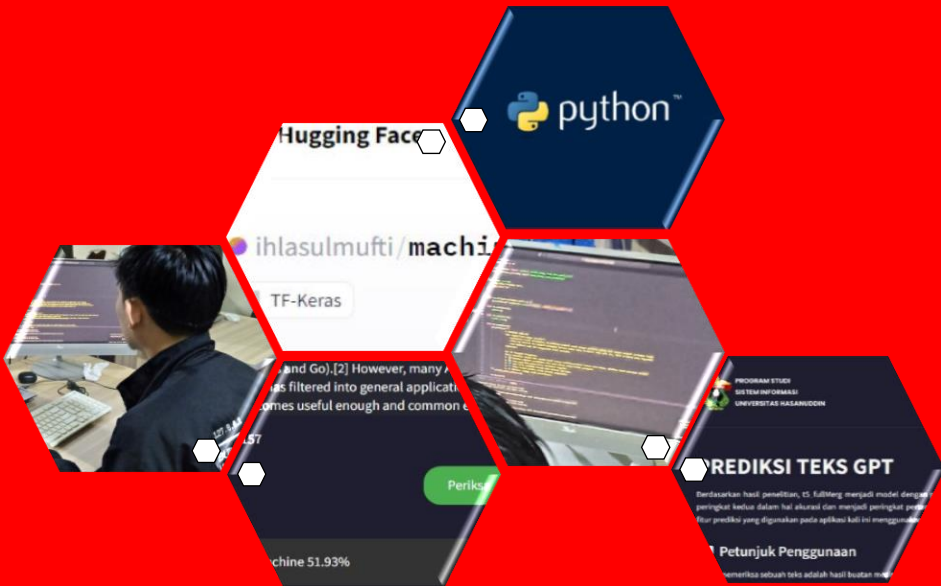


**ANALISIS PERBANDINGAN KINERJA
MODEL ARSITEKTUR XLNET, T5, DAN ERNIE
TERHADAP DETEKSI TEKS HASIL KECERDASAN BUATAN**



IHLASUL MUFTI FAQIH

H071201068



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR**

**ANALISIS PERBANDINGAN KINERJA
MODEL ARSITEKTUR XLNET, T5, DAN ERNIE
TERHADAP DETEKSI TEKS HASIL KECERDASAN BUATAN**

**IHLASUL MUFTI FAQIH
H071201068**



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2024**

**ANALISIS PERBANDINGAN KINERJA
MODEL ARSITEKTUR XLNET, T5, DAN ERNIE
TERHADAP DETEKSI TEKS HASIL KECERDASAN BUATAN**

IHLASUL MUFTI FAQIH

H071201068

Skripsi

Sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Sistem Informasi

pada

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2024**

SKRIPSI

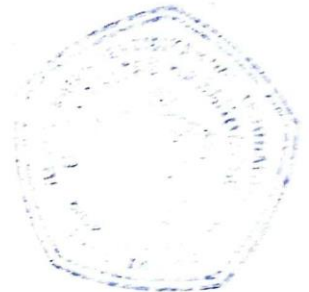
ANALISIS PERBANDINGAN KINERJA MODEL ARSITEKTUR XLNET, T5, DAN ERNIE TERHADAP DETEKSI TEKS HASIL KECERDASAN BUATAN

IHLASUL MUFTI FAQIH
H071201068

Skripsi,

telah dipertahankan di depan Panitia Ujian Sarjana Sistem Informasi
pada 23 Agustus 2024
dan dinyatakan telah memenuhi syarat kelulusan
pada

Program Studi Sistem Informasi
Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan
Universitas Hasanuddin
Makassar



Mengesahkan:
Pembimbing Tugas Akhir,

Edy Saputra Rusdi, S.Si., M.Si
NIP. 19910410 202005 3 001

Pembimbing Pendamping,

A. Muh. Amil Siddik, S.Si..M.Si
NIP. 19911003 201903 1 015

Mengetahui:
Ketua Program Studi

Prof. Dr. Jeffry Kusuma, Ph.D
NIP. 19641112 198703 1 002

PERNYATAAN KEASLIAN SKRIPSI DAN PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa, skripsi berjudul "Analisis Perbandingan Kinerja Model Arsitektur XLNet, T5, dan ERNIE terhadap Deteksi Teks Hasil Kecerdasan Buatan" adalah benar karya saya dengan arahan dari pembimbing Edy Saputra Rusdi, S.Si., M.Si sebagai Pembimbing Utama dan A. Muh. Amil Siddik, S.Si.,M.Si sebagai Pembimbing Pendamping. Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, 23 Agustus 2024



Muhammad Faqih
Muhammad Faqih
H071201068

UCAPAN TERIMA KASIH

Puji dan syukur dipanjatkan atas kehadiran Allah SWT. karena berkat rahmat dan karunia-Nya skripsi dengan judul “Analisis perbandingan kinerja model arsitektur XLNet, T5, dan ERNIE terhadap deteksi teks hasil kecerdasan buatan” dapat terselesaikan dengan baik dan tepat pada waktunya. Penulis menyadari dalam pelaksanaan dan penyusunan skripsi ini banyak melibatkan pihak yang sangat berjasa bagi penulis.

Rasa terima kasih yang besar terhadap keluarga penulis terkhusus Syamsul Bahri sebagai Ayah dan Rahmaini Amrullah sebagai Ibu yang selalu mendukung, mendoakan tiada hentinya, serta pemberian uang saku dan biaya pendidikan sejak Sekolah Dasar hingga bangku Universitas untuk memenuhi kebutuhan pendidikan penulis. Keluarga sebagai pendukung utama telah menjadi alasan kuat bagi penulis untuk menyelesaikan skripsi ini sebagai salah satu syarat menyelesaikan pendidikan di Universitas Hasanuddin.

Ucapan terima kasih juga penulis sampaikan kepada Bapak Edy Saputra Rusdi, S.Si., M.Si dan Bapak A. Muh. Amil Siddik, S.Si.,M.Si yang telah memberikan dan menyempatkan waktunya untuk membimbing dalam penyusunan skripsi ini. Terima kasih juga untuk Bapak Dr. Muhammad Hasbi, M.Sc dan Ibu Rozalina Amran, S.T., M.Eng selaku dosen penguji sidang skripsi yang telah memberikan banyak masukan sehingga melengkapi kekurangan yang ada selama penyusunan skripsi ini. Tidak lupa terima kasih kepada para dosen dan staf mulai dari lingkup program studi, departemen, fakultas, dan universitas yang telah mengajar dan mendukung penulis selama masa perkuliahan.

Tidak lupa penulis mengucapkan terima kasih kepada teman-teman Sistem Informasi 2020 (Sisfo'20) yang menjadi rekan seperjuangan. Penulis juga mengucapkan terima kasih kepada kumpulan alumni Sisfo'20: Talitha, Ave, Nilam, Nanda, Faizah, Rani, Mita, Siska, Purez, Fadil, Haerul, Ara, Qila, Dini, yang telah menyelesaikan studi mereka terlebih dahulu dan menjadi motivasi bagi penulis untuk segera menyusul mereka. Ucapan terima kasih juga ditujukan kepada Awang, Rizuki, Hamsa, Yan, Rahmah, Gau, Thoriq, Fauzan, Iman, Azhar, Firman, Miko, Resky, Ghazy, Idil, Wiwit, Aqsa, Izza, Napus, Vina, Mustika, Alim, Hajid, Hadi, Jumadil, Fuad, Jalu, Ichsan, Nisa, Ikram, dan Mamet, serta seluruh teman-teman Sisfo'20 yang tidak sempat penulis sebutkan satu per satu, yang telah menemani masa perkuliahan penulis hingga selesai. Ucapan terima kasih untuk teman-teman KKNT Gel. 111, Bangkit 2023, dan asisten serta praktikan laboratorium Sistem Informasi 2022 – 2024 yang mengisi kegiatan penulis di luar kegiatan perkuliahan.

Penulis,
Ihlasul Mufti Faqih

ABSTRAK

IHLASUL MUFTI FAQIH. **Analisis perbandingan kinerja model arsitektur XLNet, T5, dan ERNIE terhadap deteksi teks hasil kecerdasan buatan** (dibimbing oleh Edy Saputra Rusdi dan A. Muh. Amil Siddik).

Latar belakang. Perkembangan kecerdasan buatan dalam membuat teks telah berkembang pesat dan memiliki manfaat yang baik namun juga dapat disalahgunakan. **Tujuan.** Penelitian ini bertujuan untuk membandingkan model arsitektur XLNet, T5, dan ERNIE untuk melihat kinerja terbaik dalam mendeteksi teks hasil kecerdasan buatan serta mengimplementasikannya ke dalam laman web. **Metode.** Dataset yang digunakan berisikan teks abstrak penelitian (14.000 baris) dan wikipedia (210.000 baris) dengan label manusia dan mesin. Dataset kemudian digabungkan dengan jumlah keseluruhan dan jumlah seimbang sehingga dihasilkan empat jenis dataset. Penelitian ini menggunakan nilai akurasi, *precision*, *recall*, *F1score* dan sumber daya yang digunakan model dalam menilai kinerja yang terbaik. Hugging Face digunakan sebagai tempat mengunggah model dan Streamlit sebagai *framework* pengembangan *website*. **Hasil.** Model yang dilatih menggunakan dataset ukuran besar seperti gabungan penuh antara abstrak dan wikipedia memiliki hasil paling tinggi dari dataset yang lain. T5 memiliki nilai akurasi 97,2% jika dibandingkan dengan XLNet (64,62%) dan ERNIE (61,56%). Melihat dari nilai F1score menggunakan empat dataset, keseluruhan model T5 menduduki empat teratas dengan nilai 0.84 - 0.99. Hal ini berbeda jauh dengan nilai XLNet dan ERNIE yang memiliki nilai F1score di bawah 0.68. **Kesimpulan.** Model arsitektur T5 memiliki kinerja paling baik, terlebih jika dilatih menggunakan dataset yang besar seperti gabungan keseluruhan data abstrak dan wikipedia. Pengembangan *website* menggunakan Streamlit dapat diakses melalui <https://predict-gpt-sisfo.streamlit.app/>.

