Ekstraksi Informasi pada Dokumen Teks Menggunakan Metode *Named-Entity Recognition* untuk Sistem *Autoful* Formulir Lowongan SIM Magang MyITS StudentConnect

Kevin Christian Hadinata, Dini Adni Navastara dan Hadziq Fabroyir
Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: dini_navastara@if.its.ac.id

***Abstrak*—Sistem Informasi Manajemen (SIM) Magang MyITS StudentConnect adalah suatu platform yang dibuat untuk memenuhi kebutuhan penyebaran informasi magang dalam lingkungan mahasiswa Institut Teknologi Sepuluh Nopember (ITS). Dalam perkembangannya, diperlukan suatu sistem yang efektif dan efisien untuk pengisian informasi lowongan yang akan diunggah oleh pihak Pengembangan Kewirausahaan dan Karir (PK2) dalam lingkungan ITS. Oleh karena itu, suatu sistem autofill (sistem pengisian otomatis) dirasa perlu untuk dapat meningkatkan efisiensi dalam pengisian lowongan magang. Sistem ini bekerja dengan cara memindai dokumen lowongan magang yang diunggah dan mengisikan informasi yang didapat dari dokumen lowongan tersebut dalam format lowongan yang sesuai dengan modul SIM Magang myITS Student Connect.Untuk melakukan ekstraksi informasi dari dokumen yang diunggah, dilakukan pemindaian data teks menggunakan teknik Optical Character Recognition (OCR). Lalu, untuk klasifikasi, digunakan metode Named-Entity Recognition (NER), yang merupakan salah satu metode Natural Language Processing yang dapat mengklasifikasikan informasi berdasarkan entitasnya. Hasil ekstraksi informasi tersebut kemudian dimasukkan ke dalam kolom-kolom form yang tersedia sesuai dengan format modul Magang myITS StudentConnect. Hasil dari penelitian ini diharapkan dapat meningkatkan kinerja dan efisiensi SIM Magang dalam melakukan pendistribusian informasi terkait magang yang tersedia untuk kalangan mahasiswa-mahasiswa ITS. Dilakukan pengamatan performa terhadap ketepatan analisa NER dengan menggunakan data latih berupa poster lowongan magang sebanyak 24 buah. Setelah itu, didapatkan hasil berupa optimizer Adam dengan epochs sebanyak 1000 yang dapat bekerja dengan performa paling baik dengan nilai precision 0.53023, recall 0.56755, dan f1-score 0.54565.**

***Kata Kunci*—Named Entity Recognition, Natural Language Processing, Optical Character Recognition, Sistem Pengisian Otomatis.**

I. PENDAHULUAN

SAAT ini, Institut Teknologi Sepuluh Nopember (ITS) Surabaya tengah merancang suatu platform yang disebut sebagai modul Sistem Informasi Manajemen (SIM) Magang MyITS StudentConnect. Nantinya, modul ini akan dapat digunakan mahasiswa untuk mencari lowongan magang yang disediakan oleh berbagai perusahaan.

Saat melakukan input lowongan magang, tentunya akan dirasa kurang efektif dan efisien apabila dilakukan secara manual dan satu persatu, mengingat setiap perusahaan atau pembuka lowongan magang memiliki format yang berbeda dalam pemberian dokumen yang berisikan lowongan magang. Oleh karena itu, dibutuhkan suatu sistem yang dapat memindai dokumen-dokumen tersebut dan menjabarkan isinya ke dalam format lowongan yang sesuai dengan modul SIM Magang MyITS StudentConnect.

Gambar 1. Contoh Data Latih.

Gambar 2. Contoh Data Tes.

Mengacu pada penelitian sebelumnya yang telah dilakukan oleh Perera, Dehmer, dan Emmert-Streib [1], Named-Entity Recognition (NER) dapat digunakan untuk pemindaian otomatis melalui teks yang tidak terstruktur untuk mendeteksi entitas, yang digunakan untuk normalisasi dan klasifikasi dalam beberapa kategori.

Gambar 3. Contoh Hasil Praproses Gambar.

Gambar 4. Contoh Deteksi Tulisan pada Gambar.

Berdasar dari penelitian tersebut, dilakukan pemindaian dokumen berupa poster lowongan yang diunggah oleh pengguna. Setelah itu, dilakukan ekstraksi informasi dengan NER untuk memisahkan informasi yang penting berdasarkan entitasnya dan dimasukkan ke dalam kolom-kolom form yang tersedia sesuai dengan format modul Magang MyITS StudentConnect.

Karena perlu untuk menyesuaikan hasil klasifikasi data lowongan sesuai dengan kolom-kolom yang terdapat pada formulir lowongan magang, dipilihlah metode NER untuk pemecahan masalah tersebut. Metode NER dipilih karena model dapat dilatih menggunakan kategori-kategori yang diinginkan oleh pengguna, sehingga pengklasifikasian data teks dapat berjalan dengan lebih akurat dan sesuai kebutuhan.

