

# Serverless Named Entity Recognition untuk Teks Instruksional Pertanian Kota

<http://dx.doi.org/10.28932/jutisi.v8i3.5447>

Riwayat Artikel

Received: 23 September 2022 | Final Revision: 21 November 2022 | Accepted: 21 November 2022

Creative Commons License 4.0 (CC BY – NC)



Trisna Gelar<sup>#1</sup>, Aprianti Nanda<sup>✉#2</sup>, Akhmad Bakhrun<sup>#3</sup>

<sup>#</sup> D3 Teknik Informatika, Politeknik Negeri Bandung

Jl. Gegerkalong Hilir, Ciwaruga, Kec. Parongpong, Kabupaten Bandung Barat, 40559, Indonesia

<sup>1</sup> trisna.gelar@polban.ac.id

<sup>2</sup> aprianti.nanda@polban.ac.id

<sup>3</sup> abakhrun@polban.ac.id

✉ Corresponding author: aprianti.nanda@polban.ac.id

**Abstrak** — *Named Entity Recognition* (NER) merupakan bagian dari pengembangan perangkat lunak dokumen, klasifikasi dan pencarian informasi. Implementasi NER pada domain pertanian, khususnya teks instruksional atau transkripsi video tutorial, akan memudahkan masyarakat umum memahami konsep dan istilah khusus dari kegiatan pertanian kota seperti proses dan prosedur produksi tanaman, metode dan alat pertanian, siklus panen, dan penanganan hama atau penyakit tanaman. *Spacy* merupakan alat bantu NLP, memiliki dua metode pengembangan model NER, yaitu dengan *Toc2Vec* dan *Transformer*. Kedua metode memiliki kelebihan dan kekurangan, yaitu ukuran, performansi dan kecepatan prediksi yang berbeda beda sesuai kebutuhan. Model NER dapat diimplementasikan menjadi aplikasi *Serverless*, dengan menggunakan pendekatan *Fungsional as Services (FaaS)* dan *Backend as Services (BaaS)*. Pada penelitian ini telah dikembangkan tiga model NER untuk data teks instruksional pertanian sub topik budidaya tanaman buah. Model berbasis *Toc2Vec* dengan optimasi efisiensi, Model *Toc2Vec* dengan optimasi akurasi dan Model berbasis *IndoBERT*. Model berbasis *Transformer* memiliki nilai f1-score terbaik sebesar 0.71 disusul Model *Toc2Vec Efisiensi* sebesar 0.60 dan Model *Toc2Vec Efektif* dan 0.57. Model *Toc2Vec* tidak dapat memprediksi entitas numerik dengan baik, Prediksi entitas COUNT, PERIOD dan VERIETAS selalu tertukar. Selain itu Ukuran model berbanding lurus dengan kecepatan prediksi kata per detik, dalam hal ini Model *Toc2Vec* optimasi efisiensi unggul, model tersebut mudah diimplementasikan menjadi *Serverless* berbasis *FaaS* dan *BaaS*. Fungsionalitas dari *Serverless ML* telah berhasil diuji menggunakan metode *Blackbox*.

**Kata kunci**—Kustomisasi *Named Entity Recognition*; Teks Instruksional; *Serverless ML*.

## Serverless Urban Farming Instructional Text Named Entity Recognition

**Abstract** — *The evolution of document documentation, classification, and information retrieval includes named entity recognition (NER). The implementation of NER in the agricultural domain, in particular instructional texts or transcriptions of tutorial videos, will make it easier for the general public to understand the specific concepts and terms of urban agricultural activities such as crop production processes and procedures, agricultural methods and tools, harvest cycles, and handling plant pests or diseases. Spacy is an NLP tool, has two methods of developing NER models, namely with Toc2Vec and Transformer. Both methods have advantages and disadvantages, namely different sizes, performance and prediction speeds according to needs. The NER model can be implemented into a Serverless application, using the Functional as Services (FaaS) and Backend as Services (BaaS) approaches. For the subtopic of*

*cultivating fruit crops in agricultural instructional literature, three NER models have been built in this study. First, the IndoBERT-based model, the Toc2Vec-based model with efficiency optimization, and the Toc2Vec-based model with accuracy optimization. The most efficient toc2vec model, with a f1-score of 0.71, is followed by the effective toc2vec model, with a f1-score of 0.60. The COUNT, PERIOD, and VERIETAS entities are consistently predicted incorrectly by the Toc2Vec model, which is unable to forecast numeric entities well. In addition, the Toc2Vec Model's better efficiency optimization directly relates the size of the model to the speed of word prediction per second, and the model is simple to integrate into a FaaS- and BaaS-based Serverless. The capabilities of Serverless M have been successfully tested using the black box method.*

**Keywords—** Custom Named Entity Recognition, Instructional Text; Serverless ML.

## I. PENDAHULUAN

Named Entity Recognition (NER) digunakan untuk mengidentifikasi entitas dari sebuah dokumen. Secara fungsional NER adalah bagian dari proses *Natural Language Processing* (NLP) seperti pengembangan perangkat lunak dokumen, klasifikasi dokumen, pencarian informasi berdasarkan teks dan sentimen analisis [1]. Selain itu model NER dapat memperbaiki kata pada model bahasa untuk data transkripsi *speech recognition* [2]. Untuk mendapatkan NER dari berbagai tipe, diperlukan data training yang telah dianotasi, proses ini disebut *tag-ing*, standar yang dipakai seperti *POS tagger* meliputi tag; *IO*, *IOB*, *IOE*, *IOBES*, *BI*, *IE*, dan *BIES* [3] untuk mendapatkan tipe entitas dari kata.

