TESIS - IS185401

ONLINE INCREMENTAL LEARNING BERBASIS *CROWDSOURCING* UNTUK EKSTRAKSI RELASI ONTOLOGI BAHASA INDONESIA

EUNIKE ANDRIANI KARDINATA
NRP. 05211850010016

Dosen Pembimbing
Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

2020

(Halaman sengaja dikosongkan)

THESIS - IS185401

ONLINE INCREMENTAL LEARNING BASED ON CROWDSOURCING FOR INDONESIAN ONTOLOGY RELATION EXTRACTION

EUNIKE ANDRIANI KARDINATA
NRP. 05211850010016

Supervisor
Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

Department of Information Systems
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
2020

(Halaman sengaja dikosongkan)

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom)
di
Institut Teknologi Sepuluh Nopember
Oleh:

EUNIKE ANDRIANI KARDINATA
NRP: 05211850010016

Tanggal Ujian: 9 Januari 2020
Periode Wisuda: Maret 2020

Disetujui oleh:
Pembimbing:

1. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D. .....
   NIP: 198201202005012001

Penguji:

1. Dr. Apol Pribadi Subriadi, S.T., M.T. .....
   NIP: 197002252009121001

2. Ahmad Mukhlason, S.Kom., M.Sc., Ph.D. .....
   NIP: 198203022009121009

Kepala Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas

Dr. Mudjahidin, S.T., M.T.
NIP: 197010102003121001

(Halaman sengaja dikosongkan)

ONLINE INCREMENTAL LEARNING* BERBASIS *CROWDSOURCING
UNTUK EKSTRAKSI RELASI ONTOLOGI BAHASA INDONESIA

Nama Mahasiswa	: Eunike Andriani Kardinata
NRP	: 05211850010016
Pembimbing	: Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

ABSTRAK

Ontologi adalah salah satu bentuk representasi pengetahuan yang terstruktur. Ontologi banyak digunakan dan dikembangkan dalam proses *information retrieval* karena kemampuannya merepresentasikan pengetahuan ke dalam bentuk yang bisa dipahami oleh mesin dan manusia. Dengan meningkatnya skala dan kerumitan ontologi, terdapat tantangan yang lebih besar dalam identifikasi *extra-logical error*. Metode pembangunan ontologi kebanyakan menggunakan pembelajaran mesin, yang mana terdapat risiko adanya *extra-logical error* yang terlewatkan. Untuk menanganinya, digunakan *crowdsourcing*, yaitu membagi sebuah pekerjaan besar ke beberapa pekerjaan kecil dan mempekerjakan massa untuk menyelesaikannya secara *online*. Untuk memanfaatkan *crowdsourcing*, pemrosesan data yang biasanya dilakukan secara *offline* dan *batch* diubah menjadi *online* dan *incremental*. *Online incremental learning* langsung menyusun model secara iteratif setelah sebuah perubahan dilakukan, dengan memastikan bahwa pengetahuan yang sudah didapatkan sebelumnya tetap dipertahankan. Pada penelitian ini, dibangun sebuah medium interaktif untuk menyajikan relasi awal antar pasangan konsep. Partisipan *crowdsourcing* diminta untuk memvalidasi relasi tersebut secara berulang sampai tercapai nilai akurasi yang ditentukan. Dari penelitian ini, ditemukan bahwa proses *crowdsourcing* mampu memperbaiki model yang digunakan pada proses ekstraksi relasi, yaitu dari *F1-Score* 87.2% menjadi 89.8%. Perbaikan dengan menggunakan *crowdsourcing* ini mencapai hasil akhir yang sama dengan perbaikan oleh *expert*. Dengan demikian, *crowdsourcing* dinilai mampu mengoreksi *extra-logical error* dengan tepat selayaknya *expert*. Selain itu, ditemukan juga bahwa *offline incremental learning* dengan menggunakan Random Forest menghasilkan akurasi model yang lebih tinggi dibandingkan dengan *online incremental learning* dengan menggunakan Mondrian Forest. Akurasi model Random Forest memiliki akurasi akhir sebesar 90.6% sementara akurasi model Mondrian Forest sebesar 89.7%. Dari hasil ini disimpulkan bahwa *online incremental learning* tidak mampu memberikan hasil yang lebih baik daripada *offline incremental learning* untuk perbaikan proses ekstraksi relasi *meronymy*.

Kata kunci: *crowdsourcing*; *extra-logical error*; *online incremental*; relasi

(Halaman sengaja dikosongkan)

ONLINE INCREMENTAL LEARNING BASED ON CROWDSOURCING FOR INDONESIAN ONTOLOGY RELATION EXTRACTION

By : Eunike Andriani Kardinata
Student Identity Number : 05211850010016
Supervisor : Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.

ABSTRACT

Ontology is a form of structured knowledge representation. Ontology is largely used and developed in the process of information retrieval because of its ability to represent knowledge in a form that is both understandable by machine and human. With the increase of ontology scale and complexity is a greater challenge in extra-logical error identification. Most ontological engineering methods depend on machine learning where there is a risk of overlooking extra-logical error. One way to handle this is by crowdsourcing, that is dividing a large task into several smaller subtasks and employ the mass to complete them online. To utilise crowdsourcing, we change the offline and batch data processing into the online and incremental one. Online incremental learning iteratively constructs a model right after a change is made, ensuring that previously acquired knowledge is maintained. In this research, we develop an interactive medium to display the initial relation between concepts. The crowdsourcing participants will be asked to repeatedly validate those relations until the desired accuracy value is reached. From this research, we find that crowdsourcing is able to improve the model used in relation extraction process, from the F1-Score of 87.2% to 89.8%. This improvement using crowdsourcing reaches the same score as that using expert. Therefore, crowdsourcing is considered as able to correct extra-logical error accurately, just like expert. Besides, we also discover that offline incremental learning using Random Forest produces a model with higher accuracy than online incremental learning using Mondrian Forest. Random Forest model has the final accuracy value of 90.6% while Mondrian Forest model has 89.7%. From this result, we conclude that online incremental learning is unable to produce a better result than offline incremental learning in improving meronymy relation extraction process.

Keywords: crowdsourcing; extra-logical error; online incremental; relation

(Halaman sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa, karena penyertaanNya penulis menyelesaikan penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia dengan baik. Hasil penelitian ini diharapkan dapat menambah wawasan pembaca, serta memajukan pengembangan ontologi Bahasa Indonesia.

Selama proses penelitian ini, penulis telah mendapatkan banyak dukungan. Pada kesempatan ini, penulis ingin menyampaikan terima kasih kepada:

1. Ibu Nur Aini Rakhmawati, Ph.D., selaku dosen pembimbing penulis, untuk dedikasi, perhatian, dan bimbingan beliau.
2. Bapak Dr. Apol Pribadi Subriadi dan Bapak Ahmad Mukhlason, Ph.D., selaku dosen penguji yang telah memastikan kelayakan penelitian penulis.
3. Seluruh dosen yang telah mendidik dan memperlengkapi penulis dengan pengetahuan yang dibutuhkan untuk menghasilkan karya ilmiah ini.
4. Bapak Dr. Hartarto Junaedi, Bapak Joan Santoso, M.Kom., serta segenap manajemen dan dosen Institut Sains danTeknologi Terpadu Surabaya, atas segenap dukungan yang diberikan sampai penulis menyelesaikan studinya.
5. Orang tua dan segenap keluarga yang setia mendukung penulis.
6. Rekan Jurusan S-2 Sistem Informasi 2018 Gasal atas bantuan untuk penulis.
7. Saudara Ananta Tio Putra, S.Kom. atas dukungannya kepada penulis.
8. Semua pihak yang belum dapat disebutkan, yang sudah mendukung penulis.

Akhir kata, penulis menyadari bahwa setiap masukan pembaca akan sangat berguna untuk membantu penulis menghasilkan karya yang lebih baik. Demikian, diharapkan akan terdapat penelitian dengan topik serupa yang mengupas lebih dalam mengenai pengembangan ontologi Bahasa Indonesia di masa depan.

Surabaya, Januari 2020

Eunike Andriani Kardinata

(Halaman sengaja dikosongkan)

DAFTAR ISI

(Halaman sengaja dikosongkan)

DAFTAR GAMBAR

(Halaman sengaja dikosongkan)

DAFTAR TABEL

(Halaman sengaja dikosongkan)

BAB 1
PENDAHULUAN

Pada bab ini akan dijelaskan mengenai Latar Belakang, Rumusan Masalah, Tujuan, Manfaat, dan Kontribusi Penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia. Kemudian juga akan disertakan Batasan Penelitian dan Sistematika Penulisan laporan ini.

1.1 Latar Belakang

Pertumbuhan volume data pada masa kini sangat pesat. Seorang analis data dituntut untuk dapat mengolah data menjadi informasi yang berguna. Kemudian berdasarkan informasi yang ada, disimpulkan sebuah pengetahuan atau pemahaman yang memampukan seseorang membuat keputusan. Dengan perkembangan zaman, semakin banyak pula bentuk dan alat representasi infomasi maupun pengetahuan.

Salah satu perkembangan pada masa kini adalah *semantic web*. *Semantic web* adalah pengembangan dari *World Wide Web* yang bertujuan untuk menyusun data dalam web ke dalam struktur yang dapat dibaca langsung oleh mesin. Format standar dalam *semantic web* disusun oleh *World Wide Web Consortium* dengan memanfaatkan teknologi seperti *Resource Description Framework* dan *Web Ontology Language*. Ontologi merupakan bentuk representasi pengetahuan yang terstruktur dengan baik. Ontologi adalah sekumpulan konsep atau entitas dalam sebuah domain yang saling terhubung melalui relasi antar konsep. Secara sederhana, ontologi menggambarkan bagaimana objek-objek (entitas) dalam suatu ruang lingkup (domain) saling berhubungan dan jenis hubungan (relasi) tersebut.

Ontologi banyak digunakan dan dikembangkan karena kemampuannya mendefinisikan pemahaman umum pada suatu domain, sehingga memudahkan penggunaan serta pengembangan pengetahuan tersebut. Dengan menggunakan ontologi, dapat diketahui definisi standar suatu objek dan kaitannya dengan objek lain. Selain itu, ontologi juga mampu merepresentasikan pengetahuan manusia ke dalam format yang dapat dipahami oleh mesin untuk diproses lebih lanjut. Hal ini

dikarenakan ontologi memiliki format yang sudah distandarisasi, seperti layaknya format dokumen dalam bentuk digital.

Dalam pengembangan ontologi, perlu diketahui domain yang akan dicakup ontologi tersebut, contohnya domain makanan dan minuman, domain kesehatan dan kedokteran (*medical*), dan sebagainya. Selanjutnya, ditentukan juga tujuan penggunaan ontologi tersebut, misalnya untuk membantu proses akuisisi informasi, untuk digunakan pada aplikasi yang memberikan rekomendasi kombinasi makanan dan minuman pada pengguna, atau untuk membantu pengambilan keputusan tindakan medis. Dari tujuan ini, pengembang ontologi dapat mendefinisikan pertanyaan-pertanyaan yang harus bisa dijawab oleh ontologi tersebut secara lebih spesifik. Pertimbangan lain yang tidak kalah pentingnya ialah siapa yang akan menggunakan dan memelihara ontologi yang dikembangkan. Melalui keseluruhan proses ini, pengembang ontologi dapat memutuskan apakah dibutuhkan *interface* tambahan untuk memudahkan pengguna dan pemelihara ontologi.

Salah satu bentuk penggunaan ontologi adalah pada area *information retrieval*. Beberapa contohnya, antara lain *information retrieval* pada bidang akademik (Jamgade & Karale, 2015), bidang *e-commerce* (Tao & Zhao, 2012), dan bidang sains (Xinhua, Xutang, & Zhongkai, 2012). Tidak menutup kemungkinan juga untuk dilakukan penelitian pada domain terbuka, sehingga hasilnya dapat diaplikasikan secara umum pada berbagai bidang.

Semakin banyaknya penggunaan ontologi dan penambahan data baru ke dalam ontologi tersebut, maka skala dan tingkat kompleksitas ontologi pun akan meningkat. Dengan meningkatnya skala dan kerumitan ontologi, terdapat tantangan yang lebih besar dalam identifikasi *extra-logical error* (Mortensen, Crowdsourcing Ontology Verification, 2013). *Extra-logical error* merupakan jenis error yang hanya dapat dideteksi oleh manusia karena umumnya berkaitan dengan konteks kalimat bersangkutan.

Dalam bidang linguistik, Avram Noam Chomsky (Chomsky, 1957) mengusulkan bahwa sebuah kalimat dapat memiliki struktur dalam (*deep structure*) dan struktur permukaan (*surface structure*). Sebagai contoh, kalimat “Ana menyukai Budi” dan “Budi disukai Ana” memiliki struktur permukaan yang

berbeda, namun keduanya diturunkan dari struktur dalam yang mirip, sehingga makna kedua kalimat tersebut dapat dikatakan sama.

Sebuah kalimat, dapat memiliki representasi bentuk *logical* atau biasa disebut *logical form*. Bentuk *logical* adalah representasi pemahaman manusia yang murni didapatkan dari struktur permukaan sebuah kalimat. Dengan kata lain, di dalam Bahasa Indonesia, bentuk ini mirip dengan penggambaran makna tersurat sebuah kalimat. Berdasarkan pemahaman ini, *logical error* dapat didefinisikan sebagai sebuah *error* yang terjadi pada struktur permukaan sebuah kalimat atau karena kesalahan representasi pada struktur tersebut.

Error yang bukan *logical* dapat disebut sebagai *extra-logical* atau *non-logical*. Menurut perbedaan definisinya, *extra-logical* berarti melampaui batas *logical*, sedangkan *non-logical* berarti tidak berhubungan atau berdasarkan pada konsep *logical*. Pada penelitian ini, jenis *error* yang diangkat lebih tepat disebut sebagai *extra-logical error* karena masih berkaitan dengan bentuk *logical*; namun untuk mendeteksi makna sesungguhnya dari sebuah kalimat, peneliti harus melihat di luar batasan bentuk *logical* tersebut.

Berikut Gambar 1 menunjukkan contoh kasus yang dapat menimbulkan *extra-logical error*.

```
Kalimat 1: Pria itu merupakan tangan kanan direktur perusahaan
Kalimat 2: Tangan kanan direktur perusahaan dibalut perban
```

Gambar 1 Contoh *Extra-Logical Error*

Pada kalimat 1 dan kalimat 2, terdapat konteks yang berbeda untuk relasi antara konsep “tangan kanan” dan konsep “direktur perusahaan”. “Tangan kanan” pada kalimat 1 dapat diartikan sebagai “orang kepercayaan”, sedangkan pada kalimat 2, “tangan kanan” yang dimaksud adalah “anggota tubuh”. Jika konteks kalimat tidak dipertimbangkan, maka kedua kalimat tersebut akan menghasilkan relasi yang sama, padahal makna yang dimaksudkan sangat bebeda.

Saat ini, metode pembangunan ontologi cenderung bergantung pada metode pembelajaran mesin (*machine leaning*) (Losing, Hammera, & Wersing, 2018).

Metode pembelajaran mesin membutuhkan sumber daya dan waktu yang lebih sedikit dibandingkan dengan proses manual, namun terdapat risiko adanya *extra-logical error* yang terlewatkan. Error yang terlewatkan ini mempengaruhi kualitas ontologi yang dihasilkan, khususnya akurasi. Oleh sebab itu, diperlukan penelitian untuk menangani *extra-logical error* sehingga akurasi ontologi dapat ditingkatkan.

Evaluasi oleh manusia dapat dilakukan oleh tenaga ahli (*expert*) atau masyarakat umum. Kelebihan evaluasi oleh *expert* adalah kualitas evaluasi tersebut akan cenderung lebih tinggi karena *expert* sudah memiliki pengetahuan yang memadai dalam bidang yang berkaitan. Namun kelemahannya, biaya untuk mempekerjakan seorang *expert* relatif mahal, umumnya sebanding dengan kualitas evaluasi yang dihasilkan. Alternatif lain adalah dengan memanfaatkan kemampuan masyarakat umum atau *crowdsourcing*. Biaya untuk *crowdsourcing* biasanya lebih rendah daripada untuk *expert*. Selain itu, untuk menjamin validitas evaluasi massa, telah terdapat berbagai macam metode yang dapat digunakan.

Beberapa penelitian terkait telah dilakukan sebelumnya. Penelitian tentang *crowdsourcing* untuk menanggulangi *extra-logical error* pernah dilakukan oleh (Mortensen, et al., 2016) dan (Yang & Callan, Human-Guided Ontology Learning, 2008). Berkaitan dengan *crowdsourcing*, diperlukan pembelajaran *incremental* seperti yang telah diteliti oleh (Losing, Hammera, & Wersing, 2018) dan (Meng, Chen, Tong, & Zhang, 2017). Selain itu, *crowdsourcing* juga membutuhkan sebuah medium interaktif untuk penggunanya. Terdapat beberapa contoh pemanfaatan editor ontologi yang diberi nama OntoCop, seperti pada penelitian mengenai komentar publik untuk regulasi pemerintah oleh (Yang & Callan, OntoCop: Constructing Ontologies for Public Comments, 2009) dan pada penelitian mengenai ontologi untuk *information science* oleh (Sawsaa & Lu, 2010).

Saat ini belum ditemukan penelitian yang membahas secara rinci proses ekstraksi relasi pada ontologi dengan menggunakan *online incremental learning* dan *crowdsourcing* pada tahap validasi atau praproses data. Selain itu, masih terdapat banyak area dalam pengembangan ontologi Bahasa Indonesia yang belum dijelajahi. Dengan demikian diharapkan melalui penelitian ini, dapat dibangun sebuah ontologi dalam Bahasa Indonesia yang akurat.

1.2 Rumusan Masalah

Pada ontologi Bahasa Indonesia, ontologi yang dibangun dengan metode pembelajaran mesin tidak luput dari kemungkinan adanya *extra-logical error*. Hal ini dikarenakan *extra-logical error* merupakan error yang bergantung pada konteks berkaitan, sehingga diperlukan evaluasi manusia untuk deteksi yang lebih akurat.

Penggunaan *expert* dalam deteksi *extra-logical error* memang sangat membantu, namun juga sangat mahal. Solusi alternatif yang dapat digunakan adalah dengan *crowdsourcing*. Partisipan *crowdsourcing* mungkin tidak semuanya adalah ahli, namun evaluasi kolektif semua partisipan dapat dimanfaatkan untuk deteksi *extra-logical error*. Secara spesifik, *crowdsourcing* akan digunakan pada tahap validasi relasi yang dihasilkan dari *online incremental learning*. Proses ini dapat dianggap juga sebagai praproses sebelum data relasi digunakan kembali untuk *online incremental learning*.

Penggunaan *online incremental* learning adalah untuk memfasilitasi *crowdsourcing* dalam pembangunan ontologi. Metode ini dipilih karena data *crowdsourcing* tidak langsung tersedia secara bersamaan, melainkan secara bertahap. Dengan demikian, rumusan masalah yang diangkat dapat dirangkumkan dalam poin-poin berikut:

- Pengukuran seberapa besar pengaruh *crowdsourcing* dalam mengoreksi *extra-logical error* pada ontologi Bahasa Indonesia.
- Pengukuran seberapa besar pengaruh *online incremental learning* untuk memperbaiki proses ekstraksi relasi dalam ontologi Bahasa Indonesia.

1.3 Tujuan Penelitian

Berkenaan dengan masalah *extra-logical error*, untuk mengintegrasikan *crowdsourcing* ke dalam proses pembangunan ontologi, diperlukan sebuah medium interakif yang dapat memvisualisasikan data relasi untuk peserta *crowdsourcing*. Kemudian dengan mengimplementasikan metode *online incremental learning*, akan ditambahkan koreksi yang sudah didapat dari evaluasi peserta *crowdsourcing* secara bertahap.

