

Semi-Supervised Learning pada Pelabelan dalam Klasifikasi Multi-Label Data Teks

Nimas Ayuningtyas¹, Wiyli Yustanti²

^{1,2} Sistem Informasi, Universitas Negeri Surabaya

¹nimas.20056@mhs.unesa.ac.id

²wiyliyustanti@unesa.ac.id

Abstrak— Penelitian ini berfokus pada implementasi dan evaluasi hasil pseudo-label dalam konteks pelabelan semi-supervised untuk klasifikasi data teks *multi-label*. Pendekatan ini memanfaatkan sebagian data yang telah berlabel untuk melatih model, yang kemudian digunakan untuk memberi label pada data yang belum terlabel. Hasil evaluasi menunjukkan peningkatan signifikan dalam efisiensi dan akurasi klasifikasi teks *multi-label*. Teknik *pseudo-labeling* berhasil dievaluasi, dengan model mampu memprediksi label dengan tingkat keakuratan rata-rata sebesar 89,28%, menunjukkan konsistensi yang tinggi dengan pilihan label oleh responden. Selain itu, kinerja model dalam mengelola dan mengklasifikasi data teks *multi-label* juga telah diuji dan terbukti. Penggunaan kombinasi algoritma *Random Forest* dengan teknik *Label Powerset* menghasilkan *F1-Score* yang sangat baik, dengan nilai 0,9944 untuk data training dan 0,9888 untuk data testing. Hasil ini menunjukkan bahwa model mampu mempertahankan kinerja yang konsisten dan unggul dalam berbagai kondisi pengujian. Dengan demikian, hasil penelitian ini memberikan kontribusi penting dalam pengembangan metode *semi-supervised Learning* untuk klasifikasi teks *multi-label*, yang memiliki potensi untuk diterapkan dalam berbagai domain seperti analisis sentimen, kategorisasi dokumen, dan identifikasi informasi dalam big data.

Kata Kunci— *Pseudo-labeling, Semi-supervised Learning, Multi-label Classification, Random Forest, Label Powerset.*

I. PENDAHULUAN

Teknik klasifikasi yang biasanya digunakan adalah dengan menempatkan data ke dalam satu label saja. Ini dapat dilakukan apabila data yang dikelompokkan hanya memiliki satu label [1]. Namun, dalam beberapa kasus seperti pada pengelompokkan kategori yang memerlukan label lebih dari satu, *single* label kurang relevan sehingga digunakan klasifikasi *multi-label*. Klasifikasi *multi-label* adalah jenis klasifikasi dimana setiap data *point* dapat memiliki lebih dari satu label atau kategori [2]. Klasifikasi *multi-label* bisa dilakukan dengan menggunakan berbagai algoritma klasifikasi, yang ditingkatkan dengan metode transformasi data dan ekstraksi fitur. Tentunya, klasifikasi *multi-label* dapat lebih relevan digunakan dibandingkan klasifikasi *single* label dalam beberapa kejadian. Masalah klasifikasi *multi-label* diatasi dengan menerapkan pendekatan *problem transformation*. Pendekatan ini mengubah masalah *multi-label* menjadi masalah *single* label sehingga dapat dipecahkan dengan algoritma pembelajaran *single* label. Pemberian label awal akan dilakukan berdasarkan kategori tertentu yang selanjutnya diubah menjadi sejumlah kolom dengan nilai biner [3].

Terdapat tantangan yang muncul dalam klasifikasi *multi-label* dan dapat menjadi penghambat. Tantangan-tantangan ini mencakup proses persiapan data, yaitu pelabelan pada data latih yang dilakukan secara manual. Pelabelan manual dilakukan dengan menganalisis data satu per satu karena setiap kalimat dapat memiliki beragam makna. Oleh karena itu, pelabelan manual lebih akurat. Namun, proses ini cukup memakan waktu [4]. Proses pelabelan atau anotasi kelas secara manual memerlukan keahlian pakar untuk menghasilkan dataset berlabel berkualitas tinggi. Tantangan dalam proses anotasi manual meliputi waktu yang lama, biaya tinggi, dan potensi bias akibat subyektivitas pakar dalam memahami kalimat atau teks [5].

Pada penelitian terdahulu menggunakan metode *problem transformation* dengan teknik *binary relevance* untuk menentukan *keyword multi-label* secara otomatis yang menggunakan metode BRkNN [6]. Penelitian tersebut memberikan hasil evaluasi 0,70. Penelitian lain berjudul "Implementasi Metode Klasifikasi *Multi-label* Untuk Kategorisasi Materi Pembelajaran Secara Otomatis" juga menggunakan metode *problem transformation binary relevance* pada penelitiannya dan menghasilkan akurasi tertinggi 0,933. Namun, kedua penelitian tersebut masih menggunakan teknik pelabelan manual. Penelitian dengan pelabelan otomatis yang dilakukan oleh Siti Khomsah dan Agus Sasmito Aribowo pada tahun 2022 menggunakan *semi-supervised annotation* (SSA) sebagai metode untuk mengatasi sulitnya proses pelabelan data. Proses pelabelan atau anotasi menggunakan *semi-supervised annotation* (SSA) akan membangun model dari sebagian dataset berlabel. Lalu, dataset yang belum berlabel akan diberi label atau dianotasi oleh model yang sudah terbentuk. Penelitian tersebut menghasilkan nilai akurasi 0,82. Namun, penelitian tersebut menggunakan *single* label. Masih diperlukan penelitian lebih lanjut tentang pengembangan metode yang dapat digunakan pada *multi-label classification* agar memudahkan pekerjaan labeling dan mendapatkan hasil evaluasi yang lebih tinggi [7].