Beberapa penelitian mengimplementasikan NER dalam berbagai domain khusus seperti pertanian [4], kemacetan lalu lintas kota [5] dan keputusan pengadilan [6]. Topik pertanian memiliki istilah-istilah khusus yang dipahami secara terbatas oleh *stakeholder* bidang pertanian seperti proses dan prosedur produksi tanaman, metode dan alat pertanian, siklus panen, dan penanganan hama atau penyakit tanaman. Meskipun sulit untuk masyarakat memahami istilah tersebut, atensi mempelajari video tutorial atau dokumen instruksional pada domain ini, terutama pertanian kota, cukup populery [7]. Oleh karena itu NER dapat mempersempit gap pengetahuan dari *non-expert* tentang istilah-istilah pertanian.

Tiga penelitian sebelumnya tentang NER pada domain pertanian telah dilakukan. Pertama Malarkodi, et. al [8] menggunakan data pertanian berbahasa Inggris, dengan total 19 tipe entitas (*Person, Location, Organization, Chemical, Crop, Organism, Policy, Climate, Food Items, Diseases, Natural Disaster, Events, Nutrients, Count, Distance, Quantity, Money, Temperature, Date*). Kombinasi metode *Conditional Random Fields* (CRF) dan fitur kata (*Kata, POS, Frasa Benda, Kata Kunci, Imbuhan, Kapital, Pola POS dari entitas dan Numerik*). Performansi model menghasilkan nilai tertinggi saat kombinasi CRF dan fitur kata digabungkan dengan rata-rata *f-measure* tertinggi 0.82

Kedua, Ziu Liu, et. al [9], menggunakan data lebih khusus sub topik hortikultura berbahasa Inggris meliputi budidaya, pengolahan, dan penjualan buah, kacang-kacangan, sayuran, dan tanaman hias. Pada penelitian ini metode *Conditional Random Fields* (CRF) dan lima fitur kata (*Word, POS, Cue, Tanda Baca, Kapital dan Numerik*) diimplementasikan pada tujuh tipe entitas pertanian (*Crop, Trait, Method, Chemicals, Gen, Environment dan Entitas lain*). Model NER menghasilkan performansi rata-rata *f-measure* tertinggi 0.81.

Ketiga, Hercules, et. al [10], melakukan eksplorasi pembangunan model NER berbasis *Deep Learning* pada dokumen pertanian berbahasa Inggris menggunakan arsitektur *spacy.Tok2Vec.v2* dan *spacy.TransitionBasedParser.v2* dari *library Spacy NLP*. Model NER dibangun berdasarkan *transfer learning* dua model bahasa umum yaitu *en\_core\_web\_sm* dan *en\_core\_web\_lg* dengan kombinasi hyperparameter *batch size* (64,128) serta *learning rate* (0.0001, 0.01). Model NER *en\_core\_web\_sm* dengan *hyperparameter batch size* 64 dan *learning rate* 0.0001 menghasilkan nilai *f-measure* tertinggi 0.51. Kelebihan dari pengembangan model berbasis *deep learning menggunakan Spacy* adalah pemanfaatan metode *transfer learning* seperti *transformer* yang secara empiris dapat meningkatkan performansi model NER [11].

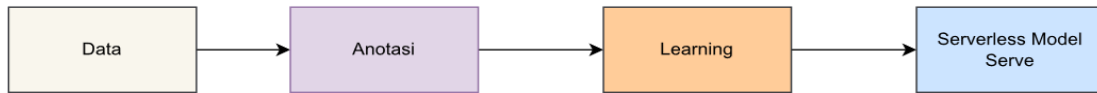
Proyek *machine learning* seperti NER dapat memanfaatkan konsep *Serverless* untuk tahapan *model serving* [12]. Dengan pendekatan ini sebuah servis dibuat modular sesuai fungsinya. Pembuatan servis terpisah akan menjaga utilitas tetap tinggi dan berbiaya rendah namun tidak mengurangi sasaran kinerja. Pengembangan aplikasi fokus pada abstraksi pemrograman dari aplikasi tanpa melakukan operasional manajemen sumber daya komputasi dan keamanan sistem [13].

Terdapat dua tipe pengembangan *Serverless* yaitu *Function as Service* (FaaS) dan *Backend as Service* (BaaS), pada FaaS, fungsi abstrak tanpa *state* dikembangkan menggunakan bahasa pemrograman pilihan (*Golang, Javascript* atau *Python*). Fungsi tersebut berinteraksi dengan *resource* komputasi awan berdasarkan *trigger* yang diberikan bisa berupa *HTTP request*, perubahan *state* pada *database* atau data sinkronisasi dari *file storage*. Sedangkan pada BaaS, perbedaan utamanya adalah *resource* yang digunakan akan lebih fleksibel karena pengembang akan menginisiasi *environment* berbasis *provisioning* seperti pengaturan *base operation system*, versi program, *library* dan kebutuhan *file storage* sesuai kebutuhan [13].

Penelitian ini mengangkat dua permasalahan, Pertama, pembangunan model NER topik dokumen instruksional pertanian perkotaan sub topik hortikultura pada berbasis *Spacy v3* [14] memanfaatkan arsitektur *Tok2Vec* dan *Transformer* bahasa Indonesia (*IndoBERT* [15]) beserta evaluasinya. Kedua, implementasi *model serving NER* menggunakan dua pendekatan *Serverless*, yaitu *FaaS* dan *BaaS*.

## II. METODE PENELITIAN

Penelitian ini berfokus pada pengembangan Model *NER* teks instruksional hortikultura berbahasa Indonesia dan *model serving* berbasis *Serverless*.



Gambar 1. Prosedur penelitian.

Prosedur penelitian ditunjukkan pada Gambar 1, yang terdiri dari empat tahapan utama. Pertama, dokumen teks instruksional tema hortikultura yang telah terkurasi dikumpulkan berdasarkan konteks proses budidaya pada tanaman buah di Indonesia. Kedua, untuk setiap dokumen, data dianotasi sesuai dengan entitas pertanian digunakan untuk keperluan eksperimen. Ketiga, proses pembelajaran model dari data teranotasi *NER* berbasis *Spacy v3* setelah itu setiap model dievaluasi. Keempat, model akan di-*deploy* secara *Serverless* menggunakan pendekatan *FaaS* dan *BaaS* pada *Google Cloud Platform*.