Penelitian ini bertujuan untuk menganalisis seberapa besar peningkatan yang ditimbulkan oleh proses *crowdsourcing* pada akurasi koreksi *extra-logical error*. Analisis akan dilakukan dengan cara mengukur akurasi relasi yang dihasilkan setelah koreksi dilakukan oleh partisipan *crowdsourcing*. Untuk setiap *learning*, jika relasi yang baru lebih akurat, maka partisipan tidak akan melakukan koreksi lagi, demikian sebaliknya.

Kemudian akan dianalisis seberapa besar peningkatan yang ditimbulkan oleh *online incremental learning* dalam memperbaiki proses ekstraksi relasi. Perbaikan yang dimaksudkan adalah akurasi yang dihasilkan dibandingkan dengan metode lain, seperti *offline incremental learning* atau metode manual.

1.4 Manfaat Penelitian

Dengan penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia, pembangunan ontologi dapat dilakukan dengan sumber daya yang lebih sedikit, namun kualitas ontologi yang dihasilkan dapat tetap terjaga. Hal ini mungkin dicapai karena *crowdsourcing* membutuhkan biaya yang relatif lebih sedikit daripada penggunaan tenaga ahli, namun evaluasinya dapat disetarakan untuk mendukung hasil evaluasi tenaga ahli (Mortensen, et al., 2016).

Dataset untuk ontologi yang dihasilkan dari penelitian ini—dalam bentuk *graph*—diharapkan memiliki kualitas yang setara dengan pelabelan manual, namun usaha yang dikeluarkan sebenarnya hanya koreksi. Koreksi membutuhkan waktu dan usaha yang lebih sedikit daripada pelabelan awal, karena secara umum lebih mudah menemukan kesalahan pada sebuah jawaban daripada menentukan jawaban.

Pada penelitian ini dikembangkan ontologi Bahasa Indonesia pada domain terbuka, yang saat ini masih belum banyak ditemukan seperti ontologi bahasa lain, contohnya Bahasa Inggris. Hasil dari penelitian ini bermanfaat untuk peneliti yang hendak mengembangkan ontologi dalam Bahasa Indonesia, mengingat Bahasa Indonesia memiliki tata bahasa yang berbeda dari bahasa lainnya. Peneliti juga dapat menggunakan ontologi yang dikembangkan untuk meningkatkan kemudahan

information reterieval pada konteks umum, misalnya dari artikel berita atau hasil pencarian pada website.

Kemudian berdasarkan proses yang dilakukan pada penelitian ini, dapat dikembangkan ontologi dalam bahasa lain. Selama dataset yang digunakan memenuhi ketentuan, baik untuk praproses maupun proses inti, maka metode dalam penelitian ini dapat diaplikasikan ke dalam bahasa lain. Data utama yang wajib ada, yaitu kalimat lengkap dan indentifikasi pasangan entitas di dalam kalimat tersebut.

1.5 Kontribusi Penelitian

Kontribusi penelitian yang dilakukan dibagi ke dalam dua jenis, yaitu kontribusi teoritis dan kontribusi praktis. Kontribusi teoritis berkaitan dengan bidang penelitian, sementara kontribusi praktis berkaitan dengan aplikasi riilnya.

1.5.1 Kontribusi Teoritis

Keterbaruan yang didapatkan dari penelitian ini ialah analisis pengaruh *crowdsourcing* pada proses ektraksi relasi ontologi Bahasa Indonesia. Data hasil *crowdsourcing* akan divalidasi dengan menggunakan *inter-annotator agreement*. Setelah itu, data yang valid akan digunakan untuk memperbaiki model yang digunakan dalam proses ekstraksi relasi. Dari percobaan ini, akan ditemukan apakah *crowdsoucing* benar-benar membantu dalam proses ekstraksi relasi. Jika memang *crowdsourcing* bermanfaat, dapat diketahui pula bagaimana cara mengaplikasikannya dan konfigurasi seperti apa yang harus digunakan.

Selain hal tersebut, akan dianalisis juga pengaruh penggunaan *online incremental learning* pada proses ekstraksi relasi. Hasil analisis ini dapat digunakan sebagai acuan, yaitu pada kondisi apa sebaiknya *online incremental learning* digunakan dan bagaimana implementasinya.

1.5.2 Kontribusi Praktis

Berkaitan dengan aplikasi di dunia nyata, penelitian ini akan menghasilkan sebuah medium interaktif yang dapat digunakan untuk pengembangan ontologi dalam bahasa atau domain lain dengan menggunakan *crowdsourcing*. Pengembang

ontologi dapat dengan lebih mudah memverifikasi ketepatan ontologi yang sudah ada dengan menggunakan alat tersebut. Selain itu, akan dihasilkan juga dataset untuk ontologi Bahasa Indonesia pada domain terbuka yang dapat divisualisasikan dalam bentuk *graph*.

1.6 Batasan Penelitian

Terdapat beberapa batasan pada penelitian ini untuk menjaga fokus dari penelitian dan untuk mendefinisikan konteks penelitian. Batasan-batasan tersebut adalah sebagai berikut:

- Penelitian akan difokuskan pada ekstraksi relasi. Proses ekstraksi konsep sudah dilakukan pada penelitian terdahulu dan tidak akan dievaluasi.
- Relasi yang diekstrak adalah relasi *meronymy* atau *part-whole* menurut standar yang telah didefinisikan pada penelitian terdahulu oleh (Phi & Matsumoto, Integrating Word Embedding Offsets into the Espresso System for Part-Whole Relation, 2016).
- Dataset yang digunakan disadur dari penelitian mengenai ekstraksi relasi menggunakan metode lain oleh (Phi, Santoso, Shimbo, & Matsumoto, 2018) yang dipublikasikan oleh Association for Computational Linguistics.
- Jika ditemukan data yang lebih baik, yaitu data dalam Bahasa Indonesia atau data yang belum pernah digunakan, sehingga meningkatkan kontribusi penelitian, maka data tersebut akan digunakan sebagai pengganti
- Penelitian tidak berfokus pada perbandingan algoritma *online incremental learning*, sehingga berdasarkan penelitian terdahulu, peneliti akan memilih satu algoritma yang terbaik.
- Evaluasi hasil penelitian dilakukan pada implementasi algoritma *online incremental* dan hasil *crowdsourcing*, bukan untuk alat yang dibangun.
- Ontologi dibuat untuk domain terbuka dalam Bahasa Indonesia. Dengan demikian, metode yang digunakan secara spesifik akan difokuskan untuk bekerja optimal pada Bahasa Indonesia. Namun tidak menutup kemungkinan jika performa model memang baik, maka model dapat diaplikasikan ke dalam bahasa lain selama ketentuan data terpenuhi.

- Dataset yang digunakan pada pembentukan model untuk *online incremental learning* sekaligus menjadi acuan subtipe relasi *meronymy*. Jika dataset acuan diganti, maka model dan hasil *learning* pun akan berubah.

1.7 Sistematika Penulisan Laporan

Pada laporan ini terdapat beberapa bab yang berisi penjabaran penelitian yang diajukan. Pada halaman berikut diberikan sistematika pembahasan laporan ini.

Bab 1: Pendahuluan

Pada bab ini akan dijelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat, dan kontribusi penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia. Kemudian juga akan disertakan batasan penelitian dan sistematika penulisan laporan ini.

Bab 2: Kajian Pustaka

Penelitian yang akan dilakukan perlu didukung dengan referensi yang dapat dipercaya dan ilmu yang memadai. Bab ini akan membahas pustaka yang dikaji untuk penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia, antara lain: *crowdsourcing*, ontologi, pembelajaran ontologi, ekstraksi relasi, *online incremental learning*, editor untuk evaluasi ontologi, dan penelitian terkait.

Bab 3: Metodologi Penelitian

Berdasarkan sumber dan penelitian terdahulu yang telah dikaji, maka dibuat sebuah rancangan penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia. Rancangan yang dibahas dalam bab ini meliputi metode penelitian dan pengembangan, usulan arsitektur sistem, dan sumber data.

Bab 4: Hasil dan Pembahasan

Pembahasan pada bab ini disusun berdasarkan metodologi penelitian, yaitu penerapan dan hasil percobaan untuk kerangka kerja *crowdsourcing*. Secara lebih detail, hal-hal yang dibahas meliputi: pengumpulan data, praproses data, ekstraksi fitur, penerapan *offline incremental learning*, penerapan *online incremental learning*, pengembangan alat interaktif untuk *crowdsourcing*, dan evaluasi kerangka kerja yang diusulkan.

Bab 5: Kesimpulan dan Saran

Pada bab ini akan dijabarkan beberapa kesimpulan dari penelitan yang dilakukan. Kemudian diberikan juga beberapa saran terkait dengan kesimpulan, guna memperbaiki penelitian-penelitian pada bidang yang sama di kemudian hari.

BAB 2
KAJIAN PUSTAKA

Penelitian yang akan dilakukan perlu didukung dengan referensi yang dapat dipercaya dan ilmu yang memadai. Bab ini akan membahas pustaka yang dikaji untuk penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia, antara lain: *crowdsourcing*, ontologi, pembelajaran ontologi, ekstraksi relasi, *online incremental learning*, editor untuk evaluasi ontologi, dan penelitian terkait.

2.1 *Crowdsourcing*

Crowdsourcing didefinisikan sebagai proses membagi sebuah pekerjaan yang besar ke dalam pekerjaan-pekerjaan lain yang lebih kecil dan mempekerjakan banyak orang lainnya untuk menyelesaikan pekerjaan tersebut (Mortensen, Crowdsourcing Ontology Verification, 2013). *Crowdsourcing* semakin diminati karena peneliti menyadari bahwa mesin tidak dapat sepenuhnya mereplikasi pengetahuan manusia, sementara penggunaan tenaga ahli pasti akan membutuhkan biaya dan usaha yang besar. Demikian, dengan pertumbuhan riset saat ini dan keterbatasan tenaga ahli, *crowdsourcing* dapat menjadi sebuah solusi.

Beberapa mekanisme *crowdsourcing* telah diusulkan selama ini, bahkan terdapat penelitian yang melakukan *crowdsourcing* untuk mendesain mekanisme *crowdsourcing* (Sakurai, Matsuda, Shinoda, & Oyama, 2017). Dua pengaturan yang dapat dipertimbangkan dalam desain mekasnisme *crowdsourcing* adalah *skip-based* dan *confidence-based* (Shah & Zhou, 2016). *Skip-based* berarti partisipan dapat memilih untuk menjawab pertanyaan atau melewati pertanyaan tersebut. Sedangkan *confidence-based* berarti partisipan dapat menentukan seberapa yakin mereka terdapat jawaban yang diberikan.

Imbalan yang diberikan pada partisipan *crowdsourcing* dapat diatur dalam berbagai skema. Skema yang paling sederhana adalah berdasarkan jumlah kontribusi yang diberikan, dalam hal ini berarti jumlah evaluasi yang benar. Lebih lanjut, dapat diberlakukan skema *double or nothing* (Shah & Zhou, 2016). Pada

skema ini, untuk setiap kelipatan jumlah evaluasi tertentu yang benar, maka imbalan akan dilipatgandakan. Namun jika ada satu saja evaluasi yang salah, maka nilai imbalan akan dikembalikan ke nilai awal yang bisa jadi adalah nol.

Setiap pengaturan dan skema pemberian imbalan yang digunakan pada proses *crowdsourcing* harus ditentukan dengan pertimbangan yang matang. Pertimbangan yang dilakukan meliputi, jenis kegiatan *crowdsourcing*, karakteristik responden, hasil data yang diharapkan, dan sebagainya.

2.2 Ontologi

Terdapat beberapa terminologi yang sering disetarakan dengan ontologi, namun pada kenyataannya, terminologi tersebut memiliki makna yang berbeda dengan ontologi. Beberapa terminologi yang perlu dipahami menurut (Verma, Kaur, & Kaur, 2017) ialah *vocabulary* (kosakata), *controlled vocabulary* (kosakata terkontrol), *glossary* (glosarium), *taxonomy* (taksonomi), *thesaurus* (tesaurus), dan yang terakhir *ontology* (ontologi).

Kosakata, secara sederhana, adalah koleksi dari istilah yang definisinya konsisten pada segala macam konteks. Sedangkan kosakata terkontrol mempunyai jumlah istilah yang terbatas. Glosarium menambahkan deskripsi tidak resmi istilah semantik dalam bahasa alami (*natural language*). Lalu taksonomi adalah kosakata yang terkontrol melalui relasi secara hirarki. Tesaurus memiliki ekuivalensi, hirarki, homograf, dan asosiasi antar istilah. Terakhir, ontologi mempunyai relasi bedasarkan konteks kosakata. Secara umum, ontologi dapat didefinisikan sebagai sebuah konseptualisasi dengan spesifikasi eksplisit yang lengkap maupun tidak, dengan perspektif pribadi maupun bersama, dan dapat berupa bahasa yang resmi maupun bahasa alami—tergantung pada penggunaannya dan area penerapannya.

Ontologi yang berbeda memiliki variasi pada konten, struktur, detail deskripsi, ruang lingkup konseptual, dan spesifikasi bahasa (Gali, Chen, Claypool, & Uceda-Sosa, 2004). Terdapat banyak jenis ontologi, di antaranya adalah:

- Ontologi generik, merepresentasikan pandangan secara umum yang mencakup berbagai bidang atau abstrak.
- Ontologi domain, menggambarkan konseptualisasi untuk domain spesifik.

- Ontologi semi informal, dibuat dalam bahasa alami terstruktur yang dapat dibaca oleh mesin.
- Ontologi formal, dibuat dalam bahasa resmi dan dapat dibaca oleh mesin.
- Ontologi statis, mendefinisikan benda-benda yang eksis.
- Ontologi dinamik, mendefinisikan aspek yang berubah seiring waktu.

Beberapa proses yang berkaitan dengan ontologi telah dibahas dalam literatur oleh (Taye, 2010) dan (Flouris, Manakanatas, Kondylakis, Plexousakis, & Antoniou, 2008), yaitu:

- Transformasi, pembangunan ontologi yang baru karena adanya kebutuhan baru dari ontologi yang sudah ada.
- Penerjemahan, menerjemahkan representasi resmi dalam ontologi dengan memastikan kesamaan semantik.
- Penggabungan, proses pembuatan sebuah ontologi dari dua atau lebih ontologi yang sudah ada pada domain yang sama.
- Integrasi, membuat ontologi baru dari ontologi lain pada domain berbeda.
- Pemetaan, mencari hubungan semantik antara konsep dari ontologi berbeda.
- Penjajaran, untuk menyamakan atau menyetarakan dua atau lebih ontologi.
- Morfisme, untuk mengidentifikasikan konsep yang berhubungan dan aksioma di antara ontologi yang berbeda.
- Evolusi, untuk mengimplementasikan perubahan pada ontologi sumber.
- Pembuatan versi, untuk menyediakan akses ke versi ontologi yang berbeda.

Pengembangan ontologi memiliki potensi yang besar karena pada ontologi, dapat dilakukan banyak proses dan perbaikan. Ontologi di dalam Bahasa Indonesia pun juga dapat terus dikembangkan dengan cara membuat ontologi baru sesuai domain spesifik atau bahkan dengan menerjemahkan ontologi dari bahasa lain lalu mengecek validitas hasilnya dengan konteks dalam Bahasa Indonesia.

2.3 Pembelajaran Ontologi

Rekayasa ontologi (*ontological engineering*) adalah sebuah disiplin yang berkaitan dengan pembangunan dan pemeliharaan ontologi (Verma, Kaur, & Kaur,

2017). Rekayasa ontologi mempelajari proses pembangunan, serta metode dan metodologi yang digunakan untuk membuat sebuah ontologi (Nicola, Missikoff, & Roberto, 2009). Lakel mennyebutkan ada beberapa metode pembangunan ontologi (Lakel & Bendella, 2015), antara lain:

- *Methods for ontology building from scratch* atau metode untuk pembangunan dari ontologi dari nol didasarkan pada ekstraksi pengetahuan secara manual. Kemudian dengan menggunakan teknik *Natural Language Processing* dan *knowledge acquisition*, dihasilkan pengetahuan yang baru.
- *Methods for cooperative construction of ontologies* atau metode konstruksi ontologi secara kooperatif mengadopsi pendekatan kolaboratif untuk mencapai kesepakatan dan penerimaan dari pengguna di dalam komunitas.
- *Methods for reengineering of ontologies* atau metode rekayasa ulang ontologi berfokus pada proses untuk membangun ulang ontologi dan menghubungkan model konseptual ontologi yang sudah diimplementasikan ke ontologi yang sedang diimplementasikan.
- *Methods of learning ontologies* atau metode pembelajaran ontologi bertujuan untuk memperbaiki pembangunan komponen ontologi dengan mengenalkan *plug-in*—baik dalam bentuk teks atau basis pengetahuan—dalam proses pembangunan ontologi.

Proses pembelajaran ontologi terdiri dari dua subproses, yaitu ekstraksi konsep dan formasi relasi (Yang & Callan, A Metric-based Framework for Automatic Taxonomy Induction, 2009). Konsep, atau dengan kata lain entitas, adalah kata benda atau frasa beda. Ekstraksi konsep ialah proses untuk mengidentifikasi konsep dalam sebuah korpus. Kemudian, formasi relasi menemukan relasi antar konsep dan membangun ontologi berdasarkan relasi yang ditemukan tersebut.

2.4 Ekstraksi Relasi

Ekstraksi relasi bertujuan untuk membangun sebuah pengetahuan yang terstruktur dari teks yang tidak terstruktur. Terdapat beberapa kategorisasi tipe relasi yang biasanya didasarkan pada konteks di mana relasi tersebut ditemukan

(Zettlemoyer, 2013). Sebagai contoh, beberapa tipe relasi yang berbeda didefinisikan menurut ACE 2003, Freebase, Geographical, dan sebagainya.

Dalam ekstraksi relasi, sangat penting untuk mempertimbangkan konteks dalam menentukan relasi apa yang dicari. Selain itu, pada konteks yang berbeda, relasi dapat memiliki makna atau definisi yang berbeda. Dengan demikian, evaluasi hasil ekstraksi relasi menjadi aspek yang signifikan dalam pembangunan ontologi.

Terdapat beberapa metode ekstraksi relasi yang didefinisikan pada (Granada, Vieira, Trojahn, & Aussenac-Gilles, 2018). Metode yang dipertimbangkan untuk digunakan dalam penelitian ini hanyalah metode *semi-supervised* atau metode *unsupervised* karena fokus penelitian ini adalah pada integrasi *crowdsourcing* ke dalam proses pembelajaran ontologi. Berikut pada Tabel 1 adalah beberapa metode ekstraksi relasi menurut kategori yang diusulkan oleh Granada et al. (Granada, Vieira, Trojahn, & Aussenac-Gilles, 2018).

