1. PENDAHULUAN

1.1 Latar Belakang

Pada klasifikasi teks multi-label atau *Multi Label Text Classification* (MLTC), sebuah teks dapat memiliki satu atau lebih label, berbeda dari klasifikasi teks *single-label* (SLTC) yang hanya memiliki satu label untuk masing-masing teks. MLTC adalah masalah di bidang NLP yang mempunyai banyak pengaplikasian di dunia nyata, diantaranya kategorisasi produk, artikel berita, jurnal penelitian, diagnosis penyakit, dll. (Huang et al., 2021). Beberapa tantangan yang dihadapi oleh MLTC adalah ketidakseimbangan dataset serta adanya korelasi antar-label (Wang & Liu, 2023). Meskipun berbagai model *deep learning* dan *machine learning* terbukti mampu mengatasi hal ini dan memberikan performa yang memuaskan, model-model ini membutuhkan data-data pelatihan yang telah dilabeli dalam jumlah besar untuk dapat mencapai performa yang optimal (Alfando & Hayami, 2023).

Dalam melakukan klasifikasi teks, proses untuk mengumpulkan data seringkali jauh lebih mudah dan murah dibandingkan dengan proses pelabelan data nya. Hal ini dikarenakan sebagian besar proses pelabelan saat ini masih membutuhkan tenaga kerja manusia dan waktu yang banyak. Rata-rata waktu yang dibutuhkan untuk melabeli sebuah data adalah 20-40 detik (Seifert et al., 2013). Selain itu, biaya yang dibutuhkan untuk proses pelabelan data juga tidak sedikit. Sebagai contoh, Hendrycks et al. (2021) menghabiskan biaya sebesar 2 juta dolar untuk biaya pelabelan data dengan bantuan pengacara pada *Contract Understanding Atticus Dataset* (CUAD) yang terdiri dari 500 kontrak. Di Indonesia, rata-rata gaji seorang *data annotator* profesional sendiri mencapai Rp 100.000 per jam (ERI Economic Research Institute, 2024). Oleh karena itu, dibutuhkan sebuah sistem yang dapat membuat proses pelabelan data ini menjadi lebih efisien, baik dari segi waktu maupun biaya tanpa mengorbankan performa dari model yang dihasilkan.

Telah terdapat berbagai cara untuk membuat proses pelabelan data menjadi lebih efisien, salah satunya adalah dengan memanfaatkan LLM seperti ChatGPT. Meskipun ChatGPT mampu memberikan label dengan tingkat akurasi yang tinggi, label yang dihasilkan masih memiliki banyak *noise*, yang ditandai dengan nilai *precision* dan *recall* yang rendah (Gielens & Sowula, 2024). Oleh karena itu, ChatGPT saat ini masih belum dapat menggantikan peran manusia dalam proses pelabelan data. Cara lain yang umum digunakan adalah *active learning*. *Active learning* adalah teknik pelatihan model *machine learning* yang bertujuan untuk mencari

subset terbaik dari keseluruhan data yang nantinya akan dilabeli oleh manusia. Dengan demikian, model *active learning* dapat memberikan performa yang optimal dengan lebih sedikit data sehingga menurunkan biaya dan waktu yang diperlukan untuk proses pelabelan data.