### A. Data

Dokumen teks instruksional merupakan hasil kurasi penelitian [16], pada penelitian ini empat buku sub tema budidaya buah (Mangga, Jeruk, Markisa, dan Buah Naga), digunakan karena memiliki kerangka pembahasan yang mirip, seperti terlihat pada Tabel 1.

TABEL 1  
DETAIL KURASI BUKU TEKS TEMA BUDIDAYA BUAH

Judul Buku	Kerangka	Hal	Kalimat
Budi Daya dan Pascapanen Mangga	Pendahuluan, Syarat Tumbuh, Varietas, Pembibitan, Penanaman, Hama, Pascapanen	33	290
Budi Daya Jeruk	Pendahuluan, Syarat Tumbuh, Bibit, Budidaya, Panen dan Pascapanen.	13	131
Budi Daya Buah Naga	Pendahuluan, Benih, Pemupukan, Hama dan Penyakit, Pemasaran.	43	115
Petunjuk Teknis Budidaya Markisa	Pendahuluan, Pembibitan, Lahan, Pemeliharaan, Hama dan Penyakit, Panen dan Pascapanen.	37	103

*Preprocessing* dilakukan secara manual untuk setiap buku, hal ini diperlukan karena hasil ekstraksi *pdf* tidak dapat mendeteksi sebuah paragraf dan kalimat secara baik, beberapa kata terpotong sehingga tidak membentuk kalimat lengkap. Hasil akhir dari proses ini adalah kumpulan kalimat *raw* terdapat total 639 kalimat. Pada penelitian ini tidak dilakukan data *cleaning* karena huruf kapital dan tanda baca memiliki makna tersendiri pada istilah pertanian, contohnya nama dari varietas, penyakit dan unsur Kimia.

### B. Annotation

Entitas pertanian merupakan adaptasi dari penelitian [8] [9] yang disesuaikan dengan konteks pada teks instruksional pertanian kota berbahasa Indonesia. Terdapat 14 entitas yang terdiri dari *CROP*, *CHEMICAL*, *QUANTITY*, *DISTANCE*, *DISEASES*, *PESTS*, *LOCATION*, *COUNT*, *TRAIT*, *VARIETIES*, *PERIOD*, *TOOLS*, *METHOD*, dan *CONDITION*. TABEL 2 merupakan deskripsi tiap entitas beserta jumlah keseluruhan entitas pada dokumen.

TABEL 2  
DAFTAR ENTITAS, DESKRIPSI DAN JUMLAH PADA DOKUMEN

Entitas	Deskripsi	Jumlah
<i>CROP</i>	Nama Buah dan Tanaman	70
<i>CHEMICAL</i>	Unsur Kimia, Pupuk, dan Pestisida	73
<i>QUANTITY</i>	Ukuran berat, seperti liter, ton, dan gram.	21
<i>DISTANCE</i>	Ukuran jarak, seperti cm, m dan Km	13

<i>DISEASES</i>	Penyakit pada buah-buahan atau tanaman.	14
<i>PESTS</i>	Hama penyebab penyakit pada buah-buahan atau tanaman.	47
<i>LOCATION</i>	Nama dari kota, negara, daerah atau sebuah tempat.	64
<i>COUNT</i>	Jumlah dari sebuah benda.	22
<i>TRAIT</i>	Karakteristik Biologis dari tanaman atau buah-buahan, seperti dahan, daun, batang, dan lain-lain.	115
<i>VARIETIES</i>	Varietas dari tanaman atau buah-buahan.	11
<i>PERIOD</i>	Frekuensi atau siklus dari sebuah metode/proses.	39
<i>TOOLS</i>	Alat-alat budidaya pertanian kota.	74
<i>METHOD</i>	Metode budidaya pertanian kota.	123
<i>CONDITION</i>	Kondisi dari tanaman atau buah-buahan setelah dilakukan metode/proses.	96

Setelah 14 entitas terdefinisi, proses anotasi dilakukan pada tiap kalimat dan tidak ada entitas yang *overlapping*. Pada penelitian ini *NER Annotator* (<https://github.com/tecoholic/ner-annotator>) sebuah alat bantu anotasi *NER* berbasis web yang menyediakan GUI untuk melakukan anotasi dan menghasilkan data anotasi berformat JSON yang siap untuk digunakan pada *Spacy v3*. Selain itu *NER Annotator* dapat dimodifikasi menjadi alat untuk menganotasi semantik dari sebuah data teks [17].

Berikut adalah data sampel, hasil anotasi, tiap elemen data terdapat teks dan posisi awal, akhir dan label dari entitas. Setelah dilakukan anotasi,

```
[ 'Penyakit yang paling ganas adalah CVPD, disusul oleh virus Tristeza\r',
  { 'entities': [[34, 38, 'DISEASES'], [53, 58, 'DISEASES'], [59, 67, 'DISEASES']] } ],
[ 'Tanaman kaktus ini tumbuh merambat
memanjat dengan batang berbentuk segi tiga dan pada bagian pinggirnya terdapat duri duri \r',
  { 'entities': [[8, 14, 'CROP'], [51, 57, 'TRAIT'], [68, 77, 'CONDITIONS'], [114, 123, 'CONDITIONS']] } ],
[ 'Pemberian kapur diberikan sebulan sebelum tanam agar kapur meresap dan bereaksi menetralkan tanah\r',
  { 'entities': [[10, 15, 'CHEMICAL'], [26, 33, 'PERIOD'], [53, 58, 'CHEMICAL']] } ]
```

Sebelum dilakukan pemodelan, proses *splitting dataset* dengan rasio 80:10:10 sebagai *input data training*, *testing* dan *validasi* dengan total kalimat masing-masing 511, 64 dan 64 kalimat. Tabel 3 mempresentasikan frekuensi entitas pada tiap data hasil pembagian.

