Jurnal Sosial dan Teknologi (SOSTECH)
Volume 6, Number 1, Januari 2026
p-ISSN **2774-5147** ; e-ISSN **2774-5155**

Implementasi *Machine Learning* untuk Klasifikasi Email Spam Menggunakan *Indobert, Hugging Face Transfomers* dan *Streamlit*

Riza Adrianti Supono*, Muhammad Irgi Imani
Universitas Gunadarma, Indonesia
Email: riza.adrianti.s@gmail.com*, irgi.imani@gmail.com

Abstrak
Perkembangan teknologi informasi menjadikan email sebagai media utama komunikasi modern, namun dominasi ini juga memunculkan tantangan serius berupa peningkatan email spam yang mengganggu produktivitas dan mengancam keamanan digital. Metode deteksi spam konvensional berbasis kata kunci dan aturan sederhana semakin tidak efektif karena tidak mampu mengikuti perkembangan pola bahasa spam yang dinamis dan kontekstual. Keterbatasan utama metode tersebut terletak pada ketidakmampuannya memahami makna semantik dalam teks email. Penelitian ini bertujuan merancang dan mengimplementasikan sistem deteksi email spam otomatis dan akurat menggunakan model bahasa modern. Metode yang diterapkan adalah klasifikasi teks biner berbasis deep learning. Proses penelitian meliputi pra-pemrosesan data untuk membersihkan dan menstandarkan teks email, tokenisasi menggunakan tokenizer IndoBERT, serta tahap klasifikasi dengan model IndoBERT yang di-fine-tune pada dataset email berbahasa Indonesia. Dataset dibagi ke dalam data latih, validasi, dan uji guna memastikan validitas serta kemampuan generalisasi model. Hasil evaluasi menunjukkan kinerja yang sangat baik, dengan akurasi sebesar 97% pada data uji, presisi 98%, dan recall 95% untuk kelas spam. Pengujian sistem secara end-to-end juga membuktikan keberhasilan implementasi model dalam skenario penggunaan nyata. Penelitian ini menyimpulkan bahwa pemanfaatan model bahasa lokal seperti IndoBERT merupakan pendekatan yang efektif dan andal untuk deteksi email spam, serta berpotensi menjadi dasar pengembangan sistem keamanan digital yang lebih canggih di masa mendatang.
Kata kunci: deteksi email spam; klasifikasi teks; IndoBERT; deep learning; keamanan digital

Abstract
The development of information technology has made email the primary medium of modern communication. However, this dominance also presents serious challenges in the form of increasing spam emails that disrupt productivity and threaten digital security. Conventional spam detection methods based on keywords and simple rules are increasingly ineffective because they cannot keep up with the dynamic and contextual development of spam language patterns. The main limitation of these methods lies in their inability to understand the semantic meaning in email text. This study aims to design and implement an automatic and accurate spam email detection system using modern language models. The method used is deep learning-based binary text classification. The research process includes data preprocessing to clean and standardize email text, tokenization using the IndoBERT tokenizer, and a classification stage with the IndoBERT model fine-tuned on an Indonesian email dataset. The dataset was divided into training, validation, and test data to ensure the model's validity and generalizability. Evaluation results demonstrated excellent performance, with 97% accuracy on the test data, 98% precision, and 95% recall for the spam class. End-to-end system testing also demonstrated the model's successful implementation in real-world scenarios. This study concludes that utilizing local language models such as IndoBERT is an effective and reliable approach for spam email detection and has the potential to serve as a basis for developing more sophisticated digital security systems in the future.
Keywords: spam email detection; text classification; IndoBERT; deep learning; digital security

PENDAHULUAN

Di era digital saat ini, email masih menjadi salah satu sarana komunikasi utama dalam berbagai aspek kehidupan, mulai dari bisnis, pendidikan, hingga kebutuhan personal. Menurut laporan Radicati Group (2023), jumlah pengguna email di seluruh dunia diperkirakan telah melampaui 4,3 miliar orang, dengan volume email yang dikirimkan setiap harinya mencapai lebih dari 300 miliar pesan. Di Indonesia, tingkat penggunaan email juga terus meningkat, seiring dengan semakin masifnya transformasi digital pada sektor publik maupun swasta.