Pengisian data otomatis atau sistem autofill umumnya dirancang dengan menyimpan informasi personal dari pengguna dan merancang suatu profil untuk pengisian otomatis pada formulir berikutnya. Beberapa aplikasi ini memiliki keterbatasan, seperti inisiasi profil awal secara manual atau tidak berlakunya autofill pada salah satu situs saat digunakan pada situs lainnya [2].

Tabel 1.
Daftar Label

Label	Kategori
JUD	Nama/Judul Magang
DED	Batas Pengumpulan Lamaran
SEM	Batas Awal Semester
POS	Posisi yang Ditawarkan
DEP	Departemen yang Dapat Mendaftar

Tabel 2.
Contoh Pelabelan Manual

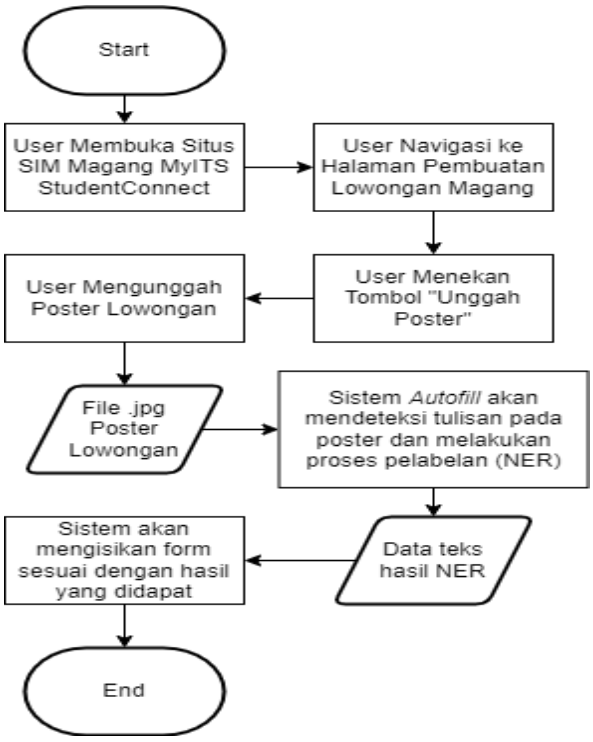
Kata	Label
PROGRAM	B-JUD
MAGANG	I-JUD
MAHASISWA	I-JUD
BERSERTIFIKAT	I-JUD
2021	I-JUD
TEKNIK	B-DEP
INFORMATIKA	I-DEP
PROGRAMMER	B-DEP
HUBUNGAN	B-DEP
MASYARAKAT	I-DEP
JURNALISTIK	B-DEP
MANAJEMEN	B-DEP
KOMUNIKASI	I-DEP
ILMU	B-DEP
KOMUNIKASI	I-DEP
DESAIN	B-DEP
KOMUNIKASI	I-DEP
VISUAL	I-DEP

```
[[[1070, 90], [1284, 90], [1284, 176], [1070, 176]],
 'FORUM HUMAN CAPITAL INDONESIA'],
 [[[1893, 83], [2103, 83], [2103, 175], [1893, 175]],
 'P U P U K INDONESIA Holding company'],
 [[[57, 69], [535, 69], [535, 181], [57, 181]], 'BuMn UNTUK INDONESIA'],
 [[[756, 288], [1405, 288], [1405, 355], [756, 355]],
 'Mari Bergabung Bersama'],
 [[[660, 346], [1504, 346], [1504, 420], [660, 420]],
 'Pupuk Kujang dengan Mengikuti'],
 [[[137, 478], [940, 478], [940, 1192], [137, 1192]],
 'PROGRAM MAGANG MAHASISWA BERSERTIFIKAT 2021'],
 [[[1125, 528], [2040, 528], [2040, 974], [1125, 974]],
 'Kriteria 1. MAHASISWA AKTIF MINIMAL SEMESTER 5 2 BERSEDIA MAGANG SELAMA 6 BULAN
3. MEMILIKI SURAT REKOMENDASI DARI FAKULTAS 4. PERGURUAN TINGGI SUDAH
MENANDATANGANI MOU DENGAN FHCI UNTUK PMMB'],
 [[[1124, 1003], [2034, 1003], [2034, 1898], [1124, 1898]],
 'Jurusan 1 TEKNIK INFORMATIKA PROGRAMMER 2 HUBUNGAN MASYARAKAT JURNALISTIK
MANAJEMEN KOMUNIKASI ILMU KOMUNIKASI / DESAIN KOMUNIKASI VISUAL Tahapan Seleksi
1.ADMINISTRASI 2. INTERVIEW USER Benefit 1. SERTIFIKAT FHCI 2. UANG SAKU SESUAI
DENGAN ATURAN YANG BERLAKU'],
 [[[116, 1684], [940, 1684], [940, 1948], [116, 1948]],
 'Pendaftaran Paling Lambat 15 Maret 2021 Informasi & Pendaftaran https:/ /tinyurl
com/PMMB2021'],
 [[[19, 2083], [491, 2083], [491, 2143], [19, 2143]],
 '#SOLUSIANDALAGRIBISNIS'],
 [[[568, 2091], [777, 2091], [777, 2135], [568, 2135]], 'pupukkujang'],
 [[[865, 2091], [1151, 2091], [1151, 2135], [865, 2135]], '@kujangcikampek'],
 [[[1243, 2094], [1661, 2094], [1661, 2136], [1243, 2136]],
 'PT Pupuk Kujang Cikampek'],
 [[[1763, 2093], [2139, 2093], [2139, 2136], [1763, 2136]],
 'PT Pupuk Kujang Official']]
```