Berbeda dengan metode pelatihan tradisional atau *passive learning* yang menggunakan seluruh data pelatihan, proses *active learning* diawali dengan mengambil bagian kecil dari data yang belum dilabeli, misal 5% dari total data. Data ini kemudian akan dilabeli terlebih dahulu sebagai bahan awal untuk pelatihan model. Setelah pelatihan dengan data awal selesai, model kemudian akan melakukan seleksi terhadap data-data yang belum dilabeli (Rouzegar & Makrehchi, 2024). Proses seleksi data ini disebut sebagai *sampling* atau *query strategy*. Tiga jenis *query strategy* yang paling sering digunakan pada proses ini adalah *uncertainty sampling*, *diversity sampling,* dan *hybrid sampling* (Mosqueira-Rey et al., 2022). *Uncertainty sampling* berfokus pada memilih data-data yang paling informatif (eksploitasi). Sementara *diversity sampling* lebih berfokus pada memilih data-data yang berbeda dari data yang sudah ada (eksplorasi). *Hybrid sampling* menggabungkan *uncertainty* dan *diversity sampling*. Setelah data diseleksi, data-data baru ini akan diserahkan kepada *data annotator* untuk dilabeli dan ditambahkan ke dalam pelatihan. Model kemudian akan dilatih kembali dengan data pelatihan yang telah ditambah dengan data-data baru. Proses akan ini dilakukan secara iteratif hingga performa dari model mencapai tingkat tertentu atau *budget* yang disediakan untuk proses pelabelan data sudah habis (Liebenlito et al., 2024). Dengan teknik ini, proses pembuatan model menjadi lebih efisien karena model bisa mencapai performa yang hampir sama atau bahkan lebih tinggi dengan menggunakan lebih sedikit data, dibandingkan dengan metode pelatihan tradisional yang menggunakan seluruh data (*passive learning*) (Tran et al., 2018).

Telah terdapat beberapa penelitian yang mencoba memanfaatkan *active learning* ini untuk melakukan klasifikasi teks pada bahasa Indonesia. Salah satunya adalah penelitian yang dilakukan oleh (Liebenlito et al., 2024), dimana peneliti mencoba menggunakan *active learning* untuk melakukan analisis sentimen pada teks berbahasa Indonesia dari Twitter. *Query strategy* yang digunakan pada penelitian ini adalah *margin sampling* yang merupakan salah satu jenis dari *uncertainty sampling*. Penelitian ini menggunakan model Logistic Regression dan Random Forest pada 2 *dataset single-label* untuk melakukan pengujian. Hasil dari penelitian ini berhasil menunjukkan bahwa *active learning* mampu memberikan akurasi yang lebih tinggi dari *passive learning* dan *random sampling* dengan hanya menggunakan 35-55% data pada *dataset* pertama dan 56-90% data pada *dataset* kedua. Selain analisis sentimen, *active learning* juga pernah digunakan untuk deteksi ujaran kebencian (Abidin et al., 2021) dan klasifikasi emosi (Asa &

Hastuti, 2023) dari teks berbahasa Indonesia dimana hasil dari kedua penelitian ini juga menyimpulkan bahwa *active learning* dengan *uncertainty sampling* mampu memberikan hasil yang lebih baik daripada *random sampling*, bahkan pada beberapa kasus melampaui *passive learning*.

Meskipun begitu, sebagian besar penelitian *active learning* yang ada saat ini masih terlalu berfokus pada klasifikasi teks single-label (SLTC). Hal ini juga terlihat pada penelitian-penelitian sebelumnya yang menggunakan teks berbahasa Indonesia. Penelitian mengenai MLTC menggunakan *active learning* pernah dicoba oleh (Wang & Liu, 2023). Dalam penelitian ini, mereka juga berusaha mencari tahu apakah *query strategy* yang sering digunakan pada SLTC dapat diterapkan pada MLTC. Beberapa *query strategy* yang digunakan pada penelitian ini adalah *random sampling* yang digunakan sebagai *baseline*, kemudian Least Confidence (LC), Deep Bayesian Active Learning by Disagreement (BALD), Monte Carlo (MC) Dropout yang merupakan bagian dari *uncertainty sampling*. Selain itu, peneliti juga menggunakan beberapa teknik *diversity sampling* seperti Coreset dan K-Means. Eksperimen yang dilakukan pada 6 buah *dataset* berbahasa Inggris menunjukkan bahwa kondisi data yang tidak seimbang dapat mempengaruhi performa *query strategy*, dimana beberapa *query strategy* bahkan mendapatkan hasil yang lebih buruk dari *random sampling*. Meskipun begitu, performa dari *diversity sampling* terlihat lebih konsisten daripada teknik-teknik *uncertainty sampling*. Kekurangan utama dari penelitian ini adalah belum adanya penggunaan *query strategy* yang khusus dirancang untuk menangani masalah-masalah pada klasifikasi teks *multi-label*.