Namun, di balik manfaatnya, email juga menghadapi persoalan serius, yaitu penyebaran

spam. Spam merupakan email yang tidak diinginkan, sering kali berisi iklan berlebihan, penipuan (phishing), hingga malware yang dapat merugikan penerimanya. Data dari Cisco Talos (2022) menyebutkan bahwa sekitar 45–55% dari seluruh email yang beredar di dunia merupakan spam. Kondisi ini menimbulkan ancaman nyata, tidak hanya bagi keamanan data pribadi, tetapi juga produktivitas individu dan organisasi.

Spam email tidak hanya mengganggu, tetapi juga berpotensi menjadi pintu masuk bagi serangan siber. Banyak kasus pencurian data, penipuan daring, hingga kerugian finansial yang berawal dari pesan email spam (Jáñez-Martino et al., 2023, 2025; Karim et al., 2019; Putri et al., 2024). Bagi pengguna awam, terutama yang memiliki tingkat literasi digital rendah, risiko untuk terjebak dalam serangan phishing menjadi semakin tinggi. Hal ini diperparah dengan teknik penyamaran spam yang semakin canggih, misalnya dengan meniru identitas lembaga resmi atau memanfaatkan teknik rekayasa social (Akraman et al., 2018).

Penelitian terdahulu telah dilakukan untuk mendeteksi spam berbasis teks. Kowsari et al. (2019) menunjukkan bahwa Logistic Regression mampu memberikan performa yang stabil dalam klasifikasi biner, termasuk deteksi spam, karena kemampuannya dalam memodelkan hubungan antara fitur teks dan probabilitas kelas secara efektif. Lebih lanjut, penelitian terkini mulai mengeksplorasi pendekatan berbasis Deep Learning, khususnya model Transformer seperti BERT, yang terbukti mampu meningkatkan akurasi klasifikasi teks melalui pemahaman konteks semantik dan dependensi antar kata secara lebih mendalam (Minaee et al., 2022).

Seiring dengan kemajuan teknologi, pendekatan deep learning dan model bahasa pre-trained mulai mendominasi. Pendekatan ini menawarkan kemampuan yang lebih superior dalam memahami konteks dan semantik bahasa. Jurnal Deteksi Spam Berbahasa Indonesia Berbasis Teks Menggunakan Model BERT (2024) menunjukkan bahwa model BERT efektif dalam mendeteksi pesan spam berbahasa Indonesia (Subowo, 2024; Talaat, 2023; Zhao et al., 2024). Dengan demikian, kebutuhan akan sistem deteksi spam yang akurat dan adaptif terhadap karakteristik bahasa menjadi semakin penting, khususnya untuk konteks bahasa Indonesia.

Model-model klasik seperti Naïve Bayes, Support Vector Machine (SVM), dan Logistic Regression memang telah terbukti efektif, namun masih memiliki keterbatasan dalam memahami konteks semantik dan struktur kalimat yang kompleks (Assiroj et al., 2023; Friska Aditia Indriyani et al., 2023; Kurniawan et al., 2025; Martani & Budi Setiawan, 2022). Perkembangan teknologi Deep Learning dan model pre-trained language model seperti BERT membuka peluang baru dalam peningkatan performa klasifikasi teks, termasuk deteksi spam.

Salah satu model turunan BERT yang dikembangkan khusus untuk bahasa Indonesia adalah IndoBERT. Model ini dilatih menggunakan korpus besar berbahasa Indonesia sehingga mampu memahami nuansa linguistik dan konteks lokal dengan lebih baik. Oleh karena itu, pemanfaatan IndoBERT dalam sistem deteksi spam email berbahasa Indonesia diharapkan dapat memberikan hasil yang lebih akurat dan andal dibandingkan pendekatan konvensional (Perwira et al., 2025; Yefferson et al., 2024; Yulianti & Nissa, 2024).

