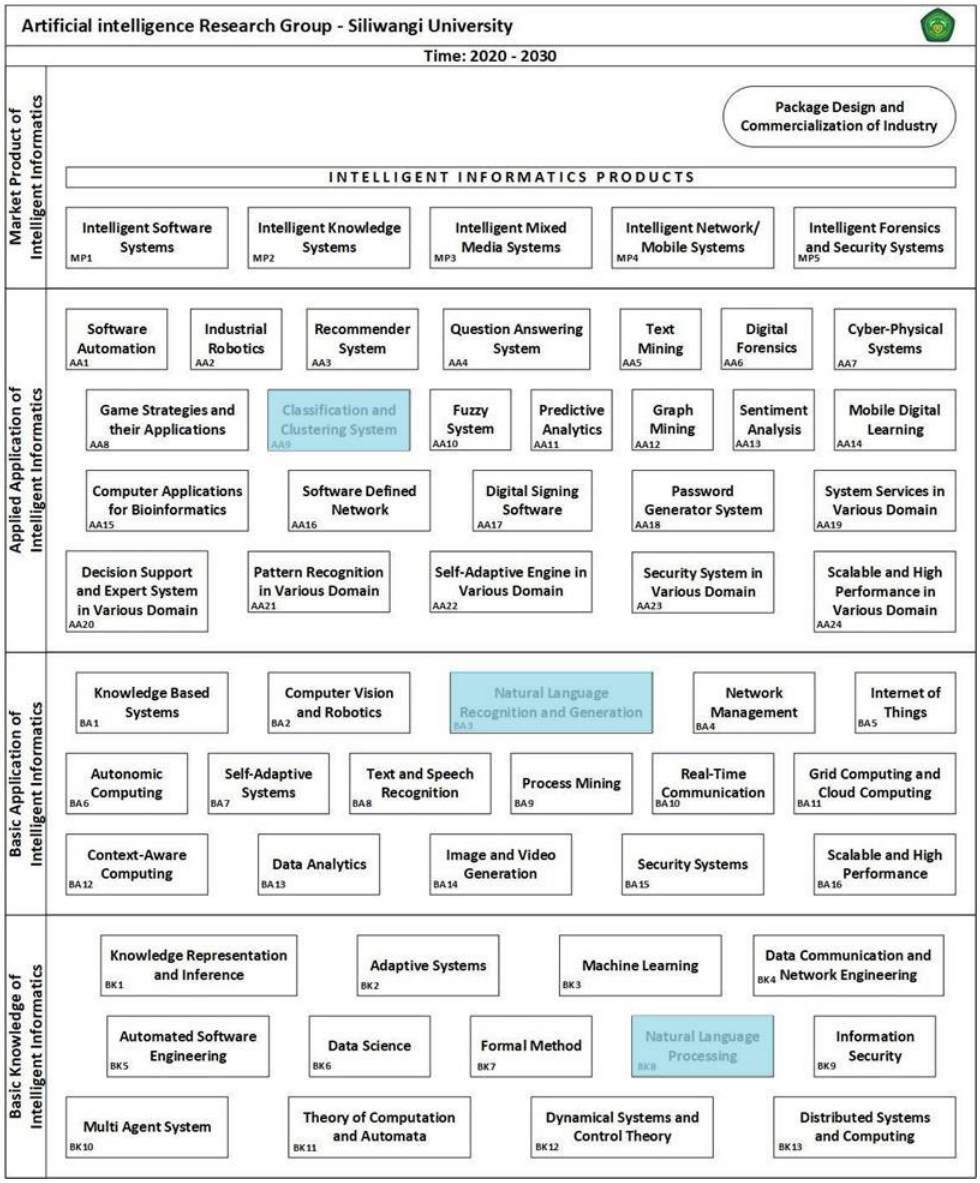
BAB III

METODOLOGI

3.1. Kontribusi Terhadap AIS

Kontribusi penelitian yang diajukan terhadap terlampir pada Gambar 3.1, yaitu roadmap produk informatika cerdas yang dikembangkan oleh kelompok penelitian Artificial Intelligence Siliwangi (AIS).