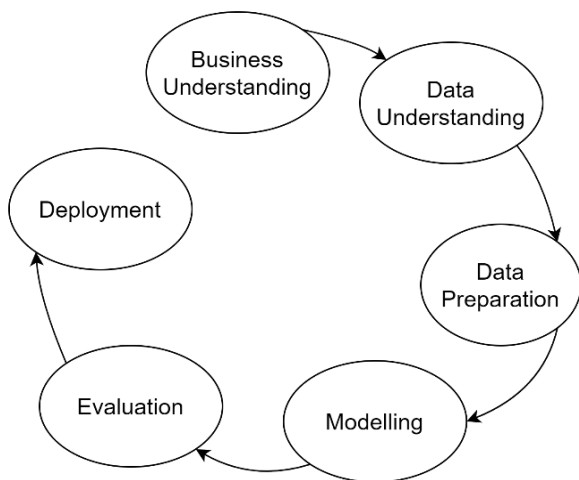
Klasifikasi kategorikal menggunakan kategori sebagai acuan untuk mengelompokkan objek, dengan tujuan utama mengklasifikasikan observasi ke dalam kelompok yang sesuai [8]. Di sisi lain, klasifikasi teks *multi-label* melibatkan proses mengelompokkan teks ke dalam satu atau lebih kategori, yang sering memberikan hasil yang lebih informatif daripada klasifikasi label tunggal [9]. Beberapa algoritma yang digunakan untuk klasifikasi *multi-label* termasuk *Label Powerset, Classifier Chain*, dan *Binary relevance*, dimana *Label Powerset* mengubah dataset *multi-label* menjadi dataset

multi-class. Dengan menggunakan transformasi masalah dengan *Label Powerset* [10], dapat diperoleh pendekatan yang lebih kompleks dalam klasifikasi teks. Selain itu, *Semi-supervised Learning* adalah metode pembelajaran mesin yang memanfaatkan berbagai jumlah data, termasuk data yang diberi label dan data yang tidak diberi label, sangat berguna ketika data yang tersedia memiliki keterbatasan dalam pelabelan yang akurat. Dengan demikian, SSL dapat menjadi alternatif yang lebih cepat, lebih murah, dan efektif untuk pelabelan dataset opini [11].

Dalam penelitian ini akan dibahas salah satu metode pada *semi-supervised Learning*, yaitu *pseudo-labeling* untuk memberikan label secara otomatis pada klasifikasi *multi-label* teks pada *Twitter (X)*. *Pseudo-labeling* merupakan salah satu metode yang digunakan untuk memberikan prediksi yang menghubungkan berbagai label pada sebuah *instance* [12]. Teknik dan algoritma yang digunakan dalam pembentukan model ini sangat cocok untuk mengidentifikasi kata kunci (label) dalam situasi di mana sebuah teks dapat diberi lebih dari satu label. Metode *pseudo-labeling* memisahkan dataset berlabel dengan dataset tidak berlabel. Pada dataset berlabel akan dibentuk model dengan beberapa algoritma yang kemudian model akan digunakan untuk memberikan label pada dataset yang belum berlabel. Tujuan penelitian ini adalah mengimplementasikan metode *pseudo-labeling* untuk *auto labeling* sehingga dapat membuat model yang bisa memprediksi label pada *instance* lainnya.

II. METODOLOGI PENELITIAN

A. Kerangka Penelitian



Gbr. 1 Metode CRISP-DM

Penelitian ini menggunakan metode CRISP-DM (*Cross-Industry Standard Process for Data Mining*), yang merupakan pendekatan metodologi terstruktur dan terorganisir untuk mengelola proyek *data Mining* mulai dari tahap awal hingga akhir. Metode ini dirancang untuk menyediakan pedoman yang terstruktur dalam mengumpulkan data. Metode ini terdiri dari enam tahapan, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan

Deployment [13]. Alur metode CRISP-DM digambarkan pada gambar 1.

TABEL I
TAHAPAN CRISP-DM

| Tahap | Aktivitas | Hasil |
|-------------------------------|---|---|
| <i>Business Understanding</i> | Identifikasi masalah dan tujuan bisnis. | Tujuan bisnis dan perumusan masalah yang jelas. |
| <i>Data Understanding</i> | Pengumpulan data melalui <i>crawling</i> dari <i>Twitter (X)</i> menggunakan <i>API Key</i> dengan bahasa pemrograman <i>Python</i> . | Data mentah dalam format CSV. |
| <i>Data Preparation</i> | Pembersihan dan transformasi data. | Data siap untuk analisis. |
| <i>Modeling</i> | Pengembangan model <i>Semi-supervised Learning</i> untuk klasifikasi <i>multi-label</i> teks. | Model yang siap diuji. |
| <i>Evaluation</i> | Evaluasi performa model. | Hasil evaluasi model yang diukur dengan metrik tertentu. |
| <i>Deployment</i> | Implementasi model ke dalam lingkungan produksi dan penyampaian hasil. | Model diimplementasikan dan hasil diterapkan dalam operasional nyata. |

Berdasarkan pada tabel I, pada tahap *Business Understanding*, masalah dan tujuan bisnis diidentifikasi untuk merumuskan tujuan yang jelas. Selanjutnya, pada tahap *Data Understanding*, data dikumpulkan melalui *crawling* dari *Twitter (X)* menggunakan *API Key* dengan bahasa pemrograman *Python*, menghasilkan data mentah dalam format CSV. Tahap *Data Preparation* melibatkan pembersihan dan transformasi data agar siap untuk analisis. Pada tahap *Modeling*, model *Semi-supervised Learning* dikembangkan untuk klasifikasi *multi-label* teks, menghasilkan model yang siap diuji. Tahap *Evaluation* kemudian mengevaluasi performa model dengan metrik tertentu, dan akhirnya, pada tahap *Deployment*, model diimplementasikan ke dalam lingkungan produksi dan hasilnya diterapkan dalam operasional nyata.