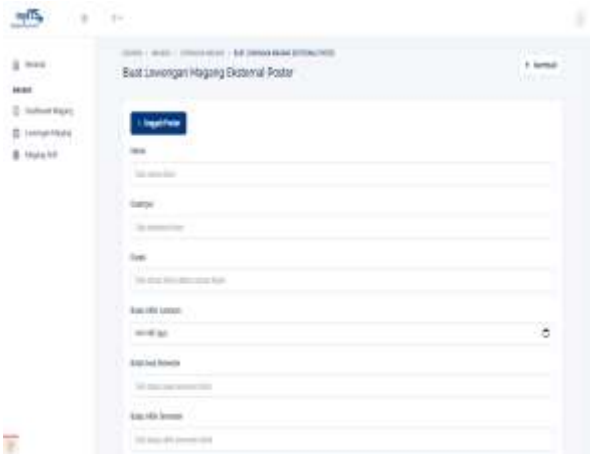
Gambar 5. Contoh Hasil OCR.

Untuk mengatasi keterbatasan tersebut, autofill yang dilakukan oleh situs MyITS StudentConnect didapatkan dari hasil pengunggahan dokumen dan pemindaian dengan NER, sehingga autofill dapat dikhususkan untuk keperluan pengunggahan lowongan magang.

Hasil dari penelitian ini diharapkan dapat menghasilkan sistem konversi dokumen dan autofill informasi dalam format SIM Magang ITS yang dapat meningkatkan kinerja dan efisiensi SIM Magang dalam melakukan pendistribusian informasi terkait magang yang tersedia dan dapat dilamar oleh mahasiswa-mahasiswa ITS.

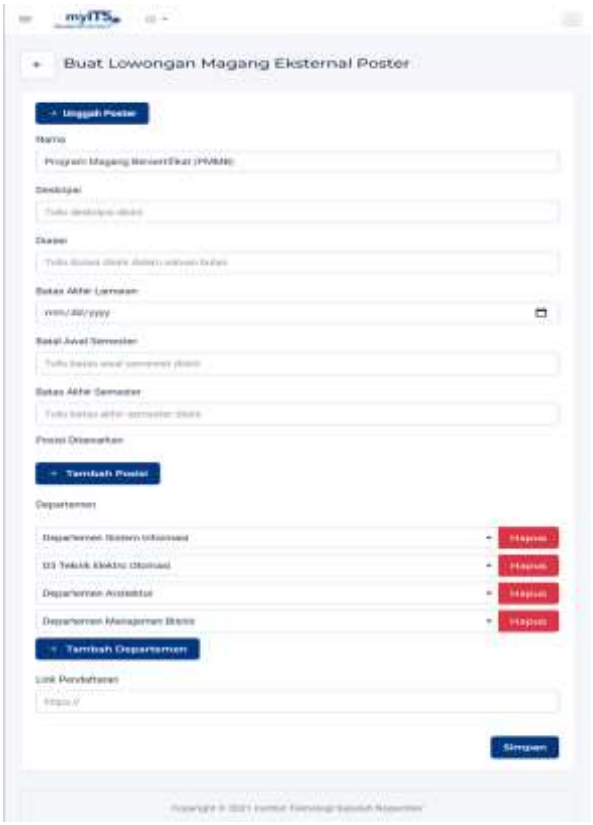


Gambar 6. Diagram Alir Proses Penggunaan Sistem.



Gambar 7. Halaman Pembuatan Lowongan Magang.

Gambar 8. Proses Pengunggahan Poster.



Gambar 9. Halaman Setelah Pengunggahan.