Tabel 1 Kategori Metode Ekstraksi Relasi

Kategori	**Metode Menurut Nama Peneliti**
Lexico-Syntactic Patterns	Hearst (Hearst, Automated discovery of wordnet relations, 1998) (Hearst, Automatic acquisition of hyponyms from large text corpora, 1992), Pantel and Pennacchiotti (Pantel & Pennacchiotti, 2006), Ponzetto and Strube (Ponzetto & Strube, 2011), Cederberg and Widdows (Cederberg & Widdows, 2003)
Head-Modifier Detection	Velardi et al. (Velardi, Missikoff, & Basili, 2001), Vossen (Vossen, 2001), Buitelaar et al. (Buitelaar, Olejnik, & Sintek, 2004), Sintek et al. (Sintek, Buitelaar, & Olejnik, 2004), Lopes (Lopes, 2012), Espinosa et al (Espinosa-Anke, Saggion, & Ronzano, 2015).
Distributional Analysis	Sanderson and Croft (Sanderson & Croft, 1999.), Njike-Fotzo and Gallinari (Njike-Fotzo & Gallinari, 20014), Caraballo (Caraballo, 1999), Liu et al. (Liu, Song, Liu, & Wang, 2012), De Knijff et al. (Knijff, Frasincar, & Hogenboom, 2013), Tan et al. (Tan, Gupta, & Genabith, 2015), Pocotales (Pocostales, 2016)

Kategori	Metode Menurut Nama Peneliti
Distributional Inclusion	Weeds et al. (Weeds, Weir, & McCarthy, 2004), Geffet and Dagan (Geffet & Dagan, 2005), Szpektor and Dagan (Szpektor & Dagan, 2008), Clarke (Clarke, 2009), Lenci and Benotto (Lenci & Benotto, 2012), Kotlerman et al. (Kotlerman, Dagan, Szpektor, & Zhitomirsky-geffet, 2010), Santus et al. (Santus, Lenci, Lu, & Walde, 2014)

Pada bagian berikut akan dijabarkan mengenai perbedaan antara setiap kategori metode ekstraksi relasi. Setiap kategori secara umum dibedakan menurut persepsi inferensi relasi antar konsep.

2.4.1 Metode Berdasarkan *Lexico-Syntactic Patterns*

Basis dari penggunaan metode berdasarkan *lexico-syntactic patterns* adalah hubungan semantik dapat diinferensikan sekalipun istilah yang bersangkutan belum pernah dijumpai. Berikut contohnya pada Gambar 2.

```
Beberapa paus, seperti beluga, adalah cetacea di Kutub Utara.
           NP    seperti    NP
```

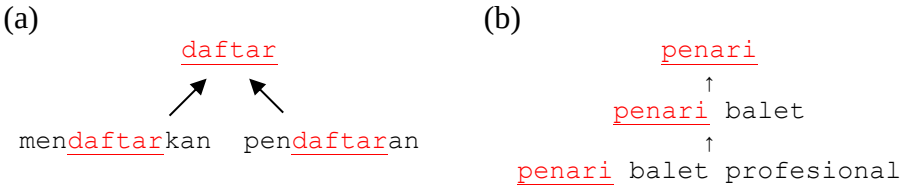
Gambar 2 Pola dalam Kalimat

Kalimat tersebut mempunyai pola frasa benda (NP—*noun phrase*) diikuti dengan kata “seperti” lalu frasa benda (NP) lainnya. Dari kalimat tersebut, dapat diduga bahwa “beluga” adalah sejenis “paus” sekalipun kata “beluga” tidak pernah ditemukan sebelumnya. Demikian, sebuah kalimat yang mengandung kata “seperti” menunjukkan bahwa kalimat tersebut digunakan untuk memberikan contoh.

2.4.2 Metode Berdasarkan Deteksi *Head-Modifier*

Pola pikir utama dalam deteksi *head-modifier* adalah kepala dari sebuah frasa merupakan kata yang paling penting secara tata bahasa, karena kata tersebut akan mendefinisikan keseluruhan frasa (Radford, 1997). Sebuah kata dapat memiliki afiksasi yang perlu dihilangkan agar dapat ditemukan istilah kepalanya, sedangkan sebuah frasa secara umum adalah istilah majemuk. Sebuah frasa

mempunyai kepala yang mendefinisikan kategori semantiknya dan elemen lain yang akan membedakan frasa tersebut dengan frasa lain dalam kategori yang sama. Berikut ini adalah contoh pencarian kepala sebuah kata dan sebuah frasa.

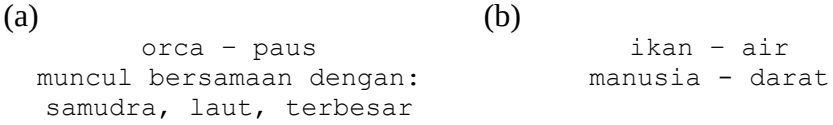


Gambar 3 (a) Kepala dari Kata; (b) Kepala dari Frasa

Pada Gambar 3, dapat diduga bahwa “mendaftarkan” dan “pendaftaran” adalah kegiatan yang berhubungan dengan “daftar”. Kemudian dapat diduga juga bahwa “penari balet profesional” adalah salah satu tipe “penari balet” dan “penari balet” adalah salah satu tipe “penari”. Afiksasi “me-an” dan “pe-an”, serta istilah tambahan “balet” dan “profesional”, membedakan sebuah kata atau sebuah frasa dengan kata atau frasa lainnya dalam hirarki yang sama.

2.4.3 Metode Berdasarkan Analisis Distribusi

Selanjutnya, model kemiripan distribusi cenderung bergantung pada sejenis hipotesis distribusional yang menyatakan bahwa sebuah kata dengan makna yang setara dapat muncul pada konteks yang sama (Harris, 1954). Dengan demikian, dapat diasumsikan jika terdapat kesamaan konteks, berarti terdapat juga kemiripan semantik antara dua kata. Ada dua macam kemiripan: kemiripan antar atribut dan kemiripan antar relasi (Medin, Goldstone, & Gentner, 1990). Gambar 4 menunjukkan contoh untuk setiap macam.

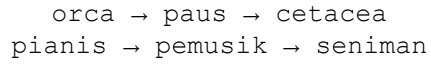


Gambar 4 (a) Kemiripan Atribut; (b) Kemiripan Relasi

Seperti yang dapat dilihat pada contoh di atas, kemiripan atribut ditemukan ketika terdapat kata-kata yang muncul secara bersamaan atau ketika pasangan kata memiliki kemiripan konteks. Kata “orca” dan “paus” sering dikelilingi kata-kata “samudra”, “laut”, dan “terbesar”. Demikian, disimpulkan bahwa kedua kata tersebut memiliki atribut yang mirip. Lalu terdapat pula kemiripan relasi antara pasangan kata “ikan” dan “ air” dengan pasangan kata “manusia” dan “darat” karena keduanya menunjukkan lingkungan hidup dari subjek bersangkutan.

2.4.4 Metode Berdasarkan Inklusi Distribusi

Terakhir, inklusi distribusi adalah ketika relasi transitif dan asimetris juga dipertimbangkan. Dalam hal ini, sebuah struktur hirarki semantik akan dihasilkan dari relasi tersebut. Artinya, sebuah kata dengan pangkat yang lebih tinggi akan mewariskan semua konsep umumnya dan kemudian bawahannya menambahkan sendiri fitur pembedanya. Contohnya dapat dilihat pada gambar di bawah ini.



Gambar 5 Relasi Transitif

Pada Gambar 5 terdapat dua contoh kalimat yang menggambarkan relasi transitif dan asimetris. “Orca” adalah sejenis “paus” dan sama halnya, “paus” adalah sejenis “cetacea”. Secara transitif berarti “orca” adalah sejenis “cetacea”, namun secara asimetris, “cetacea” bukan sejenis “orca”. Sebagai pembanding, “pianis” adalah salah satu macam “pemusik” dan “pemusik” adalah salah satu macam “seniman”. Dengan demikian, “pemusik” adalah salah satu macam “seniman”, namun “seniman” bukan salah satu macam “pianis”.

2.5 *Online Incremental Learning*

Terdapat banyak area untuk mengaplikasikan *crowdsourcing*, umumnya *crowdsourcing* digunakan untuk proses verifikasi. Dalam hubungannya dengan *extra-logical error* yang mungkin muncul dalam proses pembelajaran mesin,

crowdsourcing dapat menjadi sebuah solusi yang layak (Mortensen, et al., 2016). Untuk memasukkan *crowdsourcing* dalam proses pembelajaran ontologi, dapat digunakan metode *online incremental learning*.

Metode pembelajaran mesin digunakan untuk mendapatkan informasi yang relevan dari data yang sudah dikumpulkan. Metode ini dapat dikategorikan menjadi dua: *batch* dan *incremental* (Wang, Zhao, Hoi, & Jin, 2014). Dalam pembelajaran secara *batch*, semua data diakses secara serentak. Selain itu, *batch learning* biasanya membutuhkan ketersediaan fitur yang lengkap untuk pembelajaran. Namun dalam *incremental learning*, data yang diproses tidak harus lengkap tersedia. Dengan data dan fitur yang terbatas sekalipun, informasi yang relevan tetap dapat diekstrak dengan *incremental learning*.

Kemudian pembelajaran juga dapat dibedakan menjadi pembelajaran dengan kondisi *offline* atau *online* (Losing, Hammera, & Wersing, 2018). Pada kondisi *offline incremental learning*, terdapat beberapa kelompok data *training* yang digunakan untuk menghasilkan beberapa model *training*. *Training* dilakukan dengan cara membuat model dari satu kelompok data *training*, kemudian mengintegrasikan kelompok data berikutnya untuk memperbaiki model. Model yang dihasilkan paling akhirlah yang digunakan untuk memprediksi data *testing*.

Berbeda dengan kondisi *online incremental learning*, pada kondisi ini, data tidak dibagi menjadi data *training* dan data *testing*. Dimulai dengan menggunakan sebuah model yang sudah ditentukan sebelumnya, akan diprediksi dataset yang akan digunakan untuk memperbaiki model tersebut. Dengan memanfaatkan dataset hasil prediksi dan model yang sudah ada, dibentuk lagi model yang lebih baik. Siklus ini terus diulang sampai didapatkan model yang dikehendaki.

Online incremental learning pun mendapatkan lebih banyak perhatian di dalam dunia penelitian karena kemampuannya untuk memenuhi kebutuhan dalam menangani data dalam volum yang besar dalam satu waktu, sehingga informasi yang baru dapat terus-menerus diintegrasikan. Dalam setiap siklus *online incremental learning*, tidak perlu dilakukan pembuatan model dari awal. Model yang sudah ada dapat diadaptasikan dengan informasi yang baru secara bertahap

tanpa perlu mengulang seluruh proses *training*. Hal ini jelas menghemat waktu dan sumber daya dalam pembelajaran yang dilakukan.

Tantangan utama dalam *online incremental learning* adalah *catastrophic forgetting* (M.French, 1999) dan *concept drift* (Ditzler, Roveri, Alippi, & Polikar, 2015). Hal ini terjadi ketika model yang sudah ada gagal mempertahankan penyimpanan pengetahuan yang telah didapat sebelumnya, akibatnya akurasi model tersebut berkurang. Selain itu juga terdapat tantangan di mana jumlah data *training* yang dapat dipertahankan sangat terbatas. Demikian, pemilihan sampel yang tepat yang bisa mempertahankan dan merepresentasikan pengetahuan yang sudah didapat sebelumnya menjadi suatu hal yang penting.

Terdapat beberapa algoritma yang banyak digunakan untuk *online incremental learning*. Beberapa algoritma yang sudah diulas oleh Losing et al. (Losing, Hammera, & Wersing, 2018) adalah sebagai berikut.

- *Incremental Support Vector Machine* (ISVM) adalah versi *incremental* dari *Support Vector Machine* (SVM) (Cauwenberghs & Poggio, 2000). ISVM merupakan sebuah algoritma *lossless*; algoritma yang menghasilkan model yang sama dengan model hasil pembelajaran secara *batch* jika vektor kandidat yang digunakan mengandung semua data yang sudah pernah ada.
- LASVM adalah perkiraan *online* dari pemecah SVM (Bordes, Ertekin, Weston, & Bottou, 2005). LASVM mengecek kemungkinan bahwa contoh yang diproses adalah *support vector* dan memutuskan untuk menghapus *support vector* yang tidak terpakai.
- *Online Random Forest* (ORF) adalah versi incremental dari algoritma Random Forest (Saffari, Leistner, Santner, Godec, & Bischof, 2009). Berdasarkan sejumlah *tree* yang diberikan, setiap *tree* akan tumbuh terus menerus dengan menambahkan percabangan pada saat didapati jumlah sampel yang cukup pada satu *leaf*.
- *Incremental Learning Vector Quantization* (ILVQ) adalah adaptasi dari *Generalized Leaning Vector Quantization* (GLVQ) statis menjadi model dinamik yang selalu berkembang, yang akan menambahkan prototipe baru jika memang dibutuhkan (Sato & Yamada, 1995).

- *Learn++* (LPP) adalah sebuah algoritma yang tidak terganung pada model. LPP memproses sampel baru dalam *chunk* yang besarannya sudah ditentukan (Polikar, Upda, Upda, & Honavar, 2001). Kumpulan pengklasifikasi dasar akan melalui proses *training* dan digabungkan melalui proses *voting* dengan pembobotan untuk setiap *chunk*.
- *Incremental Extreme Machine Learning* (IELM) merumuskan kembali solusi kuadrat terkecil ELM secara *batch* ke dalam skema sekuensial (Liang, Huang, Saratchandran, & Sundararajan, 2006). Jaringan pada IELM adalah statis dan jumlah neuron yang tersembunyi harus ditentukan sebelumnya.
- *Naïve Bayes* memasangkan distribusi Gaussian yang paralel terhadap satu sumbu untuk setiap *class* dan menggunakannya sebagai kemungkinan estimasi pada algoritma *Naïve Bayes* (Zhang H. , 2004). Model yang tersebar ini memungkinkan peningkatan efisiensi waktu pemrosesan dan kebutuhan *memory* selama pembelajaran.
- *Stochastic Gradient Descent* (SGD) adalah metode optimasi yang efisien untuk pembelajaran model diskriminatif dengan cara meminimalkan fungsi *loss*. SGD yang dipadukan dengan model linier memiliki performa yang baik untuk data yang tersebar dan mencakup dimensi yang luas.

Dari eksperimen yang sudah dilakukan oleh Losing et al (Losing, Hammera, & Wersing, 2018), terdapat tiga algoritma yang bisa dikatakan sebagai yang terbaik, yaitu: ISVM, ILVQ, dan SGD. Telah dibuktikan bahwa ISVM mempunyai akurasi yang tertinggi dibandingkan dengan ILVQ dan SGD dalam eksperimen pembelajaran secara *online*. Namun eksperimen lain yang mempertimbangkan kemungkinan terjadinya *concept drift* menunjukkan bahwa ISVM tidak mampu menyelesaikan proses pembelajaran dalam batas waktu yang telah ditentukan, yaitu 24 jam. Di antara semua algoritma yang berhasil menyelesaikan proses dalam batas waktu yang ditentukan, ILVQ memiliki akurasi tertinggi, diikuti oleh SGD. Dengan mempertimbangkan kompleksitas model yang dihasilkan setelah *training*, SGD menjadi algoritma yang kompleksitasnya terendah dibandingkan algoritma lain.

2.5.1 Random Forest

Random Forest adalah sebuah metode *ensemble learning* atau metode pembelajaran yang menggabungkan beberapa algoritma *learning* untuk mendapatkan prediksi yang lebih baik daripada prediksi dari masing-masing algoritma penyusunnya. Random Forest dapat digunakan untuk berbagai macam keperluan, seperti klasifikasi, regresi dan sebagainya d (Loupe, 2014).

Secara sederhana, Random Forest bekerja dengan cara membangun banyak *decision tree* pada saat *training* dan mengembalikan *class* yang merupakan modus dari semua *class* hasil atau rata-rata prediksi. Dengan cara ini, risiko *overfitting* yang terjadi pada implementasi *decision tree* dapat diatasi karena Random Forest mempertimbangkan hasil dari semua *tree* yang ada.

Berikut adalah beberapa alasan pemilihan dan kelebihan algoritma Random Forest (Cutler, Cutler, & Stevens, 2011):

- Mampu menangani regresi dan klasifikasi dengan banyak *class*.
- Hanya memerlukan satu atau dua parameter *tuning*.
- Dapat menghitung seberapa penting sebuah variabel.

Pada dasarnya algoritma Random Forest sudah mampu bekerja dengan baik, namun terdapat beberapa parameter yang bisa digunakan untuk lebih meningkatkan akurasi, antara lain: jumlah *predictor* yang dipilih secara acak pada setiap *node*, jumlah *tree* pada *forest*, dan ukuran *tree* yang diukur dengan jumlah *node* terkecil untuk dipecah atau jumlah terbesar *terminal node*.

Selanjutnya, algoritma Random Forest juga memiliki fitur untuk melihat seberapa penting sebuah variabel atau fitur. Melalui fitur ini, dapat ditemukan fitur apa saja yang memiliki dampak besar pada hasil prediksi dan fitur apa yang tidak terlalu penting. Dengan demikian, akurasi model dapat ditingkatkan dengan mempertimbangkan fitur yang memang berpengaruh pada hasil prediksi.

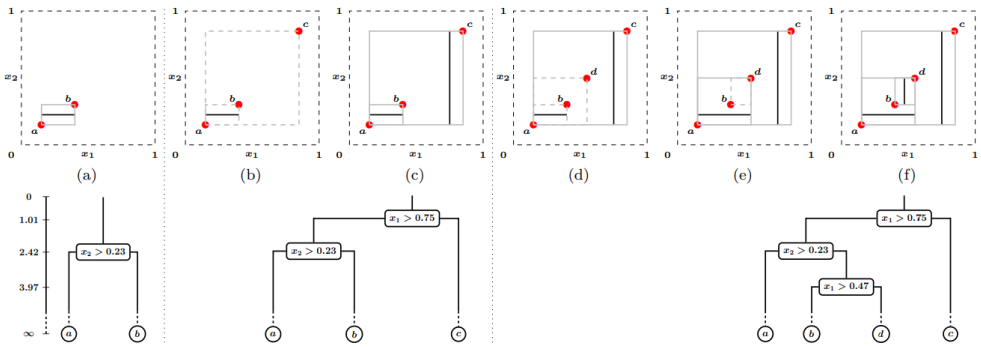
2.5.2 Mondrian Forest

Algoritma Mondrian Forest merupakan algoritma yang dikembangkan dari Random Forest. Secara spesifik, Mondrian Forest menggabungkan proses Mondrian dengan Random Forest. Mondrian Forest dikembangkan karena terdapat

observasi bahwa pelatihan ulang untuk versi *batch* pada Random Forest dapat membutuhkan waktu yang lama. Selain itu, versi *online* Random Forest yang ada (Saffari, Leistner, Santner, Godec, & Bischof, 2009) membutuhkan banyak komputasi dan data awal (Lakshminarayanan, Roy, & Teh, 2014).

Berdasarkan klaim pengembang, Mondrian Forest dapat beroperasi pada mode *batch* atau pada mode *online*, dengan kecepatan setara proses mode *online*. Dengan demikian dapat disimpulkan bahwa Mondrian Forests lebih efisien dalam mengolah data karena mampu memberikan prediksi dengan cepat. Selain itu, Mondrian Forest membangun *tree* secara konsisten dengan memanfaatkan observasi yang ada. Sebuah *tree* dapat ditambahkan percabangannya di titik manapun selama hasilnya konsisten dengan *subtree* di bawah percabangan tersebut.

Gambar 6 (Lakshminarayanan, Roy, & Teh, 2014) mengilustrasikan bagaimana Mondrian Tree dibangun. Untuk dua poin data dalam sebuah area, dibuat pembatas antara area yang menjadi area poin pertama dan area poin kedua (percabangan pertama). Saat ditambahkan data baru, area akan dibagi lagi dengan memastikan bahwa pembatas area tidak memasuki area untuk dua poin awal untuk menjaga konsistensi percabangan yang sudah ada. Demikian seterusnya pembagian area akan dilakukan untuk menambah atau mengubah percabangan dari Mondrian Tree. Percabangan dapat ditambahkan di atas percabangan yang sudah ada atau di bawah percabangan yang sudah ada. Selain itu, rentang percabangan yang sudah ada juga dapat diperbarui.