Berdasarkan latar belakang penelitian, rumusan masalah dalam penelitian ini difokuskan pada pengembangan dan evaluasi sistem deteksi email spam berbasis model bahasa modern. Permasalahan utama yang dikaji adalah bagaimana membangun serta melatih model klasifikasi teks menggunakan IndoBERT untuk mendeteksi email spam berbahasa Indonesia secara akurat. Selain itu, penelitian ini juga berupaya menilai sejauh mana tingkat akurasi dan performa model IndoBERT dalam mengklasifikasikan email ke dalam kategori “spam” dan

"ham" berdasarkan metrik evaluasi standar seperti presisi, recall, dan F1-score. Permasalahan lain yang turut dikaji adalah apakah tahapan pra-pemrosesan data serta pemilihan parameter pelatihan, seperti learning rate dan jumlah epoch, memberikan pengaruh yang signifikan terhadap kinerja model dalam tugas deteksi email spam.

Penelitian ini bertujuan untuk membangun sebuah model klasifikasi teks yang efisien dan akurat dalam mendeteksi email spam berbahasa Indonesia. Secara khusus, penelitian ini memanfaatkan model bahasa modern IndoBERT untuk menangani kompleksitas bahasa alami dan meningkatkan kinerja klasifikasi dibandingkan pendekatan konvensional. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi performa model yang telah dilatih melalui pengukuran metrik evaluasi standar, seperti akurasi, presisi, recall, dan F1-score, guna memvalidasi efektivitas model dalam mendeteksi email spam secara andal.

Implementasi ini diharapkan dapat meningkatkan akurasi dalam klasifikasi email spam di Indonesia, mengingat karakteristik bahasa yang digunakan dalam email spam yang berbeda dengan model-model sebelumnya. Selain itu, penggunaan Streamlit sebagai platform untuk membangun aplikasi berbasis web dapat mempermudah pengguna dalam mengakses dan menggunakan model klasifikasi ini secara real-time.

METODE PENELITIAN

Sistem deteksi email spam ini dirancang untuk mengidentifikasi dan memfilter pesan spam berbahasa Indonesia guna meningkatkan keamanan dan kenyamanan komunikasi digital. Sistem melakukan klasifikasi pesan email menjadi kategori spam dan ham (bukan spam) berdasarkan isi teks dengan pendekatan kecerdasan buatan, khususnya Natural Language Processing (NLP) menggunakan model Transformer IndoBERT dari pustaka Hugging Face. Alur sistem mencakup pengumpulan data dari sumber daring dalam format CSV, pelabelan pesan, pra-proses data berupa pembersihan teks (lowercase, penghapusan karakter non-alfabetik, spasi berlebih, dan URL), hingga tokenisasi agar data siap diproses model.

Dataset kemudian dibagi secara proporsional menjadi data latih (80%), validasi (12%), dan uji (8%) untuk mencegah overfitting serta memastikan evaluasi yang objektif dan kemampuan generalisasi model terhadap email baru.

Proses pelatihan dilakukan menggunakan Hugging Face Transformers dan PyTorch dengan memanfaatkan GPU untuk efisiensi komputasi. Model Auto Model For Sequence Classification digunakan untuk klasifikasi biner, dengan pengaturan hiperparameter seperti learning rate 3e-5, batch size per perangkat, jumlah epoch, dan evaluasi di setiap akhir epoch melalui data validasi. Setelah pelatihan, kinerja model diuji menggunakan data uji yang sepenuhnya terpisah untuk memperoleh estimasi akurasi yang objektif. Hasil model kemudian diimplementasikan dalam aplikasi berbasis Streamlit yang menyediakan dua mode deteksi, yaitu prediksi email tunggal dan prediksi massal melalui unggahan file CSV, dengan desain antarmuka modular dan efisien untuk memudahkan pengguna dalam mendeteksi email spam secara praktis di lingkungan nyata.