II. METODE PENELITIAN

A. *Pengumpulan Dataset*

Data yang digunakan sebagai input data latih pada sistem yang dibuat merupakan poster-poster lowongan magang, terutama lowongan yang dapat dilamar oleh mahasiswa. Proses pengumpulan data didapat dari pihak PK2 ITS sebagai distributor layanan magang di lingkungan ITS dan pengumpulan gambar manual melalui Google Images, dengan kata kunci, "lowongan magang mahasiswa", "poster magang", dan "magang mahasiswa". Berdasarkan proses ini, didapatkan 30 jumlah poster lowongan magang yang nantinya akan dibagi menjadi 24 data train dan 6 data test.Contoh data train dan data test dapat dilihat pada Gambar 1 dan Gambar 2.

B. *Praproses Gambar*

Praproses gambar dilakukan dengan merubah nilai *contrast* dan mengubah gambar yang diunggah menjadi hitam dan putih. Hal ini dilakukan untuk mengurangi *noise* pada gambar berupa *background* atau obyek-obyek lain yang dapat mempengaruhi proses *Optical Character Recognition* (OCR). Gambar 3 akan menunjukkan contoh hasil praproses dari Gambar 1.

C. *Optical Character Recognition*

Gambar yang telah dipraproses pada bagian sebelumnya akan dipindai menggunakan *library* EasyOCR. Pemindaian gambar ini akan menghasilkan beberapa *bounding box* yang berisi teks hasil *Optical Character Recognition*. Hasil dari proses OCR dapat dilihat pada Gambar 4 dan Gambar 5.

D. *Pelabelan Manual*

Penandaan per kata secara manual dilakukan dengan memperhatikan beberapa kategori yang dibutuhkan untuk tiap *field* pada form lowongan magang. Setiap kata ditandai menggunakan *BIO-Tagging* sehingga membentuk frasa yang dapat dikenali oleh sistem sebagai kategori tertentu. Kategori yang digunakan dapat dilihat pada Tabel 1.

Tabel 2 merupakan contoh untuk pelabelan manual pada hasil OCR yang didapatkan pada Gambar 5. Semua kata yang

Tabel 3.
Contoh Pelabelan Manual

Parameter	Keterangan	Nilai
img	Input image	img
paragraph	Menggabungkan hasil dalam bentuk paragraf	false
min_size	Menghilangkan *bounding box* dengan ukuran *pixel* < nilai	0
slope_ths	Nilai maksimal kemiringan (Δy/Δx) untuk digabungkan menjadi satu *bounding box*.	0.2
ycenter_ths	Nilai maksimal pergeseran pada sumbu y. Area yang ukurannya > nilai, tidak akan digabung menjadi satu *bounding box.*	0.7
contrast_ths	*Bounding box* dengan nilai *contrast* < nilai parameter akan diatur *contrast*-nya melalui parameter *adjust_contrast* (*float*, default=0.5).	0
height_ths	Nilai maksimal perbedaan pada tinggi *bounding box*. Apabila dua *bounding box* bersebelahan dan tingginya melebih nilai, maka kedua *bounding box* tersebut tidak akan dijadikan satu.	0.6
decoder	Pemilihan metode decode	beamsearch
beamWidth	Jumlah *beam* yang digunakan ketika menggunakan *decoder*.	10

Tabel 4.
Hasil Uji Coba Praproses

Metode yang Diujikan	Hasil Cosine Similarity
Enhance Contrast dengan Nilai Faktor 4	0.7459
Enhance Contrast dengan Nilai Faktor 7	0.7821
Enhance Contrast dengan Nilai Faktor 10	0.766
Adaptive Mean Thresholding	0.3144
Adaptive Gaussian Thresholding	0.3319
Otsu's Thresholding	0.7588

tergolong dalam kategori Tabel 1 akan diberikan label "B" atau "I" beserta kategorinya. Sementara itu, kata-kata lain yang tidak termasuk dalam Tabel 2 akan diberi label "O".

E. Pelatihan Model

Data yang sudah disiapkan melalui pelabelan manual akan dilatih untuk membuat model yang digunakan saat proses *named-entity recognition* (NER). *Named-entity recognition* (NER) adalah komponen utama dari ekstraksi informasi yang melabeli data input dengan kategori-kategori sesuai dengan data training. Ibaratnya, *named-entity recognition* di sini berfungsi untuk melanjutkan Manual BIO-Tagging yang dibuat pada inputan data teks yang belum dilabeli untuk menghasilkan keluaran berupa sekumpulan teks yang telah dikategorikan.

Dalam proses model *fitting*, digunakan juga *pre-trained vector* yang didapat dari Wikipedia dengan bahasa terkait. *Pre-trained vector* ini dapat diunduh secara langsung melalui situs *FastText*, Bagian inilah yang menjadi proses pembuatan *word embeddings*.