Gambar 6 Ilustrasi Pembangunan Mondrian Tree

2.6 Editor untuk Evaluasi Ontologi

Evaluasi ontologi biasanya dilakukan oleh seorang ahli. Kemampuan manusia untuk menilai sebuah konteks lebih baik daripada kemampuan mesin, sehingga ontologi yang dibangun secara manual biasanya lebih dipandang baik (Yang & Callan, Human-Guided Ontology Learning, 2008). Mesin bergantung pada data *training* yang diberikan, sementara manusia tidak demikian. Walaupun manusia mungkin melakukan kesalahan, terdapat banyak cara untuk menangani permasalahan ini—salah satunya *crowdsourcing*.

Mortensen menyebutkan dalam penelitiannya bahwa *extra-logical error*—error yang hanya dapat dideteksi oleh interpretasi manusia—dapat menyebabkan dampak negatif pada sebuah aplikasi (Mortensen, Crowdsourcing Ontology Verification, 2013). Dalam kasus ini, dapat diasumsikan bahwa *crowdsourcing* seharusnya dapat mengurangi kemunculan error ini dalam ontologi yang dihasilkan. Salah satu contoh integrasi *crowdsourcing* ke dalam pembelajaran ontologi telah dilakukan oleh Yang dengan menggunakan aplikasi yang dikembangkan yang bernama OntoCop (Yang & Callan, OntoCop: Constructing Ontologies for Public Comments, 2009).

OntoCop adalah sebuah *interface* interaktif yang dapat digunakan untuk mengoranisir konsep ke dalam ontologi (Sawsaa & Lu, 2010). Dalam OntoCop, ekstraksi konsep dilakukan dengan menggunakan pendekatan konvensional, sedangkan formasi relasi dilakukan dengan pendekatan yang lebih baru. Dengan mengkombinasikan pembelajaran mesin dan interaksi manusia, OntoCop mampu menghemat waktu dan usaha manusia dalam pembangunan ontologi.

Pertama, OntoCop akan menampilkan ontologi awal berdasarkan *head-noun matching*, WordNet *hypernyms* (Fellbaum, 1998), dan *lexico-syntactic patterns* (Hearst, Automatic acquisition of hyponyms from large text corpora, 1992). Kemudian orang akan memodifikasi ontologi tersebut dengan menggunakan *drag-and-drop interface* sederhana. Ketika OntoCop menerima perubahan tersebut, OntoCop akan menyesuaikan algoritma *clustering* yang digunakan lalu menampilkan kembali ontologi yang sudah diperbaiki. Siklus interaksi manusia dan komputer ini akan terus berulang sampai editor puas dengan hasil ontologi akhir.

Pada OntoCop, pengembang menggunakan *Cohen's Kappa statistic* untuk membandingkan *inter-annotator agreement*—yaitu kesamaan hasil antara evaluasi yang berbeda—pada dua skenario eksperimen yang berbeda. Skenario pertama adalah dua percobaan manual, sementara skenario kedua adalah satu percobaan manual dan satu percobaan interaktif.

Hasil perbandingan ini menunjukkan bahwa kedua skenario memiliki nilai yang setara, yang berarti bahwa tidak ada perbedaan kualitas yang signifikan antara kedua ontologi yang dihasilkan dari dua skenario eksperimen yang berbeda. Lebih lanjut, pada perbandingan usaha yang dilakukan dalam pembangunan ontologi, skenario interaktif terbukti membutuhkan usaha yang lebih sedikit. Terdapat lebih sedikit aktivitas yang dilakukan pada proses pembangunan dan lebih sedikit pula waktu yang digunakan. Dengan demikian, OntoCop telah memenuhi tujuannya untuk membangun ontologi berkualitas tinggi dengan usaha yang seminimumnya.

2.7 Penelitian Terkait

Beberapa penelitian yang berkaitan telah diidentifikasikan dan berikut ini adalah rangkuman dari penelitian tersebut.

Tabel 2 Rangkuman Penelitian Terkait

No	Judul	Isi Penelitian
1	Evaluating the Complementarity of Taxonomic Relation Extraction Methods Across Different Languages (Granada, Vieira, Trojahn, & Aussenac-Gilles, 2018)	Fokus: Ektraksi relasi Metode: Uji coba dengan beberapa dataset Tujuan: Evaluasi metode dengan dataset yang sama namun dalam bahasa yang berbeda Kasus: Europarl TED Talks Portugis dan Inggris Hasil: Perbandingan metrik hirarki dan evaluasi
2	Automatic relationship extraction from agricultural text for ontology construction (Kaushik & Chatterjee, 2018)	Fokus: *Ontology engineering* Metode: *RegEx* dan teknik NLP Tujuan: Mengusulkan dan evaluasi algoritma untuk pembangungan ontologi otomatis Kasus: Agrikultur Hasil: Algoritma *RelExOnt*

No	Judul	Isi Penelitian
3	Concept relation extraction using Naive Bayes classifier for ontology-based question answering systems (Kumar & Zayaraz, 2015)	Fokus: Ekstraksi relasi Metode: *Syntactic and semantic probability-based Naive Bayes classifier* Tujuan: Mengusulkan dan evaluasi algoritma ektraksi relasi Kasus: Ontologi umum Hasil: Algoritma ektraksi relasi
4	Dynamic Evaluation of Ontologies (Lakel & Bendella, 2015)	Fokus: *Ontology engineering* Metode: Uji coba beberapa dataset Tujuan: Menggabungkan beberapa alat untuk konstruksi ontologi otomatis secara kolaboratif Kasus: Dynamo dan WordNet 1.2 Hasil: Sistem *Dynamic Evaluation of Ontologies*
5	Incremental on-line learning: A review and comparison of state-of-the-art algorithms (Losing, Hammera, & Wersing, 2018)	Fokus: *Incremental online learning* Metode: Uji coba dengan beberapa dataset Tujuan: Evaluasi akurasi, konvergensi, kecepatan, dan kompleksitas model Kasus: Dataset umum Hasil: Perbandingan performa algoritma
6	Crowdsourcing Ontology Verification (Mortensen, Crowdsourcing Ontology Verification, 2013)	Fokus: *Crowdsourcing* Metode: Ulas literatur, pembuatan *framework*, aplikasi *crowdsourcing* Tujuan: Pembangunan aplikasi *crowdsourcing* Kasus: Ontologi *biomedical* Hasil: Aplikasi untuk *crowdsourcing*
7	Is the crowd better as an assistant or a replacement in ontology engineering? An exploration through the lens of the Gene Ontology (Mortensen, et al., 2016)	Fokus: *Crowdsourcing* Metode: Perbandingan hasil uji coba pada *expert* dan pada massa Tujuan: Menentukan apakah massa adalah asisten atau pengganti *expert* dalam pembangunan ontologi. Kasus: Gene ontology Hasil: Massa menjadi asisten *expert*
8	Completeness and consistency analysis for evolving knowledge bases (Rashid, et al., 2019)	Fokus: *Knowledge base* Metode: *Data profiling*, *consistency check*, uji coba beberapa dataset Tujuan: Identifikasi masalah kelengkapan dan konsistensi *knowledge base* yang berkembang Kasus: DBpedia dataset, 3cixty knowledge base Hasil: Model terbaik adalah *Random Forest*

No	Judul	Isi Penelitian
9	Noun ontology generation from Wikipedia article using Map Reduce with pattern-based approach (Santoso, Nugraha, Yuniarno, & Hariadi, 2015)	Fokus: Ektraksi relasi Metode: Menggunakan *taxonomy template information* pada Wikipedia Tujuan: Evaluasi proses ekstraksi relasi *hyponymy* dan *meronymy* Kasus: Artikel Wikipedia Indonesia Hasil: Akurasi proses yang dilakukan tinggi
10	OntoCop: A Virtual Community of Practice to Create Ontology of Information Science (IS) (Sawsaa & Lu, 2010)	Fokus: *Ontology editor* Metode: *Software engineering* Tujuan: Membangun *ontology editor* yang terstruktur dan menarik minat partisipan Kasus: Ontologi untuk *Information Science* Hasil: Aplikasi OntoCop
11	Theory of Ontological Engineering (Verma, Kaur, & Kaur, 2017)	Fokus: *Ontology engineering* Metode: Ulas literatur dan analisis Tujuan: Mendefinisikan dan evaluasi *ontology engineering* Kasus: Definisi umum Hasil: Kompilasi definisi dan evaluasi *ontology engineering*
12	Online Feature Selection and Its Applications (Wang, Zhao, Hoi, & Jin, 2014)	Fokus*: Incremental online learning* Metode: Uji coba beberapa dataset dan skenario Tujuan: Mengusulkan dan evaluasi algoritma *learning* dengan *full* dan *partial* input Kasus: *Computer vision* dan *bioinformatics* Hasil: Algoritma *online feature selection*
13	Human-Guided Ontology Learning (Yang & Callan, Human-Guided Ontology Learning, 2008)	Fokus: *Ontology engineering* Metode: *Supervised clustering algorithm* Tujuan: Memperbaiki proses pembangunan ontologi dan inkorporasi preferensi personal Kasus: Komentar publik untuk regulasi pemerintah Hasil: Ontologi memiliki kualitas setara dengan manual dan preferensi personal berhasil ditambahkan.

No	Judul	Isi Penelitian
14	OntoCop: Constructing Ontologies for Public Comments (Yang & Callan, OntoCop: Constructing Ontologies for Public Comments, 2009)	Fokus: *Ontology editor* Metode: Uji coba pada *expert* Tujuan: Evaluasi performa pembangunan ontologi secara interaktif Kasus: Komentar publik untuk regulasi pemerintah Hasil: Performa pembangunan interaktif sangat efektif dan efisien menghemat sumber daya
15	Relation Extraction (Zettlemoyer, 2013)	Fokus: Ekstraksi relasi Metode: Ulas literatur Tujuan: Mendefinisikan proses ektraksi relasi Kasus: Dataset umum Hasil: Definisi proses ekstraksi relasi
16	Game-based crowdsourcing to support collaborative customization of the definition of sustainability (Bakhta, El-Diraby, & Hossainic, 2018)	Fokus: *Crowdsourcing* Metode: Uji coba pada massa Tujuan: Mengusulkan dan evaluasi pendekatan untuk menangkap opini masyarakat Kasus: Perencanaan kota di Amerika Utara Hasil: Pendekatan yang diusulkan mampu menangkap opini masyarakat dan meningkatkan analisis sentimen
17	Crowdsourced Data Management: A Survey (Li, Wang, Zheng, & Franklin, 2016)	Fokus: *Crowdsourcing* Metode: Ulas literatur Tujuan: Survei dan sintesis pengelolaan data *crowdsourcing* Kasus: Umum Hasil: Rangkuman metode pengelolaan data *crowdsourcing* dan teknik terkait.
18	Knowledge Base Semantic Integration Using Crowdsourcing (Meng, Chen, Tong, & Zhang, 2017)	Fokus: *Knowledge Base* Metode: *Crowdsourcing* Tujuan: Investigasi dan usulan solusi masalah integrasi semantik otomatis *Knowledge Base* Kasus: YAGO dan DBPedia Hasil: Solusi sudah efektif dan efisien
19	Pilot experiments on a designed crowdsourcing decision tool (Thuan, Antunes, & Johnstone, 2016)	Fokus: *Crowdsourcing* Metode: *Software engineering* Tujuan: Membuktikan bahwa sistem yang dibangun mendukung proses *crowdsourcing* yang terintegrasi Kasus: Mahasiswa jurusan IT Hasil: Masukan untuk melakukan eksperimen, mendesain alat *crowdsourcing*

No	Judul	Isi Penelitian
20	A Crowd Wisdom Management Framework for Crowdsourcing Systems (Zhang, Shangguan, & Yuan, 2016)	Fokus: *Crowdsourcing* Metode: Perancangan dan uji coba pada dataset Tujuan: Mengusulkan framework pengelola wisdom dari massa Kasus: MovieLens dataset (rating film) Hasil: *Framework* MacroWiz

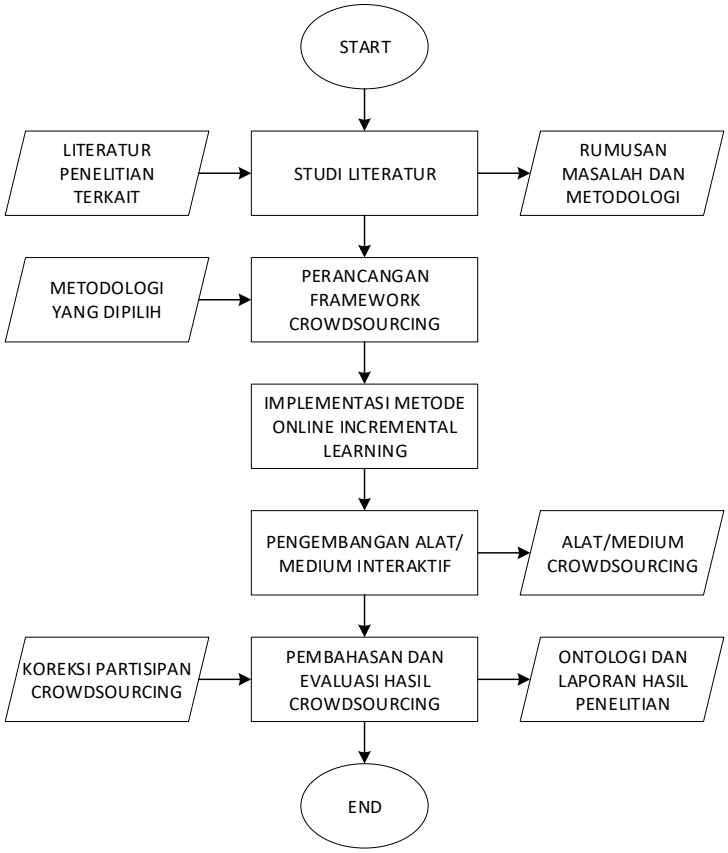
(Halaman sengaja dikosongkan)

BAB 3
METODOLOGI PENELITIAN

Berdasarkan sumber dan penelitian terdahulu yang telah dikaji, maka dibuat sebuah rancangan penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia. Rancangan yang dibahas dalam bab ini meliputi metode penelitian dan pengembangan, usulan arsitektur sistem, sumber data, dan rencana penelitian.

3.1 Metode Penelitian dan Pengembangan

Metode penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia dapat dilihat pada Gambar 7.



Gambar 7 Metode Penelitian dan Pengembangan

Penelitian akan diawali dengan studi literatur untuk mengetahui rumusan masalah yang ada, yaitu *extra-logical error* dan bagaimana metodologi untuk menangani masalah tersebut, yaitu *crowdsourcing* dengan menggunakan metode *online incremental learning*. Dari rencana yang sudah dibuat, akan dirancang sebuah kerangka kerja (*framework*) untuk implementasi usulan penanganan masalah yang dirumuskan.

Kemudian peneliti akan melakukan implementasi metode pembelajaran yang dipilih sesuai dengan kerangka kerja yang dirancang; dilanjutkan dengan pengembangan alat/medium interaktif untuk proses *crowdsourcing*. Alat ini akan diujicobakan kepada partisipan *crowdsourcing* untuk mengoreksi relasi pada ontologi. Dari hasil koreksi, performa algoritma yang dipilih dan hasil ontologi dari proses *crowdsourcing* akan dievaluasi. Terakhir, penelitian akan didokumentasikan dalam bentuk laporan dan ontologi yang sudah dikoreksi akan dirilis.

Berdasarkan metode peneilitian dan pengembangan yang diusulkan, penelitian *Online Incremental Learning* Berbasis *Crowdsourcing* untuk Ekstraksi Relasi Ontologi Bahasa Indonesia terdiri atas lima tahap berikut: (1) studi literatur, (2) perancangan kerangka kerja *crowdsourcing*, (3) implementasi metode *online incremental learning*, (4) pengembangan alat/medium interaktif, (5) pembahasan dan evaluasi hasil *crowdsourcing*. Berikut akan dijelaskan masing-masing tahapan.

3.1.1 Studi Literatur

Pada tahap ini, peneliti akan mengumpulkan dan mengulas literatur yang terkait dengan penelitian yang akan diusulkan. Beberapa topik literatur yang diulas adalah mengenai *crowdsourcing*, pembangunan ontologi, ekstraksi relasi, metode *online incremental learning*, dan editor ontologi.

Kemudian dari proses ini, peneliti mendefinisikan rumusan masalah dan usulan solusi untuk permasalahan tersebut. Adapun inti yang akan diangkat pada penelitian ini adalah implementasi metode *online incremental learning* dalam *crowdsourcing* untuk mengatasi masalah *extra-logical error* pada proses ekstraksi relasi ontologi dalam Bahasa Indonesia.

3.1.2 Perancangan Kerangka Kerja *Crowdsourcing*

Peneliti mengusulkan kerangka kerja, atau dalam hal ini arsitektur sistem, yang akan dibangun dalam penelitian. Kerangka kerja ini mencakup proses di mana data disiapkan untuk *crowdsourcing*, kemudian proses pelaksanaan *crowdsourcing*, diikuti dengan pengolahan hasil *crowdsourcing*, sampai akhirnya evaluasi untuk menentukan apakah siklus *crowdsourcing* akan diulang atau dihentikan.

3.1.3 Implementasi Metode *Online Incremental Learning*

Melanjutkan dari perancangan kerangka kerja, akan diimplementasikan *online incremental learning* menurut algoritma Mondrian Forest. Algoritma ini dipilih berdasarkan ulasan dari pustaka yang sudah diidentifikasikan sebelumnya. Beberapa kriteria yang menjadi penentu pemilihan algoritma, antara lain performa algoritma dalam hal akurasi dan kompleksitas model yang dihasilkan dari algoritma tersebut (Losing, Hammera, & Wersing, 2018).

3.1.4 Pengembangan Alat/Medium Interaktif

Lalu mengacu pada editor ontologi yang sudah ada (Yang & Callan, OntoCop: Constructing Ontologies for Public Comments, 2009), peneliti akan mengembangkan sebuah alat/medium interaktif yang dapat digunakan oleh massa untuk memverifikasi ontologi. Faktor yang dipertimbangkan dalam pembangunan alat ini, yaitu alat harus dapat memverifikasi apakah pengguna yang akan dilibatkan dalam *crowdsourcing* dapat memberikan masukan yang valid, *interface* harus dapat dipahami dengan mudah oleh pengguna yang merupakan kalangan umum, dan alat juga harus dapat mengolah masukan pengguna untuk pemrosesan lebih lanjut.