Kata kunci: deteksi ChatGPT; pengembangan website; evaluasi performa

ABSTRACT

IHLASUL MUFTI FAQIH. **Comparative Performance Analysis of XLNet, T5, and ERNIE Model Architectures for AI-Generated Text Detection** (supervised by Edy Saputra Rusdi dan A. Muh. Amil Siddik).

Background. The development of artificial intelligence in text generation has progressed rapidly, offering significant benefits but also potential for misuse. **Objective.** This study aims to compare the XLNet, T5, and ERNIE model architectures to determine the best-performing model for detecting AI-generated text and to implement it on a website. **Methods.** The datasets used includes research abstracts (14,000 rows) and Wikipedia (210,000 rows) texts, labeled as human and machine. The dataset then combined to produce a total of four types of datasets with both full and balanced quantities. The study uses accuracy, precision, recall, F1 score, and the resources used by the models to evaluate performance. Hugging Face was used to upload the models, and Streamlit was used as the website development framework. **Results.** Models trained with large datasets, such as the full combination of abstracts and Wikipedia, achieved the highest results compared to other datasets. T5 achieved an accuracy of 97.2%, compared to XLNet (64.62%) and ERNIE (61.56%). In terms of F1 score across the four datasets, the T5 model consistently ranked among the top four with scores ranging from 0.84 to 0.99. This is significantly higher than XLNet and ERNIE, which had F1 scores below 0.68. **Conclusion.** The T5 model architecture demonstrated the best performance, especially when trained with a large dataset combining abstracts and Wikipedia. The website development using Streamlit can be accessed at <https://predict-gpt-sisfo.streamlit.app/>.

Keywords: ChatGPT detection; website development; performance evaluation

DAFTAR ISI

UCAPAN TERIMA KASIH	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR RUMUS	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Landasan Teori	3
1.6.1 Al-generated text detection	3
1.6.2 Transfer learning	3
1.6.3 Text-to-Text Transfer Transformation (T5)	4
1.6.4 XLNet	4
1.6.5 ERNIE	5
1.6.6 Framework Streamlit	6
1.6.7 Tensorflow	6
1.6.8 Hugging Face	7
BAB II METODE PENELITIAN	8
2.1 Metode Kuantitatif (<i>Confusion Matrix</i>)	8
2.2 Sumber Data	9
2.3 Waktu dan Tempat Penelitian	10
2.3.1 Waktu penelitian	10
2.3.2 Tempat penelitian	10
2.4 Tahapan Penelitian	10
2.4.1 Datasets tunggal	10
2.4.2 Datasets gabungan	11
2.4.3 Detail tahapan penelitian	12

2.5 Rancangan Aplikasi Web	13
2.5.1 Halaman beranda	13
2.5.2 Halaman analisis hasil	14
2.5.3 Halaman prediksi teks	18
2.6 Instrumen Penelitian	19
2.6.1 Perangkat keras (<i>hardware</i>)	19
2.6.2 Perangkat lunak (<i>software</i>).....	20
BAB III HASIL DAN PEMBAHASAN	22
3.1 Proses Pelatihan Model	22
3.1.1 Penyesuaian dataset	22
3.1.2 Pemrosesan teks	23
3.1.3 <i>Hyperparameter</i>	24
3.1.4 Implementasi model <i>transfer learning</i>	25
3.2 Perbandingan Kinerja Model	26
3.2.1 Penggunaan sumber daya.....	26
3.2.2 Akurasi <i>training</i> dan <i>validation</i>	27
3.2.3 <i>Confusion matrix</i>	31
3.3 Publikasi Model	33
3.4 Implementasi Website	34
3.3.1 Tampilan beranda	35
3.3.2 Tampilan analisis hasil	36
3.3.3 Tampilan prediksi teks	40
3.5 Analisis Kinerja Website	42
BAB IV PENUTUP	44
4.1 Kesimpulan	44
4.2 Saran	44
DAFTAR PUSTAKA	45
LAMPIRAN	47

DAFTAR TABEL

Nomor urut	Halaman
1. Confusion matrix yang digunakan pada penelitian	8
2. Distribusi dataset.....	9
3. Waktu penelitian.....	10
4. Perangkat keras	20
5. Perangkat lunak	20
6. Pengelompokan dataset.....	22
7. Tahapan preprocessing text dan contohnya.....	23
8. Hyperparameter tuning.....	24
9. Detail transfer learning	25
10. Penggunaan sumber daya	26
11. Akurasi training	28
12. Akurasi validation	29
13. Perbandingan tingkat akurasi website yang dikembangkan dengan ZeroGPT terhadap teks AI/machine atau teks yang mengalami Humanize	42

DAFTAR GAMBAR

Nomor urut	Halaman
1. Penerapan berbagai tugas teks pada T5 (Raffel et al., 2020)	4
2. (a) Content stream attention standar seperti self-attention. (b) Query stream attention yang tidak memiliki akses informasi terhadap content. (c) Gambaran umum training permutation language modelling dengan two-stream attention (Yang et al., 2019)	5
3. (a) Menunjukkan dua encoder utama: K-Encoder dan T-Encoder. (b) Detail proses agregasi (Zhang et al., 2019).....	5
4. Tahapan mengunduh dataset dari Hugging Face.....	11
5. Tahapan menggabungkan dataset. (a) Balance Merged Dataset. (b) Full Merged Dataset.....	12
6. Detail tahapan penelitian.....	13
7. Wireframe – halaman beranda.....	14
8. Wireframe – halaman model analysis	15
9. Wireframe – model analysis menampilkan tabel dan grafik accuracy	16
10. Wireframe – model analysis menampilkan grafik confusion matrix	17
11. Wireframe – halaman prediksi GPT	18
12. Wireframe – hasil dari prediksi teks.....	19
13. Grafik akurasi training model.....	30
14. Grafik akurasi validation model	31
15. Grafik precision scores.....	32
16. Grafik recall scores.....	32
17. Grafik F1-scores.....	33
18. Publikasi model di Hugging Face	34
19. Tampilan beranda website.....	35
20. Tampilan halaman analisis hasil.....	37
21. Tampilan hasil akurasi	38
22. Tampilan hasil confusion matrix	39
23. Tampilan halaman prediksi GPT	40
24. Tampilan hasil prediksi	41

DAFTAR RUMUS

Nomor urut	Halaman
1. Pengukuran <i>confusion matrix</i> (<i>accuracy</i>)	8
2. Pengukuran <i>confusion matrix</i> (<i>precision</i>)	9
3. Pengukuran <i>confusion matrix</i> (<i>recall</i>)	9
4. Pengukuran <i>confusion matrix</i> (<i>F1 score</i>).....	9

DAFTAR LAMPIRAN

Nomor urut	Halaman
1. <i>Repository machine learning</i>	47
2. <i>Repository aplikasi website</i>	48
3. <i>Repository model T5</i>	49
4. <i>Website prediksi teks kecerdasan buatan</i>	50
5. Hasil pengujian <i>website</i>	51
6. Biodata diri	52

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam perkembangan teknologi bidang *Natural Language Processing (NLP)* terbaru, model-model berbasis *Transformer* telah banyak berkembang dan menjadikannya sebagai arsitektur untuk tugas-tugas seperti memahami dan menghasilkan teks. Salah satu model arsitektur yang paling sering digunakan dan menjadi landasan pengembangan arsitektur transformer lainnya yaitu *BERT (Bidirectional Encoder Representations from Transformer)* (Devlin et al., 2019).

Pengembangan *website* pendukung sebagai alat dalam menghasilkan *AI Generated text* sudah banyak tersebar. Beberapa *website* yang terkenal seperti ChatGPT (OpenAI, 2023) dan Gemini (Google et al., 2023) dapat menghasilkan teks dengan hasil yang sangat baik. Perkembangan ini tentu bagaikan pisau yang memiliki sisi positif dan negatif tergantung pemakainya.

Salah satu contoh positif penggunaannya seperti membantu dalam *decision making* (saran keputusan) terhadap suatu masalah (Bouschery et al., 2023; Javaid et al., 2023; S. Liu et al., 2023). Dalam hal sisi negatif, Sadasivan et al. (2023) menyimpulkan penggunaan AI dapat digunakan untuk plagiarisme, membuat berita palsu, melakukan spam, memberi ulasan palsu, dan manipulasi konten *website* yang merugikan masyarakat (Adelani et al., 2020; Weiss, 2019). Maka penelitian terus berlanjut untuk mengembangkan model yang mampu mengidentifikasi dengan baik perbedaan teks yang dihasilkan oleh kecerdasan buatan atau manusia. Hal ini untuk mencegah penyalahgunaan terhadap model-model NLP.

Salah satu model arsitektur hasil pengembangan dari *BERT* yaitu *RoBERTa* memiliki kinerja yang baik dalam melakukan tugas *AI-Generated Text Detection*, bahkan dalam studi yang dilakukan Y. Liu et al (2023) memiliki akurasi 99% pada tingkat dokumen dan akurasi 93% untuk tingkat kalimat. *RoBERTa* juga terbukti berhasil memiliki performa yang baik dalam mendeteksi teks berita yang dibuat oleh kecerdasan buatan (Bhattacharjee et al., 2023). Terdapat juga beberapa *website* yang telah dikembangkan untuk mendeteksi teks hasil kecerdasan buatan seperti ZeroGPT (ZeroGPT, 2024).