Untuk dapat melakukan *named-entity recognition*, data latih yang telah dilakukan manual BIO-tagging, bersamaan dengan hasil *word embeddings*, akan di-fit ke dalam model. Model tersebut dibuat dengan *optimizer* dan pengulangan (epochs) tertentu untuk menghasilkan model yang bagus dan siap pakai.

Tabel 5.
Contoh Pengujian Label Dataset

Dengan Deskripsi		Tanpa Deskripsi	
ketentuan	B-DES	ketentuan	O
sebagai	I-DES	sebagai	O
berikut	I-DES	berikut	O
1.	I-DES	1.	O
Mahasiswa	I-DES	Mahasiswa	O
angkatan	I-DES	angkatan	O
2018/2019/2020	I-DES	2018/2019/2020	O
2.	I-DES	2.	O
Memiliki	I-DES	Memiliki	O
kemampuan	I-DES	kemampuan	O
videografi,	I-DES	videografi,	O
audio,	I-DES	audio,	O
fotografi,	I-DES	fotografi,	O
editing,	I-DES	editing,	O
copywriting,	I-DES	copywriting,	O
dan	I-DES	dan	O
ilustrasi;	I-DES	ilustrasi;	O

Tabel 6.
Hasil Pengujian Label Dataset

	Dengan Deskripsi	Tanpa Deskripsi
Precision	0.46835	0.6
Recall	0.55224	0.52174
F1-Score	0.50685	0.55814

Tabel 7.
Hasil Pengujian Label Dataset

	RMSProp	Adam	Adadelta
150	0.57143	0.48649	0.2439
300	0.5	0.53333	0.48148
600	0.59259	0.55814	0.39216
1000	0.5	0.60465	0.45614

F. Pemasangan Situs SIM Magang

Setelah program *autofill* telah dibuat, langkah berikutnya adalah pemasangan pada situs MyITS StudentConnect. Prosesnya dapat dilihat pada Gambar 6.

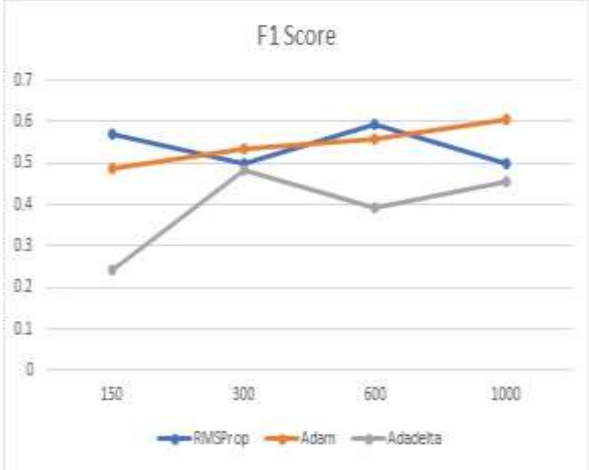
Alur yang tertera pada Gambar 6 dimulai dari pengguna diminta untuk mengakses akun MyITS terlebih dahulu melalui situs *my.its.ac.id*. Setelah itu, pengguna dapat mengakses menu MyITS StudentConnect pada salah satu pilihan menu yang tersedia pada *dashboard*.Pada bagian menu navigasi, pengguna dapat mengakses *sub-menu* Magang dan halaman pembuatan lowongan magang. Dari sana, akan muncul tombol untuk mengunggah poster dan poster yang telah diunggah akan secara otomatis terproses oleh sistem dan melakukan pengisian secara otomatis pada formulir lowongan yang tersedia.

Pada praktik ini, tentunya hanya pengguna dengan *role* tertentu yang dapat melakukan akses terhadap halaman lowongan magang, contohnya admin PK2. Admin PK2 akan dapat mengecek kembali isian otomatis yang telah dimasukkan oleh sistem dan mengisikan bagian-bagian yang masih kurang lengkap apabila diperlukan.

Nantinya, pengguna berupa admin PK2 akan dapat memasukkan lowongan magang pada SIM Magang MyITS StudentConnect dan mengunggah poster lowongan magang. Halaman situs akan tampak seperti pada Gambar 7 dan 8.

Dalam kasus ini, dicoba untuk menggunakan pengunggahan contoh data tes (Gambar 1) pada sistem. Sistem akan melakukan proses pemindaian gambar dan melakukan klasifikasi pada teks yang ditemukan pada gambar. Hasil klasifikasi tersebut akan di-*append* satu

persatu berdasarkan kolom pada formulir lowongan yang



Gambar 10. Grafik F1-Score Uji Coba Model.

Gambar 11. Contoh Data Tes yang Diujikan.

sesuai. Tampilan halaman yang telah diisikan dapat dilihat pada Gambar 9.