3.1.5 Pembahasan dan Evaluasi Hasil *Crowdsourcing*

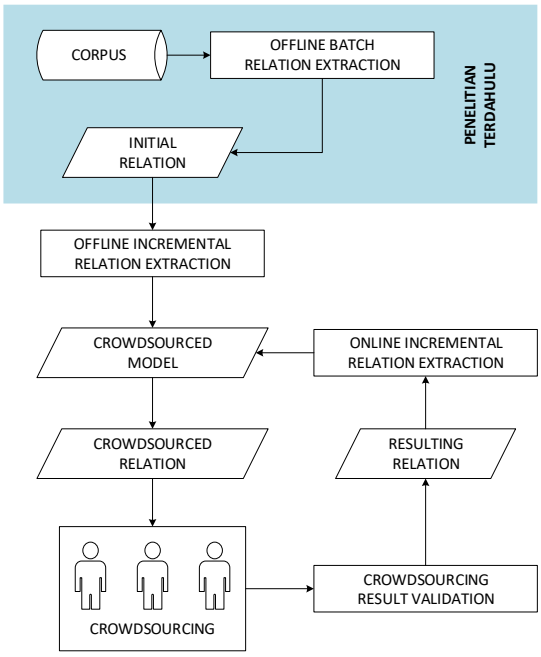
Dari implementasi metode pembelajaran dan alat yang dibangun, akan dilakukan evaluasi terhadap hasil data dari algoritma yang dipilih dan hasil akhir ontologi. Evaluasi ini bertujuan untuk menilai apakah algoritma yang dipilih dapat digunakan untuk memperbaiki pembangunan ontologi Bahasa Indonesia dengan memanfaatkan *crowdsourcing* untuk mendeteksi *extra-logical error*. Evaluasi akan

dilakukan dengan *precision*, *recall*, dan *F1 Score* (Powers, 2011). Adapun skenario evaluasi direncanakan sebagai berikut:

- Skenario 1: Evaluasi pengaruh *crowdsourcing* dalam mengoreksi *extra-logical error*, membandingkan relasi yang dihasilkan pada setiap siklus *crowdsourcing* dengan relasi yang diidentifikasikan sebagai benar.
- Skenario 2: Evaluasi performa metode online incremental learning dalam memperbaiki proses ekstraksi relasi ontologi, membandingkan akurasi relasi yang dihasilkan dari proses online dan offline incremental learning.

3.2 Usulan Arsitektur Sistem

Dalam penelitiannya, Mortensen (Mortensen, Crowdsourcing Ontology Verification, 2013) mengusulkan sebuah ikhtisar alur kerja untuk *crowdsourcing* dan Losing et al. (Losing, Hammera, & Wersing, 2018) juga menyajikan skema pembelajaran *online* yang digunakan dalam eksperimen *online incremental learning*. Dengan memperhatikan kedua kerangka kerja ini, maka dirancanglah arsitektur sistem seperti pada Gambar 8.



Gambar 8 Arsitektur Sistem

Proses dalam sistem dimulai dengan ekstraksi relasi secara *offline* dari korpus yang sudah ada. Pada korpus ini, semua konsep sudah dilabeli dan relasi awal dibuat menggunakan metode ekstraksi relasi secara *offline* dan *batch*. Tahapan tersebut tidak dicakup dalam penelitian ini, tetapi peneliti akan menggunakan data yang dihasilkan dengan merujuk kepada penelitian terdahulu oleh Phi et al. (Phi, Santoso, Shimbo, & Matsumoto, 2018).

Kemudian dengan menggunakan relasi awal, dilakukan ekstraksi relasi secara *offline* dan *incremental* untuk menghasilkan model *crowdsourcing*. Proses ini diawali dengan mengekstrak beberapa fitur dari data yang sudah ada. Fitur-fitur tersebut adalah:

1. Probabilitas pattern menunjukkan sebuah relasi,
2. Probabilitas pasangan entitas menunjukkan sebuah relasi,
3. Kemiripan pasangan entitas (*entity matching*), dan
4. *Cosine similarity* antara pasangan entitas.

Fitur-fitur yang sudah didapatkan akan dilatih dengan algoritma Random Forest untuk membuat model awal yang akan memprediksi relasi untuk dikoreksi oleh partisipan *crowdsourcing*.

Selanjutnya akan diambil contoh relasi untuk dievaluasi oleh partisipan *crowdsourcing*. Relasi yang diambil didasarkan pada subtipe relasi dengan akurasi yang paling rendah. Relasi tersebut akan disajikan kepada partisipan dengan menggunakan alat yang dibangun untuk merepresentasikan relasi tersebut secara visual. Representasi yang umum adalah *graph*.

Proses *crowdsourcing* akan diperuntukkan untuk partisipan yang sudah dewasa dengan asumsi partisipan tersebut mampu mengidentifikasi entitas yang diberikan. Namun partisipan tidak harus paham akan konsep relasi *meronymy*. Hal ini dipertimbangkan dengan tujuan melakukan simulasi *crowdsourcing* yang nyata, di mana mayoritas partisipan mungkin tidak pernah mempelajari konsep relasi yang akan diekstrak.

Hasil dari proses *crowdsourcing* akan melalui proses validasi sebelum digunakan untuk proses ekstraksi relasi secara *online* dan *incremental*. Validasi yang dilakukan berupa pengukuran *inter-annotator agreement*, yaitu seberapa

besar persetujuan antara sekian banyak anotasi (Artstein, 2017). Untuk data yang homogen, maka perhitungan koefisien *inter-annotator agreement* dapat dilakukan secara langsung. Sedangkan untuk data yang heterogen, perhitungan koefisien *inter-annotator agreement* harus dibandingkan menurut masing-masing kelompok data yang homogen dan tidak bisa langsung mengambil dari keseluruhan data.

Perhitungan *inter-annotator agreement* yang digunakan adalah Fleiss' Kappa. Perhitungan ini dipilih karena Fleiss' Kappa dapat digunakan pada kondisi di mana terdapat lebih dari dua anotator. Berbeda dengan Cohen's Kappa yang diperuntukkan untuk pelabelan dengan dua anotator saja.

Dengan *target class* baru yang dihasilkan melalui koreksi partisipan *crowdsourcing*, akan dibangun model *online* dengan menambahkan koreksi yang valid. Lalu akan diambil kembali contoh relasi untuk dievaluasi oleh partisipan. Siklus ini akan diulang untuk memperbarui model *online* yang dibangun.

Proses *crowdsourcing* akan terus diiterasi sampai partisipan memutuskan bahwa relasi yang dihasilkan oleh model tidak perlu dimodifikasi lebih lanjut atau sampai tercapai level akurasi yang dikehendaki. Dengan kata lain, model yang digunakan untuk proses ekstraksi relasi sudah memenuhi kriteria akurasi seperti evaluasi secara manual.

Hasil akhir dari proses crowdsourcing adalah ontologi sederhana dengan relasi yang sudah dikoreksi. Ontologi yang dihasilkan dalam bentuk *graph* ini dapat digunakan untuk membantu proses *information retrieval* pada domain terbuka.

3.3 Sumber data

Dataset yang digunakan berasal dari penelitian yang dilakukan oleh Phi et al. (Phi, Santoso, Shimbo, & Matsumoto, 2018). Pada penelitian tersebut, data yang digunakan dihimpun dari berbagai sumber, seperti artikel-artikel dari berbagai jenis website, dan diolah sehingga sudah diidentifikasikan konsep dan relasi awalnya.

Kategori relasi yang digunakan pada dataset adalah relasi *part-whole* atau relasi *meronymy*, yang dapat didefinisikan sebagai relasi yang menunjukkan apakah sebuah konsep merupakan anggota dari konsep lain. Berikut subtipe relasi *part-*

whole yang diidentifikasikan menurut (Phi & Matsumoto, Integrating Word Embedding Offsets into the Espresso System for Part-Whole Relation, 2016).

- *Component-Of* menunjukkan relasi antara integral dengan komponen fungsionalnya, contohnya: jari dan tangan.
- *Member-Of* menunjukkan relasi antara objek fisik atau peran dengan sebuah agregasi (tim, organisasi, dan sebagainya), contohnya: pemain dan tim.
- *Portion-Of* atau *Sub-Quantity-Of* menunjukkan relasi antara jumlah materi atau unit (masih merupakan objek yang sama), contohnya: oksigen dan air.
- *Stuff-Of* atau *Substance-Of* atau *Constituted-Of* menunjukkan relasi antara objek fisik yang berbeda atau dengan materi, contohnya: besi dan mobil.
- *Located-In* menunjukkan relasi antar konsep dengan region dua dimensi, contohnya: Jakarta dan Indonesia.
- *Contained-In* menunjukkan relasi antar konsep dengan region tiga dimensi, contohnya: chip dan prosesor.
- *Phase-Of* atau *Involved-In* atau *Feature-Activity* menunjukkan relasi antara sebuah fase dengan sebuah proses, contohnya: mengunyah dan makan.
- *Participates-In* menunjukkan relasi antara konsep dan proses, contohnya: enzim dan reaksi.

Terdapat sekitar 5700 baris dalam dataset yang digunakan. Setiap baris data memiliki format sebagai berikut:

- 1stEntity → konsep pertama
- Pattern → frasa atau kata yang menunjukkan relasi
- 2ndEntity → konsep kedua
- 1stEntity - normalized → normalisasi konsep pertama
- Pattern - normalized → normalisasi frasa atau kata yang menunjukkan relasi
- 2ndEntity - normalized → normalisasi konsep kedua
- Kalimat sumber yang mengandung *triple* (1stEntity, Pattern, 2ndEntity)
- URL sumber artikel atau paragraf yang mengandung kalimat sumber
- Subtipe relasi *part-whole*

- Urutan yang menunjukkan hubungan 1stEntity dan 2ndEntity: *Part-Whole* atau *Whole-Part*

Berikut ini adalah contoh satu baris data sesuai dengan format yang dijelaskan sebelumnya:

- 1stEntity → the committee
- Pattern → is made up of
- 2ndEntity → eight members
- 1stEntity - normalized → the committee
- Pattern - normalized → be make up of
- 2ndEntity - normalized → eight member
- Kalimat sumber → the committee is made up of eight members, appointed by the board of voluntary planning.
- URL sumber → http://www.dominionpaper.ca/articles/2147
- Subtipe relasi → Member-Of
- Urutan → Whole-Part

Dari sumber aslinya, dataset yang digunakan masih dalam Bahasa Inggris. Oleh sebab itu, dataset ini akan diterjemahkan dahulu secara manual ke dalam Bahasa Indonesia agar dapat digunakan untuk pemrosesan lebih lanjut dalam penelitian ini. Validitas terjemahan ke dalam Bahasa Indonesia akan diperiksa dengan pengukuran *inter-annotator agreement* memakai *Cohen's Kappa statistic* (Artstein, 2017) atau dapat dijamin apabila dataset tersebut sudah diterima untuk dipublikasikan dalam penelitian terdahulu.

BAB 4
HASIL DAN PEMBAHASAN

Pembahasan pada bab ini disusun berdasarkan metodologi penelitian, yaitu penerapan dan hasil percobaan untuk kerangka kerja *crowdsourcing*. Secara lebih detail, hal-hal yang dibahas meliputi: pengumpulan data, praproses data, ekstraksi fitur, penerapan *offline incremental learning*, penerapan *online incremental learning*, pengembangan alat interaktif untuk *crowdsourcing*, dan evaluasi kerangka kerja yang diusulkan.

5.1 Pengumpulan Data

Data yang digunakan pada penelitian ini bersumber dari penelitian terdahulu yang berjudul Ranking-Based Automatic Seed Selection and Noise Reduction for Weakly Supervised Relation Extraction oleh Phi, et al. (Phi, Santoso, Shimbo, & Matsumoto, 2018). Data asli adalah dalam bahasa Inggris, yang kemudian diterjemahkan secara manual ke dalam bahasa Indonesia. Data dalam bahasa Indonesia masih mempertahankan format asli seperti data dalam bahasa Inggris. Berikut salah satu contoh perbandingan data sebelum dan sesudah diterjemahkan.

Tabel 3 Perbandingan Format Data

No	Field	Bahasa Inggris (Asli)	Bahasa Indonesia (Terjemahan)
1	1st Entity	proteins	-
2	Pattern	are made up of	-
3	2nd Entity	amino acids	-
4	1st Entity (normalized)	protein	protein
5	Pattern (normalized)	be make up of	terdiri dari
6	2nd Entity (normalized)	amino acid	asam amino
7	Kalimat Sumber	proteins are made up of amino acids	protein terdiri dari asam amino

8	URL Sumber	http://swiftweb.com/ holistic/badbreathcause treatment.html	-
9	Subtipe Relasi	Component-Of	Component-Of
10	Urutan	Whole-Part	T2-T1

Pada data yang diterjemahkan, 1st Entity atau entitas pertama disingkat menjadi T1, sementara 2nd Entity atau entitas kedua disingkat menjadi T2. Urutan yang menunjukkan bagian *part* dan *whole* selalu dituliskan dengan cara berikut: bagian *part* diikuti dengan bagian *whole*. Jadi jika 1st Entity adalah bagian (*part*) dari 2nd Entity, maka akan dituliskan T1-T2. Sebaliknya jika 2nd Entity adalah bagian (*part*) dari 1st Entity, maka akan dituliskan T2-T1. Contoh data di atas menunjukkan bahwa entitas kedua "asam amino" adalah bagian dari entitas pertama "protein", maka pada urutannya dituliskan "T2-T1".

Data yang diambil ke dalam bahasa Indonesia hanya entitas pertama, pola yang menunjukkan relasi, entitas kedua, kalimat sumber, subtipe relasi, dan urutan bagian yang menunjukkan *part* dan *whole*. Tidak semua data diterjemahkan dengan pertimbangan, hanya data yang akan diproses saja yang diterjemahkan. Namun jika memang dibutuhkan, tidak menutup kemungkinan peneliti akan merilis data lengkap dalam bahasa Indonesia atau menambahkan data lain untuk percobaan.

5.2 Praproses Data

Pada tahap ini, data yang sudah diterjemahkan akan dipersiapkan ke dalam format yang sesuai sebelum dilakukan ekstraksi fitur. Terdapat beberapa praproses yang dilakukan, antara memuat data ke dalam program, menyesuaikan tipe data untuk ekstraksi fitur, dan mengategorikan data untuk memudahkan proses ekstraksi fitur. Berikut akan dijelaskan masing-masing proses tersebut.

5.2.1 Memuat Data

Data hasil terjemahan tersedia dalam dua format, yaitu .ann dan .txt. Data dengan ekstensi .ann adalah data yang sudah dianotasi oleh peneliti terdahulu

dengan menggunakan bantuan alat anotasi BRAT[1]. File ini memuat tipe entitas pertama dan kedua, posis indeks entitas pertama dan kedua di dalam kalimat (dimulai dari 0), entitas pertama dan kedua, subtipe relasi, dan urutan seperti yang dijelaskan pada bagian Pengumpulan Data. Kemudian data dengan ekstensi .txt memuat kalimat sumber. Gambar 9 di bawah ini menunjukkan contoh file dengan ektensi .ann.

```
T1  MISC 0 7    Protein
T2  MISC 21 31  asam amino
R1  Component-Of Arg1:T2 Arg2:T1
```

Gambar 9 Contoh File .ann

Pemuatan data dilakukan dengan memanfaatkan *library* Python glob dan os. Berikut adalah kode yang dibuat untuk memuat data pada Gambar 10.

```
01:  def load_annotated_data():
02:      corpus = []
03:      path_name   =   os.path.join(os.getcwd(),   'data_text',
     '*.ann')
04:      file_list = glob.glob(path_name)
05:      for file_path in file_list:
06:          with open(file_path) as file_input:
07:              string_data = file_input.read()
08:              # Kode untuk memasukkan file sesuai format
09:      return corpus
```

Gambar 10 Kode untuk Memuat Data

Baris 1 mendefinisikan fungsi yang digunakan untuk memuat data. Lalu baris 2 mendeklarasikan *array* yang akan digunakan untuk menampung data. Baris 3 dan 4 digunakan untuk mencari *path* dari file data. Dilanjutkan baris 5, untuk setiap file yang ditemukan di dalam *path*, baris 6 membuka file tersebut, dan baris 7 membaca file. Pada baris 8 dibuat code untuk menyesuaikan format data yang mana akan dijelaskan pada Kode 2. Baris 9 mengembalikan *array* yang sudah terisi data sesuai format.

[1] https://brat.nlplab.org/

5.2.2 Menyesuaikan Tipe Data

Proses untuk menyesuaikan tipe data dilakukan pada saat memuat data. Setiap file yang dibaca akan dipecah berdasarkan baris, kemudian berdasarkan *delimiter whitespace*. Setiap data yang ditemukan dicek tipenya dan ditampung ke dalam *array* sesuai urutan yang ditentukan. Dengan menggunakan cara yang sama, data teks kalimat asal juga dimuat dan digabungkan ke dalam *array* data keseluruhan. Dengan demikian, peneliti dapat mengambil data yang dibutuhkan dari *array* data keseluruhan.

5.2.3 Mengategorikan Data

Proses ini dilakukan dengan tujuan mengelompokkan data yang merupakan pasangan entitas dan pattern. Daftar pasangan entitas dapat diperoleh berdasarkan label pada data sumber, sementara daftar pattern diambil dari kata-kata yang terdapat di antara dua entitas pada kalimat sumber.

Dari data yang sudah lengkap, dibuat lagi dua macam *array* yang masing-masing berfungsi menyimpan entitas saja atau pattern saja. Secara umum, cara pembuatan *array* entitas dan *array* pattern adalah sama, yang membedakan hanya di bagian pengecekan saja. Perbedaan ini dikarenakan data Bahasa Indonesia tidak menyediakan pattern secara langsung, namun peneliti harus mendapatkan pattern tersebut dari kalimat sumbernya. Algoritma pada Gambar 11 berikut adalah bagian yang sama pada proses kategorisasi entitas dan pattern.

```
01:   Memuat data dengan anotasi
02:   Memuat data kalimat asal
03:   Deklarasi array penampung setiap subtipe relasi
04:   Untuk setiap row pada array
05:       Cek tipe entitas atau pattern
06:       Simpan data pada array sesuai subtipe relasi
```

Gambar 11 Algoritma untuk Kategorisasi Data

Pengambilan data entitas dilakukan sekaligus dengan pengecekan arah relasi untuk memudahkan penggunaan data pada tahap selanjutnya. Dengan demikian, data yang tersimpan sudah dibuat dalam format arah relasi dari entitas

pertama ke entitas kedua. Sebagai contoh, jika terdapat kalimat “protein terdiri dari asam amino” di mana asam amino adalah komponen penyusun protein, maka arah relasinya adalah “T2-T1” atau dari entitas kedua ke entitas pertama. Dalam hal ini data akan disimpan dengan urutan “asam amino” diikuti “protein” supaya arah semua relasi menjadi sama “T1-T2”. Untuk menghilangkan duplikasi pasangan entitas yang sudah tersimpan, digunakan algoritma pada Gambar 12.

```
01:   Untuk setiap subtipe relasi
02:       Urutkan semua pasangan entitas
03:       Cek setiap pasangan entitas yang ditemukan
04:           Jika pasangan entitas kembar, hapus salah satu
05:           Jika pasangan entitas unik, simpan
```

Gambar 12 Algoritma untuk Mengecek Duplikasi Entitas

Pengambilan data pattern dilakukan dengan algoritma pada Gambar 13 berikut ini. Pada proses ini, dilakukan pemotongan kalimat sumber dengan cara mengambil kata-kata yang ada di antara kedua entitas. Jika di antara kedua entitas tidak ada kata apapun, maka diasumsikan tidak ada pattern.

```
01:   Mencari indeks terakhir entitas pertama
02:   Mencari indeks pertama entitas kedua
03:   Mengecek indeks mana yang lebih besar
04:       Memotong kalimat mulai dari indeks kecil ke indeks besar
```

Gambar 13 Algoritma untuk Kategorisasi Pattern

Pengecekan duplikasi pattern lebih sederhana karena *array* kumpulan pattern adalah sebuah vektor. Peneliti cukup mengonversi tipe *array* dari *list* yang mengijinkan duplikasi data ke *dictionary* yang mengharuskan data unik. Kemudian kembalikan *dictionary* tersebut ke dalam bentuk *list*. Berikut Gambar 14 adalah kode yang menunjukkan proses ini untuk satu subtipe relasi. Kode ini diulang untuk semua subtipe relasi.

```
01:   list_componentOf = list(dict.fromkeys(list_componentOf))
```

Gambar 14 Kode untuk Mengecek Duplikasi Pattern

Dengan demikian praproses data telah selesai dan selanjutnya data akan diolah dalam proses ekstraksi fitur yang akan dijabarkan pada bagian berikut.