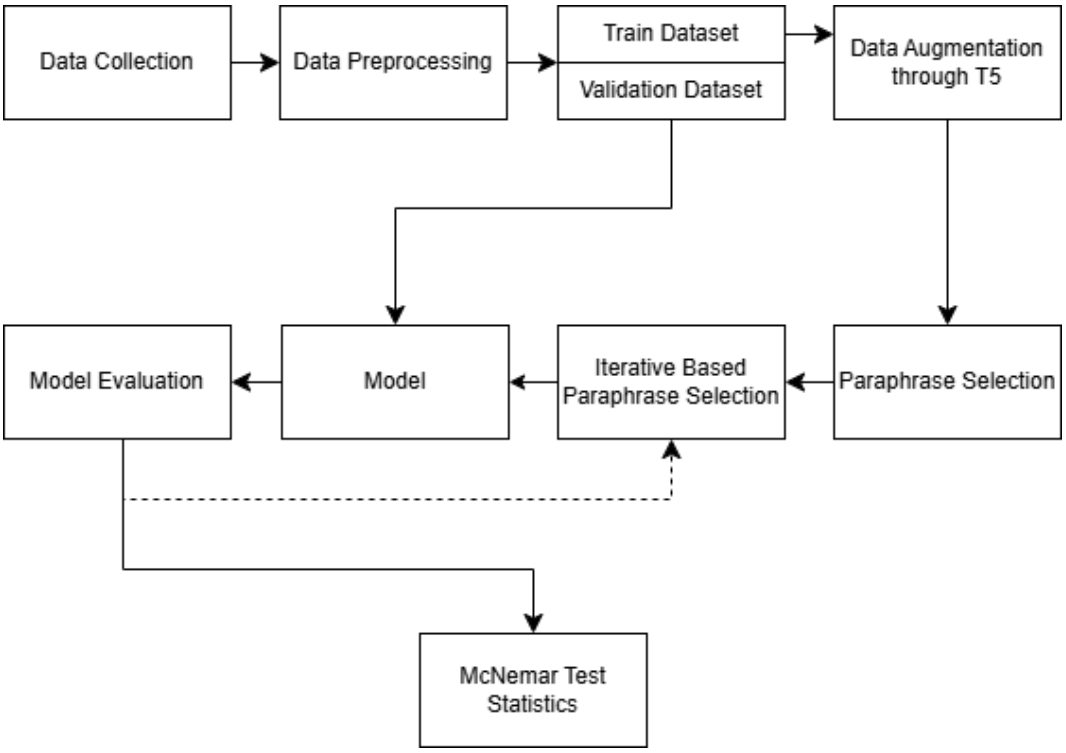


Gambar 3.1 *Roadmap* Penelitian (Artificial Intelligence Research Group, 2024)

Kontribusi penelitian ini terhadap Artificial Intelligence Siliwangi masuk ke dalam beberapa bidang, yaitu *Basic Knowledge of Intelligent Informatics* di bidang *Natural Language Processing* karena *RoBERTa* memiliki kemampuan yang revolusioner dalam memahami konteks dua arah dalam sebuah kalimat. Dengan arsitektur *transformer*, *RoBERTa* dapat menangani berbagai tugas *NLP* seperti klasifikasi teks, penjawaban pertanyaan, ekstraksi informasi, dan lain-lain. Selanjutnya, masuk ke *Basic Application of Intelligent Informatics* pada bidang *Natural Language Recognition and Generation* karena kemampuannya untuk memahami konteks *bidirectional* dari kata-kata dalam teks. Kemudian, masuk ke *Applied Application of Intelligent Informatic* pada bidang *Classification and Clustering System* karena proses akhirnya adalah pengklasifikasian teks berdasarkan bias.