TABEL 3  
FREKUENSI ENTITAS PADA DATA TRAINING, TESTING DAN VALIDASI

Entitas	Train	Test	Val
<i>CROP</i>	29	20	21
<i>CHEMICAL</i>	16	21	36
<i>QUANTITY</i>	7	6	8
<i>DISTANCE</i>	2	7	4
<i>DISEASES</i>	10	2	2
<i>PESTS</i>	13	20	14
<i>LOCATION</i>	32	28	4
<i>COUNT</i>	7	9	6
<i>TRAIT</i>	37	34	44
<i>VARIETIES</i>	5	5	1
<i>PERIOD</i>	11	15	13
<i>TOOLS</i>	28	11	35
<i>METHOD</i>	43	30	50
<i>CONDITION</i>	34	34	28

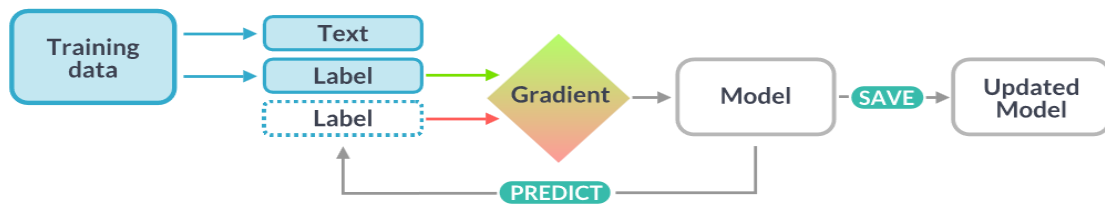
### C. Learning

Terdapat dua proses pada tahap ini, Pertama, model *NER* berbasis *Spacy* dikembangkan dengan tiga skenario berdasarkan pilihan optimasi atau penggunaan *Transformer*. Kedua, evaluasi dari model secara keseluruhan dan untuk setiap entitas.

#### 1) Learning

Gambar 2 merupakan ilustrasi proses pembelajaran pada *Spacy v3*, proses ini berlangsung secara terus menerus, membandingkan hasil prediksi dengan *ground truth* anotasi untuk mengukur nilai *gradient* yang berasal dari optimasi *loss*

function. Melalui algoritma *backpropagation*, nilai *gradient* yang diperoleh kemudian digunakan untuk mengupdate parameter dari model. Hal ini terus dikerjakan untuk setiap *instance* dari training data hingga diperoleh nilai *loss* yang kecil



Gambar 2. Proses Training Model pada Spacy v3. (<https://spacy.io/usage/training>)

*Spacy library* belum menyediakan model dalam bahasa Indonesia, sehingga pengembangan model *NER* dibuat secara khusus melalui. Terdapat dua alternatif pengembangan model *NER* yaitu menggunakan arsitektur *Tok2Vec* dan menggunakan arsitektur *Transformer*, secara empiris pemanfaatan arsitektur ini telah diimplementasikan pada domain pariwisata dengan memanfaatkan arsitektur *Transformer BERT*. Hasilnya dapat meningkatkan performansi akurasi karena dengan bantuan *transfer learning* beberapa entitas yang jumlah datanya terbatas, dapat diprediksi dengan baik[11]. Opsi pertama, untuk membangun model *NER* diperlukan arsitektur *Tok2Vec(spacy.Tok2Vec.v2)* dan *pipeline NER*, serta pada opsi ini disediakan optimasi *default* yaitu mengejar akurasi atau efisiensi. Model yang efisien, lebih cepat dibangun, diinferensi namun kurang akurat, sedangkan model yang akurat sebaliknya. Opsi kedua, menggunakan model bahasa *Transformer* melalui *transfer learning*, untuk model bahasa Indonesia, *Spacy* dapat menggunakan *transformer indolem/indobert*.

Tabel 4 merupakan deskripsi dari Tiga skenario pembangunan model *NER* pertanian kota akan dibangun.

TABEL 4  
PERBANDINGAN KONFIGURASI *HYPERPARAMETER* DAN ARSITEKTUR 3 MODEL *NER*

Configuration	Model 1/2	Model 3
System	<code>gpu_allocator = null</code> <code>seed = 0</code>	<code>gpu_allocator = "pytorch"</code> <code>seed = 0</code>
NLP	<code>lang = "id"</code> <code>pipeline = ["tok2vec", "ner"]</code> <code>batch_size = 1000</code>	<code>lang = "id"</code> <code>pipeline = ["transformer", "ner"]</code> <code>batch_size = 128</code>
NER Model	<code>@architectures = "spacy.Tok2VecListener.v1"</code> <code>width</code> <code>width = {components.tok2vec.model.encode.width}</code>	<code>@architectures = "spacy-transformers.TransformerListener.v1"</code> <code>grad_factor = 1.0</code> <code>pooling = {"@layers": "reduce_mean.v1"}</code>
Architecture	<code>[components.tok2vec]</code> <code>factory = "tok2vec"</code>	<code>[components.transformer]</code> <code>factory = "transformer"</code> <code>max_batch_items = 4096</code>  <code>[components.transformer.model]</code> <code>@architectures = "spacy-transformers.TransformerModel.v3"</code> <code>name = "indolem/indobert-base-uncased"</code>
Optimization	<b>Model 1</b> <code>[components.tok2vec.model.encode]</code> <code>@architectures = "spacy.MaxoutWindowEncoder.v2"</code> <code>width = 96</code> <code>depth = 4</code>	
	<b>Model 2</b> <code>[components.tok2vec.model.encode]</code> <code>@architectures = "spacy.MaxoutWindowEncoder.v2"</code> <code>width = 256</code> <code>depth = 8</code>	

## 2) Evaluasi, :

Pada penelitian ini tiga model *NER* akan dievaluasi performansinya menggunakan pengukuran *precision*, *recall* dan *f1-score* pada data testing dengan total 64 kalimat teks instruksional pertanian. Selain itu tabel *confusion matrix* untuk entitas-entitas dengan nilai *f-score* kurang dari 0.5 akan dipersentasekan dan dianalisis. Selain itu ukuran model yang dihasilkan dan kecepatan prediksi akan diukur.