5.3 Ekstraksi Fitur

Proses ekstraksi fitur dilakukan untuk membentuk model awal dalam proses prediksi subtipe relasi. Beberapa fitur yang digunakan untuk proses pembelajaran, antara lain: probabilitas pattern menunjukkan sebuah relasi, probabilitas pasangan entitas menunjukkan sebuah relasi, kemiripan pasangan entitas (*entity matching*), dan *cosine similarity* antara pasangan entitas.

5.3.1 Probabilitas Pattern

Fitur ini didapatkan dengan cara membandingkan pattern dari kalimat yang akan diprediksi relasinya dengan pattern yang sudah pernah ditemukan. Jika pattern ditemukan menunjukkan suatu subtipe relasi, maka probabilitasnya untuk subtipe relasi tersebut adalah 1. Jika pattern tidak ditemukan menunjukkan subtipe relasi tersebut, maka probabilitasnya adalah 0. Kode pada Gambar 15 menunjukkan proses pengecekan probabilitas pattern dengan hasil vektor sepanjang delapan.

```
01:   def check_pattern(pattern, list_pattern):
02:       result = []
03:       for i in list_pattern:
04:           if pattern in i:
05:               result.append("1")
06:           else:
07:               result.append("0")
08:       return result
```

Gambar 15 Kode untuk Fitur Probabilitas Pattern

5.3.2 Probabilitas Pasangan Entitas

Dengan metode yang hampir sama dengan pengecekan probabilitas pattern, dilakukan juga pengecekan probabilitas untuk pasangan entitas. Hasil dari proses ini sama dengan hasil dari pengecekan probabilitas pattern, yaitu sebuah vektor dengan panjang delapan untuk delapan subtipe relasi. Berikut kode pengecekan probabilitas pasangan entitas pada Gambar 16.

```
01:   def check_entity(entity_1, entity_2, list_entity):
02:       result = []
03:       compare = [entity_1, entity_2]
04:       for i in list_entity:
05:           if compare in i:
06:               result.append("1")
07:           else:
08:               result.append("0")
09:       return result
```

Gambar 16 Kode untuk Fitur Probabilitas Pasangan Entitas

5.3.3 *Entity Matching* antara Pasangan Entitas

Fitur selanjutnya adalah kemiripan pasangan entitas berdasarkan *string distance* antara kedua entitas. *String distance* dihitung menggunakan implementasi algoritma Needleman-Wunsch. Pertama, peneliti akan membuat sebuah matriks yang menunjukkan nilai perbandingan setiap karakter di dalam string kedua entitas. Kemudian dari matriks tersebut dicari sebuah *path* dengan nilai tertinggi. Berdasarkan *path* ini, didapatkan nilai *string distance* yang dapat dikonversi menjadi sebuah nilai antara 0 sampai dengan 1, di mana 0 berarti kedua entitas tidak memiliki kesamaan sama sekali dan 1 berarti kedua entitas sama persis.

5.3.4 *Cosine Similarity* Pasangan Entitas

Fitur ini didapatkan dengan cara merepresentasikan entitas ke dalam vektor, lalu menghitung kemiripannya dengan menggunakan *cosine similarity*. *Word embedding* yang digunakan pada penelitian ini adalah BERT [2] (*Bidirectional Encoder Representations from Transformers*) yang dipublikasikan oleh Google. *Embedding* ini dipilih karena dianggap mampu merepresentasikan informasi kontekstual dengan lebih baik dibandingkan dengan *embedding* yang statis.

Secara sederhana, embedding yang dihasilkan akan memiliki dimensi 12 layer dengan 768 value untuk masing-masing layer. Dari 12 layer ini, akan diambil value dari 4 layer terakhir karena *embedding* dari 4 layer ini memiliki akurasi yang paling tinggi. Berikut akan dijelaskan tahap pemrosesan mulai dari embedding sampai dengan perhitungan *cosine similarity*.

[2] https://github.com/google-research/bert

Pertama, setiap kalimat akan melalui proses *tokenizing* seperti *embedding* lain pada umumnya. *Token* khusus yang digunakan untuk *embedding* pada penelitian ini adalah "[CLS]" untuk menandai awal kalimat dan "[SEP]" untuk menandai akhir kalimat. *Embedding* dilakukan dengan menggunakan model yang sudah dirilis untuk *embedding* multilingual, yaitu *bert-base-multilingual-cased*.

Hasil *embedding* sebuah kalimat adalah *tensor* dengan dimensi n x 12 x 768, di mana n adalah jumlah token. Perlu diperhatikan bahwa sebuah kata dapat dipecah menjadi beberapa token, tergantung dari model yang digunakan. Oleh sebab itu, diberikan proses tambahan untuk mendapatkan indeks awal dan akhir sebuah kata. Lalu sebuah entitas juga dapat terdiri dari satu atau lebih kata. Sehingga ditambahkan juga proses untuk menemukan indeks awal dan akhir token yang menunjukkan sebuah entitas.

Untuk entitas yang terdiri dari beberapa token, maka *embedding* untuk masing-masing token akan diambil nilai rata-ratanya (*mean*). Setelah itu barulah akan dijumlahkan value dari 4 layer terakhir (*sum*). Hasil dari proses ini adalah sebuah vektor dengan panjang 768 elemen. Vektor inilah yang digunakan untuk perhitungan nilai *cosine similarity*.

5.4 Penerapan *Offline Incremental Learning*

Sesuai dengan arsitektur sistem yang telah dirancang, proses ini bertujuan membangun model awal yang digunakan untuk menghasilkan relasi pada tahap *crowdsourcing*. Penerapan *offline incremental learning* dilakukan dua kali dengan menggunakan alat bantu RapidMiner[3] dan *library* scikit-learn[4] untuk memastikan bahwa performa model benar-benar baik. Dari proses ini ditemukan bobot relevansi sebuah fitur dalam menentukan subtipe relasi dan ditemukan juga akurasi model awal yang akan digunakan untuk *crowdsourcing*.

Fitur RapidMiner yang digunakan pada tahap ini adalah Auto Model. Peneliti pertama memilih dataset berisi fitur yang akan digunakan, kemudian menentukan task yang hendak dilakukan: prediksi, *clustering*, atau identifikasi

[3] https://rapidminer.com/
[4] https://scikit-learn.org/stable/index.html

outliers. Dari dataset yang digunakan, tentukan kolom yang menjadi hasil prediksi. Lalu peneliti juga dapat mengecek dan mengatur *target class* yang sudah dipilih, dilanjutkan dengan memilih input data fitur. Setelah itu peneliti menentukan model apa yang digunakan dan bagaimana optimasinya, kemudian mengeksekusi pembangunan model.

Pembuatan model dengan *library* scikit-learn (sklearn) dilakukan dengan mempersiapkan data fitur ke dalam format yang sesuai untuk *training*, lalu menjalankan fungsi-fungsi yang disediakan. Data mentah semua fitur akan dicek kelengkapannya dan diubah ke dalam format float32. Dari proses ini, didapatkan 5651 baris data yang layak dipakai dari 5727 baris data awal. Lalu data akan dibagi menjadi *training* dan *testing set* dengan menggunakan kode pada Gambar 17.

```
01:   train_features, test_features, train_labels, test_labels =
   train_test_split(features,       labels,       test_size=0.25,
   random_state=27)
```

Gambar 17 Kode untuk Pembagian Training dan Testing Set

Kelebihan *library* sklearn adalah peneliti dapat menentukan besarnya proporsi data untuk *training* dan *testing*, serta peneliti dapat menentukan *random state* mana yang dipakai agar hasil klasifikasi konsisten. Dalam percobaan ini, digunakan proporsi 3:1 untuk *training* dan *testing*, serta *random state* 27.

Pembangunan dan pengujian model dilakukan dengan menjalankan kode pada Gambar 18 berikut ini.

```
01:   clf = RandomForestClassifier(n_estimators=100, max_depth=30,
   random_state=27)
02:   clf.fit(train_features, train_labels)
03:   predictions = clf.predict(test_features)
```

Gambar 18 Kode untuk Pembangunan dan Pengujian Model *Offline*

Input yang digunakan untuk pembangunan model adalah semua fitur yang sudah diekstrak, dengan output atau *target class* sebanyak delapan subtipe relasi. Model yang digunakan adalah Random Forest dengan optimasi jumlah maksimum

tree sebanyak 100 tree dan maksimum kedalaman 30 untuk sebuah tree. Berikut didapatkan hasil pembobotan fitur dan performa model yang dihasilkan.

Terdapat empat fitur yang digunakan, yaitu f1, f2, f3, dan f4. Fitur f1 dan f2 merupakan probabilitas berdasarkan delapan subtipe relasi, sehingga masing-masing terbagi menjadi delapan field. Kemudian f3 merupakan hasil perhitungan *string distance* dan nilai *similarity* antar entitas. Dilanjutkan f4 adalah nilai *cosine similarity* pasangan entitas. Tabel 4 di bawah ini menunjukkan perbandingan nilai pembobotan masing-masing fitur dari RapidMiner dan scikit-learn.

Tabel 4 Nilai Pembobotan Fitur

No	Fitur	RapidMiner		scikit-learn	
1	f1_pattern_1	0.02835444	0.27002076	0.02714647	0.34615267
2	f1_pattern_2	0.08703606		0.05089895	
3	f1_pattern_3	0.01196931		0.03671897	
4	f1_pattern_4	0.02377111		0.07756600	
5	f1_pattern_5	0.03146719		0.03775412	
6	f1_pattern_6	0.02483880		0.02908938	
7	f1_pattern_7	0.01769112		0.03326593	
8	f1_pattern_8	0.04489271		0.05371285	
9	f2_entity_1	0.02689664	0.25837125	0.04005497	0.39902870
10	f2_entity_2	0.03119324		0.00979597	
11	f2_entity_3	0.01287090		0.07202484	
12	f2_entity_4	0.02200766		0.01758003	
13	f2_entity_5	0.08021434		0.11850090	
14	f2_entity_6	0.04922273		0.06651649	
15	f2_entity_7	0.01845392		0.01922606	
16	f2_entity_8	0.01751182		0.05532944	
17	f3_dist_e1_e2	0.12380934	0.19218480	0.05558232	0.12985352
18	f3_sim_score	0.06837546		0.07427120	
19	f4_cosine_sim	0.08072881	0.08072881	0.12496506	0.12496506

Besarnya nilai pembobotan menunjukkan seberapa relevan sebuah fitur untuk menentukan subtipe relasi. Dapat dilihat bahwa masing-masing metode memiliki pembobotan fitur yang berbeda dan hasil akurasi masing-masing metode pun berbeda. Model yang dihasilkan RapidMiner memiliki akurasi sebesar 84.3% sementara model dari sklearn memiliki akurasi sebesar 87.3%. Demikian model yang akan digunakan untuk *crowdsourcing* adalah model dari sklearn.

Berdasarkan model sklearn, ditemukan bahwa fitur yang paling relevan adalah fitur probabilitas entitas dengan total pembobotan 0.39902870, diikuti fitur probabilitas pattern dengan total pembobotan 0.34615267, fitur *string distance* dengan total pembobotan 0.12985352, dan fitur *cosine similarity* dengan total pembobotan 0.12496506. Nilai pembobotan dihitung menurut rata-rata nilai yang dihasilkan dari 100 tree yang dibuat pada Random Forest. Demikian, terdapat kemungkinan total semua nilai bobot tidak mencapai nilai penuh sebesar 1. Dalam percobaan ini, model RapidMiner kehilangan bobot sebesar 0.1986944, sedangkan model sklearn mencapai nilai penuh sebesar 1.

Performa model yang dihasilkan sklearn berdasarkan nilai *precision* dan *recall* tercantum pada Tabel 5 berikut. Secara berurutan subtipe relasi yang dimaksudkan dalam Tabel 5 adalah:

- R1: Component-Of
- R2: Contained-In
- R3: Located-In
- R4: Member-Of
- R5: Participates-In
- R6: Involved-In
- R7: Subquantity-Of
- R8: Constituted-Of

Tabel 5 Performa Model yang Dihasilkan

	true R1	**true R2**	**true R3**	**true R4**	**true R5**	**true R6**	**true R7**	**true R8**	**Preci-sion**
pr R1	108	1	0	18	0	1	7	8	0.755
pr R2	0	57	0	2	0	0	0	0	0.966
pr R3	0	0	139	2	0	0	0	0	0.986
pr R4	31	1	1	290	0	7	32	3	0.795
pr R5	1	0	0	0	203	0	1	0	0.990
pr R6	1	0	0	0	2	103	0	1	0.963
pr R7	8	0	0	17	0	1	95	6	0.748
pr R8	11	0	0	4	1	1	10	239	0.898
Recall	0.675	0.966	0.993	0.871	0.985	0.912	0.655	0.930	

Secara umum model yang dihasilkan memiliki akurasi yang baik. Nilai *precision* tertinggi adalah 99.0% untuk subtipe relasi Participates-In atau yang menunjukkan relasi konsep-proses dan nilai *precision* terendah adalah 74.8% untuk subtipe relasi Subquantity-Of atau yang menunjukkan relasi dengan unit materi. Nilai *recall* tertinggi adalah 99.3% untuk subtipe Located-In yang menunjukkan lokasi entitas dan nilai *recall* terendah adalah 65.5% untuk subtipe Subquantity-Of.

Di bawah ini terdapat Tabel 6 yang menunjukkan jumlah awal pasangan entitas, jumlah pasangan entitas setelah duplikasi dihapus, jumlah awal pattern, dan jumlah pattern setelah duplikasi dihapus untuk masing-masing subtipe relasi. Disertakan juga persentase data unik terhadap seluruh data awal.

Tabel 6 Jumlah Data Input

	Pattern			**Entity**		
	Awal	**Unik**	**%**	**Awal**	**Unik**	**%**
Component-Of	643	205	0.319	643	262	0.407
Contained-In	272	43	0.158	272	243	0.893
Located-In	534	49	0.092	534	516	0.966
Member-Of	1272	241	0.189	1272	412	0.324
Participates-In	872	310	0.356	872	194	0.222
Involved-In	497	232	0.467	497	175	0.352
Subquantity-Of	555	134	0.241	555	455	0.820
Constituted-Of	1082	402	0.372	1082	302	0.279

Nilai *precision* menunjukkan seberapa banyak prediksi suatu relasi yang benar-benar tepat menunjuk ke relasi tersebut dibandingkan dengan semua prediksi yang mengarah ke relasi tersebut. Dengan kata lain terdapat banyak relasi yang diprediksikan sebagai Subquantity-Of, namun ternyata bukan menunjukkan relasi tersebut. Subtipe relasi Subquantity-Of memiliki variasi pattern yang tidak terlalu banyak, sementara variasi entitasnya sangat banyak.

Nilai *recall* menunjukkan seberapa banyak prediksi suatu relasi yang benar-benar tepat menunjukkan relasi tersebut dibandingkan dengan semua prediksi yang dibuat untuk data suatu relasi. Berarti, untuk subtipe relasi Subquantity-Of dan Component-Of, banyak prediksi yang mengarah ke relasi lainnya—dalam hal ini, prediksi terbanyak mengarah ke subtipe relasi Member-Of. Jika data awal diamati,

dari 1272 pattern yang ditemukan menunjuk ke subtipe relasi Member-Of, sebanyak 840 juga menunjuk ke subtipe relasi Subquantity-Of dan Component-Of. Sehingga diperkirakan terjadi *overlap* antar ketiga subtipe relasi ini.

5.5 Penerapan *Online Incremental Learning*

Proses penerapan *online incremental learning* mencakup penggunaan model yang dihasilkan dari penerapan *offline incremental learning*, proses koreksi label atau *crowdsourcing*, dilanjutkan dengan pengolahan hasil koreksi tersebut dan validasi *inter-annotator agreement*, serta penentuan apakah hasil koreksi akan digunakan untuk memperbarui model awal. Percobaan yang dilakukan melibatkan sebanyak tiga anotator yang akan mengecek klasifikasi delapan subtipe relasi.

Berdasarkan hasil penerapan *offline incremental learning*, ditemukan bahwa subtipe relasi yang akurasinya rendah adalah Subquantity-Of. Dengan demikian percobaan akan dilakukan untuk memperbaiki model, sehingga akurasi subtipe relasi Subquantity-Of dapat ditingkatkan. Terdapat total 536 baris data yang valid untuk digunakan pada percobaan ini.

Metode untuk memuat data, melakukan praproses data, dan penggunaan model adalah sama seperti yang telah dijelaskan pada bagian sebelumnya. Lalu proses koreksi label pada percobaan dilakukan secara manual, namun dapat pula dilakukan dengan menggunakan alat interaktif yang dikembangkan. Penjelasan mengenai alat tersebut dicantumkan pada subbab berikutnya. Hasil koreksi akan diolah dan divalidasi dengan perhitungan *inter-annotator agreement*. Metode perhitungan yang digunakan adalah Fleiss' Kappa karena Fleiss Kappa mempertimbangkan jumlah anotator. Berbeda dengan perhitungan lain, seperti Cohen's Kappa yang digunakan saat jumlah anotator hanya dua orang.

Perhitungan Fleiss Kappa dapat diilustrasikan seperti pada Tabel 7 berikut, di mana n adalah jumlah anotator, N adalah jumlah label atau kalimat, dan k adalah jumlah subtipe relasi. Dalam percobaan ini, n = 3, N = 536, dan k = 8.

Tabel 7 Ilustrasi Perhitungan Fleiss Kappa

	R_1	R_2	…	R_k	P_i
L_1	0	0			1.0
L_3	0	0			1.0
…					
L_N				$x_{i,j}$	
P_j	0.004	0.0			

Untuk setiap label L akan dihitung frekuensi anotator memasukkannya ke dalam kategori R. Nilai P_i untuk setiap kalimat dan P_j untuk setiap kategori dihitung dengan menggunakan Rumus (1) dan (2) berikut. Nilai P_i menunjukkan tingkat *agreement* setiap anotator untuk sebuah label L, sementara nilai P_j menunjukkan proporsi setiap kategori R di seluruh dokumen.

$$P_i = \frac{1}{n(n-1)} \times \left(\sum_{j=1}^{k} {x_{i,j}}^2 - n \right) \quad (1)$$

$$P_j = \frac{1}{N \times n} \times \sum_{i=1}^{N} x_{i,j} \quad (2)$$

Selanjutnya dihitung nilai P_{bar}, yaitu rata-rata nilai P_i dan dihitung juga nilai $P_{e\text{-}bar}$, yaitu jumlah kuadrat nilai P_j, dengan Rumus (3) dan (4).

$$\bar{P} = \frac{1}{N} \times \sum_{i=1}^{N} P_i \quad (3)$$

$$\bar{P_e} = \sum_{j=1}^{k} {P_j}^2 \quad (4)$$

Nilai *inter-annotator agreement* secara keseluruhan atau nilai Fleiss' Kappa didapatkan dengan menggunakan Rumus (5) berikut.

$$kappa = \frac{\bar{P} - \bar{P_e}}{1 - \bar{P_e}} \tag{5}$$

Pada percobaan yang dilakukan, dihitung nilai *inter-annotator agreement* untuk hasil koreksi subtipe relasi yang diprediksi dengan menggunakan model *offline*. Nilai yang digunakan adalah nilai *agreement* untuk label pada setiap pasangan entitas (P_i). Jika nilai *agreement* di atas 0.7, diasumsikan koreksi dari mayoritas anotator adalah tepat. Dengan demikian hasil koreksi akan digunakan untuk memperbarui data yang digunakan untuk memperbaiki model *online*.