3.2. Tahapan Penelitian

Capaian penelitian ini diharapkan mampu mengoptimalkan metode *RoBERTa/DA-RoBERTa* dengan metrik evaluasi yang lebih baik pada kasus deteksi bias media. Dengan demikian, dilakukan optimasi dengan augmentasi data berbasis parafrase menggunakan model *T5*, yang dapat mereformula kalimat dalam dataset tanpa mengubah maknanya. Tahapan penelitian dalam studi ini mengadaptasi dan mengembangkan pendekatan dari beberapa penelitian sebelumnya, antara lain yang dilakukan oleh (Bird dkk., 2023; Ghorbani & Zou, 2019; Krieger dkk., 2022; Lee dkk., 2022; Spinde dkk., 2021). Berikut merupakan rangkaian tahapan penelitian yang telah dikembangkan dan akan dilakukan dalam penelitian ini terlampir pada Gambar 3.2.



Gambar 3.2 Tahapan Penelitian

3.2.1. Pengumpulan Data

Dataset yang digunakan berasal dari repositori *Paper With Code* pada dataset *BABE* (Bias Annotations By Experts) (https://paperswithcode.com/dataset/babe). Dataset ini terdiri dari 3.700 kalimat yang tersebar merata di berbagai topik dan sumber berita. Setiap kalimat dalam dataset ini telah dianotasi untuk mendeteksi bias media. Salah satu aspek utama dari dataset *BABE* adalah penggunaan anotasi dari para ahli untuk memastikan kualitas data yang tinggi, dirancang untuk menghasilkan label bias yang lebih andal dan berkualitas tinggi (Spinde dkk., 2021). *BABE* memuat beberapa atribut penting, diantaranya adalah *text*, *news_link*, *outlet*, *topic*, *type*, *label_bias*, *label_opinion* dan *biased_words*. Sesuai dengan penelitian sebelumnya (Krieger dkk., 2022; Spinde dkk., 2021), fitur yang

digunakan adalah *text*, sedangkan labelnya adalah *label_bias* (Biased, Non-biased, No Agreement). Pendekatan yang sama akan diterapkan dalam penelitian ini.

3.2.2. Pra-pemrosesan Data

Preprocessing data bertujuan untuk memastikan bahwa data teks dalam kondisi yang optimal sebelum diolah oleh model (Siino dkk., 2024). Namun, model *transformer* biasanya tidak memerlukan *preprocessing* berbasis bahasa seperti *lemmatization* atau *stemming* karena umumnya dirancang untuk belajar langsung dari representasi teks mentah yang telah di-tokenisasi (Rahali & Akhloufi, 2023). Oleh karena itu, dalam tahap ini dilakukan *Data Cleaning* yaitu menghapus data yang tidak memiliki kesepakatan label, mengonversi label menjadi format numerik. Kemudian, tokenisasi menggunakan tokenizer dari *RoBERTa*, tidak ada *preprocessing* tambahan.

3.2.3. Pembagian Data

Pembagian data dilakukan dengan menggunakan teknik *Stratified 5-Fold Cross-Validation*. *K-Fold Cross-Validation* adalah teknik evaluasi model yang membagi dataset menjadi K bagian (fold). Setiap *fold* akan bergiliran digunakan sebagai data validasi sementara *K-Fold* lainnya digunakan sebagai data pelatihan.

3.2.4. Augmentasi Data

Proses augmentasi data dilakukan bersamaan dengan teknik *k-fold cross-validation*, di mana dataset dibagi menjadi beberapa *fold*. Augmentasi hanya diterapkan pada data latih di setiap iterasi fold, sementara data validasi tetap menggunakan data asli untuk menjaga kualitas data. Pada setiap iterasi *fold*, data

latih diperluas secara bergantian menggunakan teknik parafrase. Setiap indeks pada data latih diumpankan ke model parafrase, sehingga menghasilkan beberapa alternatif parafrase. Setelah itu, hasil augmentasi kemudian ditambahkan ke data latih asli (Bird dkk., 2023).

