Deep Learning Untuk *Entity Matching* Produk Kamera Antar *Online Store* Menggunakan DeepMatcher

Adam Akbar[1], Nisrina Fadhilah Fano[2] dan Nur Aini Rakhmawati[3]
[1,2,3]*Jurusan Sistem Informasi; Fakultas Teknologi Elektro dan Informatika Cerdas;Institut Teknologi Sepuluh Nopember;*
Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Kota SBY, Jawa Timur 60111
E-mail : adamakbar.id@gmail.com[1], nisrina.fano@gmail.com[2], nur.aini@is.its.ac.id[3]

Abstract—In Computer Science field, Entity Matching has become a challenge for some researchers. Some have tried to develop an entity matching algorithm to improve accuracy. This study will test DeepMatcher as a representation of Entity Matching using Deep Learning by matching entities against case studies of matching camera products at two online stores using four different learning algorithms that DeepMatcher has, namely Smooth Inverse Frequency, Bidirectional RNN, Decomposable Attention Model, and Hybrid. Model. By building a dataset and a learning model, DeepMatcher can independently match data that has not been previously entered. The matching results will be measured using an f-measure to then analyze its reliability. The test results show that the type of learning on DeepMatcher that is most suitable for using in entity matching on camera products between online stores is Bidirectional RNN with an average resulting F1 score of 61.546.

Abstrak—Dalam bidang ilmu *Computer Science*, *Entity Matching* telah menjadi tantangan tersendiri bagi beberapa peneliti. Beberapa berusaha mengembangkan algoritma *entitiy matching* untuk meningkatkan akurasi. Penelitian ini akan menguji DeepMatcher sebagai representasi *Entity Matching* yang menggunakan Deep Learning dengan melakukan pencocokan entitas terhadap studi kasus pencocokan produk kamera pada dua *online store* menggunakan empat algoritma pemelajaran berbeda yang dimiliki oleh DeepMatcher yakni *Smooth Inverse Frequency*, *Bidirectional RNN*, *Decomposable Attention Model*, dan *Hybrid Model*. Dengan membangun *dataset* dan model pemelajaran, DeepMatcher dapat melakukan pencocokan secara mandiri pada data yang belum dimasukkan sebelumnya. Hasil pencocokan tersebut akan diukur menggunakan *f-measure* untuk kemudian dianalisa kehandalannya. Hasil pengujian menunjukkan bahwa janis pemelajaran pada DeepMatcher yang paling cocok untuk digunakan dalam melakukan *entity matching* pada produk kamera antar *online store* adalah *Bidirectional RNN* dengan rata-rata skor F1 yang dihasilkan adalah 61,546.

Kata Kunci : *smooth inverse frequency, bidirectional rnn, decomposable attention model, hybrid model, entity matching, deep learning, DeepMatcher*.

I. PENDAHULUAN

INTEGRASI data merupakan hal yang sangat penting untuk memadukan data dari banyak sumber agar tersedianya informasi yang lebih luas [1]. Salah satu teknik yang digunakan dalam integrasi data adalah *entity matching* untuk memastikan bahwa data-data yang berasal dari banyak sumber tersebut merupakan entitas yang sama di dunia nyata [2] [3]. Sebagai contoh, sebuah produk pada satu *website* dapat tersajikan dengan nama dan deskripsi yang berbeda pada *website* yang lain walau produk tersebut merupakan barang yang serupa. Masalah tersebut sering dialami calon pembeli ketika ingin mencari harga termurah untuk suatu produk pada banyak *online store* [3].

Kegiatan *entitiy matching* dapat dibagi menjadi lima langkah: *data pre-processing*, *indexing*, *record-pair comparison*, klasifikasi, dan evaluasi [4]. Langkah pertama dimaksudkan untuk memastikan data semua sumber yang akan diolah telah sesuai dengan format standar yang ditetapkan. Tiga langkah berikutnya merupakan langkah utama dari *entity matching*. Langkah terakhir yaitu evaluasi hasil untuk mengukur akurasi pencocokan entitas yang telah dilakukan [3]. Selain pencocokan data, *entity matching* juga digunakan untuk membersihkan data yang *redundan* [2]-[4].