B. Business Understanding

Tahap awal ini berfokus pada pemahaman tujuan data mining serta evaluasi semua kebutuhan terkait dari perspektif bisnis. Informasi tersebut kemudian diubah menjadi definisi masalah dalam aktivitas *data mining* dan perencanaan awal disusun untuk mencapai tujuan yang telah ditetapkan [14]. Langkah ini mencakup penentuan tujuan bisnis proyek, kebutuhan pihak-pihak terkait, serta perumusan masalah *data Mining* yang jelas. Dalam penelitian ini, fokusnya adalah mengotomatisasi pelabelan dalam klasifikasi *multi-label* pada data teks untuk mengurangi waktu dan tenaga yang

dibutuhkan dalam proses manual. *Semi-supervised Learning* digunakan untuk memecahkan masalah ini. Studi literatur dilakukan untuk mempelajari teori terkait seperti *Keyword text classification*, *multi-label problem transformation*, *Binary relevance*, *Classifier Chain*, *Label Powerset*, *Semi-supervised Learning*, dan *pseudo-labeling*.

C. Data Understanding

Tahap berikutnya dalam metode CRISP-DM adalah memahami kebutuhan data yang diperlukan untuk mencapai tujuan yang telah ditetapkan [15]. Tahap awal ini berfokus pada memahami tujuan *data mining* serta mengevaluasi semua kebutuhan terkait dari perspektif bisnis. Informasi ini kemudian diubah menjadi definisi masalah dalam kegiatan *data mining*, serta disusun rencana awal untuk mencapai tujuan yang ditetapkan. Pada tahap ini, data yang relevan dikumpulkan dan dipahami. Proses pengumpulan data dilakukan melalui *crawling* data dari platform *Twitter (X)* (sekarang dikenal sebagai X) menggunakan *API Key Twitter (X)* dengan bahasa pemrograman *Python*. Data yang diambil akan disimpan dalam *file CSV*. Proses *crawling* dilakukan dengan menggunakan *Google Colab* untuk eksekusi kode *Python* secara *cloud-based*, memungkinkan aksesibilitas dan kolaborasi yang lebih baik dalam pengumpulan data.

D. Data Preparation

Tahap *preprocessing*, juga dikenal sebagai pemrosesan data, melibatkan dua kegiatan utama: pembangunan dan pembersihan data. Pembangunan data melibatkan pengumpulan dari berbagai sumber dan penggabungannya menjadi satu dataset. Pembersihan data bertujuan untuk mengidentifikasi dan menangani data yang tidak lengkap, tidak akurat, atau tidak relevan. Tujuannya adalah memastikan kualitas data yang digunakan untuk analisis dan kesiapan untuk tahap selanjutnya, yaitu pemodelan data [16]. Data yang telah dikumpulkan kemudian disiapkan untuk analisis. Proses ini melibatkan pembersihan, transformasi, dan integrasi data untuk memastikan bahwa data dalam kondisi siap untuk tahap pemodelan. Teknik pembersihan data termasuk menghapus duplikat, mengatasi *missing values*, dan normalisasi data teks menggunakan teknik seperti *tokenization* dan *stemming*.

E. Modeling

Langkah awal adalah memilih teknik pemodelan, yang kemudian diikuti dengan pembuatan skenario pengujian untuk memvalidasi kualitas model tersebut [17]. Pada tahap pemodelan, dikembangkan model statistik atau algoritma *machine learning* yang sesuai untuk menghasilkan prediksi atau solusi yang sesuai dengan tujuan bisnis. Penelitian ini menggunakan metode *Semi-supervised Learning* untuk klasifikasi *multi-label* teks. Model *Random Forest*, *Support Vector Machine*, dan *K-Nearest Neighbor* dipilih karena kemampuannya dalam menangani data yang kompleks dan non-linear, sedangkan metode *Label Powerset*, *Binary relevance*, dan *Classifier Chain* digunakan untuk menangani keterkaitan antar label dalam *multi-label classification*.

F. Evaluation

Pada tahap ini, dilakukan analisis atau pengukuran untuk mengevaluasi keakuratan dari pemodelan yang telah dibuat. Evaluasi ini penting untuk memastikan bahwa model yang dikembangkan memberikan hasil yang sesuai dengan tujuan yang telah ditetapkan dan dapat diandalkan dalam penggunaan praktis [18]. Evaluasi model dilakukan untuk memastikan kinerja model optimal dan sesuai dengan kriteria keberhasilan yang telah ditetapkan. Hasil evaluasi digunakan untuk mengukur performa model sebelum diterapkan ke lingkungan produksi. Metrik evaluasi yang digunakan dalam penelitian ini termasuk *Accuracy* dan *F1-Score* untuk mengukur keakuratan model dalam klasifikasi *multi-label*.

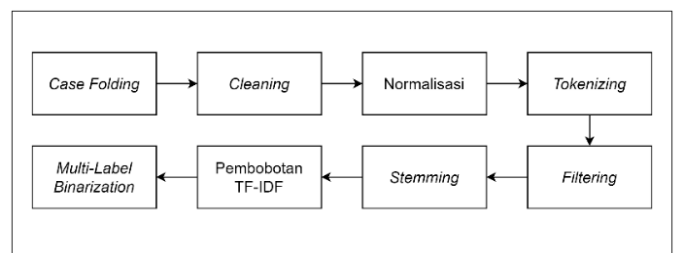
G. Deployment

Tahap terakhir adalah penerapan solusi yang ditemukan ke dalam lingkungan produksi dan penyampaian hasil kepada pihak-pihak terkait. Langkah ini memastikan bahwa model yang dikembangkan dapat diimplementasikan secara efektif dalam operasional nyata. Proses *Deployment* melibatkan integrasi model ke dalam sistem yang ada dan pemantauan performa model secara kontinu untuk memastikan stabilitas dan akurasi prediksi. Bagian *Deployment* melibatkan peningkatan model yang telah dikembangkan dengan mempertimbangkan aspek kemudahan penggunaan serta pemeliharannya di masa yang akan datang, sehingga memastikan model dapat diterapkan secara efisien dalam lingkungan praktis. Tahap ini penting untuk memastikan bahwa implementasi model dapat berjalan dengan lancar dan memberikan manfaat maksimal bagi pengguna [19].