3.2.4.1.Parafrase yang Dihasilkan

Teknik parafrase digunakan untuk menghasilkan dataset baru pada data pelatihan. Model yang digunakan adalah sebuah model berbasis *T5* yang telah di *fine-tune* khusus untuk tugas parafrase (*Parrot Paraphraser*). Model ini diambil dari repositori GitHub (https://github.com/PrithivirajDamodaran/Parrot_Paraphraser) dengan konfigurasi parameter diversity_ranker="levenshtein", do_diverse=False, max_return_phrases=10, max_length=100, adequacy_threshold=0.88, fluency_threshold=0.80. *Levenshtein distance* digunakan untuk mengukur perbedaan antara dua string, yaitu data asli dan data hasil parafrase. Setiap kalimat pada data latih diparafrasekan sebanyak 10 kali untuk menghasilkan kalimat yang variase dan untuk menghemat penggunaan sumber daya komputasi seperti memori dan waktu. Kemudian, panjang parafrase dibatasi maksimal 100 token dan parafrase harus memenuhi *adequacy threshold* ≥ 88% dan *fluency threshold* ≥ 80% untuk memastikan kesamaan makna kefasihan bahasa.

3.2.4.2.Penggabungan Dataset

Dataset hasil parafrase digabungkan dengan dataset latih asli. Penggabungan ini bertujuan untuk meningkatkan variasi data sehingga model dapat belajar dari berbagai ekspresi kalimat yang memiliki makna serupa.

3.2.5. Seleksi Parafrase

(Lee dkk., 2022) menunjukan bahwa keberadaan teks yang identik pada dataset dapat menurunkan performa model serta meningkatkan kecenderungan model menghafal teks tertentu dan melemahkan pemahaman pola dalam data. Oleh karena itu, dilakukan seleksi terhadap data hasil parafrase dengan cara menghilangkan data yang identik dengan data asli maupun data parafrase lainnya.

Penghapusan data yang identik dengan data asli dilakukan berdasarkan skor *Levenshtein*, dimulai dari skor terkecil (0). Penghapusan dilakukan berdasarkan kelipatan 10 untuk menghemat waktu komputasi. Kemudian, untuk mengatasi data hasil parafrase yang identik dengan data parafrase lainnya, hanya satu data parafrase yang diambil dengan *Levenshtein* terkecil pada setiap kelompok data penghapusan. Berikut merupakan *pseudo code* dari tahap seleksi parafrase.

```
INPUT:
- data ← DataFrame['text', 'Label_bias_0-1']
- settings:
    - paraphrase ← True
    - paraphrase_selection ← True
    - min_score ← 10

PROCESS:
FOR each fold in kfold.split(data):
    train_data ← data[train_idx]
```

```
    test_data  ← data[test_idx]


    IF settings.paraphrase:
        paraphrase_file ← "paraphrase_fold_<fold>.csv"


        IF file_exists(paraphrase_file):
            combined ← load_csv(paraphrase_file)


            IF settings. paraphrase_selection:
                original     ← rows with index_source = NULL
                paraphrased  ← rows with index_source ≠ NULL AND
score ≥ min_score
                selected     ← select one best paraphrase (lowest
score) per source
                train_data   ← original + selected
        ELSE:
            generate_new_paraphrase()
```

Augmentasi data dilakukan dengan rasio 1:1 (satu parafrase baru per satu sampel asli) setelah proses seleksi, langkah ini bertujuan untuk mengoptimalkan efisiensi komputasi karena adanya keterbatasan sumber daya *GPU* pada platform Kaggle.

3.2.6. Seleksi Parafrase Berbasis Iteratif

Tahap ini bertujuan untuk menghasilkan dataset parafrase pelatihan final yang lebih bersih. Parafrasa yang dihasilkan oleh model seperti *transformer* tidak selalu berkualitas baik (Zhou & Bhat, 2021). Oleh karena itu, dilakukan penghapusan dataset yang berpotensi menurunkan performa model.