### D. Serverless Model Serving

Tiga model *NER* akan *deploy* secara *Serverless* dengan dua pendekatan *FaaS* dan *BaaS* menggunakan *Google Cloud Platform* melalui *Google Cloud Shell SDK*.

#### 1) Model *Serveless FaaS* menggunakan *Google Cloud Function* (<https://cloud.google.com/functions>)

Untuk membuat sebuah fungsi pada *Google Cloud Function* minimal diperlukan sebuah main program, file *requirements.txt* dan model *NER*. Pada pendekatan *FaaS*, main program cukup berisi abstraksi program berbahasa *python* dan sebuah fungsi utama yang berhubungan dengan *trigger* seperti *HTTP request* yang terotorisasi.

#### 2) Model *Serveless BaaS* menggunakan *Google Cloud Run* (<https://cloud.google.com/run>)

Untuk membuat sebuah fungsi pada *Google Cloud Run*, pada pendekatan *BaaS*, selain model *NER*, *requirements.txt* dan main program yang berisi abstraksi program berbahasa *python* diperlukan file *provision* berbasis *Docker* yang akan menginisiasi instalasi program, library dan mengeksekusi perintah untuk menjalankan server yang berhubungan dengan *trigger* seperti *HTTP request* yang terotorisasi.

Hasil implementasi akan diuji secara *black box* untuk mengamati dan memeriksa fungsionalitas *NER* apakah berhasil melakukan tag-ing pada kalimat uji. Total ada 5 pengujian, Model 1, Model 2 berbasis *FaaS* dan Model 1, Model 2, Model 3 berbasis *BaaS*, fungsi akan diuji menggunakan *API Platform Postman* (<https://www.postman.com>).

## III. HASIL DAN PEMBAHASAN

Terdapat dua kajian utama pada hasil dan pembahasan yaitu bagaimana perbandingan performansi dari tiga model *NER* yang telah dibangun dan Hasil implementasi *Serverless Model Serve* pada *Google Cloud Function* dan *Google Cloud Run*.

### A. Evaluasi Model

TABEL 5  
DAFTAR PERFORMANSI MODEL: *PRECISION*, *RECALL* DAN *F1-SCORE*

Model	Precision	Recall	F1-Score	Ukuran Model	Pred Speed wps
Model 1 Toc2Vec efisiensi	0.63	0.58	0.60	6 Mb	15205
Model 2 Toc2Vec akurasi	0.58	0.56	0.57	36 Mb	3586
Model 3 transformer	0.71	0.71	0.71	401 Mb	1505

Secara makro (performansi keseluruhan), terlihat pada Tabel 5, Model *NER* ke-3, *Transformer* yang mengadaptasi model bahasa Indonesia *indolem/indobert*, memiliki nilai performansi *precision*, *recall* dan *f-score* paling tinggi dengan rata-rata 0.71 dibandingkan dengan dua model lainnya namun memiliki ukuran model terbesar dan kecepatan prediksi terlambat (*word per second*). Sedangkan terjadi anomali pada Model *NER* ke-2, dengan nilai performansinya paling rendah dibandingkan dengan Model *NER* ke-1, perbedaan kedua model adalah ukuran lebar dan dalamnya arsitektur *spacy.MaxoutWindowEncoder.v2* lebih besar.

TABEL 6  
DAFTAR PERFORMANSI MODEL *F1 SCORE* UNTUK TIAP ENTITAS

Tag	Model 1	Model 2	Model 3
<i>CROP</i>	0.97	0.95	0.90
<i>CHEMICAL</i>	0.77	0.85	0.84
<i>QUANTITY</i>	0.71	0.5	1
<i>DISTANCE</i>	0.83	0.57	0.86
<i>DISEASES</i>	0.4	0.67	0.67
<i>PESTS</i>	0.65	0.47	0.59
<i>LOCATION</i>	0.59	0.67	0.91
<i>COUNT</i>	0.24	0.13	0.57
<i>TRAIT</i>	0.49	0.56	0.65
<i>VARIETIES</i>	0.8	0.5	0.33
<i>PERIOD</i>	0.79	0.57	0.90
<i>TOOLS</i>	0.48	0.31	0.44
<i>METHOD</i>	0.46	0.51	0.67
<i>CONDITION</i>	0.52	0.47	0.67

Secara mikro (performansi per-entitas), terlihat pada Tabel 6 secara rata-rata untuk entitas yang berhubungan dengan istilah pertanian *DISEASES*, *PESTS*, *TRAIT*, *TOOLS*, *METHOD*, *CONDITION* dan *TRAIT* nilai *f1-score* beragam kurang dari 0.70. Namun untuk entitas *CROP* dan *CHEMICAL* tiga model dapat memprediksi dengan baik. Terdapat Anomali untuk entitas *VARIETIES*, dengan Model NER ke-1 dapat memprediksi entitas tersebut dengan baik dibandingkan dengan model lainnya.

Model NER ke-1 dan Model NER Ke-3 dapat memprediksi entitas numerik seperti *QUANTITY*, *PERIOD* dan *DISTANCE* dengan baik dengan nilai performansi lebih dari 0.7. Secara keseluruhan Model NER ke-2 memiliki nilai akurasi terendah untuk tipe entitas numerik tersebut yaitu 0.5, 0.57, dan 0.57.