Penelitian yang dilakukan oleh (Tan et al., 2024) berusaha mengatasi hal ini dengan mengembangkan sebuah *query strategy* yang memanfaatkan *beta scoring* untuk mencari data yang paling informatif. Setelah itu, *clustering* dilakukan untuk memastikan data yang dipilih tetap beragam. *Query strategy* ini dinamakan BESRA. Hasil dari penelitian yang dilakukan terhadap 6 *dataset* berbahasa Inggris menunjukkan bahwa BESRA mampu memberikan nilai f1-mikro yang lebih tinggi dari beberapa *query strategy* lainnya, dimana dalam beberapa *dataset* selisih yang didapatkan mencapai 3%. Sayangnya, *query strategy* ini membutuhkan daya komputasi yang cukup tinggi, karena di dalam proses nya membutuhkan beberapa model (*ensemble*) untuk melakukan *beta scoring*. Pendekatan lain yang dilakukan oleh (Peng et al., 2024) menggabungkan *contrastive learning* dengan dua *query strategy* (CoMAL) untuk memastikan data yang dipilih tetap informatif dan beragam juga terbukti efektif untuk melakukan klasifikasi teks *multi-label*. Hasil pengujian yang dilakukan menunjukkan CoMAL

mampu memberikan nilai f1-mikro yang lebih tinggi daripada 9 *query strategy* lainnya pada 4 *dataset* yang berbeda.

Skripsi ini akan mencoba menggunakan *query strategy* yang telah dikembangkan oleh Tan et al. dan Peng et al. Pemilihan *query strategy* ini diharapkan dapat mengatasi *dataset* untuk klasifikasi *multi-label* yang seringkali tidak seimbang. Selain itu, skripsi ini juga akan menggunakan beberapa *query strategy* yang digunakan pada penelitian Wang & Liu, seperti Least Confidence, Monte Carlo Dropout, K-Means, Coreset, dan Random sebagai representasi dari *query strategy* yang berbasis *uncertainty* dan *diversity*. Eksperimen akan dilakukan pada beberapa *dataset* teks *multi-label* berbahasa Indonesia untuk menguji efektivitas dari *active learning* ini. Beberapa *dataset* yang akan digunakan antara lain CASA (Ilmania et al., 2018) dan HoASA (Azhar et al., 2019) yang berfokus pada *Aspect-Based Sentiment Analysis*, kemudian juga Netifier (Ahmadizzan, 2018) dan *Multi-label Hate Speech and Abusive Language Detection Dataset* (Ibrohim & Budi, 2019) yang berfokus untuk mendeteksi ujaran kebencian dan komentar *toxic* di sosial media. Terakhir, skripsi ini juga menggunakan *Doctor's Answer Text Dataset* (Juanita et al., 2023) yang berisi jawaban-jawaban dokter dari *website* alodokter.com dan kemudian dikategorikan menjadi 6 macam jawaban.

Model *deep learning* IndoBERT juga akan digunakan pada skripsi ini. IndoBERT dipilih karena model ini telah dilatih secara khusus pada teks berbahasa Indonesia sehingga model dapat menangkap makna dan konteks dari teks berbahasa Indonesia dengan lebih baik. Hal ini menjadi penting karena morfologi bahasa Indonesia yang lebih kompleks dari bahasa lainnya, seperti bahasa Inggris sehingga berpotensi mempengaruhi performa dari *active learning* (Wertz et al., 2023)(Denistia & Baayen, 2022). Selain itu, IndoBERT juga telah menunjukkan performa yang baik dalam berbagai tugas MLTC, seperti pada deteksi ujaran kebencian dimana IndoBERT berhasil memperoleh akurasi 81.38% (Alzahrani et al., 2024), jauh lebih baik daripada model *machine learning* tradisional seperti *Naive Bayes, SVM,* dan *Random Forest* yang memperoleh akurasi tertinggi di angka 66.12% (Ibrohim & Budi, 2019).