Penelitian yang selalu berkembang menghasilkan beberapa model Arsitektur lain seperti *XLNet*, *T5*, dan *ERNIE*. Dalam sebuah penelitian, *XLNet* menggabungkan metode *autoregressive (AR)* dan *autoencoding (AE)* memiliki performa yang lebih bagus terhadap beberapa tugas jika dibandingkan dengan model *BERT* dan *RoBERTa* (Yang et al., 2019). Adapun model *ERNIE* yang juga merupakan pengembangan dari *BERT*, dimana struktur *BERT* dikombinasikan dengan *K-encoder (knowledge encoder)* yang terhubung dengan tumpukan agregator. Hasil eksperimen menunjukkan bahwa *ERNIE* jauh lebih unggul melakukan *denoising distantly supervised data* dan *fine-tuning* untuk data yang terbatas dibandingkan *BERT* (Zhang et al., 2019). *T5* atau *text-to-text transfer transformer* yang mencapai

hasil *state-of-the-art* dengan skalabilitas sehingga menyederhanakan berbagai macam tugas *Natural Language Processor* atau *NLP* (Raffel et al., 2020).

Setiap model tentunya memiliki kelebihan dan kekurangannya masing-masing. Penggunaan model arsitektur *XLNet*, *T5*, dan *ERNIE* dalam *AI-Generated Text Detection* perlu diuji dan dibandingkan secara komprehensif. Hal ini demi kemajuan penelitian untuk mencegah penyalahgunaan *Text Generated* di masa yang akan datang. Selain itu, Kekurangan dari penelitian sebelumnya mengenai pendeteksian teks hasil kecerdasan buatan yaitu tidak terdapatnya *website* ataupun hasil yang dapat digunakan langsung. Adapun ketika terdapat *website* untuk digunakan namun tidak memiliki publikasi untuk dipelajari untuk menjaga rahasia dari perusahaan tersebut.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan dan membandingkan model arsitektur *XLNet*, *T5*, dan *ERNIE* dalam *AI-Generated Text Detection*. Dengan menggabungkan pemahaman mendalam tentang bahasa yang dimiliki oleh masing-masing model, serta kekuatan pemrosesan teks mereka, penelitian ini diharapkan dapat memberikan wawasan berharga tentang kinerja dari ketiga model tersebut dalam mendeteksi *Generated Text*.

1.2 Rumusan Masalah

Adapun beberapa masalah yang ada dapat dirumuskan menjadi:

1. Bagaimana perbandingan kinerja model *XLNet*, *T5*, dan *ERNIE* terhadap tugas *AI-Generated Text Detection*?
2. Bagaimana cara mengaplikasikan model menjadi sebuah *website*?
3. Bagaimana kinerja model setelah diterapkan ke dalam sebuah *website*?

1.3 Batasan Masalah

Penelitian ini berfokus pada kinerja model yang digunakan untuk mendeteksi teks kecerdasan buatan. Batasan berikut dibuat untuk menjaga fokus dari penelitian.

1. Arsitektur model yang digunakan yaitu *XLNet*, *T5*, dan *ERNIE*, sehingga penelitian ini akan menerapkan metode *transfer learning*.
2. Menggunakan *Streamlit* sebagai *framework* untuk pengaplikasian model ke dalam sebuah *website* sederhana.
3. *AI-Generated Text Detection* hanya pada tingkat kalimat dan berbasis Bahasa Inggris.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu:

1. Mengetahui perbandingan kinerja model *XLNet*, *T5*, dan *ERNIE* terhadap tugas *AI-Generated Text Detection*.
2. Mengaplikasikan model yang sudah ada ke dalam sebuah *website*.
3. Melihat kinerja model setelah diterapkan ke dalam sebuah *website*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memiliki manfaat sebagai berikut:

1. Membantu dalam mengenali sebuah teks adalah hasil kecerdasan buatan.
2. Mengatasi kekhawatiran masyarakat terkait keaslian sebuah teks (berita, esai, tugas siswa, dll).
3. Pembelajaran dan referensi untuk penelitian-penelitian selanjutnya.

1.6 Landasan Teori

1.6.1 AI-generated text detection

AI-Generated Text atau *Machine Generated Texts* juga disebut sebagai *neural fake texts* adalah teks yang dibuat menggunakan model bahasa dan menirukan gaya penulisan manusia sehingga manusia cukup sulit untuk membedakannya (Ippolito et al., 2020). Kemudian, menurut Kamus Besar Bahasa Indonesia (KBBI) deteksi adalah usaha menemukan dan menentukan keberadaan, anggapan, atau kenyataan (KBBI, n.d.).

Berdasarkan kedua pengertian tersebut, *AI-Generated Text Detection* adalah sebuah proses atau usaha dalam menemukan dan mengidentifikasi sebuah teks yang dihasilkan oleh kecerdasan buatan yang menirukan gaya penulisan manusia dengan sedemikian rupa sehingga sulit dibedakan secara langsung dengan teks yang dihasilkan oleh manusia. Proses ini menggunakan teknik atau alat untuk mengidentifikasi atau membedakan teks hasil kecerdasan buatan dari teks manusia, seringkali untuk tujuan seperti verifikasi keaslian atau pemantauan terhadap teks yang mungkin tidak sah atau merugikan.

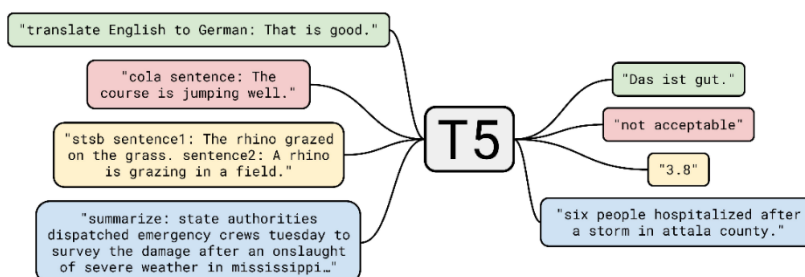
1.6.2 Transfer learning

Menurut (Goodfellow et al., 2016) dalam bukunya *Deep Learning (Adaptive Computation and Machine Learning series)*, *Transfer Learning* dan *domain adaption* merujuk pada situasi dimana sesuatu yang sudah dipelajari pada suatu pengaturan akan dieksplorasi untuk meningkatkan generalisasi pada pengaturan yang lain, artinya suatu model yang telah mengalami *training* pada suatu tugas akan digunakan kembali pada tugas lain yang mirip atau terkait (Brownlee, 2019). *Transfer Learning* akan membantu peneliti dalam melakukan *training* yang lebih cepat serta keuntungan lainnya seperti mengurangi kebutuhan data *training*, *multitask learning* yang memungkinkan model menangani beberapa tugas sekaligus serta hal lainnya.

Transfer Learning, dimana model telah menjalani *pre-training* pada banyak data dengan berbagai tugas sebelum nantinya akan dilakukan penyesuaian kembali (*fine-tuning*) pada tugas lain memiliki kekuatan yang luar biasa untuk digunakan pada tugas NLP. Keefektifan dari *transfer learning* telah melahirkan keragaman pendekatan, metodologi, dan praktik dalam penelitian (Raffel et al., 2020).

1.6.3 Text-to-Text Transfer Transformation (T5)

Sesuai dengan namanya, T5 atau *Text-to-Text Transfer Transformation* menerapkan metode *text-to-text* dimana model akan mengambil teks sebagai input dan memproduksi teks baru sebagai output. Kerangka kerja *text-to-text* memungkinkan untuk secara langsung menerapkan model, tujuan, prosedur training, dan proses decoding yang sama untuk setiap tugas. Memanfaatkan fleksibilitas dari cara kerja tersebut, T5 dapat menangani berbagai macam masalah NLP seperti menjawab pertanyaan, meringkas dokumen, dan klasifikasi sentiment (Raffel et al., 2020). T5 menggunakan “*Colossal Clean Crawled Corpus*” (C4) datasets dalam model pre-trained.



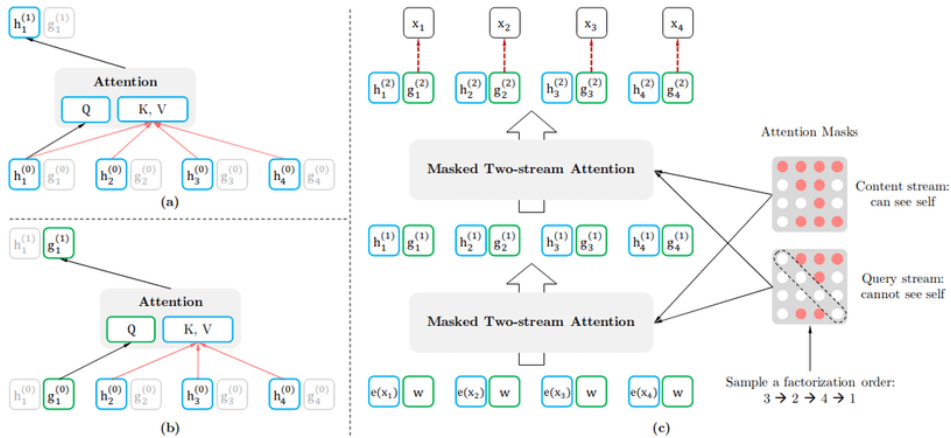
Gambar 1 Penerapan berbagai tugas teks pada T5 (Raffel et al., 2020)

Gambar 1 memberikan contoh penerapan berbagai tugas NLP seperti *translation* (penerjemahan), *classification* (klasifikasi), *similarity* (kesamaan), dan *summarized* (menyimpulkan) pada satu model T5. Setiap *input* (masukan) yang diberikan pada model T5 berupa teks dan juga akan menghasilkan *output* (keluaran) berupa teks. Hal tersebut dapat dilihat pada kasus *similarity* yang menghasilkan output angka yang tetap berbentuk teks.