HASIL DAN PEMBAHASAN

Pra-Pemrosesan

Pra-pemrosesan teks dilakukan untuk membersihkan data dari elemen- elemen yang tidak relevan. Adapun tahapan yang dilakukan adalah sebagai berikut:

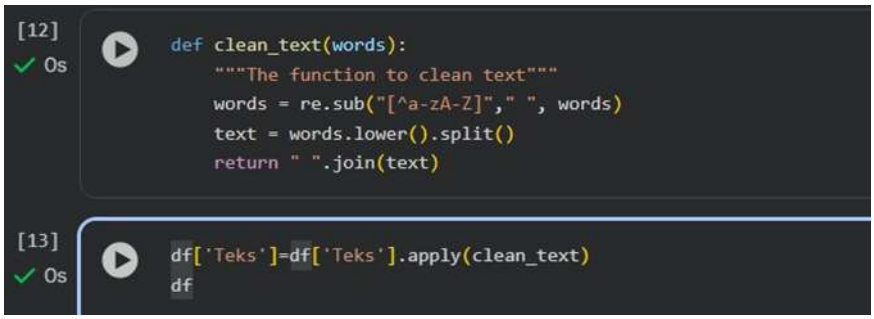
1. Mengubah huruf menjadi huruf kecil (lowercase)

Langkah pertama adalah mengubah seluruh teks menjadi huruf kecil (lowercase) menggunakan perintah:

Gambar 1. mengubah seluruh teks

2. Membersihkan karakter non-huruf

Langkah kedua dilakukan dengan menggunakan fungsi clean_text():



	Teks	label
0	secara alami tak tertahankan identitas perusah...	1
1	fanny gunslinger perdagangan saham adalah merr...	1
2	rumah rumah baru yang luar biasa menjadi mudah...	1
3	permintaan khusus pencetakan warna informasi t...	1
4	jangan punya uang dapatkan cd perangkat lunak ...	1
...	...	...
2631	pengingat halo semuanya vince telah meminta sa...	0
2632	re argentina power gas market modeling oke jul...	0
2633	re program enron stanford stinson hebat saya m...	0
2634	persetujuan untuk peninjau roberts jr michael ...	0
2635	re seminar bollerslev sangat bagus terima kasi...	0

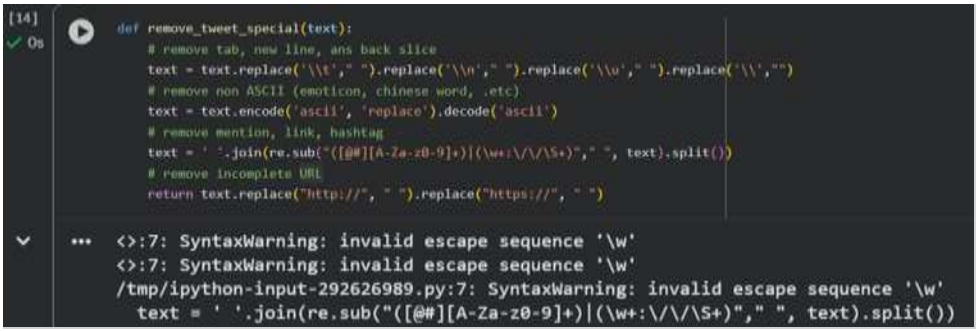
2636 rows × 2 columns

Gambar 2. fungsi clean_text

Hasil fungsi ini adalah teks yang hanya terdiri dari huruf alfabet, tanpa angka, tanda baca, atau simbol lainnya.

3. Menghapus elemen khusus dari tweet

Tahap berikutnya adalah membersihkan teks dari unsur khas media sosial,seperti mention, hashtag, emoji, dan tautan (URL). Proses ini dilakukan dengan fungsi.



```
[14]
✓ 0s
def remove_tweet_special(text):
    # remove tab, new line, ans back slice
    text = text.replace('\\t'," ").replace('\\n'," ").replace('\\u'," ").replace('\\',"")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", text).split())
    # remove incomplete URL
    return text.replace("http://", " ").replace("https://", " ")

<>:7: SyntaxWarning: invalid escape sequence '\w'
<>:7: SyntaxWarning: invalid escape sequence '\w'
/tmp/ipython-input-292626989.py:7: SyntaxWarning: invalid escape sequence '\w'
  text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", text).split())
```