III. HASIL DAN PEMBAHASAN

Penelitian ini membandingkan performa beberapa algoritma klasifikasi dan metode klasifikasi *multi-label* untuk menemukan kombinasi terbaik dalam pelabelan teks otomatis. Hasil penelitian menunjukkan bahwa kombinasi algoritma *Random Forest* dengan metode *Label Powerset* menghasilkan performa terbaik.

A. Prapemrosesan Data



Gbr. 2 Tahap Prapemrosesan Data

Pra-pemrosesan teks merupakan elemen krusial dalam algoritma *Text Mining*, yang bertujuan untuk mengelola teks sehingga dapat diubah menjadi data terstruktur yang sesuai dengan kebutuhan analisis lebih lanjut. Tahap pra-pemrosesan teks dalam proses klasifikasi berperan dalam meningkatkan akurasi klasifikasi data [20]. *Data Preparation* dalam

B. Performansi Model pada Data Training dan Testing

TABEL IV
HASIL EVALUASI SKENARIO

| 80:20 | | | |
|-----------------------|------------------------|----------------------|---------------------|
| Algoritma Multi-label | Model | F1 Score (Train set) | F1 Score (Test set) |
| Classifier Chain | Random Forest | 0.9944 | 0.9790 |
| Label Powerset | | 0.9944 | 0.9888 |
| Binary relevance | | 0.9951 | 0.9730 |
| Classifier Chain | Support Vector Machine | 0.9755 | 0.9302 |
| Label Powerset | | 0.9879 | 0.9283 |
| Binary relevance | | 0.9744 | 0.9274 |
| Classifier Chain | K-Nearest Neighbor | 0.9122 | 0.8828 |
| Label Powerset | | 0.9188 | 0.8664 |
| Binary relevance | | 0.9233 | 0.8783 |
| 75:25 | | | |
| Algoritma Multi-label | Model | F1 Score (Train set) | F1 Score (Test set) |
| Classifier Chain | Random Forest | 0.9944 | 0.9684 |
| Label Powerset | | 0.9944 | 0.9840 |
| Binary relevance | | 0.9951 | 0.9644 |
| Classifier Chain | Support Vector Machine | 0.9765 | 0.9368 |
| Label Powerset | | 0.9888 | 0.9335 |
| Binary relevance | | 0.9747 | 0.9318 |
| Classifier Chain | K-Nearest Neighbor | 0.9160 | 0.8771 |
| Label Powerset | | 0.9161 | 0.8655 |
| Binary relevance | | 0.9215 | 0.8789 |
| 60:40 | | | |
| Algoritma Multi-label | Model | F1 Score (Train set) | F1 Score (Test set) |
| Classifier Chain | Random Forest | 0.9947 | 0.9653 |
| Label Powerset | | 0.9947 | 0.9761 |
| Binary relevance | | 0.9955 | 0.9646 |
| Classifier Chain | Support Vector Machine | 0.9765 | 0.9294 |
| Label Powerset | | 0.9918 | 0.9349 |
| Binary relevance | | 0.9748 | 0.9330 |
| Classifier Chain | K-Nearest Neighbor | 0.9167 | 0.8692 |
| Label Powerset | | 0.9133 | 0.8522 |
| Binary relevance | | 0.9186 | 0.8741 |
| 50:50 | | | |
| Algoritma Multi-label | Model | F1 Score (Train set) | F1 Score (Test set) |
| Classifier Chain | Random Forest | 0.9838 | 0.7961 |
| Label Powerset | | 0.9838 | 0.9061 |
| Binary relevance | | 0.9805 | 0.7346 |

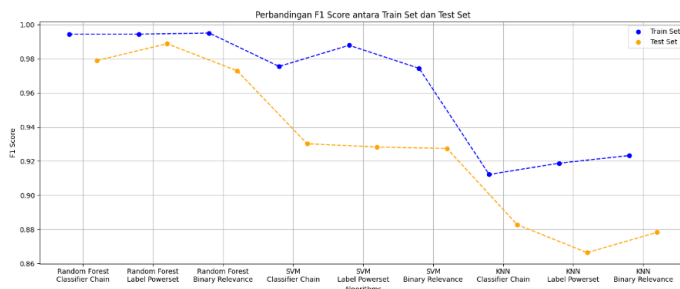
| | | | |
|------------------|------------------------|--------|--------|
| Classifier Chain | Support Vector Machine | 0.9026 | 0.6440 |
| Label Powerset | | 0.9675 | 0.7346 |
| Binary relevance | | 0.8636 | 0.6149 |
| Classifier Chain | K-Nearest Neighbor | 0.6851 | 0.5696 |
| Label Powerset | | 0.7045 | 0.5307 |
| Binary relevance | | 0.5844 | 0.4660 |

Tabel IV menyajikan hasil evaluasi performa berbagai algoritma dalam konteks *multi-label classification* dengan empat skema pembagian data yang berbeda, yaitu 80:20, 75:25, 60:40, dan 50:50 untuk data pelatihan dan pengujian. Dari hasil evaluasi, terlihat bahwa performa algoritma cenderung menurun seiring dengan peningkatan rasio pembagian data uji. Namun, perbedaan performa antara algoritma dan pendekatan *multi-label classification* juga terlihat jelas. Misalnya, *Random Forest* dengan pendekatan *Label Powerset* menunjukkan F1-Score tertinggi pada skema 80:20 dengan nilai 0.9888, sementara SVM dengan pendekatan *Classifier Chain* menampilkan F1-Score terendah pada skema 50:50 dengan nilai 0.6440. Evaluasi ini memberikan gambaran komprehensif tentang keefektifan berbagai algoritma dan pendekatan dalam menghadapi tantangan *multi-label classification*, serta memberikan panduan berharga dalam pemilihan strategi pembagian data yang tepat.