1.6.4 XLNet

XLNet adalah metode *pre-training* generalisasi *auto-regressive* (AR) yang menggabungkan keunggulan dari kedua pendekatan *auto-regresif* (AR) dan *auto-encoding* (AE) melalui objektif *permutation language modelling*. Arsitektur *neural network* *XLNet* dirancang secara cermat untuk berkolaborasi dengan objektif AR, dengan mengintegrasikan elemen seperti Transformer-XL dan mekanisme *two-stream* yang dirancang dengan baik. Arsitektur *XLNet* dapat dilihat pada gambar 2.

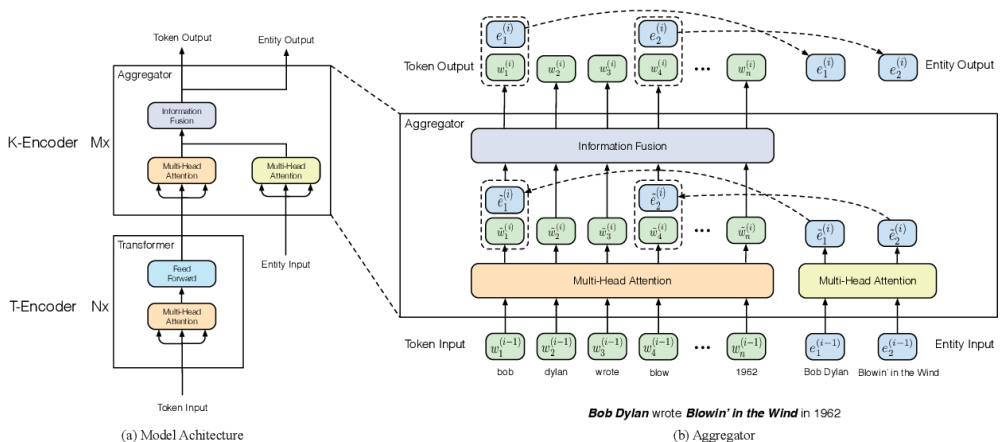
Hasilnya, *XLNet* menunjukkan peningkatan kinerja yang signifikan dibandingkan dengan objektif *pre-training* sebelumnya dalam berbagai tugas NLP. Pendekatan ini mencerminkan kemajuan berkelanjutan dalam teknik-teknik NLP dan menunjukkan potensi untuk lebih meningkatkan pemahaman bahasa dalam model-model yang ada (Yang et al., 2019). *XLNet* merupakan model arsitektur *pre-trained* yang menggunakan datasets *bookcorpus* dan *wikipedia*.



Gambar 2 (a) Content stream attention standar seperti self-attention. (b) Query stream attention yang tidak memiliki akses informasi terhadap content. (c) Gambaran umum training permutation language modelling dengan two-stream attention (Yang et al., 2019)

1.6.5 ERNIE

ERNIE atau Enhanced Language Representation with Informative Entities merupakan *pre-trained* model berdasarkan *large-scale textual corpora* atau kumpulan data teks yang sangat besar disertai dengan *knowledge graph*. ERNIE menggunakan *english wikipedia* sebagai *pre-training* datanya. Model ini dapat mengambil leksikal (makna/pemakaian kata), sintaksis (struktur/susunan kalimat), dan informasi pengetahuan dari teks secara bersamaan. ERNIE mencapai peningkatan performa terhadap tugas NLP jika dibandingkan dengan BERT yang memiliki performa hampir sama (Zhang et al., 2019). Untuk melihat cara kerja arsitektur ERNIE dapat dilihat pada gambar berikut.



Gambar 3 (a) Menunjukkan dua *encoder* utama: K-Encoder dan T-Encoder. (b) Detail proses agregasi (Zhang et al., 2019)

Model ERNIE memiliki arsitektur yang dirancang untuk meningkatkan representasi bahasa dengan mengintegrasikan entitas informatif. K-Encoder dirancang untuk menangani *input* entitas dengan modul *Information Fusion* dan *Multi-Head Attention*. Sementara itu, T-Encoder menangani *input* token dengan menggunakan modul *Transformer* dan *Multi-Head Attention*. Kedua *encoder* ini mengeluarkan token dan entitas yang kemudian digabungkan oleh agregator. Agregator ini bertugas untuk menggabungkan informasi dari token dan entitas menggunakan mekanisme *fusion* dan *attention*, yang memungkinkan model untuk memahami hubungan antara kata-kata dan entitas secara lebih mendalam.

Input token dan entitas pertama kali diproses oleh lapisan *Multi-Head Attention* untuk menangkap informasi kontekstual. Selanjutnya, lapisan *Information Fusion* menggabungkan informasi dari berbagai sumber, menghasilkan *output* token dan entitas yang lebih kaya akan informasi. Misalnya, kalimat "**Bob Dylan wrote Blowin' in the Wind in 1962**" diproses untuk mengenali entitas seperti "**Bob Dylan**" dan "**Blowin' in the Wind**", serta hubungan temporal dengan tahun **1962**. Dengan pendekatan ini, ERNIE mampu menangkap informasi yang lebih kompleks dan kontekstual, meningkatkan pemahaman bahasa yang dihasilkan oleh model.

1.6.6 Framework Streamlit

Streamlit adalah sebuah python library *open-source* yang berfungsi mempermudah dalam membuat dan membagikan web untuk machine learning dan data science (Streamlit Inc, 2023). Streamlit didesain sehingga data scientists dan engineers untuk mengubah secara cepat dan mudah data scripts menjadi web interaktif tanpa perlu memiliki keahlian pengembangan web yang luas. Streamlit sangat populer karena ramah pengguna dan efisien untuk membangun *website* yang berpusat pada data.

1.6.7 Tensorflow

TensorFlow adalah sebuah framework untuk membangun algoritma machine learning serta menjalankan algoritma-algoritma tersebut. Sebuah komputasi yang dibangun menggunakan TensorFlow dapat dijalankan dengan sedikit atau tanpa perubahan pada beragam sistem yang berbeda, mulai dari perangkat mobile seperti ponsel dan tablet hingga sistem terdistribusi berskala besar yang terdiri dari ratusan mesin dan ribuan perangkat komputasi seperti GPU. Sistem ini fleksibel dan dapat digunakan untuk menyatakan berbagai algoritma, termasuk algoritma *training* dan *inference* untuk model *deep neural networks*, dan telah digunakan untuk melakukan penelitian serta menerapkan sistem machine learning dalam produksi di lebih dari selusin bidang ilmu komputer dan bidang lainnya, termasuk *speech recognition*, *computer vision*, *robotics*, pencarian informasi, *natural language processing*, ekstraksi informasi geografis, dan penemuan obat komputasional (Abadi et al., 2016)

1.6.8 Hugging Face

Hugging Face merupakan sebuah platform berbasis *website* komunitas yang berfokus pada pengembangan model kecerdasan buatan (umumnya NLP). Pada Hugging Face seseorang dapat membagikan model, datasets, atau pengaplikasian hasil dari penelitian atau pekerjaan mereka sehingga orang lain dapat melihat, mempelajari, atau menggunakan hasil pekerjaan mereka (Hugging Face, n.d.).

BAB II METODE PENELITIAN

2.1 Metode Kuantitatif (*Confusion Matrix*)

Metode yang dilakukan dalam penelitian ini menggunakan metode kuantitatif. Salah satu cara dalam melakukan evaluasi terhadap model *machine learning* kasus klasifikasi biner (hanya terdapat dua kelas/label) yaitu dengan *confusion matrix* (matriks kerancuan). Detail dari *confusion matrix* dapat dilihat pada tabel berikut.

Tabel 1 *Confusion matrix* yang digunakan pada penelitian

Confusion Matrix		Actual Values	
		Positive (AI/Machine)	Negative (Human)
Predicted Values	Positive (AI/Machine)	True Positive (TP)	False Positive (FP)
	Negative (Human)	False Negative (FN)	True Negative (TN)

Powers (2008) mendeskripsikan setiap baris pada matriks merepresentasikan nilai prediksi dan setiap kolom merepresentasikan nilai asli. *Confusion matrix* memberikan jumlah perbandingan *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Dalam penelitian ini nilai positif representasi untuk kelas *AI/machine* sedangkan nilai negatif representasi untuk kelas *human*.

Pada tabel 1 warna hijau menandakan model berhasil memprediksi dengan tepat sedangkan warna merah menandakan model salah dalam memprediksi kelas yang sebenarnya. Misalkan pada *true positive* yang berwarna hijau artinya teks yang memiliki kelas *AI/machine* diprediksi oleh model *machine learning* sebagai *AI/machine*. Pada kasus salah prediksi yang ditandakan dengan warna merah, misalkan *false negative* artinya teks yang memiliki kelas *AI/machine* diprediksi oleh model *machine learning* sebagai *human*.

Pengukuran pada *confusion matrix* terdapat empat penilaian yaitu *accuracy*, *precision*, *recall*, dan *F1-Score*. Setiap penilaian memiliki fungsi dan perhitungannya masing-masing.

