dalam pemrosesan bahasa daerah seperti Bali. Terakhir, implementasi model dalam aplikasi nyata seperti digitalisasi cerita rakyat Bali juga sangat direkomendasikan sebagai bentuk kontribusi terhadap pelestarian bahasa dan budaya lokal.

Daftar Pustaka

[1] I. K. Mustika, 'Pergeseran Bahasa Bali sebagai Bahasa Ibu di Era Global (Kajian Pemertahanan Bahasa)', Purwadita, vol. 2, no. 1, pp. 94–102, 2018, [Online]. Available: http://jurnal.stahnmpukuturan.ac.id/index.php/Purwadita/article/view/26.
[2] I. W. P. A. Wiguna, I. K. R. Arthana, and I. M. Putrama, 'Pengembangan Game Edukasi Satua Bali "Pan Cubling" Berbasis Android', J. Nas. Pendidik. Tek. Inform., vol. 6, no. 2, p. 192, 2017, doi: 10.23887/janapati.v6i2.11440.
[3] K. Kurniadi and N. A. S. ER, 'Person Named Entity Recognition in Balinese', JELIKU (Jurnal Elektron. Ilmu Komput. Udayana), vol. 10, no. 1, p. 99, 2021, doi: 10.24843/jlk.2021.v10.i01.p13.
[4] W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, 'Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs', Procedia Comput. Sci., vol. 135, pp. 425–432, 2018, doi: 10.1016/j.procs.2018.08.193.
[5] A. S. Wibawa and A. Purwarianti, 'Indonesian Named-entity Recognition for 15 Classes Using Ensemble Supervised Learning', Procedia Comput. Sci., vol. 81, no. May, pp. 221–228, 2016, doi: 10.1016/j.procs.2016.04.053.
[6] S. Briandoko, A. Ratna Dewi, M. Akbar Setiawan, and S. Widya Utama, 'Perbandingan Algoritma Conditional Random Field dan Hidden Markov Model pada Pos Tagging Bahasa Indonesia', J. Tek., vol. 2, no. 2, pp. 2598–294, 2018.
[7] I. N. A. R. Arintyo Archamadi, Rita Magdalena, 'Analisis dan Simulasi Identifikasi Judul Lagu dari Senandung Manusia Menggunakan Ekstraksi Ciri DCT (Discreate Cosine Transform)', vol. 3, no. 3, pp. 4575–4584, 2016.
[8] N. Yusliani, M. R. P. Sufa, A. Firdaus, Abdiansah, and S. Yoppy, 'Named-Entity Recognition Pada Teks Berbahasa Indonesia Menggunakan Metode Hidden Markov Model Dan Part-of-Speech Tagging', Linguist. Komputasional, vol. 4, no. 1, pp. 13–20, 2020.
[9] N. Jaariyah and E. Rainarli, 'Conditional Random Fields Untuk Pengenalan Entitas Bernama Pada Teks Bahasa Indonesia', Komputa J. Ilm. Komput. dan Inform., vol. 6, no. 1, pp. 29–34, 2017, doi: 10.34010/komputa.v6i1.2474.
[10] A. Ekbal and S. Bandyopadhyay, 'A Hidden Markov Model based named entity recognition system: Bengali and Hindi as case studies', Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4815 LNCS, pp. 545–552, 2007, doi: 10.1007/978-3-540-77046-6_67.
[11] A. Ekbal and S. Bandyopadhyay, 'A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi', Linguist. Issues Lang. Technol., vol. 2, no. 1, Nov. 2009, doi: 10.33011/lilt.v2i.1203.
[12] D. Chopra, N. Joshi, and I. Mathur, 'Named entity recognition in Hindi using hidden markov model', Proc. - 2016 2nd Int. Conf. Comput. Intell. Commun. Technol. CICT 2016, no. Table I, pp. 581–586, Aug. 2016, doi: 10.1109/CICT.2016.121.
[13] I. M. S. Bimantara, D. Purwitasari, N. A. S. ER, and P. G. S. Natha, 'Balinese story texts dataset for narrative text analyses', Data Br., vol. 56, 2024, doi: 10.1016/j.dib.2024.110781.

[4] Sang, E. F., & Buchholz, S. (2000). Introduction to the CoNLL-2000 Shared Task: Chunking. In Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning (CoNLL), 127-132.
[5] Explosion AI. (2022). spaCy: Industrial-strength Natural Language Processing in Python. [Online]. Available: https://spacy.io/
[6] Haddi, E., Liu, X., & Shi, Y. (2019). Deep Learning for Natural Language Processing. Springer.