Analisis terhadap tabel tersebut juga mengungkapkan pentingnya pemilihan algoritma dan pembagian data yang sesuai untuk mencapai performa yang optimal dalam tugas klasifikasi *multi-label*. Dalam konteks ini, penggunaan algoritma *Random Forest* dengan pendekatan *Label Powerset* terlihat sangat efektif pada skema pembagian data 80:20, di mana F1 Score tertinggi tercapai. Namun, penting untuk dicatat bahwa performa algoritma dapat bervariasi tergantung pada struktur data dan karakteristik masalah yang dihadapi. Oleh karena itu, evaluasi performa seperti yang ditunjukkan dalam tabel tersebut sangatlah penting dalam memandu pengambilan keputusan terkait dengan pemilihan algoritma dan strategi pembagian data untuk memastikan kesuksesan sistem klasifikasi *multi-label* dalam berbagai kondisi dan lingkungan. Kombinasi *Random Forest* dengan *Label Powerset* mampu mempertahankan kualitas prediksi yang tinggi bahkan pada data yang belum pernah dilihat sebelumnya (*testing set*). Hal ini menunjukkan model dapat menjaga generalisasi yang baik.

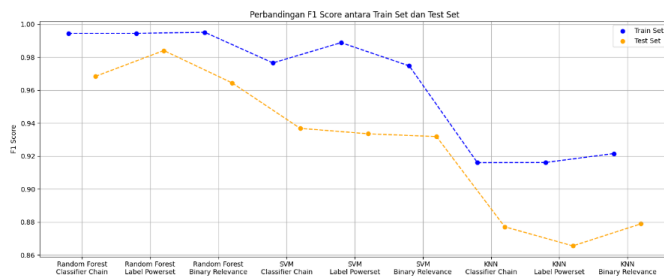
C. Grafik F1-Score pada Berbagai Pembagian Data

Grafik F1-Score pada pembagian data 80:20, 75:25, 60:40, dan 50:50 menunjukkan bahwa kombinasi *Random Forest* dan *Label Powerset* tetap konsisten menghasilkan F1-Score yang tinggi pada berbagai skenario pembagian data.



Gbr. 4 Grafik Perbandingan 80:20

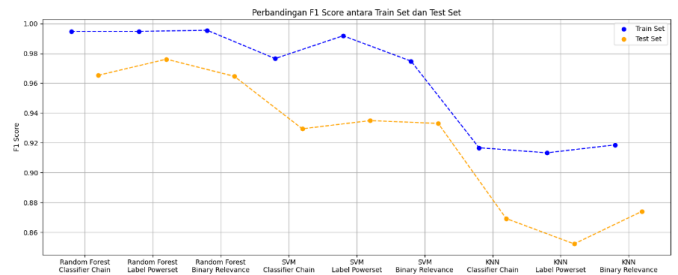
Gambar 4 menunjukkan perbandingan F1 Score antara *Train set* dan *Test set* untuk berbagai algoritma klasifikasi *multi-label*. Algoritma yang diuji termasuk *Random Forest*, *Support Vector Machine* (SVM), dan *K-Nearest Neighbor* (KNN), masing-masing dengan pendekatan *Classifier Chain*, *Label Powerset*, dan *Binary relevance*. Grafik memperlihatkan bahwa F1 Score untuk *Train set* umumnya lebih tinggi dibandingkan *Test set*, yang menunjukkan *overfitting* pada beberapa algoritma. *Random Forest* dengan *Classifier Chain* dan *Label Powerset* memiliki F1 Score tertinggi pada *Train set*, namun terdapat penurunan performa yang signifikan pada *Test set*. Sebaliknya, algoritma KNN menunjukkan performa yang lebih konsisten antara *Train set* dan *Test set* meskipun dengan skor yang lebih rendah. Perbedaan ini menyoroti tantangan dalam menjaga generalisasi model ketika dihadapkan pada data yang belum pernah dilihat sebelumnya.



Gbr. 5 Grafik Perbandingan 75:25

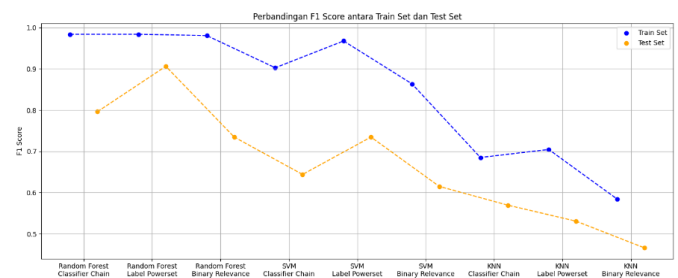
Gambar 5 menampilkan perbandingan F1 Score antara *Train set* dan *Test set* untuk berbagai algoritma dan pendekatan dalam klasifikasi *multi-label*. Algoritma yang diuji meliputi *Random Forest*, *Support Vector Machine* (SVM), dan *K-Nearest Neighbor* (KNN), masing-masing dengan pendekatan *Classifier Chain*, *Label Powerset*, dan *Binary relevance*. Dari grafik tersebut, terlihat bahwa F1-Score untuk *Train set* (ditandai dengan garis biru) umumnya lebih tinggi dibandingkan dengan *Test set* (ditandai dengan garis oranye), menunjukkan bahwa model mengalami *overfitting*. *Random Forest* dengan pendekatan *Label Powerset* memiliki F1-Score tertinggi pada *Train set*, tetapi menunjukkan penurunan yang signifikan pada *Test set*, mengindikasikan masalah generalisasi. Sementara itu, KNN dengan berbagai pendekatan memiliki performa yang lebih rendah dan lebih konsisten antara *Train set* dan *Test set*, tetapi tetap menunjukkan perbedaan yang mencolok. Grafik ini menekankan pentingnya

pemilihan algoritma dan pendekatan yang tepat untuk mencapai keseimbangan antara kinerja model pada data pelatihan dan data pengujian.