Accuracy. Penilaian paling umum pada model *machine learning* yaitu menggunakan *accuracy*. Menghitung *accuracy* dapat dilakukan dengan menjumlahkan seluruh prediksi yang benar dibagi dengan jumlah data. Rumus dalam menghitung *accuracy* dapat dilihat melalui persamaan berikut.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Precision. Nilai *precision* berfungsi saat suatu model sebisa mungkin ingin menghindari kesalahan dalam memprediksi hasil positif. Salah satu contohnya yaitu mendeteksi *email spam*. Nilai *precision* yang tinggi dibutuhkan agar terhindar dari kesalahan mendeteksi *email* penting sebagai sebuah *spam*. Rumus dalam menghitung *precision* dapat dilihat pada persamaan berikut.

$$precision = \frac{TP}{TP + FP} \quad (2)$$

Recall. Nilai *recall* berfungsi saat suatu model sebisa mungkin ingin menghindari kesalahan dalam memprediksi hasil negatif. Salah satu contohnya yaitu mendeteksi penyakit kritis seperti kanker. Nilai *recall* yang tinggi dibutuhkan agar terhindar dari kesalahan mendeteksi seseorang yang terkena kanker sebagai seseorang yang aman dari kanker. Rumus dalam menghitung *recall* dapat dilihat pada persamaan berikut.

$$recall = \frac{TP}{TP + FN} \quad (3)$$

F1 score. Nilai harmoni atau keseimbangan diantara *precision* dan *recall* merupakan definisi dari *F1 score*. Penggunaan *F1 score* biasanya ketika model dilatih menggunakan data yang tidak seimbang. Selain itu, *F1 score* juga digunakan ketika ingin mendapatkan nilai terbaik dari *precision* dan *recall*. Rumus dalam menghitung *F1 score* dapat dilihat pada persamaan berikut.

$$F1\ score = 2 * \frac{(precision * recall)}{(precision + recall)} \quad (4)$$

Dalam penelitian ini penggunaan matriks *recall* dan *F1 score* sangat penting. Penggunaan *recall* diperlukan karena model *machine learning* yang dilatih sebisa mungkin dapat menghindari kesalahan dengan melewati teks buatan *AI/machine* sebagai teks buatan *human*. Selain itu, *F1 score* juga dibutuhkan karena dalam penelitian ini akan dilakukan *training* pada dataset yang tidak seimbang jumlahnya.

2.2 Sumber Data

Penelitian ini berfokus dalam mengklasifikasi *human-machine text*. Data yang akan digunakan pada penelitian ini menggunakan *human-vs-machine datasets* yang tersedia pada Hugging Face (Sivesind & Winje, 2023). Dataset ini terbagi menjadi dua tipe yaitu *research-abstracts* dan *wikipedia* teks. Detail dataset pada tabel 2.

Tabel 2 Distribusi dataset

No.	Type	Label	Train	Validation	Test	Total
1.	Research Abstract	Human	7.000	1.500	1.500	10.000
		Machine	7.000	1.500	1.500	10.000
2	Wikipedia	Human	105.000	22.500	22.500	150.000
		Machine	105.000	22.500	22.500	150.000

Pembagian data seimbang antara *human* dan *machine* yaitu 50%, penelitian ini nantinya akan mengetes kinerja terhadap data tipe *research abstract*, *wikipedia*, serta gabungan keduanya.

2.3 Waktu dan Tempat Penelitian

2.3.1 Waktu penelitian

Penelitian ini berlangsung kurang lebih selama empat bulan. Adapun untuk detail waktu penelitian dapat dilihat pada tabel berikut.

Tabel 3 Waktu penelitian

Kegiatan	April				Maret				Mei				Juni						
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4			
<i>Data Collecting</i>																			
<i>Data Preprocessing</i>																			
<i>Build and Train Model</i>																			
<i>Model Evaluation</i>																			
<i>Model Deploying</i>																			
<i>Build Streamlit Website</i>																			
<i>Website Testing</i>																			

2.3.2 Tempat penelitian

Penelitian ini akan dilaksanakan di Lab Rekayasa Perangkat Lunak (RPL), Program Studi Sistem Informasi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin.

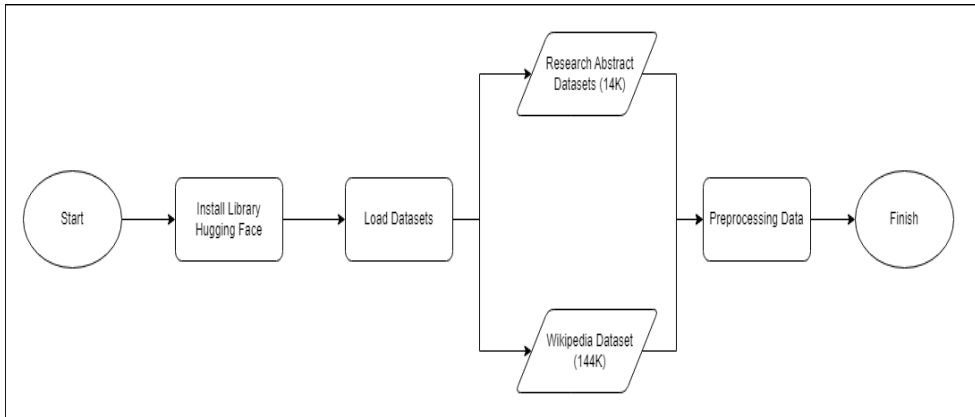
2.4 Tahapan Penelitian

Pada penelitian kali ini terdapat empat tipe dataset yang akan diuji. Setiap tahapan yang dialami dataset tentunya sama. Tahapan-tahapan tersebut dapat dilihat pada penjelasan lebih lanjut pada bagian ini.

2.4.1 Datasets tunggal

Dataset *human-vs-machine* yang terdapat pada Hugging Face terdiri dari *wikipedia* dan *research abstract text*. Pada penelitian ini akan menguji kinerja ketiga model terhadap kedua dataset tersebut secara terpisah. Tahap pertama yaitu mengunduh

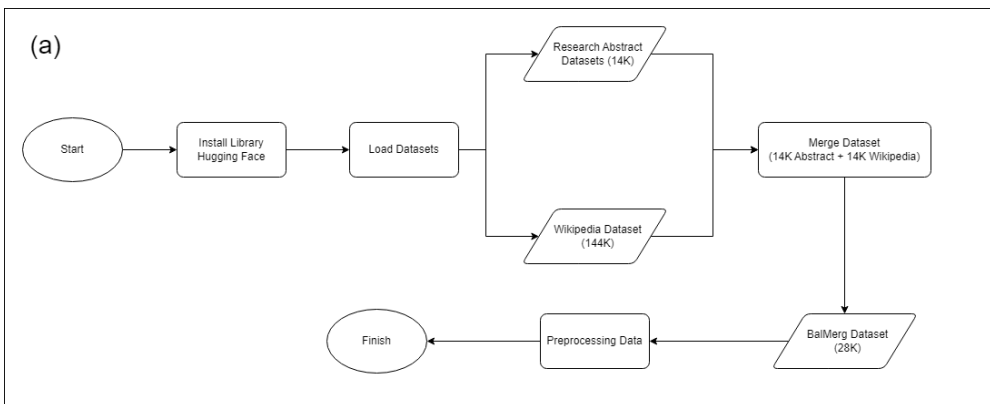
dataset yang ada pada Hugging Face. Tahapan alur dari proses tersebut dapat dilihat pada gambar di bawah.

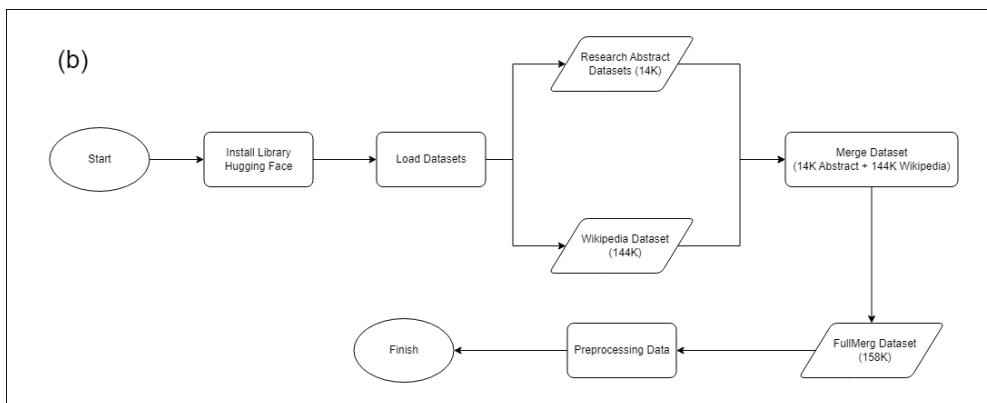


Gambar 4 Tahapan mengunduh dataset dari Hugging Face

2.4.2 Datasets gabungan

Penelitian ini juga akan menguji kinerja ketiga model dengan menggabungkan kedua jenis dataset (*research abstract* dan *wikipedia*). Penggabungan pertama akan menggabungkan semua jumlah data dan untuk penggabungan kedua akan diseimbangkan jumlah kedua jenis teks. Untuk alur kerja penggabungan dataset dapat dilihat pada gambar di bawah ini.