Proses penyaringan dilakukan dengan berbasis iterative, mengadopsi dari *Data Shapley* (Ghorbani & Zou, 2019), dimana tujuannya untuk mengevaluasi

kontribusi setiap titik data pelatihan terhadap kinerja model secara keseluruhan. Pendekatan tersebut digunakan dalam penelitian ini, jika kinerja model (Weighted F1-Score) meningkat setelah penghapusan titik data tertentu, maka data tersebut tidak digunakan lagi. Pada iterasi berikutnya, metrik evaluasi terbaru yang lebih tinggi dijadikan inisialisasi baru, proses ini berulang hingga iterasi selesai. Berikut adalah *pseudo code* dari tahap seleksi parafrase berbasis iteratif.

```
INPUT:
- data ← DataFrame['text', 'Label_bias_0-1']
- settings:
    - paraphrase ← True
    - paraphrase_selection ← True
    - iterative_selection ← True
    - min_score ← 50


INIT:
- selected_for_removal ← []
- initial_performance ← initial model performance (Weighted F1-
score)


PROCESS:
FOR each fold in kfold.split(data):
    train_data ← data[train_idx]
    test_data  ← data[test_idx]


    IF settings.paraphrase AND settings.paraphrase_selection:
        train_data ← original data + selected paraphrases


    IF settings.iterative_selection:


        all_candidates ← collect all index_source values from
train_data
```

```
        FOR each candidate IN all_candidates:

            IF candidate is in selected_for_removal:
                CONTINUE

            temp_removal ← selected_for_removal + [candidate]
            fold_scores ← []

            FOR each fold IN kfold_splits:
                fold_data ← copy of train_data for this fold
                remove paraphrases where index_source is in
temp_removal
                model ← train_model(fold_data)
                score ← evaluate_model(model, test_data)
                fold_scores.APPEND(score)

            avg_score ← average of fold_scores

            IF avg_score > initial_performance:
                selected_for_removal.APPEND(candidate)
                initial_performance ← avg_score

OUTPUT:
- Final training dataset ← original + selected -
selected_for_removal
```

3.2.7. Model

3.2.7.1.*Pre-Training*

Model dasar yang digunakan adalah *RoBERTa* dan *DA-RoBERTa* dengan bobot *pre-training* yang telah di-*fine-tune* sebelumnya oleh (Krieger dkk., 2022).

Konfigurasi pelatihan mengikuti pendekatan yang sama, dengan batch size 32 kalimat per batch, optimizer AdamW (learning rate = 1e-5), fungsi *loss Binary Cross-Entropy* dan *early stopping* diaktifkan dengan pemantauan *loss* validasi. Arsitektur dan *hyperparameter* dipertahankan tanpa modifikasi.

3.2.7.2.*Fine-Tuning*

Penelitian ini menerapkan augmentasi data dengan model *T5* yang diimplementasikan melalui *framework Parrot* untuk meningkatkan kinerja model dalam tugas klasifikasi. Proses parafrase dilakukan terhadap kalimat-kalimat dalam dataset untuk menghasilkan variasi sintaksis yang tetap mempertahankan makna asli. Dataset final hasil gabungan data latih asal dan data latih hasil parafrase digunakan untuk *fine-tuning* model. Proses *fine-tuning* dilakukan dengan konfigurasi yang konsisten dengan *pre-training*, yaitu batch size 32, optimizer AdamW (learning rate = 1e-5), fungsi *loss Binary Cross-Entropy*, dan penerapan *early stopping* yang memantau *loss* validasi untuk mencegah overfitting.

3.2.8. Evaluasi Model

Pengujian evaluasi model dilakukan menggunakan *Stratified 5-Fold Cross-Validation*. Metrik evaluasi yang digunakan secara konsisten di seluruh eksperimen mencakup *Accuracy*, *Weighted F1-Score* (*Standar Deviation*), *Precision*, *Recall*, dan *Loss* (Pedregosa dkk., 2011; Shannon, 1948; Sokolova, 2009). Penelitian ini menggunakan *Weighted F1-Score*, dimana dalam penelitian sebelumnya menyebutkan metrik ini sebagai 'Macro F1', meskipun implementasinya mengarah pada perhitungan *Weighted F1-Score* (average='weighted').