Gbr. 6 Grafik Perbandingan 60:40

Grafik pada gambar 6 menampilkan analisis perbandingan nilai F1 antara data latih dan data uji untuk berbagai algoritma klasifikasi. Algoritma yang dievaluasi meliputi *Random Forest*, *Classifier Chain*, *Label Powerset*, *Binary relevance*, serta kombinasi dari metode ini dengan SVM (*Support Vector Machine*) dan KNN (*K-Nearest Neighbors*). Garis biru mewakili nilai F1 untuk data latih yang konsisten tinggi pada semua algoritma, menunjukkan kinerja yang baik pada data latih. Sebaliknya, garis oranye menunjukkan nilai F1 untuk data uji yang umumnya lebih rendah dan menunjukkan lebih banyak variasi, menyoroti kemungkinan *overfitting* pada beberapa model, di mana kinerja pada data yang tidak terlihat tidak sekuat itu. Perbandingan ini menekankan pentingnya menyeimbangkan kompleksitas model dan generalisasi untuk mencapai kinerja optimal pada data baru.



Gbr. 7 Grafik Perbandingan 50:50

Gambar 7 menampilkan grafik perbandingan F1-Score antara *train set* dan *test set* untuk berbagai algoritma *machine learning*, seperti *Random Forest* dan SVM dengan pendekatan *Classifier Chain*, *Label Powerset*, dan *Binary relevance*. Sumbu y menunjukkan nilai F1-Score, sementara sumbu x menunjukkan algoritma yang digunakan. Garis biru dengan simbol kotak menunjukkan F1-Score pada *train set*, sedangkan garis kuning dengan simbol segitiga menunjukkan F1-Score pada *test set*. Hasilnya menunjukkan bahwa F1-Score pada *train set* umumnya lebih tinggi dibandingkan dengan *test set*, mengindikasikan adanya *overfitting*, dengan penurunan performa yang lebih jelas pada beberapa algoritma.

D. Perbandingan Label Aktual dan Prediksi

TABEL V
PERBANDINGAN LABEL

| Tweet | Label Aktual | Label Prediksi |
|---|----------------------------------|----------------------------------|
| Jalur alternatif Semarang-Bandung via limbang sumowono macet parah arah Bandung mobil motor kuat jalan licin lepas hujan | arus, cuaca, insiden, kendaraan | arus, cuaca, insiden, kendaraan |
| Jalan unair hujan kadung deres jalan airlangga karmen banjir dorong motor rsua sampek kosku banjir parah | cuaa, insiden, kendaraan, lokasi | cuaa, insiden, kendaraan, lokasi |
| Transjakarta emang sebenarnya pengaruh membludak armada faktor alam hujan deres dimana banjir lintas benar benar sendat mobil motor masuk jalur bus | arus, cuaca, kendaraan | arus, cuaca, kendaraan |

Tabel V menunjukkan perbandingan antara label aktual dan label prediksi untuk beberapa *Tweet*. *Tweet* pertama tentang kemacetan parah di jalur alternatif Semarang-Bandung setelah hujan memiliki label aktual dan prediksi yang sama, yaitu arus, cuaca, insiden, dan kendaraan. *Tweet* kedua mengenai banjir di Jalan Airlangga juga memiliki kecocokan antara label aktual dan prediksi, yaitu cuaca, insiden, kendaraan, dan lokasi. *Tweet* ketiga yang membahas dampak hujan deras terhadap lalu lintas Transjakarta juga menunjukkan kesesuaian antara label aktual dan prediksi, yaitu arus, cuaca, dan kendaraan. Hal ini menunjukkan bahwa model prediksi mampu mengidentifikasi label yang tepat untuk setiap *Tweet* dengan akurasi yang baik.