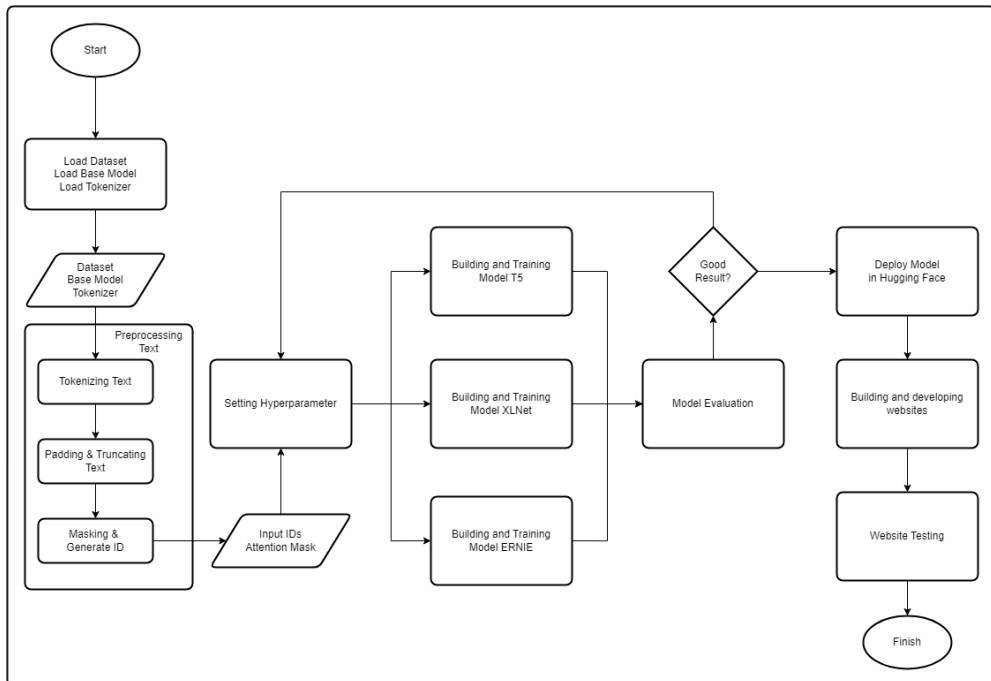
Gambar 5 Tahapan menggabungkan dataset. (a) *Balance Merged Dataset*. (b) *Full Merged Dataset*

2.4.3 Detail tahapan penelitian

Setiap tahapan yang dilalui oleh masing-masing jenis datasets sama. Pada bagian ini akan membahas alur kerja penelitian secara mendalam mulai dari *preprocessing text* (prapemrosesan teks) yang terdiri dari *tokenizing*, *padding*, *truncating*, *masking*, dan *generate ID*. Tahap *preprocessing text* ini akan menghasilkan *input IDs* dan *attention mask* yang menjadi *input* pada model.

Tahap selanjutnya yaitu mengatur parameter yang akan mempengaruhi jalannya proses *training*. Selanjutnya, melakukan *training* dan pada ketiga model dan menganalisa hasilnya. Model terbaik setelah dilakukan analisa akan diunggah pada Hugging Face agar dapat digunakan pada *website* yang akan dikembangkan. Model juga dapat digunakan oleh masyarakat secara umum.

Tahap terakhir dalam penelitian ini yaitu mengembangkan *website*. Pengembangan ini bertujuan agar model dapat digunakan secara langsung dengan tampilan yang mudah digunakan. Tahap pengujian juga akan dilakukan pada *website* yang telah dikembangkan untuk mengetahui performa dan kinerjanya. Tahapan pada penelitian ini dapat dilihat lebih jelas pada gambar berikut.



Gambar 6 Detail tahapan penelitian

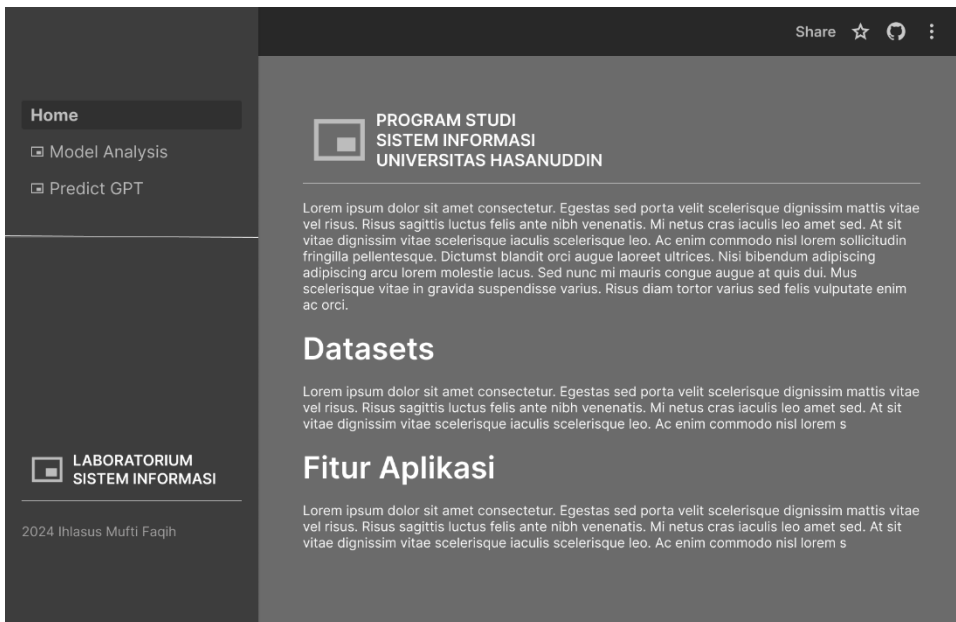
2.5 Rancangan Aplikasi Web

Pengembangan *website* pada penelitian ini menggunakan bahasa pemrograman *python* sebagai bahasa utama. Penggunaan *python* disebabkan karena *framework* yang digunakan yaitu Streamlit. Selain itu, bahasa HTML dan CSS juga digunakan untuk melakukan kustom pada tampilan *website*.

Halaman tampilan yang dikembangkan pada *website* ada tiga. Pertama, halaman beranda yang menjadi tampilan pertama saat *website* diakses. Kedua, halaman model analisis yang berfungsi sebagai *dashboard* tampilan hasil penelitian. Ketiga, halaman prediksi GPT yang berfungsi sebagai fitur agar pengguna *website* dapat memasukkan teks untuk dilakukan pemeriksaan *AI/machine* atau *human*.

2.5.1 Halaman beranda

Dalam mengembangkan *website* dibutuhkan rancangan awal yang disebut *wireframe*. Pada halaman beranda akan dtampilkan berbagai informasi umum yang menjelaskan *website* yang dikembangkan. Rincian *wireframe* pada halaman beranda dapat dilihat pada gambar berikut.

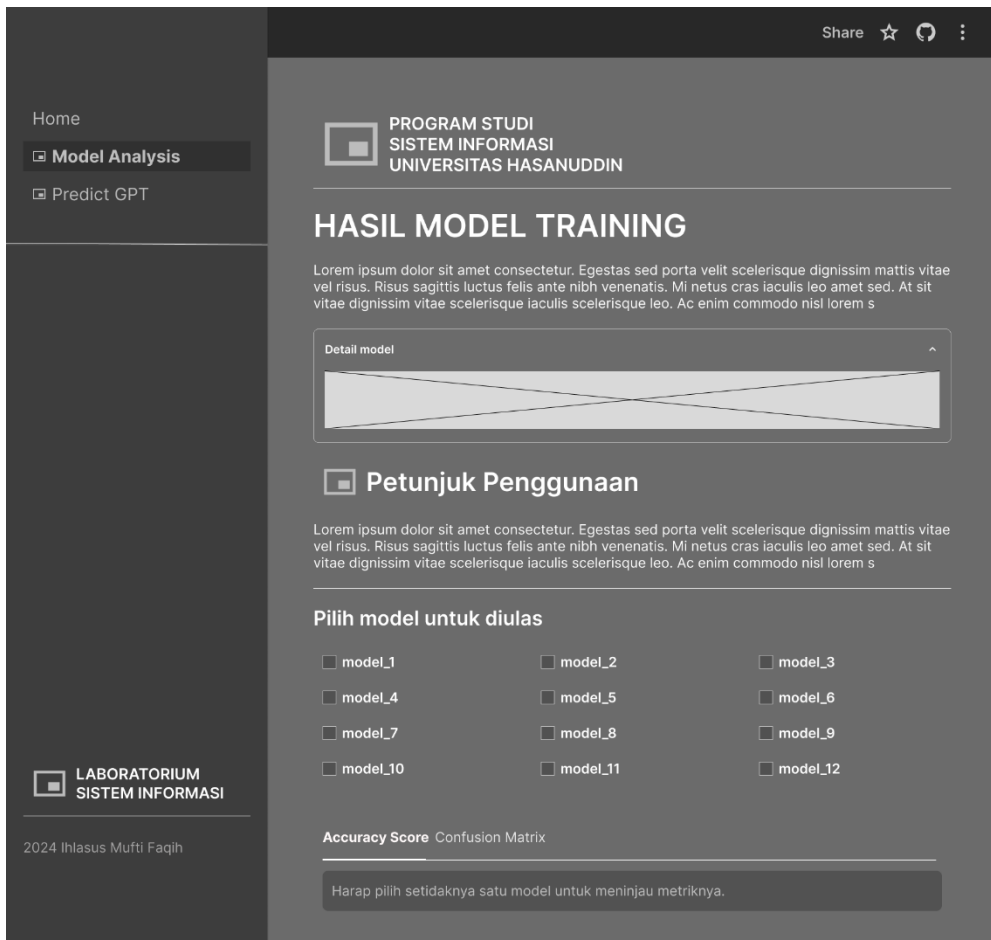


Gambar 7 Wireframe – halaman beranda

Gambar 7 memperlihatkan *wireframe* halaman beranda yang dibagi menjadi beberapa bagian utama. Tampilan sebelah kiri *website* terdapat panel navigasi dengan menu seperti **Home**, **Model Analysis**, dan **Predict GPT** serta logo laboratorium dan informasi hak cipta. Pada bagian tengah dan kanan halaman *website* menampilkan konten utama, termasuk nama program studi dan universitas di bagian atas diikuti dengan beberapa paragraf untuk deskripsi *website*. Selanjutnya, terdapat dua bagian tambahan berjudul **Datasets** dan **Fitur Aplikasi** beserta dengan setiap deskripsinya. *Wireframe* ini juga mencakup ikon berbagi di pojok kanan atas, menunjukkan bahwa pengguna dapat dengan mudah membagikan konten website ini atau mengakses *source code* pada ikon GitHub.