Dalam bidang ilmu *Computer Science*, *entity matching* telah menjadi tantangan tersendiri bagi beberapa peneliti. Beberapa berusaha mengembangkan algoritma *entitiy matching* untuk meningkatkan akurasi. Beberapa lagi mengembangkan *entity matching* dengan menerapkan prinsip pemelajaran mesin sebagai inti sistem seperti grup AnHai Doan yang telah membangun sebuah program komputer bernama DeepMatcher.

DeepMatcher merupakan *package* untuk bahasa pemrograman *Python* yang dibangun oleh grup AnHai Doan untuk melakukan *Entity Matching* dengan menggunakan *Deep Learning* [5]. grup AnHai Doan merupakan sebuah grup dari Universitas Wisconsin-Madison yang berfokus pada penelitian dan pengembangan dalam bidang manajemen data [6]. Salah satu proyek dari grup AnHai Doan adalah Magellan yang diambil dari nama Ferdinand Magellan. Proyek tersebut berfokus pada pengembangan konsep *Entity Matching* yang merupakan permasalahan

mendasar dalam integrasi data [7]. Salah satu produk dari proyek Magellan adalah DeepMatcher.

Penelitian ini akan menguji DeepMatcher sebagai representasi *Entity Matching* yang menggunakan *Deep Learning* dengan melakukan pencocokan entitas terhadap studi kasus pencocokan produk kamera pada dua *online store* berbeda.

II. METODE PENELITIAN

A. Dataset

Untuk membangun *dataset* yang diperlukan oleh *DeepMatcher* sebagai pemelajaran dan pengujian, penelitian ini mengumpulkan data dari dua *online store* yang menjual produk kamera yaitu galaxycamera.id (*S1*) dan tokocamzone.com (*S2*) dengan menggunakan teknik *web crawling*, hasil pengumpulan data tersebut diselaraskan dengan menghapus kolom-kolom yang tidak beririsan antar dua sumber lalu dipasangkan dengan kombinasi setiap baris data pada *S1* dipasangkan dengan setiap baris data pada *S2* untuk mencapai semua probabilitas yang ada. Setelah itu, pasangan data yang benar-benar berbeda akan dieliminasi (*blocking*) untuk mengurangi pemelajaran yang tidak diperlukan oleh DeepMatcher.

Eliminasi baris pasangan data pada penelitian ini dilakukan dengan menggunakan *package python* dari AnHai's Group juga yang bernama py_entitymatching dengan pengaturan *overleap* adalah 5 yang berarti py_entitymatching akan menghapus baris pasangan data jika tidak terdapat 5 kata yang sama pada atribut yang sama.

Baris-baris pasangan data yang tidak tereliminasi pada langkah sebelumnya akan menjadi kandidat *dataset* yang kemudian akan dipecah menjadi tiga *dataset* yang diperlukan oleh DeepMatcher yaitu *train* dan *validation* untuk proses pemelajaran dan *test* untuk pengukuran hasil pemelajaran, pemecahan *dataset* kandidat dilakukan secara acak dengan rasio 3:1:1 di mana 3 untuk *dataset train* dan 1 untuk *dataset validation*, dan *test*.

Setiap baris pasangan data pada tiga *dataset tersebut* akan dilabeli secara manual dengan nilai 0 untuk pasangan data yang tidak serupa atau 1 untuk pasangan data yang serupa untuk digunakan oleh DeepMatcher sebagai acuan dalam pembentukan model pemelajaran.

B. Model Pemelajaran

DeepMatcher menyediakan empat jenis pemelajaran *deep learning* yaitu *Smooth Inverse Frequency* (*sif*) yang mempertimbangkan kecocokan antar entitas berdasarkan nilai atribut tanpa memperhitungkan urutannya, *Bidirectional RNN* (*rnn)* yang mempertimbangkan urutan nilai atribut antar entitas, *Decomposable Attention Model* (*att*) yang mempertimbangkan keselarasan nilai setiap atribut antar entitas, dan *Hybrid Model* (*hybrid*) yang mempertimbangkan keselarasan urutan nilai atribut antar entitas [2].