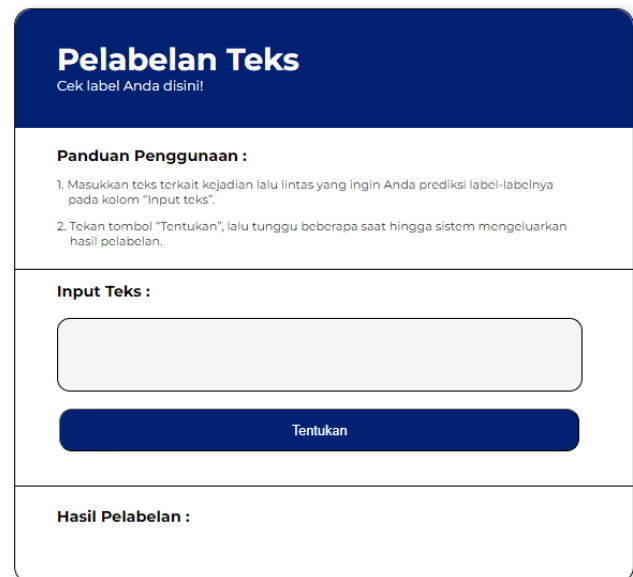
E. Perancangan Sistem Prediksi

Tahap *Deployment* dimulai dengan analisis kebutuhan, di mana data tentang kejadian lalu lintas dari platform *Twitter (X)* dikumpulkan melalui observasi lapangan. Langkah selanjutnya adalah desain sistem, yang mencakup pembuatan *wireframe* sebagai kerangka dasar antarmuka pengguna dan *flowchart* untuk memvisualisasikan alur proses sistem prediksi label. Setelah desain selesai, implementasi dilakukan menggunakan lingkungan pengembangan *Google Colab* dan *Anaconda*, dengan pemilihan algoritma seperti *Random Forest*, *SVM*, dan *KNN* untuk prediksi label pada teks. Pendekatan *multi-label classification* seperti *Multi-label Binary relevance*, *Classifier Chain*, dan *Label Powerset* juga diterapkan untuk menangani masalah *multi-label*. Penggunaan teknologi seperti *HTML*, *CSS*, dan *JavaScript* dalam desain antarmuka bertujuan untuk menciptakan tampilan yang responsif dan interaktif. Dengan demikian, tahap *Deployment* ini tidak hanya memperhatikan teknologi pengembangan perangkat lunak, tetapi juga pemilihan algoritma yang sesuai dengan kompleksitas analisis teks yang dihadapi.



Gbr. 8 Wireframe

Proses desain sistem dimulai dengan pembuatan konsep menggunakan *wireframe* pada gambar 8 sebagai kerangka dasar antarmuka. Ini melibatkan penempatan kasar elemen kunci seperti tombol dan bidang input untuk memberikan gambaran tata letak keseluruhan. Desain sistem mencakup *Flowchart* untuk visualisasi alur proses dan desain antarmuka pengguna, memberikan pandangan tentang struktur dan fungsi sistem. Hal ini memungkinkan pengembang untuk lebih efisien mengimplementasikan setiap fitur dengan panduan yang konkret, memastikan pengembangan yang lancar dan sesuai kebutuhan pengguna.



Gbr. 9 Tampilan Sistem Prediksi

Desain *interface* seperti pada gambar 9 memberikan pandangan pada pengguna terkait sistem prediksi yang akan dibuat. Sistem memiliki beberapa bagian penting, seperti judul, panduan penggunaan, input teks, tombol perintah aksi, dan

kolom hasil pelabelan. Sistem prediksi terdiri atas satu halaman untuk efisiensi dan memudahkan pengguna dalam menggunakan sistem.

F. Pengujian Sistem Prediksi

Sistem diuji menggunakan 30 data tentang kejadian lalu lintas yang dikumpulkan dari aplikasi *Twitter (X)*. Data tersebut diprediksi labelnya menggunakan sistem yang telah dibangun. Data dibagi menjadi tiga kelompok untuk mencakup berbagai skenario:

- 10 data dengan 2 label
- 10 data dengan 3 label
- 10 data dengan 4 label

Kemudian, 10 responden diminta secara independen untuk memberikan label pada setiap data berdasarkan pemahaman mereka. Pilihan label dari responden dibandingkan dengan hasil prediksi label sistem untuk menilai kesesuaian antara keduanya. Selanjutnya, rata-rata hasil perbandingan dihitung untuk mengevaluasi kinerja sistem secara komprehensif. Akurasi dihitung dengan rata-rata kesesuaian label antara sistem dan responden. Jika 1 menunjukkan kesesuaian penuh dan 0 menunjukkan ketidaksesuaian, maka kita menghitung persentase rata-rata dari semua responden.

TABEL VI
PERSENTASE RESPONDEN

| Responden | Persentase |
|----------------------|------------|
| 1 | 92,78% |
| 2 | 89,44% |
| 3 | 94,72% |
| 4 | 89,17% |
| 5 | 88,61% |
| 6 | 91,94% |
| 7 | 88,89% |
| 8 | 80,55% |
| 9 | 89,44% |
| 10 | 87,22% |
| Rata-rata Persentase | 89,28% |

Tabel VI menunjukkan hasil analisis tanggapan dari sepuluh responden terkait pemberian label yang akurat terhadap sebuah set data teks. Dengan rata-rata persentase kebenaran sebesar 89,28%, dapat disimpulkan bahwa sebagian besar responden memiliki tingkat kebenaran yang tinggi. Responden yang mencapai persentase kebenaran tertinggi adalah responden 3, dengan persentase kebenaran sebesar 94,72%, diikuti oleh responden 1 dengan 92,78%. Namun, terdapat juga responden yang memiliki persentase kebenaran lebih rendah, seperti responden 8 dengan 80,55%. Meskipun begitu, secara keseluruhan, mayoritas responden menunjukkan kemampuan yang baik dalam memilih label secara akurat.

IV. KESIMPULAN

Dari langkah-langkah penelitian yang telah dilakukan, dapat disimpulkan hal berikut:

1. Penelitian menunjukkan kombinasi *Random Forest* dengan *Label Powerset* memberikan hasil terbaik dengan F1-Score 0.9944 pada *training set* dan 0.9888 pada *testing set*. Metode ini dipilih berdasarkan eksperimen yang menunjukkan performa terbaik dalam akurasi prediksi.
2. Implementasi metode terbaik, yaitu *Label Powerset* dengan *Random Forest*, digunakan dalam proses *pseudo-labeling* pada dataset teks *Twitter (X)*. Hasil evaluasi *testing set* menunjukkan prediksi *pseudo-label* yang diterima dengan persentase keakuratan rata-rata 89,28%, menunjukkan efektivitas *pseudo-labeling* untuk pelabelan otomatis pada klasifikasi *multi-label* data teks.

UCAPAN TERIMA KASIH

Dengan rasa syukur kepada Allah SWT atas segala rahmat dan petunjuk-Nya, penelitian ini dapat berjalan dengan lancar. Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan dan dukungan selama proses penelitian serta dalam penyusunan laporan dan artikel ini. Semoga hasil penelitian ini memberikan manfaat besar, khususnya dalam bidang informatika, untuk kemajuan ilmu pengetahuan dan teknologi.