Penelitian yang dilakukan oleh Wang & Liu telah memberikan gambaran awal mengenai efektivitas *active learning* untuk klasifikasi teks multi-label, dengan menggunakan berbagai *query strategy* berbasis *uncertainty* dan *diversity*. Meskipun begitu, hasil yang didapat pada penelitian tersebut belum tentu sama apabila diterapkan pada bahasa Indonesia, mengingat karakteristik bahasa Indonesia yang berbeda dengan bahasa Inggris. Oleh karena itu, skripsi ini akan mengisi kekosongan penelitian yang ada pada bahasa Indonesia dengan menggunakan pendekatan yang serupa. Lebih dari itu, skripsi ini juga akan mencoba memanfaatkan 2 *query*

strategy hybrid yang menggabungkan *uncertainty* dan *diversity* serta model IndoBERT yang memang dirancang khusus untuk bahasa Indonesia, untuk menguji performa *active learning* dalam melakukan klasifikasi teks multi-label pada data berbahasa Indonesia. Terakhir, skripsi ini juga akan berusaha mencari tahu jumlah minimal data yang dibutuhkan untuk mencapai performa yang setara dengan *passive learning* sehingga dapat memberikan gambaran konkrit mengenai efektivitas yang dihasilkan oleh *active learning*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka rumusan masalahnya adalah:

1. Berapa jumlah minimal data yang dibutuhkan agar model *active learning* yang dibuat untuk klasifikasi teks *multi-label* dapat memberikan hasil yang serupa dengan *passive learning*?
2. *Query strategy* manakah yang dapat memberikan nilai f1-mikro paling baik pada data *validation* masing-masing *dataset*?

1.3 Tujuan Penelitian

Tujuan skripsi ini adalah mengurangi jumlah data yang harus dilabeli dalam klasifikasi teks multi-label berbahasa Indonesia dengan menggunakan *active learning*. *Active learning* digunakan untuk mencari subset terbaik dari data-data yang belum dilabeli sehingga model tetap dapat memberikan performa yang optimal dengan jumlah data yang lebih kecil.

1.4 Ruang Lingkup

Ruang lingkup pada skripsi ini dibatasi pada:

1. Bahasa pemrograman yang digunakan adalah Python.
2. Menggunakan *library* PyTorch dan HuggingFace untuk pembuatan model.
3. Menggunakan *library* matplotlib dan seaborn untuk visualisasi data.
4. Teks yang digunakan hanya dalam bahasa Indonesia.
5. Proses *preprocessing* data meliputi:
    - *Case folding*
    - *Cleansing*
    - *Slang word replacement*
    - *Tokenization*
6. Dataset yang akan digunakan meliputi:

- CASA (1080 data)
- HoASA (2854 data)
- Netifier (7773 data)
- *Multi-label Hate Speech and Abusive Language Detection Dataset* (13169 data)
- *Doctor's Answer Text Dataset* (500 data)