Rumus *Accuracy* sebagai pada persamaan 3,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Accuracy mengukur persentase prediksi benar dari total sampel.

Rumus *Precision* sebagai pada persamaan 4,

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Precision merepresentasikan akurasi prediksi kelas positif.

Rumus *Recall* sebagai pada persamaan 5,

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

Recall mengukur kemampuan model dalam mengidentifikasi semua sampel positif.

Rumus *F1-Score* sebagai pada persamaan 6,

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

F1-Score adalah rata-rata harmonik dari *precision* dan *recall*.

Rumus *Weighted F1-Score* sebagai pada persamaan 7,

$$Weighted\ F1 - Score = \sum_{i=1}^{N} w_i \times F1_i \tag{7}$$

Weighted F1 menghitung rata-rata *F1-Score* dengan bobot (w_i) sesuai jumlah sampel tiap kelas.

Rumus *Cross-Entropy Loss* sebagai pada persamaan 8,

$$Loss = -\left[y.\log(\hat{y}) + (1-y)\log(1-\hat{y})\right] \quad (8)$$

Loss mengukur kesalahan prediksi probabilitas, di mana y adalah label sebenarnya (0 atau 1) dan $\hat{y}$ adalah probabilitas prediksi model.

3.2.8.1.Pengujian Model *Baseline*

Pengujian awal dilakukan terhadap model *RoBERTa* dan *DA-RoBERTa*. Model tersebut sebelumnya telah diuji dalam penelitian (Krieger dkk., 2022; Spinde dkk., 2021). Pengujian ini dilakukan guna memastikan konsistensi hasil dengan penelitian sebelumnya, kode diperoleh dari Github (https://github.com/Media-Bias-Group/A-Domain-adaptive-Pre-training-Approach-for-Language-BiasDetection-in-News).

3.2.8.2.Pengujian Model + Augmentasi Data

Pengujian kedua dilakukan terhadap model *RoBERTa* dan *DA-RoBERTa* dengan integrasi augmentasi data. Pengujian ini dilakukan untuk meningkatkan kemampuan kinerja model dengan menambah variasi data pelatihan serta mengatasi keterbatasan jumlah data yang ada. Proses augmentasi dilakukan dengan menerapkan teknik parafrase oleh model *T5* pada data latih di setiap *fold*, sehingga memperkaya variasi linguistik dalam dataset.

3.2.8.3.Pengujian Model + Augmentasi Data + Seleksi Parafrase

Pengujian ketiga dilakukan terhadap model *RoBERTa* dan *DA-RoBERTa* dengan integrasi seleksi parafrase. Pengujian ini dilakukan guna menghilangkan data identik/duplikasi dan meningkatkan efisiensi komputasi.

3.2.8.4.Pengujian Model + Augmentasi Data + Seleksi Parafrase + Berbasis Iteratif

Pengujian keempat dilakukan terhadap model *RoBERTa* dan *DA-RoBERTa* dengan integrasi seleksi parafrase berbasis iteratif. Semua tahapan ini menjadi Augmentasi Data: *All Task* yang meliputi augmentasi data, seleksi parafrase, dan seleksi parafrase berbasis iteratif. Pengujian ini bertujuan untuk menghasilkan dataset pelatihan final yang lebih bersih dan berkualitas.

3.2.8.5.Statistik Uji McNemar

Uji McNemar adalah suatu metode untuk menguji apakah dua perlakuan menghasilkan perbedaan yang signifikan dalam respon biner dari subjek yang sama. Uji ini diterapkan pada tabel kontingensi 2 × 2, yang menunjukkan hasil dari dua algoritme pada sampel *n* contoh (Krieger dkk., 2022; Pembury Smith & Ruxton, 2020).