2.5.2 Halaman analisis hasil

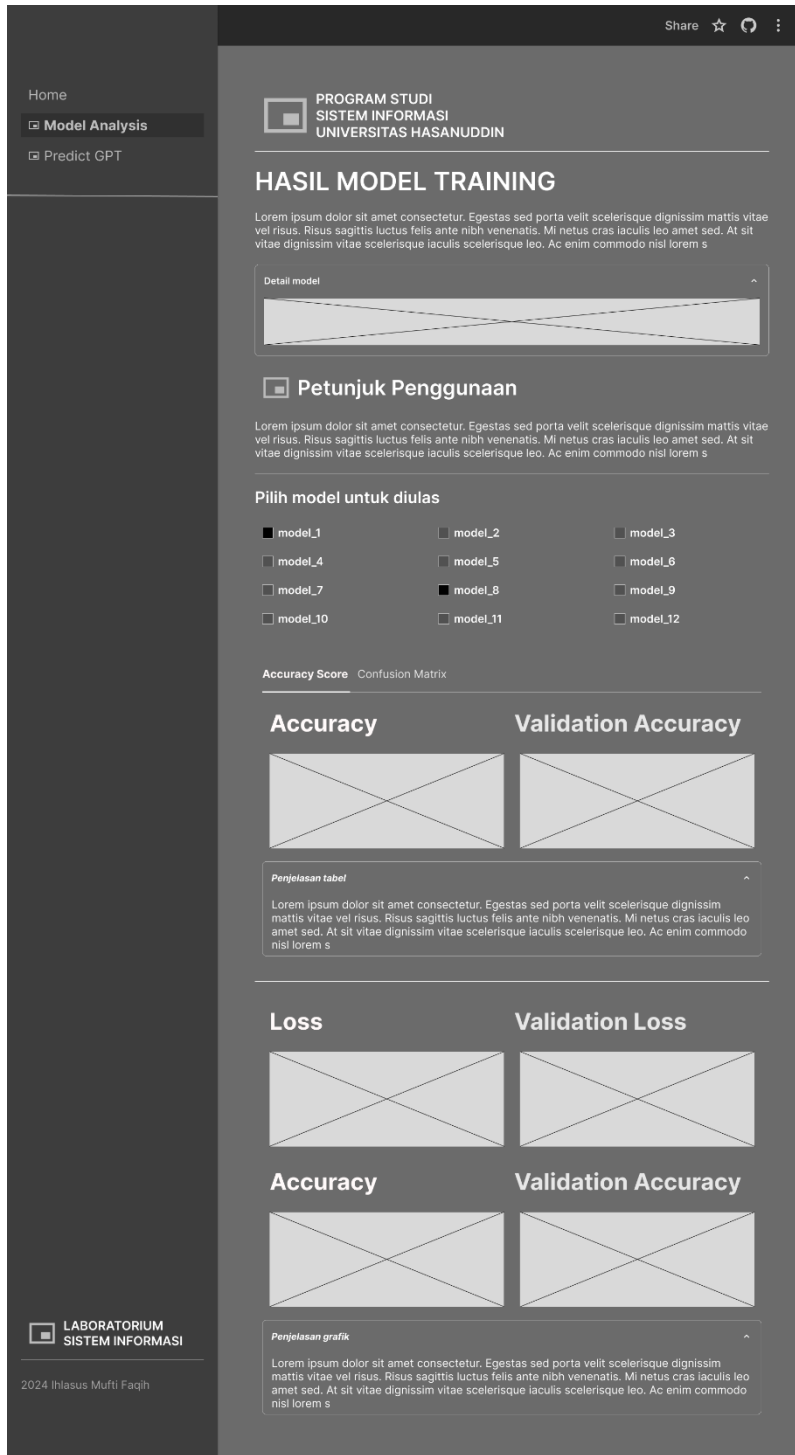
Pada *wireframe* halaman analisis hasil menggambarkan bagaimana hasil analisis model ditampilkan kepada pengguna, termasuk fitur-fitur yang memungkinkan interaksi dan eksplorasi data model. Fitur ini biasa juga disebut sebagai tampilan *dashboard*. Untuk rincian *wireframe* pada halaman analisis hasil dapat dilihat pada gambar di bawah ini.



Gambar 8 Wireframe – halaman model analysis

Gambar 8 memperlihatkan *wireframe* halaman model analysis dengan berbagai elemen interaktif. Pada sebelah kiri halaman *website* terdapat panel navigasi yang sama pada halaman beranda dengan menu **Model Analysis** yang disorot bersama dengan opsi **Home** dan **Predict GPT**. Bagian tengah dan kanan halaman menampilkan konten utama dengan judul **Hasil Model Training**. Setelah judul, terdapat deskripsi singkat dan detail model yang dapat diperluas (*expand*) untuk menampilkan tabel detail model. Terdapat juga informasi mengenai petunjuk penggunaan yang memberikan instruksi kepada pengguna dalam menggunakan *dashboard*.

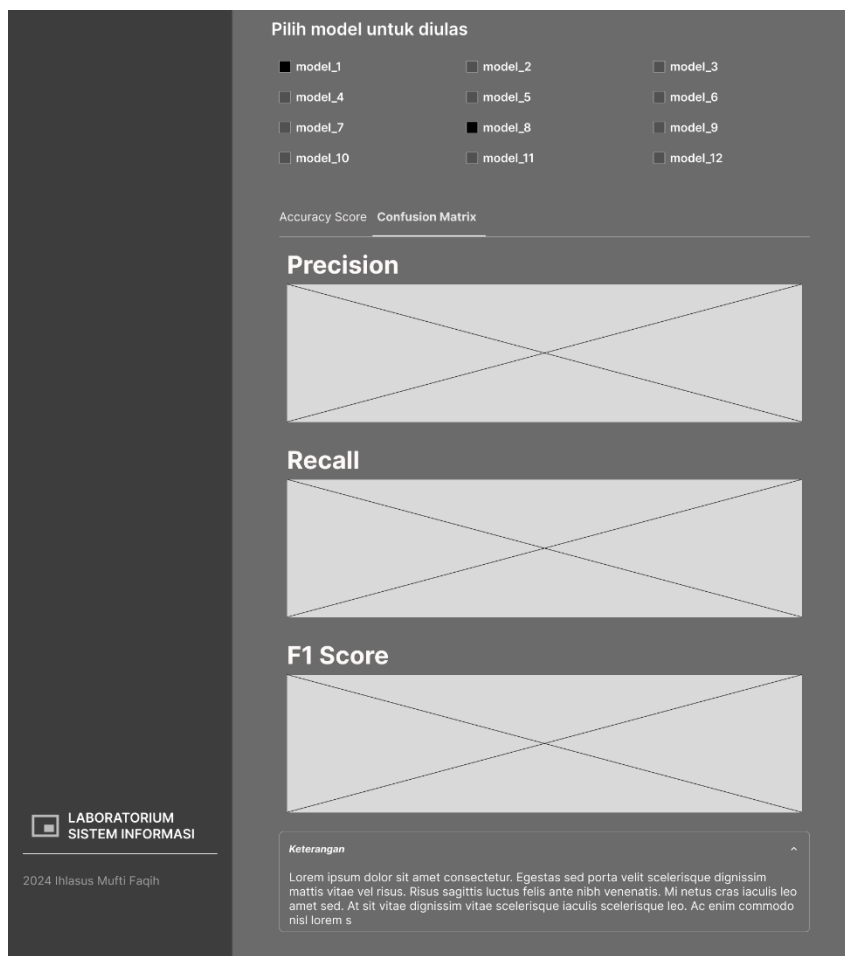
Bagian lain pada halaman terdapat daftar *checkbox* untuk memilih berbagai model yang akan dianalisis, dengan model-model bernomor dari `model_1` hingga `model_12`. Di bagian bawah halaman, terdapat *accuracy score* dan *confusion matrix* yang memberikan rincian lebih lanjut mengenai performa model. Rincian dan contoh saat memilih model dan menampilkan hasilnya dapat dilihat pada gambar berikut.



Gambar 9 Wireframe – model analysis menampilkan tabel dan grafik accuracy

Pada gambar 9, *wireframe* memperlihatkan contoh ketika terdapat model yang dipilih, maka akan menampilkan hasil. Hasil yang ditampilkan berupa tabel dan grafik. Pada tampilan tabel akan terdapat nilai *accuracy* dan *validation accuracy* dan juga penjelasan tabel yang dapat dilihat dengan memperluas (*expand*) tampilannya. Selanjutnya, pada tampilan grafik terdapat empat nilai yang ditampilkan. Nilai-nilai yang ditampilkan berupa *loss* dan *accuracy* begitu juga pada nilai *validation*. Terdapat juga penjelasan mengenai grafik yang dapat dilihat dengan cara memperluas (*expand*) tampilannya.