TABEL 7  
CONFUSION MATRIX ENTITAS TIPE NUMERIK MODEL NER KE-2

	CHEM	COND	COUNT	CROP	DISEASES	DIST	LOC	METHOD	PERIOD	PESTS	QUANT	TOOLS	TRAIT	VAR	O
QUANTITY			2						2		6				2
DISTANCE						8			2		2				3
PERIOD						2			8		2				4

Tabel 7 merupakan tabel *confusion matrix* dari entitas numerik pada data test dengan Model NER ke-2, terlihat bahwa terjadi kesalahan prediksi untuk entitas *DISTANCE* dan *PERIOD* menjadi entitas *QUANTITY*.

Sebagai contoh frasa 8x per minggu diprediksi menjadi *QUANTITY*, seharusnya entitas frasa tersebut adalah *PERIOD*. Huruf x memiliki beberapa makna dalam entitas numerik, pada *DISTANCE* digunakan sebagai ukuran persegi atau kubik sebagai contoh “Jarak tanam yang biasa digunakan adalah 5x5 m sampai 8x8 m”. untuk entitas *QUANTITY*, “pemupukan dengan pupuk NPK dosis 3x.”. Selain itu beberapa entitas O atau other (kata yang seharusnya tidak dianotasi kedalam kelompok entitas) diprediksi menjadi entitas numerik hal ini terjadi untuk beberapa frasa numerik yang memiliki tanda baca koma (,), strip (-) dan titik (.) untuk ukuran sebuah senyawa atau unsur hara. Diperlukan mekanisme khusus untuk membedakan huruf-huruf atau tanda baca yang membuat ambigu dengan menggunakan *Pipeline Rule Based Matching* berbasis *Token*, *Phrase* atau *Dependency Matcher* (<https://spacy.io/usage/rule-based-matching>) untuk membedakan setiap kasus entitas numerik.

Tiga model memiliki kelebihan dan kekurangan, hal yang perlu didalami adalah entitas *COUNT* pada Model NER ke-1 dan Model NER ke-2 memiliki nilai *f1-score* sangat rendah yaitu 0.24 dan 0.13, Model 3 memiliki nilai *f-score* lebih kecil untuk entitas istilah pertanian *VARIETIES* dan *TOOLS*.

TABEL 8  
CONFUSION MATRIX ENTITAS COUNT UNTUK MODEL NER KE-1 DAN MODEL NER KE-2

	CHEM	COND	COUNT	CROP	DISEASES	DIST	LOC	METHOD	PERIOD	PESTS	QUANT	TOOLS	TRAIT	VAR	O
COUNT Model 1		1	6								2		1		7
COUNT Model 2		1	5								2				9

Tabel 8 merupakan tabel *confusion matrix* dari entitas *COUNT* pada Model 1 dan Model 2. Entitas ini tertukar dengan *QUANTITY* dan *OTHER*. Karena *COUNT* merupakan entitas numerik, solusinya dapat menggunakan *Pipeline Regular Expression* atau melakukan preprocessing huruf atau tanda baca yang ambigu dari simbol menjadi sebuah kata.

TABEL 9  
CONFUSION MATRIX ENTITAS VARIETIES DAN TOOLS UNTUK MODEL 3

	CHEM	COND	COUNT	CROP	DISEASES	DIST	LOC	METHOD	PERIOD	PESTS	QUANT	TOOLS	TRAIT	VAR	O
VAR				3					0					2	3
TOOLS									0			7	4		5

Tabel 9 merepresentasikan kesalahan prediksi entitas *VARIETIES* dan *TOOLS*. Terlihat bahwa beberapa entitas *VARIETIES* tertukar dengan entitas *CROP*, karakter dari kata atau frasa entitas *VARIETIES* ada yang berupa nama buah-buahan (Markisa Konyal, Markisa Kuning, dan Markisa Ungu), nama species (*Passiflora edulis* Sims), atau nama buah-buahan dengan penomoran (Arumanis-143, Golek-31 dan Manalagi-69). Untuk membedakan entitas *CROP* dan *VARIETIES* agar tidak ambigu diperlukan penggunaan Regular Expression dengan Rule Based Matching Untuk entitas *TOOLS* yang diprediksi kurang tepat menjadi *TRAIT*, diperlukan pemahaman kembali mengenai domain masalah, seperti menambahkan hirarki pada proses anotasi untuk membedakan mana yang termasuk *TOOLS* dan mana yang termasuk *TRAIT*.