REFERENSI

- [1] W. Lubis, Y. A. Sari, dan M. A. Fauzi, "Klasifikasi Multilabel Menggunakan Metode Fuzzy Similarity *K-Nearest Neighbor* Untuk Rekomendasi Pencarian Artikel Online," vol. 3, no. 1, 2019.
- [2] R. T. Aldisa dan Azizah, "Penerapan Metode Scrum pada Sistem Kehadiran Asisten di Laboratorium Berbasis Android," *Journal of Information System Research (JOSH)*, vol. 3, no. 3, pp. 227–233, 2022, DOI: 10.47065/josh.v3i3.1481.
- [3] A. P. J. Dwitama dan S. Hidayat, "Identifikasi Ujaran Kebencian Multilabel pada Teks *Twitter (X)* Berbahasa Indonesia Menggunakan Convolution Neural Network," *Jurnal Sistem Komputer dan Informatika (Json)*, vol. 3, no. 2, pp. 117, 2021, DOI: 10.30865/json.v3i2.3610.
- [4] F. Sukma Alfiani dan U. Laili Yuhana, "Implementasi Metode Klasifikasi Multilabel Untuk Kategorisasi Materi Pembelajaran Secara Otomatis," vol. 8, no. 4, pp. 1750–1758, 2021.
- [5] S. Khomsah dan A. S. Aribowo, "Ledger: Journal Informatic and Information Technology Model *Semi-supervised Learning* Menggunakan Logistic Regression Untuk Anotasi Sentimen," *Open Access Ledger*, vol. 1, no. 4, 2022.
- [6] Manueke, et al., "Evaluasi *Multi-label Classification* pada Teks *Twitter (X)* Menggunakan *Binary relevance* dan BRkNN," 2022.
- [7] I. A. Kim, H. C. Park, J. H. Kim, D. W. Lee, "Multilabel Naïve Bayes *Classification* Considering Label Dependence," *Pattern Recognition Letters*, vol. 136, pp. 279–285, 2020.
- [8] M. N. Karsana, M. Muslim, and W. Astuti, "*Single-Label and Multi-label Text Classification* Using Ann and Comparison With Naïve Bayes and Svm," *MIB*, vol. 7, no. 2, pp. 857-863, 2023. doi: 10.30865/mib.v7i2.6024.
- [9] Winda Kurnia Sari, D. P. Rini, Reza Firsandaya Malik, and Iman Saladin B. Azhar, "Multilabel *Text Classification* in News Articles Using Long-Term Memory with Word2Vec", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 4, no. 2, pp. 276 - 285, Apr. 2020.
- [10] E. E. Fitriani and W. Yustanti, "Perbandingan Kinerja Metode *Problem transformation-Knn* dan *Algorithm Adaptation-Knn* pada Klasifikasi *Multi-label* Pertanyaan Kotakode," 2022.

- [11] W. Wisnalmawati, A. S. Aribowo, and Y. Herawati, "Semi-supervised Learning Models for Sentiment Analysis on Marketplace Dataset", Int. J. Artif. Intell. Robot., vol. 4, no. 2, pp. 78-85, Dec. 2022.
- [12] H. Lee, "Pseudo-label: The simple and efficient semi-supervised Learning method for deep neural networks," ICML Workshop on Challenges in Representation Learning, 2013.
- [13] S. Navisa, Luqman Hakim, and Aulia Nabilah, "Komparasi Algoritma Klasifikasi Genre Musik pada Spotify Menggunakan CRISP-DM: Indonesia", JSC, vol. 4, no. 2, pp. 114 - 125, Aug. 2021.
- [14] Nafisah Nurul Hakim, "Implementasi Machine learning pada Sistem Prediksi Kejadian dan Lokasi Patah Rel Kereta Api di Indonesia", JSC, vol. 3, no. 1, pp. 25 - 35, May 2020.
- [15] S. Sutisna and M. N. Yuniar, "Klasifikasi Kualitas Air Bersih Menggunakan Metode Naïve baiyes", SAINTEK, vol. 5, no. 1, pp. 243-246, Aug. 2023.
- [16] E. D. Nurhazizah and I. Puspitasari, "Opinion Mining Fungsi KPI (Key Performance Indikator) Dengan Algoritma Naïve Bayes Clasifier dan Support Vector Machine (SVM)", JCM, Vol. 3, No. 2, Pp. 290-302, May 2023.
- [17] L. Hermawati, V. Berland, A. Rahmadiyah, E. Hutabarat, and D. . Dwi Saputra, "Komparasi Metode Text Mining Terhadap Masalah Pengklasifikasian Narasi Informatif & Non Informatif Pada Twitter (X) @PLN_123", jsisfotek, vol. 5, no. 1, pp. 109–120, May 2023.
- [18] S. Lestari and A. Damaiyanti, "Penerapan Data Mining Menggunakan Metode Algoritma Apriori Untuk Menentukan Penjualan Barang IT pada PT. Javas Karya Tungga Jakarta Selatan: Penerapan Data Mining Menggunakan Metode Algoritma Apriori Untuk Menentukan Penjualan Barang IT pada PT. Javas Karya Tungga Jakarta Selatan", SAINTEK, vol. 5, no. 1, pp. 237-242, Aug. 2023.
- [19] A. D. Sidik and A. . . Ansawarman, "Prediksi Jumlah Kendaraan Bermotor Menggunakan Machine learning", FJMR, vol. 1, no. 3, pp. 559–568, Jul. 2022.
- [20] T. Ridwansyah, "Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter (X) Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier," KLIK: Kajian Ilmiah Informatika dan Komputer, vol. 2, no. 5, pp. 178-185, April 2022.