Pembangunan model online dilakukan dengan menggunakan bantuan *library* scikit-garden[5]. Pada *library* ini terdapat implementasi dari Mondrian Forest, yaitu salah satu versi *online* dari Random Forest. Sintaks untuk membangun model dan mendapatkan prediksi dengan menggunakan Mondrian Forest dapat dilihat pada Gambar 19 berikut ini.

```
01:   mondrian_clf   =   MondrianForestClassifier(n_estimators=100,
   max_depth=30, random_state=27)
02:   mondrian_clf.fit(features_all, labels_all)
03:   predictions = mondrian_clf.predict(features_exp)
```

Gambar 19 Kode untuk Pembangunan dan Pengujian Model *Online*

Digunakan konfigurasi yang sama untuk membangun model *online*, yaitu jumlah tree sebanyak 100 tree dan maksimum kedalaman 30 untuk sebuah tree. Data yang digunakan untuk membuat model *online* adalah seluruh data pasangan entitas dan relasi yang valid (5651 baris data). Sementara untuk pengujian model, digunakan data subtipe relasi Subquantity-Of saja (536 baris data).

[5] https://scikit-garden.github.io/

Setelah model selesai dibangun dan didapatkan prediksi relasi yang baru, maka proses *crowdsourcing*, pengolahan hasil koreksi, dan pembaruan model akan diulang dengan cara yang sama. Iterasi akan dihentikan sampai jumlah perulangan tertentu atau sampai nilai akurasi model yang dikehendaki tercapai.

Berkaitan dengan *inter-annotator* agreement, standar nilai kappa yang diterima dapat berbeda-beda tergantung pada berbagai faktor, salah satunya jumlah kategori. Pada percobaan yang dilakukan, secara umum, tingkat *agreement* tiap anotator untuk satu baris data dapat dikatakan sangat tinggi karena hasil prediksi model cukup akurat sehingga tidak banyak koreksi yang dilakukan. Namun sehubungan dengan perbedaan persepsi antar anotator, ditambah dengan konsep relasi *meronymy* yang tidak terlalu umum, nilai kappa secara keseluruhan yang didapat cukup rendah. Berikut Tabel 8 menunjukan hasil perhitungan *inter-annotator agreement* pada percobaan yang dilakukan.

Tabel 8 Perhitungan *Inter-Annotator Agreement*

Percobaan	**P_{bar}**	**$P_{e\text{-}bar}$**	**kappa**
0	0.97979798	0.97995670	-0.00791855
1	0.74410774	0.71998322	0.08615385
2	0.95959596	0.92245689	0.47894736
3	0.73737374	0.75047898	-0.05252158
4	1.0	1.0	n.a
5	0.96190476	0.96263039	-0.01941748

Percobaan 0 adalah koreksi untuk anotasi yang dihasilkan dari model *offline* dan percobaan 1 sampai dengan 5 adalah koreksi untuk anotasi yang dihasilkan dari model *online*. Seperti yang disebutkan pada bagian sebelumnya, nilai yang dipakai pada percobaan adalah nilai P_{bar} atau nilai *agreement* untuk sebuah label subtipe relasi pasangan entitas.

Koreksi pada percobaan 0, 2, dan 5 tidak terlalu banyak, sehingga nilai P_{bar} cukup tinggi. Sedangkan pada percobaan 4, ditemukan koreksi yang dilakukan oleh partisipan *crowdsourcing* adalah sama, sehingga didapatkan nilai P_{bar} yang penuh sebanyak satu. Nilai P_{bar} yang paling rendah adalah pada percobaan 1 dan 3. Pada

percobaan tersebut, ditemukan cukup banyak pasangan entitas yang subtipe relasinya cukup ambigu, sehingga banyak perubahan yang dilakukan.

Akurasi model yang dihasilkan pada *online learning* memiliki akurasi yang hampir sama, namun secara umum, akurasi model *online* meningkat dibandingkan dengan model *offline*. Berikut ini Tabel 9 adalah nilai akurasi model untuk setiap percobaan yang dilakukan. Nilai akurasi dihitung untuk dua kondisi, yaitu akurasi untuk semua data dan akurasi untuk data subtipe relasi yang dikoreksi.

Tabel 9 Perhitungan Akurasi Model

Percobaan	Akurasi Keseluruhan	Akurasi Subtipe
	0.87331918	0.90858209
0	0.88605803	0.96641791
1	0.89525832	0.99067164
2	0.89313517	0.98507463
3	0.89667374	0.99440299
4	0.89313517	0.98507463
5	0.89738146	0.99626866

Model yang digunakan pada percobaan 0 adalah model yang didapatkan dari penerapan *offline incremental learning* dengan akurasi keseluruhan 87.3%. Jika model ini diaplikasikan hanya pada subtipe relasi Subquantity-Of, didapatkan akurasi sebesar 90.9%. Secara umum, dapat diasumsikan bahwa mayoritas hasil prediksi adalah benar, sehingga diperkirakan tidak akan ada banyak koreksi yang diberikan oleh partisipan *crowdsourcing*.

Setelah koreksi pertama, yaitu pada percobaan 0, akurasi model langsung meningkat. Akurasi keseluruhan dalah 88.6% dan akurasi untuk subtipe adalah 96.6%. Pada percobaan-percobaan selanjutnya, nilai akurasi keseluruhan masih berkisar di nilai 89% dan nilai akurasi subtipe di antara 96% sampai dengan 99%.

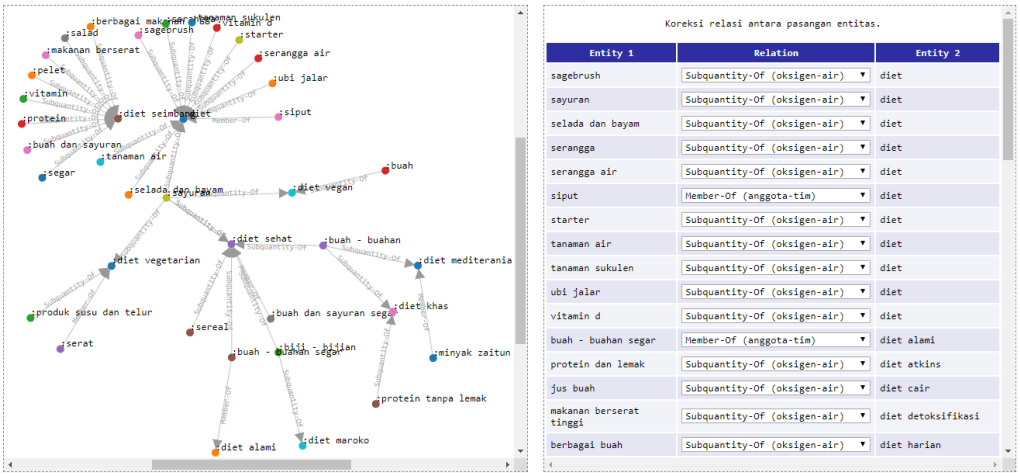
Peningkatan nilai akurasi ini menunjukkan bahwa proses *crowdsourcing* berperan untuk meningkatkan akurasi model, sehingga relasi yang diprediksi untuk pasangan entitas menjadi lebih tepat. Nilai akurasi model untuk setiap percobaan dapat berubah menjadi lebih tinggi ataupun lebih rendah. Namun perubahan tersebut masih dalam batas yang wajar, yaitu berkisar antara 0% sampai dengan

1%. Hal ini mungkin terjadi karena koreksi partisipan *crowdsourcing* mungkin tidak selalu benar, sehingga terjadi penurunan akurasi model yang dihasilkan.

Model akhir yang didapatkan dari penerapan *online incremental learning* memiliki akurasi keseluruhan 89.7% dan akurasi subtipe 99.6%. Nilai tersebut merupakan nilai akurasi yang tertinggi dari semua siklus percobaan. Dengan demikian, percobaan yang dilakukan dapat memperbaiki akurasi prediksi relasi.

5.6 Pengembangan Alat Interaktif untuk *Crowdsourcing*

Alat yang digunakan untuk *crowdsourcing*, memiliki dua komponen utama, yaitu sebuah grafik yang akan memvisualisasikan entitas dengan relasi, dan tabel untuk mengecek atau mengubah relasi. Berikut Gambar 20 adalah tampilan alat yang sudah dikembangkan.



Gambar 20 Alat *Crowdsourcing*

Visualisasi bentuk grafik dibuat dengan memanfaatkan *library* D3.js[6] yang berbasis JavaScript. Partisipan *crowdsourcing* dapat mengecek pasangan entitas dan subtipe relasi untuk pasangan tersebut. Jika subtipe relasi dinilai tidak tepat, partisipan dapat mengubah secara langsung pada tabel yang disediakan. Setelah

[6] https://d3js.org/

koreksi selesai, hasil koreksi akan disimpan untuk digunakan dalam proses pembaruan model.

Data yang akan dikoreksi disimpan ke dalam format JSON untuk dibuat visualisasinya dan dimuat ke dalam tabel koreksi. Berikut Gambar 21 adalah contoh format data JSON tersebut. Struktur data JSON terbagi atas daftar semua entitas atau *nodes* dan daftar relasi atau *links*. Daftar entitas mencatat nama (jenis) entitas, label, dan ID. Daftar relasi mencatat asal dan tujuan relasi, serta subtipe relasinya.


```
{
  "nodes": [
    {
      "name": "",
      "label": "diet",
      "id": 11
    },
    {
      "name": "",
      "label": "serangga air",
      "id": 54
    },
    {
      "name": "",
      "label": "siput",
      "id": 57
    }
  ],
  "links": [
    {
      "source": 54,
      "target": 11,
      "type": "Subquantity-Of"
    },
    {
      "source": 57,
      "target": 11,
      "type": "Member-Of"
    }
  ]
}
```


Gambar 21 Contoh Data JSON

Kemudian data hasil koreksi akan disimpan ke dalam format CSV yang standar untuk memudahkan pengolahan selanjutnya. Demikian alat interaktif yang dibuat cukup sederhana namun dapat berfungsi sebagaimana diharapkan.

5.7 Evaluasi Kerangka Kerja

Percobaan telah dilakukan berdasarkan kerangka kerja yang dirancang. Dimulai dari penerapan *offline incremental learning* untuk membuat model awal, lalu koreksi hasil prediksi oleh partisipan *crowdsourcing*, kemudian pengolahan hasil *crowdsourcing* untuk membuat atau memperbarui model yang dihasilkan dari penerapan *online incremental learning*. Pada bagian ini akan diberikan evaluasi setelah kerangka kerja tersebut diterapkan.

Pada setiap siklus percobaan yang melibatkan *crowdsourcing*, peneliti juga menghitung nilai *precision*, *recall*, dan *F1 Score*. Berikut nilai-nilai tersebut pada Tabel 10. Nilai *precision*, *recall*, dan *F1 Score* dihitung secara *weighted* untuk seluruh data. Perhitungan ini mengambil rata-rata nilai untuk setiap label dengan memberikan pembobotan sesuai jumlah label yang benar. Berbeda dengan perhitungan lain, misalnya secara makro, di mana nilai rata-rata langsung diambil secara keseluruhan tanpa mempertimbangkan proporsi label.

Tabel 10 Nilai *Precision*, *Recall*, dan *F1 Score*

Percobaan	***Precision***	***Recall***	***F1 Score***
	0.87230908	0.87331918	0.87172440
0	0.89010103	0.88605803	0.88685501
1	0.89926822	0.89525832	0.89588022
2	0.89663699	0.89313517	0.89375848
3	0.90101937	0.89667374	0.89728498
4	0.89663699	0.89313517	0.89375848
5	0.90186440	0.89738146	0.89797315

Nilai *precision*, *recall*, dan *F1 Score* pada *online learning* secara umum berada dalam rentang 88% sampai dengan 89%. Nilai tersebut menunjukkan bahwa performa model yang dibuat secara umum cukup baik, namun perubahannya tidak terlalu signifikan. Hasil akhir model memiliki nilai *precision*, *recall*, dan *F1 Score* yang tertinggi, sehingga dapat disimpulkan bahwa pada akhir proses *online learning*, model berhasil diperbaiki menjadi lebih akurat.

Peneliti juga menghitung nilai *precision*, *recall*, dan *F1 Score* pada setiap percobaan, di mana koreksi dilakukan oleh *expert*. Nilai-nilai tersebut juga dihitung secara *weighted* dan data yang digunakan pada setiap siklus dipastikan sama. Berikut hasil perhitungan pada Tabel 11. Dari hasil tersebut, ditemukan bahwa pengaruh koreksi *expert* dibandingkan dengan pengaruh koreksi partisipan *crowdsourcing* pada model yang digunakan untuk memprediksi subtipe relasi tidak berbeda jauh. Dengan demikian, koreksi dari partisipan *crowdsourcing* dapat digunakan sebagai alternatif koreksi dari *expert* karena pengaruhnya sama baiknya.

Tabel 11 Nilai *Precision*, *Recall*, dan *F1 Score Expert*

Percobaan	*Precision*	*Recall*	*F1 Score*
	0.87230908	0.87331918	0.87172440
0	0.89010103	0.88605803	0.88685501
1	0.89663699	0.89313517	0.89375848
2	0.89663699	0.89313517	0.89375848
3	0.89663699	0.89313517	0.89375848
4	0.89663699	0.89313517	0.89375848
5	0.90186440	0.89738146	0.89797315

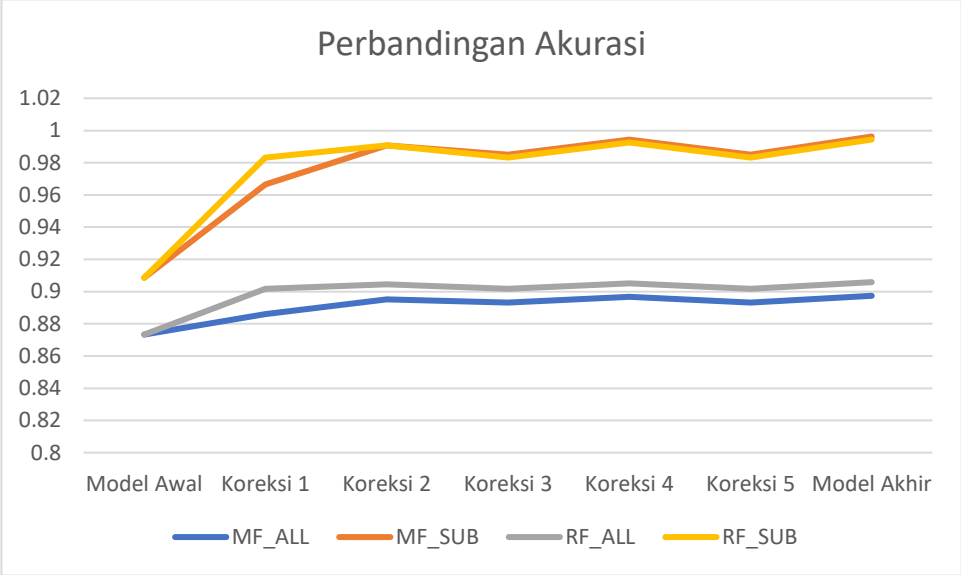
Berkaitan dengan pengaruh *crowdsourcing* untuk mengoreksi *extra-logical error*, peneliti melakukan eksperimen serupa, namun model yang digunakan tidak diubah menjadi model *online*. Pada percobaan ini, peneliti hanya menguji apakah koreksi dari partisipan *crowdsourcing* benar-benar mampu memperbaiki performa model. Berikut hasil akurasi model *offline*, setelah data yang digunakan dikoreksi melalui *crowdsourcing*, pada Tabel 12.

Tabel 12 Perhitungan Akurasi Model *Offline*

Percobaan	Akurasi Keseluruhan	Akurasi Subtipe
	0.87331918	0.90858209
0	0.90162774	0.98320896
1	0.90445860	0.99067164
2	0.90162774	0.98320896
3	0.90516631	0.99253731
4	0.90162774	0.98320896
5	0.90587403	0.99440299

Dengan menggunakan model Random Forest yang dibangun pada awal percobaan, data yang baru dari hasil *crowdsourcing* mampu meningkatkan akurasi model. Akurasi model untuk keseluruhan data meningkat dari 87.3% menjadi 90.6%, sementara akurasi model untuk data subtipe relasi Subquantity-Of meningkat dari 90.9% menjadi 99.4%. Dari peningkatan ini, proses *crowdsourcing* yang divalidasi dengan perhitungan *inter-annotator agreement* disimpulkan mampu memperbaiki performa model, dengan cara memperbaiki data yang digunakan untuk membentuk model.

Peneliti membandingkan hasil akurasi model *offline* dan model *online* pada percobaan yang melibatkan *crowdsourcing*. Model awal yang digunakan untuk prediksi relasi yang pertama kali adalah sama, yaitu model *offline*. Berikut grafik perbandingan akurasi model pada Gambar 22. Sesuai dengan algoritma yang digunakan, pada Gambar 22, MF adalah untuk Mondrian Forest dan RF adalah untuk Random Forest. Akurasi untuk data keseluruhan ditandai dengan ALL dan akurasi untuk data sebuah subtipe relasi ditandai dengan SUB.



Gambar 22 Perbandingan Akurasi

Ditemukan bahwa ternyata akurasi model *offline* yang menggunakan Random Forest untuk data keseluruhan setara dengan akurasi model *online* yang menggunakan Mondrian Forest. Namun akurasi model *offline* untuk data sebuah subtipe relasi lebih tinggi dibandingkan dengan akurasi model *online*. Dengan demikian, disimpulkan bahwa performa model *offline* Random Forest lebih baik daripada performa model *online* Mondrian Forest. Dari hasil ini, dapat disimpulkan juga bahwa *online incremental learning*, dalam kasus perbaikan proses ekstraksi relasi *meronymy*, tidak mampu memberikan hasil yang lebih baik daripada *offline incremental learning* dari sudut pandang akurasi.

BAB 5
KESIMPULAN DAN SARAN

Pada bab ini akan dijabarkan beberapa kesimpulan dari penelitan yang dilakukan. Kemudian diberikan juga beberapa saran terkait dengan kesimpulan, guna memperbaiki penelitian-penelitian pada bidang yang sama di kemudian hari.

5.1 Kesimpulan

Beberapa kesimpulan setelah menjalankan penelitian ini mencakup hasil dari penelitian, kelebihan, serta kekurangan atau kesulitan dalam penelitian ini. Berikut penjelasan kesimpulan penelitian.

1. Berdasarkan percobaan yang telah dilakukan, *crowdsourcing* mampu memperbaiki prediksi relasi *meronymy* untuk sebuah pasangan entitas dengan cara memperbaiki data yang digunakan untuk memperbarui model. Perbaikan data yang dihasilkan dari *crowdsourcing* dapat divalidasi menggunakan perhitungan *inter-annotator agreement* sehingga koreksi yang disimpan adalah koreksi yang sudah dipastikan valid.
2. Perbaikan model yang ditimbulkan dari koreksi partisipan *crowdsourcing* ditemukan setara dengan perbaikan model yang ditimbulkan dari koreksi *expert*. Dengan demikian, *crowdsourcing* dapat menjadi alternatif untuk memvalidasi hasil prediksi mesin.
3. Performa model dari *offline incremental learning* ternyata sedikit lebih baik daripada performa model dari *online incremental learning*. Hal ini dibuktikan dari tingkat akurasi yang model *offline* yang ternyata lebih tinggi daripada model *online*.
4. Peneliti menemukan dua jenis algoritma untuk *online learning* dengan basis Random Forest, yaitu Online Random Forest[7] dan Mondrian Forest[8]. Kedua implementasi algoritma ini telah dicoba digunakan, namun pembaruan

[7] https://github.com/amirsaffari/online-random-forests
[8] https://github.com/balajiln/mondrianforest

terakhir dilakukan sekitar 5 tahun yang lalu. Beberapa dependensi Online Random Forest ditemukan bermasalah, sehingga penelitian dilanjutkan dengan menggunakan Mondrian Forest dengan *library* scikit-garden.