III. UJI COBA DAN ANALISA

A. *Uji Coba Praproses Dataset*

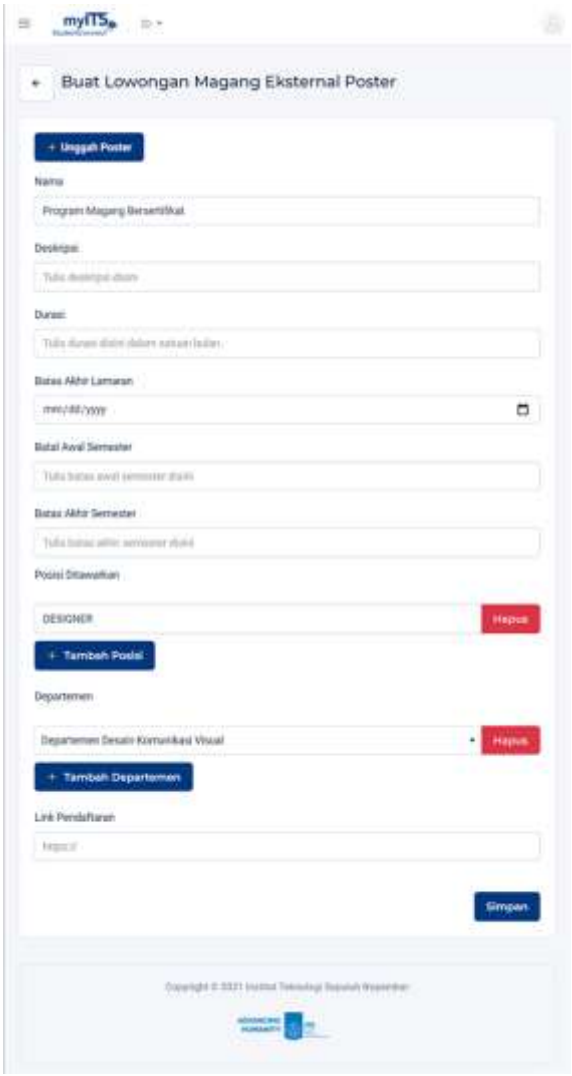
Uji coba praproses dataset dilakukan melalui penggantian cara gambar diolah sebelum dimasukkan pada proses *optical character recognition* (OCR). Proses ini akan digunakan untuk membersihkan *noise* pada *dataset* yang akan diolah dalam OCR, sehingga penting untuk diujicobakan.

Uji coba dilakukan menggunakan beberapa variasi praproses dengan variasi sebagai berikut:

1. Metode Enhance Contrast dengan Nilai Faktor 4
2. Metode Enhance Contrast dengan Nilai Faktor 7
3. Metode Enhance Contrast dengan Nilai Faktor 10
4. Metode Adaptive Mean Thresholding
5. Metode Adaptive Gaussian Thresholding
6. Metode Otsu's Thresholding

Otsu's Thresholding bersifat dinamis dan dapat secara otomatis mengomputasi nilai *threshold* paling optimal untuk gambar yang diujikan. Sehingga, berapapun nilai *threshold* yang dimasukkan ke dalam parameternya, hasilnya akan tetap sama.

6 metode uji coba ini diujikan melalui proses *Optical*



Gambar 12. Hasil Pengunggahan Data Tes.

Character Recognition dengan parameter seperti pada Tabel 3 pada 6 gambar data *test*, diambil nilai vektornya menggunakan TF-IDF Vectorizer dengan membandingkan string hasil OCR dengan string yang benar, dan dikeluarkan nilai *cosine similarity*-nya.

Pemilihan parameter yang digunakan untuk proses pembuatan *bounding box* oleh EasyOCR dianggap tidak terlalu signifikan pada hasil atau performa aplikasi. Hal ini dikarenakan data hasil pembacaan yang nantinya akan dipisah per kata, bukan per kalimat atau per frasa.

Selanjutnya, urutan pemindaian kata dalam proses ini juga tidak mengganggu kinerja sistem. Proses pelabelan manual nantinya dapat membuat tiap kata tergolongkan secara manual dan pada proses klasifikasi teks dan *input* pada formulir magang juga akan di-*append* per kata.

Sementara itu, nilai *contrast_ths* pada parameter *bounding box* sengaja diisi dengan nilai 0 karena gambar *input* sudah melalui praproses perubahan *contrast* terlebih dahulu sebelum dipindai menggunakan EasyOCR.

Percobaan pada tiap data tes memberikan hasil seperti pada Tabel 4 untuk rata-rata hasil pada tiap metode yang diujikan. Maka dari itu, akan digunakan metode praproses gambar dengan metode *enhance contrast* dengan nilai faktor sebesar 7.

Gambar 13. Kesalahan Pengisian Akibat Desain Poster.