Setelah menentukan jenis pemelajaran, selain *dataset* yang sudah dibangun sebelumnya, DeepMatcher juga memiliki *paramater* lain sebagai syarat utama untuk melakukan pemodelan yaitu *epochs, batch_size*, dan *pos_neg_ratio* yang semuanya dalam bentuk bilangan desimal. *Epoch* dimaksudkan untuk jumlah iterasi latihan yang dilakukan oleh DeepMatcher, *batch_size* mengacu pada jumlah baris data yang akan dipelajari oleh DeepMatcher untuk setiap langkah pada satu latihan (*training step)*, dan *pos_neg_ratio* adalah nilai bobot untuk *dataset train* yang ditentukan berdasarkan rasio jumlah baris berlabel 0 dan 1 pada *dataset* tersebut sebagai penyeimbang proses pemelajaran. Semua *parameter* yang sudah ditentukan tadi akan mempengaruhi kinerja dan akurasi hasil pemelajaran.

Pada tahap latihan, pembentukan model pemelajaran didasarkan pada *dataset train* dengan kata lain DeepMatcher 'mempelajari' pelabelan yang dilakukan oleh penulis berdasarkan atribut yang terdapat pada *dataset* tersebut. *Dataset validation* digunakan untuk mengevaluasi hasil latihan sebagai acuan untuk menentukan model pemelajaran terbaik berdasarkan skor F1 dari *dataset validation* tersebut. DeepMatcher akan menyimpan model pemelajaran terbaik dari semua pelatihan yang telah dilakukan untuk nanti digunakan dalam melakukan prediksi kecocokan.

$$prec = \frac{TP}{(TP + FP)} \tag{1}$$

$$rec = \frac{TP}{(TP + FN)} \tag{2}$$

$$F1 = \frac{2 \times prec \times recall}{(prec + recall)} \tag{3}$$

Precision (*prec*), *recall* (*rec*), dan *F1-score* (*f-measure*) merupakan pengukuran yang populer dalam *machine learning* [8]. Secara sederhana *prec* dapat diartikan sebagai perbandingan antara prediksi positif yang tepat (*TP*)(*true positive*) dan semua prediksi positif yang dihasilkan mesin (*TP + FP)*(*false positive*). Persamaan untuk nilai *prec* dapat dilihat pada persamaan (1). Selanjutnya, *rec* dapat diartikan sebagai perbandingan antara prediksi positif yang tepat (*TP*) dan semua baris yang sebenarnya positif (*TP + FN*)(*false negative*). Persamaan untuk nilai *rec* dapat dilihat pada persamaan (2). Skor F1 dapat diartikan sebagai nilai *mean* dari *prec* dan *rec* [8], dengan persamaan yang dapat dilihat pada persamaan (3). Berdasarkan pengukuran tersebut maka dapat disimpulkan bahwa model pemelajaran yang paling akurat dalam artian dapat memprediksi kecocokan tanpa kesalahan adalah model yang memiliki nilai 1 untuk skor F1-nya atau 100 jika disajikan dalam bentuk persen.

C. Pengujian

Pada penelitian ini, pengujian dilakukan pada layanan Colaboratory milik Google yang merupakan layanan untuk mengeksekusi program Phyton berbasis *cloud* yang ditujukan untuk peneliti khususnya untuk profesi *data*

scientist dan *AI researcher*. Colaboratory tersedia dengan spesifikasi mesin: prosesor Intel(R) Xeon(R) CPU @ 2.30GHz Dual Core, RAM 13GB dan OS Ubuntu 18.04. Dengan menjalankan pelatihan sebanyak 10 sesi untuk setiap jenis pemelajaran, model pemelajaran hasil pelatihan akan diukur akurasinya pada setiap sesi dan model pemelajaran terbaik pada sesi sebelumnya akan disimpan untuk dilatih lagi pada sesi selanjutnya per jenis pemelajaran.

D. *Analisa*

Hasil penghitungan skor F1 untuk setiap jenis pemelajaran pada setiap sesi akan dicatat untuk dianalisis kinerja dan akurasi dari DeepMatcher. Proses analisis dilakukan dengan membandingkan nilai rata-rata, nilai maksimum, dan nilai minimum Skor F1 yang didapatkan. Hasil analisis kemudian digunakan untuk menyimpulkan seberapa baik DeepMatcher dalam melakukan prediksi kecocokan untuk studi kasus pencocokan produk kamera pada dua *online store* yang berbeda.