7. *Query strategy* yang akan digunakan meliputi:
    - BESRA
    - CoMAL
    - Least Confidence
    - MC Dropout
    - Coreset
    - K-Means
    - *Random sampling*
8. Model yang digunakan adalah IndoBERT dari *library* HuggingFace.
9. Untuk *passive learning* dan *active learning*, *dataset* akan dibagi menjadi 80% untuk *training* atau *pool* dan 20% untuk *testing*.
10. Untuk proses *active learning*, 5% data akan digunakan untuk data awal pelatihan.
11. Pada setiap iterasi *active learning*, jumlah data akan ditambahkan secara dinamik dengan *threshold* sebagai berikut:
    - Least Confidence: 90th *percentile uncertainties*
    - MC Dropout : 90th *percentile variance*
    - Coreset : 90th *percentile* jarak antar data
    - KMeans : 90th percentile jarak data terhadap pusat masing-masing *cluster*
    - BESRA : 90th *percentile* dari jarak data terhadap pusat masing-masing *cluster*
    - CoMAL : 90th *percentile* dari skor
12. Proses *active learning* akan terus dilakukan hingga semua data telah digunakan dengan 4 *checkpoint* di angka 50%, 60%, 70%, dan 80% data.
13. Pengujian akan dilakukan pada masing-masing *dataset* dan *query strategy* sebanyak 5 kali. Rata-rata dari 5 kali pengujian ini akan diambil untuk dibandingkan dengan *query strategy* lainnya.

14. *Passive learning* akan dilakukan secara mandiri dengan prosedur pelatihan model pada umumnya, yaitu dengan menggunakan semua data secara langsung (tanpa *active learning*). Performa dari *passive learning* akan digunakan sebagai pembanding dengan performa yang dihasilkan menggunakan *active learning*.
15. Performa model akan diukur menggunakan nilai F1-mikro.

1.5 Metodologi Penelitian

Langkah-langkah dalam pengerjaan skripsi:

1. Studi Literatur
    - 1.1. Teori mengenai klasifikasi teks multi-label
    - 1.2. Teori mengenai *active learning*
    - 1.3. Teori mengenai *query strategy* yang didalamnya meliputi:
        - *1.3.1. Uncertainty sampling*
        - *1.3.2. Diversity sampling*
        - *1.3.3. Hybrid sampling*
    - 1.4. Teori mengenai *preprocessing* teks yang di dalamnya meliputi:
        - 1.4.1. *Case Folding*
        - 1.4.2. *Cleaning*
        - 1.4.3. *Slang word replacement*
        - 1.4.4. *Tokenization*
    - 1.5. Teori mengenai arsitektur *Transformer*
2. Pembuatan program
    - 2.1. Melakukan *preprocessing* pada data dengan melakukan *case folding, cleansing, slang word replacement,* dan *tokenization.*
    - 2.2. Pembuatan berbagai algoritma *query strategy* yang akan digunakan untuk pengujian
    - 2.3. Penyesuaian model IndoBERT agar dapat menangani klasifikasi teks *multi-label.*
3. Pengujian dan analisis program
    - 3.1. Melakukan pelatihan pada model yang telah dibuat terhadap masing-masing *dataset* dan *query strategy*
    - 3.2. Melakukan analisa terhadap *output* program

3.3. Melakukan pengukuran performa program dengan melihat tingkat F1-mikro terhadap data pengujian

4. Pengambilan kesimpulan
   4.1. Merangkum hasil pengujian
   4.2. Membuat kesimpulan tentang hasil penelitian dan analisa yang telah dilakukan
5. Pembuatan laporan
   5.1. Membuat laporan dari hasil pengujian dan kesimpulan yang diperoleh

1.6 Sistematika Penulisan

Penulisan skripsi ini dibagi menjadi beberapa bab yaitu:

BAB I : PENDAHULUAN

Pada bab ini berisikan latar belakang, rumusan masalah, ruang lingkup, tujuan penelitian, dan metodologi penelitian yang akan digunakan dalam skripsi ini.

BAB II : LANDASAN TEORI

Bab ini berisikan teori-teori yang digunakan untuk menjadi referensi dalam pembuatan skripsi dan diterapkan dalam skripsi ini.

BAB III : ANALISA DAN DESAIN SISTEM

Bab ini membahas tentang analisis dan desain sistem yang akan dibuat.

BAB IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas tentang implementasi dan pengujian sistem yang telah dibuat.

BAB V : KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan dari penelitian yang telah dilakukan serta saran bagi penelitian selanjutnya.