Tabel 8.
Hasil Scoring Optimizer Adam

	Precision	Recall	F1-Score
150	0.45	0.52941	0.48649
300	0.48	0.6	0.53333
600	0.6	0.52174	0.55814
1000	0.59091	0.61905	0.60465
Mean	0.53023	0.56755	0.54565

B. Uji Coba Pelabelan Dataset

Untuk menguji coba pelabelan *dataset* digunakan 2 skenario yang mempengaruhi hasil akhir pembuatan sistem, yaitu diikutkannya label DES (Deskripsi) atau tidak pada sistem *named-entity recognition*. Pelabelan data DES (Deskripsi) ini dilakukan pada keseluruhan *dataset*. Contoh perbedaannya dapat dilihat pada tabel 5.

Dilakukan juga perubahan terhadap keseluruhan dataset, baik data *train* maupun data *test* dengan total 30 jumlah data teks. Masing-masing disesuaikan dengan label-label yang digunakan untuk keperluan pelabelan manual seperti yang tertera pada Tabel 1.

Setelah dilakukan *data fitting* dengan menggunakan *optimizer* Adam dan *epochs*=600, model diujikan terhadap data tes dengan syarat yang sama (model dengan deskripsi diujikan dengan data tes dengan deskripsi, model tanpa deskripsi diujikan dengan data tes tanpa deskripsi).

Epochs sebanyak 600 dipilih sebagai titik tengah jumlah pelatihan, tujuannya agar *training* tidak selesai terlalu awal sehingga menimbulkan *underfitting* dan tidak dilatih terlalu banyak sehingga menimbulkan *overfitting*.

Metode pengujian dilakukan dengan membandingkan hasil NER untuk masing-masing model dengan data tes asli untuk memvalidasi, menghasilkan F1-score sesuai dengan Tabel 6. Nilai pada tabel 6 tersebut diperoleh dari model yang dilatih menggunakan 24 data *training* terhadap 6 data *testing*.

Maka dari itu, akan digunakan *dataset* yang tidak menyertakan label deskripsi. Hal ini dianalisa akibat data yang memiliki deskripsi akan menghasilkan model dengan klasifikasi yang terlalu umum sehingga menimbulkan *bias* kepada kategori deskripsi yang kontennya dapat mencakup kategori-kategori lain.

C. Uji Coba Pembuatan Model

Untuk menguji coba metode pembuatan model, perlu diteliti nilai akhir kesesuaian model dengan data tes. Untuk keperluan pengujian digunakan beberapa variabel sebagai berikut:

1. Variabel Bebas: Jumlah epoch saat pelatihan model (150, 300, 600, 1000) dan jenis metode optimisasi (Adam, RMSProp, dan Adadelta).
2. Variabel Terikat: Hasil F1-score terhadap data tes asli.
3. Variabel Kontrol: Data latih tanpa deskripsi (mengacu pada subbab sebelumnya), data tes, parameter lain pada pembuatan model, dan lingkungan uji coba.

Untuk perhitungan skor pada pengujian ini, digunakan metode yang sama dengan subbab sebelumnya, yaitu membandingkan data tes hasil NER dan data tes hasil pelabelan secara manual.

Pengujian dilakukan menggunakan model yang telah dilatih dari 24 data *train*, lalu diujikan terhadap 6 data *test*. Didapatkan hasil berupa *F1-Score* pada Gambar 10 dengan detail hasil pada Tabel 7.

Maka dari itu, akan digunakan *optimizer* berupa Adam tanpa menyertakan label deskripsi. *Epochs* yang digunakan untuk melatih model berjumlah 1000, memungkinkan untuk melakukan *training* dengan *epochs* yang lebih besar, namun harus didukung dengan banyaknya data latih juga agar tidak terjadi *overfitting* pada model. Detail perhitungan skor *optimizer* Adam dengan parameter sesuai dengan variabel uji coba dapat dilihat pada Tabel 8.

Optimizer Adam merupakan algoritma optimisasi yang merupakan perkembangan dari stochastic gradient descent klasik yang dapat memperbarui weight antar jaringan secara iteratif berdasarkan data latih. Dalam praktiknya, Adam menggunakan momentum dan *scaling*, yang merupakan keunggulan dari RMSProp dan SGD dengan Momentum. Adam dirancang untuk bentuk masalah dengan gradien yang sangat berantakan [3].

D. Uji Coba Kinerja Autofill pada Sistem

Setelah dilakukan pelatihan pada model dan diaplikasikan pada situs, perlu ada pengujian terhadap jalannya *autofill* pada sistem. Oleh karena itu, dilakukan pengunggahan 6 data *test* untuk mengetes keseluruhan program, apakah sudah dapat berjalan dengan baik atau tidak. Salah satu gambar yang diujikan terlihat pada Gambar 11 sebagai salah satu dari 6 data tes.