III. HASIL DAN PEMBAHASAN

A. *Dataset*

Teknik *web crawling* yang digunakan untuk mengumpulkan data menghasilkan 1158 baris data pada *S1* dan 1751 baris data pada *S2*. Data dari dua sumber tersebut menghasilkan kombinasi 2.027.658 baris pasangan data dengan atribut yang beririsan yaitu nama produk. Atribut lain yang terdapat pada *dataset* adalah url produk dan harga produk. Kemudian eliminasi baris pasangan data yang benar-benar berbeda menyisakan 592 baris yang menjadi *dataset* kandidat.

Dataset kandidat tersebut kemudian dibagi ke dalam 3 bagian secara acak untuk dipergunakan sebagai *dataset training*, *dataset testing* dan *dataset validation*. Jumlah data yang masuk ke dalam *dataset training* adalah 355 baris, jumlah data yang masuk ke dalam *dataset testing* adalah 119 baris, dan jumlah data yang masuk ke *dataset validation adalah 118*.

B. *Model Pemelajaran*

Terdapat tiga jenis variabel pada penelitian ini, yaitu variabel bebas, variabel tetap, dan variabel terikat. Variabel bebas pada penelitian ini adalah jenis pemelajaran yang digunakan. Dengan adanya perbedaan jenis pemelajaran yang digunakan, maka akan dilakukan pengujian untuk mendapatkan nilai dari variabel terikat, yaitu skor F1 hasil pemelajaran mesin. Sedangkan untuk variabel tetap yang nilainya sama dalam seluruh proses pengujian adalah nilai *epochs*, nilai *batch_size*, dan nilai *pos_neg_ratio*. Nilai *epochs* yang digunakan adalah 10 dan nilai *batch_size* yang digunakan adalah 16. Dua nilai ini dipilih dengan pertimbangan durasi pengujian di mana semakin besar nilai *epochs* yang digunakan, maka semakin lama durasi latihan yang dilakukan. Namun berbanding terbalik dengan *batch_size* di mana semakin besar nilai *batch_size* maka semakin cepat durasi pelatihan. Dan yang terakhir, nilai *pos_neg_ratio* yang digunakan adalah 7 berdasarkan rasio label positif dan negatif pada *dataset*.

C. *Pengujian*

Pengujian dilakukan dengan menjalankan 10 sesi pelatihan pada masing-masing jenis pemelajaran seperti yang sudah dipaparkan pada bab II.C. Hasil utama pengujian merupakan skor F1 dan durasi latihan disajikan dalam bentuk persen untuk skor F1 dan detik untuk durasi.

Hasil percobaan dengan menggunakan jenis pemelajaran *sif* dapat dilihat pada Tabel 1. Pada Tabel 1 dapat terlihat bahwa nilai skor F1 terbaik didapatkan pada sesi latihan ke-4 dan ke-5 dengan nilai 45,7143. Sedangkan nilai skor F1 terendah didapatkan pada sesi latihan ke-1 dengan nilai 27,0677. Sedangkan untuk durasi latihan tercepat didapatkan pada sesi ke-1 dengan waktu 5,496 detik. Sedangkan durasi latihan terlama didapatkan pada sesi pemelajaran ke-2 dengan waktu 5,657 detik.

Tabel 1. Hasil pengujian jenis pemelajaran *sif*

Sesi ke-	Skor F1	Durasi
1	27,0677	5,496241928
2	37,1429	5,657258996
3	44,4444	5,553248059
4	45,7143	5,535472307
5	45,7143	5,639346096
6	44,4444	5,558882805
7	33,3333	5,5201457
8	30,7692	5,567615156
9	34,4828	5,59575437
10	34,4828	5,502403132

Hasil percobaan dengan menggunakan jenis pemelajaran *rnn* dapat dilihat pada Tabel 2. Pada Tabel 2 dapat terlihat bahwa nilai skor F1 tertinggi diperoleh pada sesi pemelajaran ke-3 dan ke-7 dengan nilai 66,667. Sedangkan nilai skor F1 terendah diperoleh pada sesi pemelajaran ke-1 dengan nilai 54,167. Untuk nilai durasi latihan tercepat diperoleh pada sesi pemelajaran ke 9 dengan waktu 6.873 detik. Sedangkan durasi latihan terlama diperoleh pada sesi pemelajaran ke-1 dengan waktu 7,227 detik.