Selain menampilkan hasil *accuracy*, halaman ini juga menampilkan nilai *confusion matrix* dengan cara menekan **tab Confusion Matrix**. Rincian tampilan pada bagian *confusion matrix* dapat dilihat pada gambar berikut.

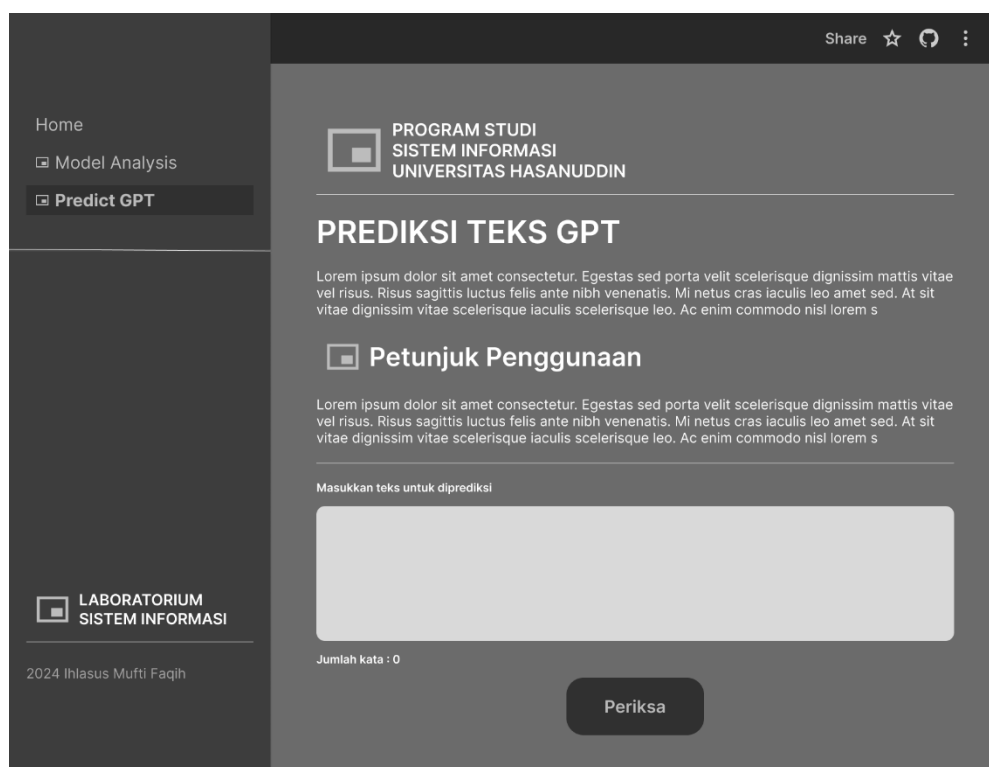


Gambar 10 Wireframe – model analysis menampilkan grafik *confusion matrix*

Masih pada halaman yang sama dan hanya berbeda *tab* saja, gambar 10 menampilkan bagian *confusion matrix* yang terdiri dari tiga nilai yaitu *precision*, *recall*, dan *F1 score*. Setiap hasil nilai *confusion matrix* ditampilkan dalam bentuk grafik. Pada bagian bawah juga terdapat keterangan mengenai hasil yang dapat dilihat dengan cara memperluas (*expand*) tampilannya.

2.5.3 Halaman prediksi teks

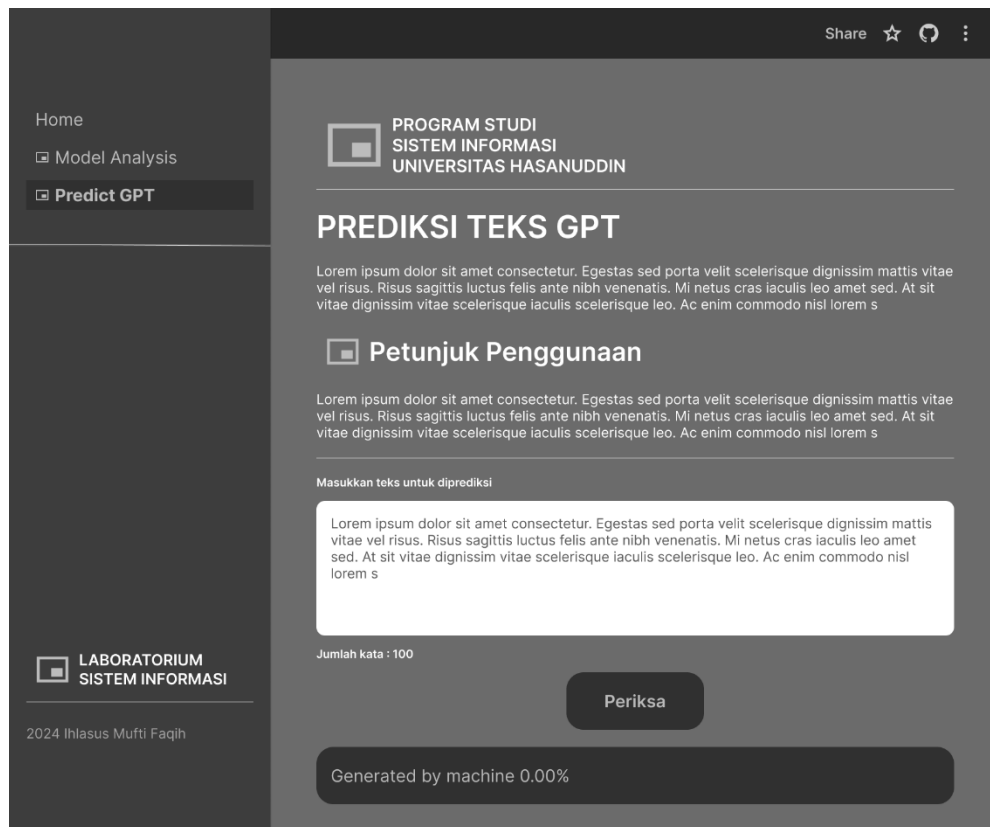
Pada *wireframe* halaman Predict GPT menggambarkan tampilan fitur yang dapat digunakan pengguna ketika ingin memeriksa teks adalah *AI/machine* atau bukan. Untuk rincian *wireframe* pada halaman Predict GPT dapat dilihat pada gambar di bawah ini.



Gambar 11 Wireframe – halaman prediksi GPT

Gambar 11 memperlihatkan *wireframe* halaman Predict GPT dengan berbagai elemen yang tidak jauh berbeda pada halaman lainnya. Hal berbeda dapat dilihat pada bagian tengah atau konten utama pada halaman ini. Pada bagian atas konten terdapat judul **Prediksi Teks GPT** beserta dengan deskripsi mengenai halaman. Selanjutnya terdapat juga petunjuk penggunaan yang memberikan instruksi atau tahapan ketika pengguna ingin melakukan pemeriksaan terhadap teks. Pada bagian bawah halaman terdapat *text area* untuk memasukkan teks beserta dengan tampilan

yang memberikan informasi mengenai jumlah kata yang dimasukkan. Terakhir yaitu terdapat tombol periksa untuk memulai proses prediksi terhadap teks adalah *AI/machine* atau *human*. Tampilan ketika menggunakan fitur prediksi tesk dapat dilihat pada gambar berikut.



Gambar 12 Wireframe – hasil dari prediksi teks

Pada gambar 12, *text area* yang diberi isian teks serta akan menampilkan jumlah kata yang dimasukkan. Selanjutnya, hasil prediksi akan ditampilkan di bawah tombol periksa. Hasil prediksi berupa nilai persentase probabilitas teks adalah *AI/machine*.

2.6 Instrumen Penelitian

Dalam penelitian ini terdapat beberapa instrument dalam membantu proses penelitian agar berjalan secara lancar. Instrumen tersebut terbagi menjadi dua yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*).

2.6.1 Perangkat keras (*hardware*)

Perangkat keras beserta dengan spesifikasi yang digunakan dalam penelitian kali ini yaitu:

Tabel 4 Perangkat keras

No.	Perangkat Keras	Spesifikasi
1.	Operasi System (OS)	Windows 10
2.	Processor	Intel® Core™ i3-2370M CPU @ 2.40GHz 2.40GHz
3.	RAM	6 GB
4.	GPU	NVIDIA GeForce 610M 2GB

2.6.2 Perangkat lunak (*software*)

Terdapat beberapa perangkat lunak atau *software* yang digunakan dalam penelitian ini yang bertujuan untuk mempermudah proses kerja. Perangkat lunak yang digunakan sebagai berikut:

Tabel 5 Perangkat lunak

No.	Perangkat Lunak	Fungsi
1.	Google Colaboratory	Jupyter Notebook yang disediakan oleh Google dengan fitur TPU dan GPU secara gratis.
2.	Keras	Library / API yang dapat digunakan pada Python untuk <i>Deep Learning</i> .
3.	Matplotlib	Library yang membantu dalam visualisasi data dalam bentuk grafik, bagan, plot, dll.
4.	Pandas	Library Python untuk pengolahan datasets yang berupa dataframe.
5.	Python	Bahasa Pemrograman utama yang digunakan dalam penelitian ini.
6.	Seaborn	Membantu Matplotlib dalam visualisasi data.
7.	Streamlit	Framework machine learning yang membantu dalam membuat dashboard atau membagikan visual data ke <i>website</i> sederhana.
8.	TensorFlow	Library Python yang berfungsi dalam pengembangan Machine Learning.

No.	Perangkat Lunak	Fungsi
9.	Trasformer	Library Python yang digunakan untuk memakai model Transfer Learning.
10.	Visual Studio Code	Code Editor yang digunakan dalam penelitian.