Pada Gambar 12, terlihat bahwa *field* yang diisikan pada formulir lowongan magang cukup akurat. Pada kolom judul magang, "Program Magang Bersertifikat" lebih dipilih ketimbang "Designer Internship Program" dikarenakan hubungan data latih yang lebih kuat untuk kata "Program", "Magang", dan "Bersertifikat", dibandingkan "Designer", "Internship", dan "Program".

Di sisi lain, kata "Designer" terindikasi sebagai *Posisi Ditawarkan* dan memang "designer" juga dianggap sebagai posisi yang dicari pada lowongan magang tersebut. Setelah itu, pemindaian "Desain Komunikasi Visual" juga dengan akurat tergolongkan sebagai *Departemen* yang dapat melamar pada lowongan tersebut.

Permasalahan yang terindikasi muncul dari pengujian unggah data tes terlihat pada Gambar 13. Pada Gambar 13, sistem membaca kata "BERSERTIFIKATU" daripada

"BERSERTIFIKAT". Hal ini diakibatkan kesalahan pada desain poster awal, dimana terdapat ilustrasi yang terlalu dekat dengan tulisan dan menyerupai abjad "U" sehingga terindikasi sebagai satu kesatuan frasa. Munculnya masalah tersebut juga ditambah dengan pewarnaan putih yang digunakan untuk ilustrasi, menyebabkan *pre-processing* pada gambar tidak dapat menghilangkan ilustrasi tersebut.

E. Analisa Uji Coba

Pada sub bab analisa uji coba, akan dibahas mengenai kesimpulan dari performa uji coba yang dilakukan.

1. Kesalahan pada analisa saat NER dijalankan terjadi dikarenakan lemahnya model. Model dapat diperbaiki dengan membuat data latih yang lebih banyak dan lebih bervariasi.
2. Penggunaan data latih tanpa mengikutsertakan deskripsi dapat mencapai hasil yang lebih baik karena sifat deskripsi yang terlalu umum dan dapat menimbulkan bias saat dijadikan data latih.
3. Penggunaan metode pembuatan yang paling baik adalah dengan menggunakan Adam dengan epochs sejumlah 1000.
4. Kesalahan atau *error* dalam jalannya sistem dapat dikarenakan kesesuaian desain poster (peletakkan objek, warna, dan sebagainya) dan lemahnya model.

IV. KESIMPULAN DAN SARAN

Dalam pengerjaan penelitian ini setelah melalui tahap perancangan aplikasi, implementasi metode, serta uji coba, diperoleh kesimpulan sebagai berikut: (1)Praproses gambar dapat dilakukan dengan menggunakan pengubahan *contrast* dan perubahan ke dalam bentuk *grayscale*. Berdasarkan pengujian, hasil terbaik diperoleh dengan menggunakan metode *enhance contrast* dengan nilai faktor 7, memiliki nilai *cosine similarity* sebanyak 0.7821 setelah diujikan pada 6 data tes. (2)Pengenalan entitas menggunakan *named-entity recognition* menghasilkan performa yang cukup baik didasari dengan *F1-Score* sebesar 0.60465 untuk *epochs* sebanyak 1000, meskipun memiliki data latih yang relatif sedikit.(3)Penggunaan *optimizer* Adam saat membuat model dapat menimbulkan hasil yang cukup baik, yaitu rata-rata 0.54565 dari penggunaan *epochs* 150, 300, 600, dan 1000.

Adapun saran yang diberikan untuk pengembangan penelitian selanjutnya, yaitu;(1)Memasukkan cara melakukan perbaikan citra setelah di-*compress* agar *size* gambar saat diolah lebih kecil namun tidak merusak hasil OCR.(2)Penambahan *dataset* yang lebih bervariasi untuk pembuatan model NER sehingga dapat menghasilkan keluaran hasil yang lebih baik.

DAFTAR PUSTAKA

[1] N. Perera, M. Dehmer, and F. Emmert-Streib, “Named Entity recognition and relation detection for biomedical information extraction,” *Front. Cell Dev. Biol.*, vol. 8, pp. 1–126, 2020, doi: 10.3389/fcell.2020.00673.

[2] S. Wang, Y. Zou, I. Keivanloo, B. Upahyaya, and J. Ng, “An intelligent framework for auto-filling web forms from different web applications,” *Int. J. Bus. Process Integr. Manag.*, vol. 8, no. 1, pp. 2–16, 2017, doi: 10.1504/IJBPIM.2017.082747.

[3] J. L. B. Diederik P. Kingma, “A Method for Stochastic Optimization,” in *International Conference on Learning Representations*, San Diego 2015, pp. 1–15.