Tabel 2. Hasil pengujian jenis pemelajaran *rnn*

Sesi ke-	Skor F1	Durasi
1	54.1667	7.22701975
2	57.1429	7.002871031
3	66.6667	7.139425133
4	62.8571	7.100219364
5	58.8235	7.039652078
6	56.25	7.012675778
7	66.6667	7.190518061
8	64.8649	7.163365998
9	63.1579	6.71901748
10	64.8649	6.873993939

Hasil percobaan dengan menggunakan jenis pemelajaran *att* dapat dilihat pada Tabel 3. Pada Tabel 3Tabel 3 dapat terlihat bahwa nilai skor F1 tertinggi diperoleh pada sesi latihan ke-8 dengan nilai 60. Sedangkan nilai skor F1 terendah diperoleh pada sesi latihan ke-1 dengan nilai 46,376. Untuk durasi latihan tercepat diperoleh pada sesi latihan ke-4 dengan waktu 9,677 detik. Sedangkan durasi terlama diperoleh pada sesi latihan ke-2 dengan waktu 10,036 detik.

Tabel 3. Hasil pengujian jenis pemelajaran *att*

Sesi ke-	Skor F1	Durasi
1	46,3768	9,714021311
2	48,9796	10,03607196
3	55,5556	9,903285294
4	52,381	9,677281717
5	54,5455	9,745093192
6	57,1429	9,796491472
7	54,1667	9,735787343
8	60	9,789084804
9	52,6316	9,832504111
10	50	9,718260275

Hasil percobaan dengan menggunakan jenis pemelajaran *hybrid* dapat dilihat pada Tabel 4. Pada Tabel 4 dapat terlihat bahwa nilai skor F1 tertinggi diperoleh pada sesi latihan ke-7 dan ke-10 dengan nilai 54,545. Sedangkan nilai skor F1 terendah dapat diperoleh pada sesi latihan ke-4 dan ke-6 dengan nilai 41,379. Untuk durasi tercepat diperoleh pada sesi latihan ke-4 dengan waktu 14,707 detik. Sedangkan untuk durasi terlama diperoleh pada sesi latihan ke-1 dengan waktu 15,189 detik.

Tabel 4. Hasil pengujian jenis pemelajaran *hybrid*

Sesi ke-	Skor F1	Durasi
1	41,5094	15,18901654
2	50	15,09044278
3	51,6129	14,96605027
4	41,3793	14,70723351
5	46,6667	15,03843949
6	41,3793	14,74860145
7	54,5455	14,71124027
8	42,8571	14,81217186
9	53,3333	14,80007555
10	54,5455	14,93722671

D. Analisa

Dari 40 sesi latihan yang dilakukan menggunakan empat jenis pemelajaran, rangkuman performa masing-masing jenis pemelajaran dapat terlihat pada Tabel 5 yang berisi tentang rangkuman hasil percobaan. Pada jenis pemelajaran *sif* dapat terlihat bahwa rata-rata skor F1 sebesar 37,759 dengan nilai skor F1 tertinggi sebesar 45,7143 dan nilai skor F1 terendah sebesar 27,0677. Sedangkan untuk jenis pemelajaran *rnn* dapat terlihat bahwa nilai rata-rata skor F1 adalah 61,546 Nilai skor F1 tertinggi untuk jenis pemelajaran tersebut adalah 66,667 dan nilai skor F1 terendah untuk jenis pemelajaran tersebut adalah 54,167. Untuk jenis pemelajaran *att* rata-rata nilai skor F1 adalah 53,177 dengan nilai skor F1 tertinggi sebesar 60 dan nilai skor F1 terendah adalah 46,376. Sedangkan pada jenis pemelajaran *hybrid* nilai rata-rata skor F1 adalah 47,783 dengan nilai skor F1 tertinggi adalah 54,545 dan nilai skor F1 terendah adalah 41,379.