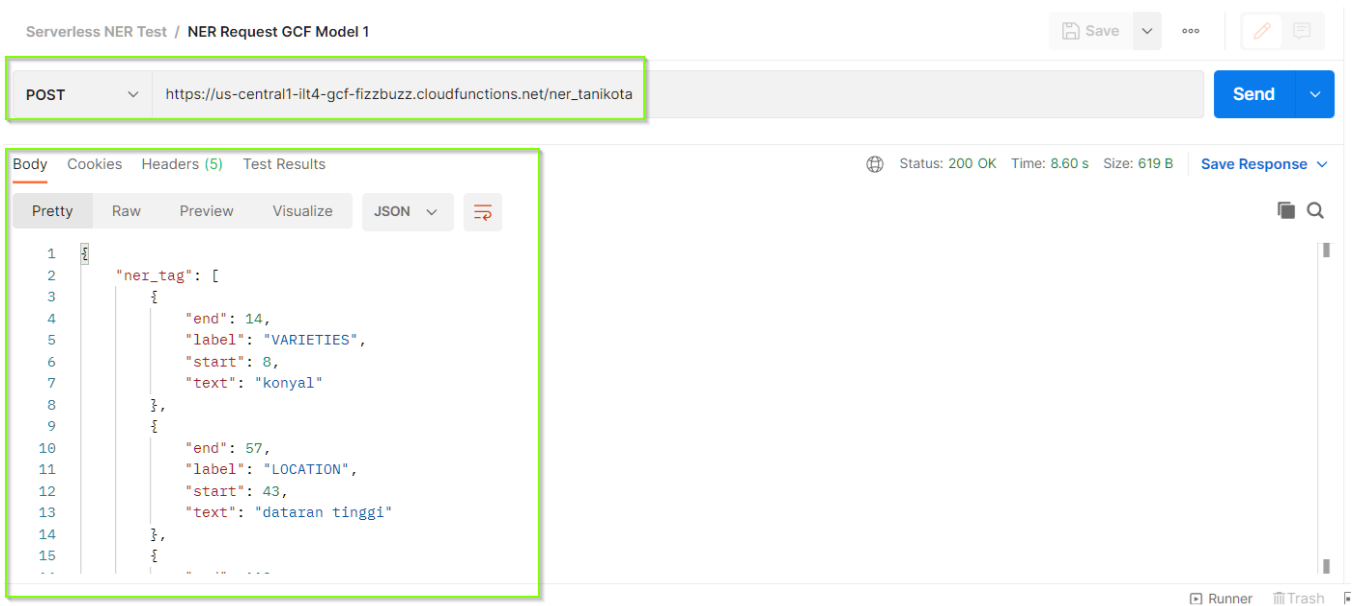
#### B. Implementasi Serverless Model Serve NER

Tabel 10 merupakan hasil implementasi telah diuji secara *black box* untuk setiap fungsionalitas *NER* pada *Serverless ML*, khusus model 3, deployment hanya dilakukan menggunakan *BaaS*, karena ukuran *model NER* lebih dari 100 Mb. Untuk implementasi pada *FaaS*, model perlu disimpan ke dalam pada *storage external* seperti *Google Cloud Storage* (<https://cloud.google.com/storage/>) tidak dilakukan pada penelitian ini.

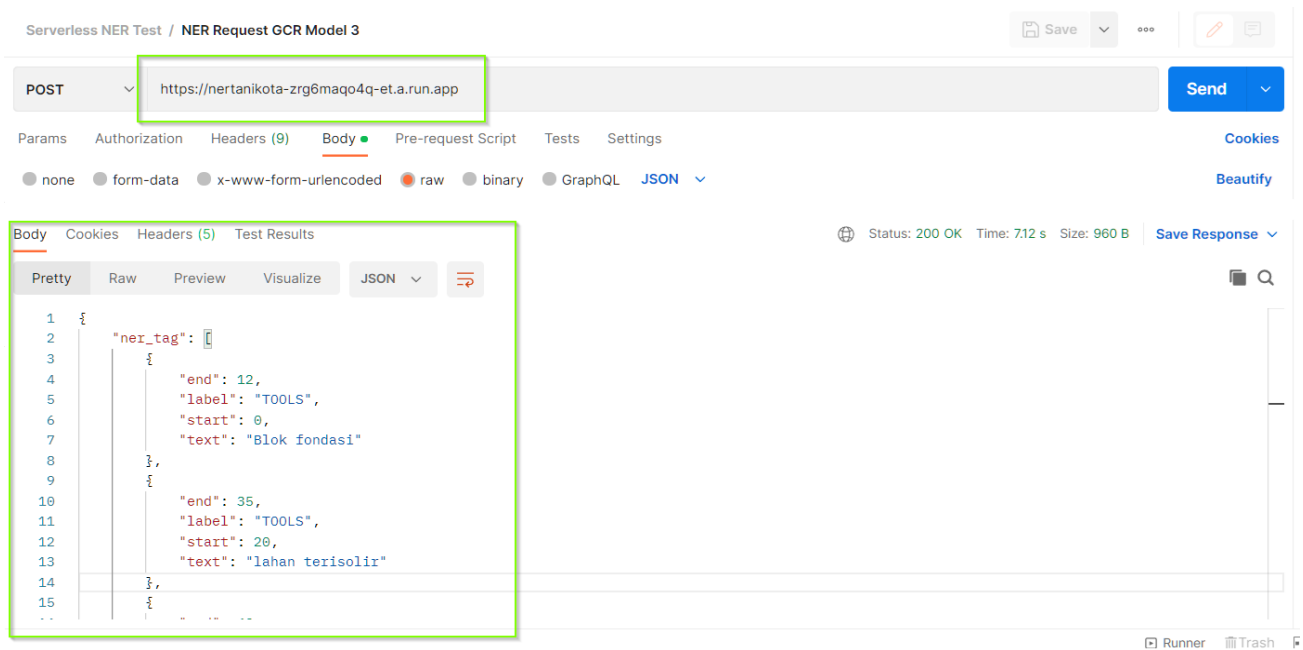
TABEL 10  
BLACK BOX TESTING FUNGSIONALITAS SERVERLESS NER

Model dan Platform	Trigger	Hasil
Model 1 FaaS	Trigger HTTP Request	Berhasil
Model 1 BaaS	Trigger HTTP Request	Berhasil
Model 2 FaaS	Trigger HTTP Request	Berhasil
Model 2 BaaS	Trigger HTTP Request	Berhasil
Model 3 BaaS	Trigger HTTP Request	Berhasil





Gambar 3. Pengujian Serverless NER FaaS Google Cloud Function pada Postman



Gambar 4. Pengujian Serverless NER Google BaaS Google Cloud Run pada Postman.

Gambar 3 dan Gambar 4 merupakan ilustrasi pengujian *Serverless NER* pendekatan *FaaS* dan *BaaS* memanfaatkan *Postman*. Kalimat teks pertanian yang diuji adalah “Tanaman konyal dapat tumbuh dengan baik di dataran tinggi dan berproduksi setelah berumur kurang lebih 1 tahun.” dan “Blok fondasi adalah lahan terisolir yang ditanami pohon induk bebas penyakit sebagai sumber mata tempel untuk penggandaan mata tempel”. *NER* Servis akan mengirimkan tag *NER* berupa awal, akhir huruf, label dan kata/frasa yang diprediksi.