5. Metode *online learning* juga dapat didefinisikan secara berbeda: *online* dalam arti model yang ada diperbarui pada saat ada fitur baru dari data lama atau *online* dalam arti model yang ada diperbarui pada saat ada data baru. Pada penelitian ini, definisi yang digunakan adalah definisi kedua. Model yang digunakan diperbarui pada saat ada koreksi subtipe relasi atau dengan kata lain, saat ada data subtipe relasi yang baru.
6. Pada tahap *crowdsourcing*, fokus penelitian hanya pada koreksi subtipe relasi tanpa memeriksa arah relasi. Namun ditemukan juga bahwa terdapat arah relasi yang kurang tepat dan belum dapat dicakup dalam penelitian ini. Arah relasi secara tidak langsung juga mempengaruhi kemudahan anotator dalam memahami suatu subtipe relasi yang berkaitan.
7. Tantangan yang dihadapi peserta *crowdsourcing* adalah kesulitan memahami perbedaan antara masing-masing subtipe relasi karena konsep *meronymy* umumnya dianggap sebagai satu jenis saja. Terdapat risiko anotasi yang kurang tepat, namun dengan menggunakan *inter-annotator agreement*, hal tersebut dapat diminimalisir.
8. Nilai *inter-annotator agreement* secara keseluruhan dapat berbanding terbalik dengan nilai *agreement* pada masing-masing relasi. Oleh sebab itu, berdasarkan kebutuhannya, peneliti dapat memilih nilai mana yang akan dipakai. Pada penelitian ini, nilai yang dipakai adalah nilai *agreeement* untuk masing-masing relasi, karena peneliti hanya akan mengambil relasi dengan nilai *agreeement* di atas 0.7 untuk dipakai memperbarui model.

5.2 Saran

Dengan mengamati kekurangan dan kesulitan pada penelitian ini, berikut beberapa saran untuk meningkatkan hasil penelitian pada area yang sama di kemudian hari. Peneliti juga mencantumkan potensi pengembangan yang untuk saat ini belum dapat dicakup pada penelitian ini.

1. Penelitian selanjutnya dapat dilakukan dengan pengaturan yang sepenuhnya *online*, yaitu dengan mencakup kedua kemungkinan pembaruan model. Model dapat diperbarui tanpa mengulang seluruh *training* pada saat ditambahkan fitur baru atau data baru.
2. Koreksi yang dilakukan dapat mencakup juga arah relasi sehingga hasil ontologi yang berupa *directed graph* menjadi lebih akurat. Perbaikan ini tentunya juga mencakup perbaikan alat yang digunakan sebagai medium interaktif untuk *crowdsourcing*.
3. Validasi koreksi dapat ditambahkan dengan menggunakan standar lain sehingga pada kondisi di mana ada jauh lebih banyak partisipan *crowdsourcing*, sebuah koreksi dapat divalidasi dengan lebih teliti. Sebagai contoh, peneliti dapat menambahkan pembobotan berdasarkan kemampuan atau pemahaman partisipan terhadap koreksi yang diminta. Kemampuan partisipan ini dapat dinilai dengan memberikan beberapa contoh data untuk dikoreksi dahulu.
4. Percobaan dapat dilakukan dengan menggunakan algoritma *learning* yang lainnya karena setiap algoritma memiliki karakteristik yang membedakan performanya pada berbagai macam kasus. Dalam hal ini, algoritma Random Forest dan Mondrian Forest ditemukan sudah cukup baik untuk memprediksi relasi *meronymy*.

(Halaman sengaja dikosongkan)

DAFTAR PUSTAKA

Artstein, R. (2017). Inter-annotator Agreement. In *Handbook of Linguistic Annotation* (pp. 297-313). Dordrecht: Springer.

Bakhta, M. N., El-Diraby, T. E., & Hossainic, M. (2018). Game-based crowdsourcing to support collaborative customization of the definition of sustainability. *Advanced Engineering Informatics, 38*, 501-513.

Bordes, A., Ertekin, S., Weston, J., & Bottou, L. (2005). Fast Kernel Classifiers with Online and Active Learning. *The Journal of Machine Learning Research, 6*, 1579-1619.

Buitelaar, P., Olejnik, D., & Sintek, M. (2004). A protege plug-in for ontology extraction from text based on linguistic analysis. *Proceedings of the 1st European Semantic Web Symposium, ESWS 2004* (pp. 31–44). Berlin, Heidelberg: Springer-Verlag.

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, ACL '99* (pp. 120–126). Stroudsburg,: Association for Computational Linguistics.

Cauwenberghs, G., & Poggio, T. (2000). Incremental and decremental support vector machine learning. *Proceedings of the 13th International Conference on Neural Information Processing Systems* (pp. 388-394). Denver, CO: MIT Press Cambridge, MA, USA.

Cederberg, S., & Widdows, D. (2003). Using lsa and noun coordination information to improve the precision and recall of automatic hyponymy extraction. *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, CoNLL-2003* (pp. 111–118). Stroudsburg, PA, USA: Association for Computational Linguistics.

Chomsky, A. N. (1957). *Syntactic Structures.* Walter de Gruyter.

Clarke, D. (2009). Context-theoretic semantics for natural language: an overview. *Proceedings of the Workshop on Geometrical Models of Natural Language*

Semantics, GEMS '09 (pp. 112–119). Stroudsburg, PA, USA: Association for Computational Linguistics.

Cutler, A., Cutler, D. R., & Stevens, J. R. (2011). Random Forests. In *Ensemble Machine Learning* (pp. 157-176). Springer.

Ditzler, G., Roveri, M., Alippi, C., & Polikar, R. (2015, November). Learning in Nonstationary Environments: A Survey. *IEEE Computational Intelligence Magazine, 10*(4), 12-25.

Espinosa-Anke, L., Saggion, H., & Ronzano, F. (2015). Taln-upf: taxonomy learning exploiting crf-based hypernym extraction on encyclopedic definitions. *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval 2015* (pp. 949–954). Stroudsburg, PA, USA: Association for Computational Linguistics.

Fellbaum, C. (1998). WordNet:An Electronic Lexical Database. In *MIT Press.*

Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., & Antoniou, G. (2008). Ontology change: classification and survey. *The Knowledge Engineering Review, 23*, 117-152.

Gali, A., Chen, C. X., Claypool, K. T., & Uceda-Sosa, R. (2004). From Ontology to Relational Databases. *ER 2004: Conceptual Modeling for Advanced Application Domains* (pp. 278-289). Springer.

Geffet, M., & Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL'05* (pp. 107–114). Stroudsburg, PA, USA: Association for Computational Linguistics.

Granada, R., Vieira, R., Trojahn, C., & Aussenac-Gilles, N. (2018, November 8). Evaluating the Complementarity of Taxonomic Relation Extraction Methods Across Different Languages. *arXiv*.

Harris, Z. S. (1954). Distributional structure. *Word, 10*(23), 146-162.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th conference on Computational linguistics - Volume 2, COLING '92* (pp. 539–545). Stroudsburg, PA, USA: Association for Computational Linguistics.

Hearst, M. A. (1998). Automated discovery of wordnet relations. In *WordNet: An electronic lexical database and some of its applications* (pp. 131–153).

Jamgade, A. N., & Karale, S. J. (2015). Ontology based information retrieval system for Academic Library. *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. Coimbatore, India: Institute of Electrical and Electronics Engineers.

Kaushik, N., & Chatterjee, N. (2018). Automatic relationship extraction from agricultural text for ontology construction. *Information Processing in Agriculture, 5*, 60-73.

Knijff, J. D., Frasincar, F., & Hogenboom, F. (2013, January). Domain taxonomy learning from text: The subsumption method versus hierarchical clustering. *Data & Knowledge Engineering, 83*, 54–69.

Kotlerman, L., Dagan, I., Szpektor, I., & Zhitomirsky-geffet, M. (2010, October). Directional distributional similarity for lexical inference. *Natural Language Engineering, 16*(4), 359-389.

Kumar, G. S., & Zayaraz, G. (2015). Concept relation extraction using Naive Bayes classifier for ontology-based question answering systems. *Journal of King Saud University – Computer and Information Sciences, 27*, 13-24.

Lakel, K., & Bendella, F. (2015). Dynamic Evaluation of Ontologies. *Procedia Computer Science, 73*, 16-23.

Lakshminarayanan, B., Roy, D. M., & Teh, Y. W. (2014, June). Mondrian Forests: Efficient Online Random Forests. *Advances in neural information processing systems 4*.

Lenci, A., & Benotto, G. (2012). Identifying hypernyms in distributional semantic spaces. *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, SemEval '12* (pp. 75–79). Stroudsburg, PA, USA: Association for Computational Linguistics.

Li, G., Wang, J., Zheng, Y., & Franklin, M. J. (2016, September 1). Crowdsourced Data Management: A Survey. *IEEE Transactions on Knowledge and Data Engineering, 28*(9), 2296 - 2319.

Liang, N.-y., Huang, G.-b., Saratchandran, P., & Sundararajan, N. (2006, November). A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Transactions on Neural Networks, 17*(6), 1411 - 1423.

Liu, X., Song, Y., Liu, S., & Wang, H. (2012). Automatic taxonomy construction from keywords. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'12* (pp. 1433–1441). New York, NY, USA: Association for Computing Machinery.

Lopes, L. (2012). *Extração automática de conceitos a partir de textos em língua portuguesa*. Porto Alegre, Brazil: PhD thesis, PUCRS University - Computer Science Department.

Losing, V., Hammera, B., & Wersing, H. (2018, January 31). Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing, 275*, 1261-1274.

Loupe, G. (2014). *Understanding Random Forests.* Belgium: University of Liege.

M.French, R. (1999, April 1). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences, 3*(4), 128-135.

Medin, D. L., Goldstone, R. L., & Gentner, D. (1990). Similarity involving attributes and relations: judgments of similarity and difference are not inverses. *Psychological Science, 1*(1), 64-69.

Meng, R., Chen, L., Tong, Y., & Zhang, C. (2017, May 1). Knowledge Base Semantic Integration Using Crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering, 29*(5), 1087-1100.

Mortensen, J. M. (2013). Crowdsourcing Ontology Verification. *ISWC 2013: The Semantic Web – ISWC 2013* (pp. 448-455). Springer, Berlin, Heidelberg.

Mortensen, J. M., Telis, N., Hughey, J. J., Fan-Minogue, H., Auken, K. V., Dumontier, M., & Musen, M. A. (2016). Is the crowd better as an assistant

or a replacement in ontology engineering? An exploration through the lens of the Gene Ontology. *Journal of Biomedical Informatics, 60*, 199-209.

Nicola, A. D., Missikoff, M., & Roberto, N. (2009, April). A Software Engineering Approach to Ontology Building. *Information Systems, 34*(2), 258-275.

Njike-Fotzo, H., & Gallinari, P. (20014). Learning generalization/specialization relations between concepts: application for automatically building thematic document hierarchies. *Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval, RIAO-2004* (pp. 143–155). Paris, France: Le Centre de Hautes Etudes Internationales d'Informatique Documentaire.

Pantel, P., & Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, ACL-44* (pp. 113–120). Stroudsburg, PA,: Association for Computational Linguistics.

Phi, V.-T., & Matsumoto, Y. (2016). Integrating Word Embedding Offsets into the Espresso System for Part-Whole Relation. *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Oral Papers* (pp. 173–181). Seoul, South Korea: ACL Anthology.

Phi, V.-T., Santoso, J., Shimbo, M., & Matsumoto, Y. (2018). Ranking-Based Automatic Seed Selection and Noise Reduction for Weakly Supervised Relation Extraction. *56th Annual Meeting of the Association for Computational Linguistics* (pp. 89-95). Melbourne, Australia: Association for Computational Linguistics.

Pocostales, J. (2016). Nuig-unlp at semeval-2016 task 13: A simple word embedding-based approach for taxonomy extraction. *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval 2016* (pp. 1298–1302). Stroudsburg, PA, USA: Association for Computational Linguistics.

Polikar, R., Upda, L., Upda, S., & Honavar, V. (2001, November). Learn++: an incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 31*(4), 497 - 508.

Ponzetto, S. P., & Strube, M. (2011, June). Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence, 175*(9), 1737-1756.

Powers, D. M. (2011). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Radford, A. (1997). *Syntax: A minimalist introduction.* Cambridge University Press.

Rashid, M. R., Rizzo, G., Torchiano, M., Mihindukulasooriya, N., Corcho, O., & García-Castro, R. (2019). Completeness and consistency analysis for evolving knowledge bases. *Web Semantics: Science, Services and Agents on the World Wide Web, 54*, 48-71.

Saffari, A., Leistner, C., Santner, J., Godec, M., & Bischof, H. (2009). On-line Random Forests. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops.* Kyoto, Japan: Institute of Electrical and Electronics Engineers (IEEE).

Sakurai, Y., Matsuda, M., Shinoda, M., & Oyama, S. (2017). Crowdsourcing Mechanism Design. *International Conference on Principles and Practice of Multi-Agent Systems: PRIMA 2017* (pp. 495-503). Springer.

Sanderson, M., & Croft, B. (1999.). Deriving concept hierarchies from text. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99* (pp. 206–213). New York, NY, USA: Association for Computing Machinery.

Santoso, J., Nugraha, J. N., Yuniarno, E. M., & Hariadi, M. (2015). Noun ontology generation from Wikipedia article using Map Reduce with pattern based approach. *2015 International Seminar on Intelligent Technology and Its*

Applications (ISITIA). Surabaya, Indonesia: Institute of Electrical and Electronics Engineers (IEEE).

Santus, E., Lenci, A., Lu, Q., & Walde, S. S. (2014). Chasing hypernyms in vector spaces with entropy. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2, EACL 2014* (pp. 38–42). Association for Computational Linguistics.

Sato, A., & Yamada, K. (1995). Generalized learning vector quantization. *Proceedings of the 8th International Conference on Neural Information Processing Systems* (pp. 423-429). Denver, Colorado: MIT Press Cambridge, MA, USA.

Sawsaa, A., & Lu, J. (2010). Ontocop: A Virtual Community of Practice to Create Ontology of Information Science (IS). *Proceedings of the 2010 International Conference on Internet Computing, ICOMP 2010.* Las Vegas Nevada, USA.

Shah, N., & Zhou, D. (2016, September 16). Double or Nothing: Multiplicative Incentive Mechanisms for Crowdsourcing. *Journal of Machine Learning Research, 17*, 1-52.

Sintek, M., Buitelaar, P., & Olejnik, D. (2004). A formalization of ontology learning from text. *Proceedings of the 3rd International Workshop on Evaluation of Ontology-based Tools, EON 2004.* Hiroshima, Japan.

Szpektor, I., & Dagan, I. (2008). Learning entailment rules for unary templates. *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1, COLING '08* (pp. 849–856). Stroudsburg, PA, USA: Association for Computational Linguistics.

Tan, L., Gupta, R., & Genabith, J. v. (2015). Usaar-wlv: hypernym generation with deep neural nets. *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval 2015* (pp. 932–937). Stroudsburg, PA, USA: Association for Computational Linguistics.

Tao, T.-y., & Zhao, M. (2012, May). An Ontology-Based Information Retrieval Model for Vegetables E-Commerce. *Journal of Integrative Agriculture, 11*(5), 800-807.

Taye, M. M. (2010, June). Understanding Semantic Web and Ontologies: Theory and Applications. *Journal of Computing, 2*(6).

Thuan, N. H., Antunes, P., & Johnstone, D. (2016). Pilot experiments on a designed crowdsourcing decision tool. *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD).* Nanchang, China: Institute of Electrical and Electronics Engineers (IEEE).

Velardi, P., Missikoff, M., & Basili, R. (2001). Identification of relevant terms to support the construction of domain ontologies. *Proceedings of the workshop on Human Language Technology and Knowledge Management - Volume 2001, HLTKM '01* (pp. 5:1–5:8). Stroudsburg, PA, USA: Association for Computational Linguistics.

Verma, A., Kaur, A., & Kaur, K. (2017, June). Theory of Ontological Engineering. *International Journal on Computer Science and Engineering (IJCSE), 9*(06), 397-403.

Vossen, P. (2001). Extending, trimming and fusing wordnet for technical documents. *Proceedings of the Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics, NAACL 2001*, (pp. 125–131). Association for Computational Linguistics.

Wang, J., Zhao, P., Hoi, S., & Jin, R. (2014, March). Online Feature Selection and Its Applications. *IEEE Transactions on Knowledge and Data Engineering, 26*(3), 698-710.

Weeds, J., Weir, D., & McCarthy, D. (2004). Characterising measures of lexical distributional similarity. *Proceedings of the 20th International Conference on Computational Linguistics, COLING '04* (pp. 1015–1021). Stroudsburg, PA, USA: Association for Computational Linguistics.

Xinhua, L., Xutang, Z., & Zhongkai, L. (2012). A Domain Ontology-based Information Retrieval Approach for Technique Preparation. *Physics Procedia, 25*, 1582-1588.

Yang, H., & Callan, J. (2008). Human-Guided Ontology Learning. *Second Workshop on Human-Computer Interaction and Information Retrieval (HCIR2008).* Redmond, WA: Microsoft Research.

Yang, H., & Callan, J. (2009). A Metric-based Framework for Automatic Taxonomy Induction. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 271–279). Singapore: Association for Computational Linguistics.

Yang, H., & Callan, J. (2009). OntoCop: Constructing Ontologies for Public Comments. *Intelligent Systems, IEEE (IEEE INTELL SYST), 24*(5).

Zettlemoyer, L. (2013). Relation Extraction. Department of Computer Science & Engineering, University of Washington.

Zhang, H. (2004). The Optimality of Naive Bayes. *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference.* Miami Beach, Florida, USA.

Zhang, X., Shangguan, L., & Yuan, Y. (2016, June 15). A Crowd Wisdom Management Framework for Crowdsourcing Systems. *IEEE Access, 4*, 9764 - 9774.

(Halaman sengaja dikosongkan)

BIODATA PENULIS

Eunike Andriani Kardinata lahir di Surabaya, pada tanggal 7 Mei 1994. Penulis telah menempuh pendidikan formal di SD Kristen Petra 1 Tegalsari Surabaya dan SMP Kristen Petra 3 Surabaya. Kemudian penulis berkesempatan melanjutkan pendidikan ke Cedar Girls' Secondary School dan Nanyang Junior College di Singapura. Pada tahun 2014, penulis meneruskan pendidikan Strata-1 di Sekolah Tinggi Teknik Surabaya, dengan jurusan Sistem Informasi dan major pada bidang *Enterprise Information System*. Penulis lulus pada tahun 2018 dengan Tugas Akhir berjudul "Studi Pengkajian iDempiere Business Suite untuk Proses Pembelian, Pengelolaan Stok, dan Penjualan". Berdasarkan minat penulis, penulis bergabung ke dalam Laboratorium Akuisisi Data dan Diseminasi Informasi (ADDI) dan melakukan penelitian untuk tesis dengan topik pengembangan ontologi Bahasa Indonesia. Segenap masukan untuk penulis dapat disampaikan melalui e-mail eunike.kardinata@gmail.com.

(Halaman sengaja dikosongkan)