Tabel 5. Rangkuman Hasil Pengujian

Pengukuran	Jenis Pemelajaran	Skor F1
Rata-rata	*Smooth Inverse Frequency* (SIF)	37,75961
	Bidirectional RNN	61,54613
	Decomposable Attention Model	53,17797
	Hybrid Model	47,7829
Nilai Maksimum	*Smooth Inverse Frequency* (SIF)	45,7143
	Bidirectional RNN	66,6667
	Decomposable Attention Model	60
	Hybrid Model	54,5455
Nilai Minimum	*Smooth Inverse Frequency* (SIF)	27,0677
	Bidirectional RNN	54,1667
	Decomposable Attention Model	46,3768
	Hybrid Model	41,3793

IV. KESIMPULAN

Terdapat empat jenis pemelajaran yang digunakan pada penelitian ini seperti yang sudah dipaparkan pada bab **Error! Reference source not found.**, *sif*, *rnn*, *att*, dan *hybrid*. Pada masing-masing jenis pemelajaran, dilakukan pelatihan sebanyak sepuluh sesi dengan iterasi 10 kali latihan untuk setiap sesinya untuk mendapatkan nilai skor F1 tertinggi. Dari hasil percobaan dan pembahasan yang telah dilakukan dapat disimpulkan bahwa jenis pemelajaran yang tepat untuk studi kasus *Entity Matching* produk kamera dengan satu atribut pembanding adalah *rnn*. Hal ini dapat dilihat dari nilai skor F1 tertinggi pada jenis pemelajaran *rnn* dengan nilai 66,667. Sedangkan untuk jenis pemelajaran yang kurang tepat untuk studi kasus dalam penelitian ini adalah model *sif* karena nilai skor F1 tertinggi yang mampu capai *sif* hanya sebesar 45,714 yang merupakan paling rendah dari nilai terbaik yang mampu dicapai semua jenis. Dari hasil tersebut juga dapat disimpulkan bahwa jenis pemelajaran yang dipilih dapat mempengaruhi akurasi yang yang dilakukan oleh mesin.

Hasil pengujian juga menunjukkan bahwa skor F1 terendah pada jenis pemelajaran *sif*, *rnn*, dan *att* terdapat pada sesi ke-1. Dapat disimpulkan bahwa pemelajaran dengan *sif*, *rnn*, dan *att* sesi awal tidak cukup optimal untuk memprediksi kecocokan pada penelitan ini. diperlukan sesi

pemelajaran berulang untuk mencapai akurasi yang optimal. Sedangkan *hybrid* tidak menunjukkan hasil yang sama namun memiliki pola yang mirip di mana semakin lama dilakukan pemelajaran maka akurasi akan menjadi semakin baik.

DAFTAR PUSTAKA

[1] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2002.

[2] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute dan V. Raghavendra, "Deep learning for entity matching: A design space exploration," dalam *Proceedings of the 2018 International Conference on Management of Data*, 2018.

[3] L. F. Carvalho, A. H. Laender and W. Meira Jr, "Entity matching: A case study in the medical domain," in *Alberto Mendelzon International Workshop on Foundations of Data Management*, Lima, 2015.

[4] P. Christen, Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection, Springer: Data-centric systems and applications, 2012.

[5] S. Mudgal, "deepmatcher/README.rst at master · anhaidgroup/deepmatcher," 9 July 2018. [Online]. Available: https://github.com/anhaidgroup/deepmatcher/blob/master/README.rst. [Diakses 18 November 2020].

[6] A. Doan, "AnHai's Group," 30 September 2017. [Online]. Available: https://sites.google.com/site/anhaidgroup/home. [Diakses 19 November 2020].

[7] A. Doan, "Magellan - AnHai's Group," 11 Febuary 2020. [Online]. Available: https://sites.google.com/site/anhaidgroup/projects/magellan. [Diakses 19 November 2020].

[8] R. a. M. G. Garreta, Learning scikit-learn: machine learning in python, Packt Publishing Ltd, 2013.