#### IV. SIMPULAN

Pengembangan model *Named Entity Recognition* untuk data kustom teks instruksional pertanian berbasis *Spacy* dikombinasikan dengan *Transfer Learning* berbasis *Transformer* IndoBERT menghasilkan performansi *f1-score* tertinggi.

Setelah itu model berbasis *Toc2Vec* dengan optimasi efisiensi performansinya *f1-score* yang lebih baik optimasi akurasi. Hal ini disebabkan kedua model tidak dapat mengidentifikasi dengan baik entitas numerik dan entitas istilah pertanian dengan baik. Prediksi entitas *COUNT*, *PERIOD* dan *VERIETAS* selalu tertukar. Ukuran model berbanding lurus dengan kecepatan prediksi kata per detik, dalam hal ini Model *Toc2Vec* optimasi efisiensi unggul, model tersebut mudah diimplementasikan menjadi *Serverless* berbasis *FaaS* dan *BaaS*, sedangkan model jika ingin diimplementasikan menjadi *FaaS* dibutuhkan komponen lain untuk menyimpan model. Lima skenario pengujian secara *black box* telah dilakukan untuk *Serverless ML*. Untuk penelitian selanjutnya, agar performansi model meningkat diperlukan mekanisme prediksi entitas numerik seperti penggunaan *Regular Expression* dengan *Rule Based Matching*.

#### UCAPAN TERIMA KASIH

Program Penelitian bersumber dari dana DIPA Politeknik Negeri Bandung dengan surat perjanjian PM No: B/114.90/PL1.R7/PG.00.03/2022. Terima kasih atas partisipasi semua pihak membantu terlaksananya kegiatan ini.

#### DAFTAR PUSTAKA

- [1] Z. Nasar, S. W. Jaffry and M. K. Malik, "Named Entity Recognition and Relation Extraction: State-of-the-Art," *ACM Computing Surveys*, vol. 54, no. 1, p. 39, 2022.
- [2] M. Ehrmann, D. Nouvel and S. Rosset, "Named Entity Resources - Overview and Outlook," in *Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016.
- [3] N. Alshammari and S. Alanazi, "The impact of using different annotation schemes on named entity recognition," *Egyptian Informatics Journal*, vol. 22, no. 3, p. 295–302, 2021.
- [4] B. Drury and M. Roche, "A Survey of the Applications of Text Mining for Agriculture," *Computers and Electronics in Agriculture*, vol. 163, p. 104864, 2019.
- [5] M. T. Zulfikar and Suharjo, "Detection Traffic Congestion Based on Twitter Data using Machine Learning," *Procedia Computer Science*, vol. 157, pp. 118-124, 2019.
- [6] E. Q. Nuranti and E. Yulianti, "Legal Entity Recognition in Indonesian Court Decision Documents Using Bi-LSTM and CRF Approaches," in *International Conference on Advanced Computer Science and Information Systems*, 2020.
- [7] T. Gelar and A. N. Sari, "Klasifikasi Komentar Video Instruksional Populer Bertemakan Pekerjaan Perkotaan menggunakan Auto-Keras," *Jurnal of Software Engineering Information and Communication Technology*, vol. 1, no. 1, 2020.
- [8] M. Nathan, E. Lex and S. Lalithadevi, "Named Entity Recognition for the Agricultural Domain," *Research in Computing Science*, vol. 117, p. 121–132, 2016.
- [9] Z. Liu, M. Luo, H. Yang and X. Liu, "Named Entity Recognition for the Horticultural Domain," *Journal of Physics Conference Series*, vol. 1631, 2020.
- [10] H. Panoutsopoulos, C. Brewster and B. E. García, "Developing a Model for the Automated Identification and Extraction of Agricultural Terms from Unstructured Text," vol. 10, p. 94, 2022.
- [11] C. Chantrapornchai and A. Tunsakul, "Information Extraction on Tourism Domain using SpaCy and BERT," *ECTI TRANSACTIONS ON COMPUTER AND INFORMATION TECHNOLOGY*, vol. 15, no. 1, p. 108–122, 2021.
- [12] Y. Wu, T. T. A. Dinh, G. Hu, M. Zhang and Y. M. Chee, "Serverless Data Science -- Are We There Yet? A Case Study of Model Serving," *Proceedings of the 2022 International Conference on Management of Data*, vol. 1, no. 1, 2021..
- [13] J. Schleier-Smith, V. Sreekanti, A. Khandelwal, J. Carreira, N. J. Yadwadkar, R. A. Popa, J. E. Gonzalez, I. Stoica and D. A. Patterson, "What Serverless Computing is and Should Become: The next Phase of Cloud Computing," *Communications of the ACM*, vol. 64, no. 2, p. 76–84, 2021.
- [14] H. Shelar, G. Kaur, N. Heda and P. Agrawal, "Named Entity Recognition Approaches and Their Comparison for Custom NER Model," *Science & Technology Libraries*, vol. 39, no. 3, p. 324–337, 2020.
- [15] F. Koto, A. Rahimi, J. H. Lau and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," 2020. [Online]. Available: <https://arxiv.org/abs/2011.00677>.
- [16] T. Gelar and A. Nanda, "Eksplorasi Pengembangan Korpus Pembicaraan Spontan pada Video Instruksional Pertanian Perkotaan," *Jurnal of Software Engineering Information and Communication Technology*, vol. 3, no. 1, p. 111–120, 2022.
- [17] A. Loreggia, S. Mosco and A. Zerbini, "SenTag: A Web-Based Tool for Semantic Annotation of Textual Documents," *The Thirty-Sixth AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, p. 13191–13193, 2022.