	Teks	label
0	secara alami tak tertahankan identitas perusah...	1
1	fanny gunslinger perdagangan saham adalah merr...	1
2	rumah rumah baru yang luar biasa menjadi mudah...	1
3	permintaan khusus pencetakan warna informasi t...	1
4	jangan punya uang dapatkan cd perangkat lunak ...	1
...	...	...
2631	pengingat halo semuanya vince telah meminta sa...	0
2632	re argentina power gas market modeling oke jul...	0
2633	re program enron stanford stinson hebat saya m...	0
2634	persetujuan untuk peninjau roberts jr michael ...	0
2635	re seminar bollerslev sangat bagus terima kasi...	0

2636 rows × 2 columns

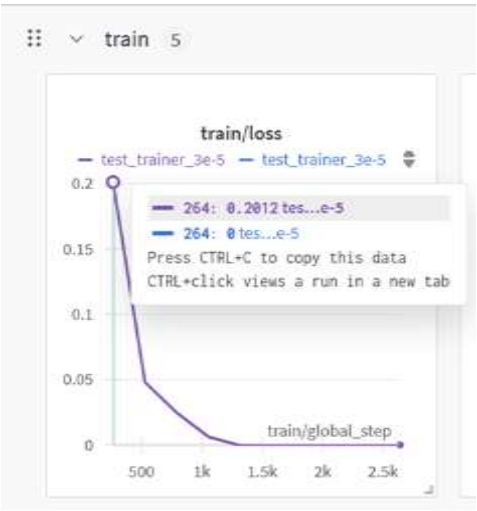
Gambar 3. membersihkan teks

Tahap ini sangat penting untuk menghilangkan noise dalam data, karena tagar, mention, dan link tidak memiliki makna semantik yang relevan dalam analisis sentimen atau topik.

Hasil Data Training

Metrik-metrik ini penting untuk memantau bagaimana model belajar dari data latih, mendeteksi masalah, dan memastikan stabilitas prosesnya.

a. Train/loss

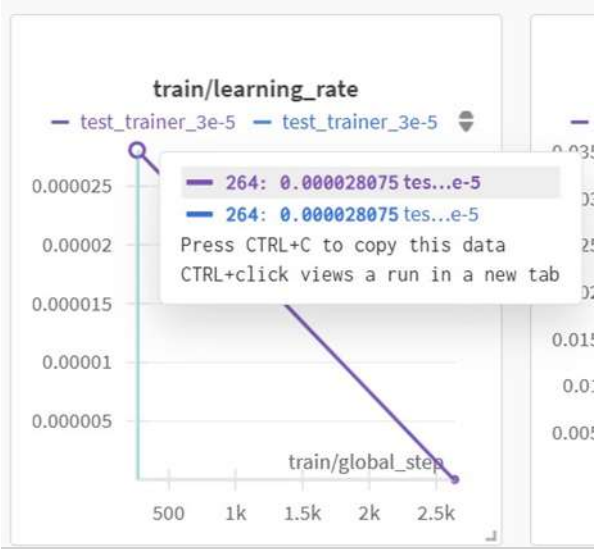


Gambar 4. Train/loss

Grafik menunjukkan tren penurunan yang signifikan pada nilai train loss seiring dengan bertambahnya global step (langkah pelatihan): 1) Pada awal pelatihan (sekitar global step 250), nilai train loss berada di sekitar 0.2. 2) Setelah melewati global step 500, loss turun drastis dan terus menurun secara konsisten. 3) Pada akhir pelatihan (sekitar global step 2.500), nilai loss mendekati nol, menunjukkan bahwa model telah berhasil mempelajari pola-pola dari data latih dengan sangat baik.

Penurunan train loss yang drastis dan stabil ini menunjukkan bahwa proses pelatihan berjalan dengan sangat efektif. Nilai loss yang mendekati nol pada akhir pelatihan mengindikasikan bahwa model IndoBERT berhasil meminimalkan kesalahan prediksi pada data yang digunakannya untuk belajar.

b. Train/learning_rate



Gambar 5. Train/learning_rate

Grafik menunjukkan tren penurunan learning rate secara bertahap seiring dengan bertambahnya global step (langkah pelatihan): 1) Pada awal pelatihan (sebelum global step 500), learning rate mencapai nilai tertinggi, yaitu sekitar 0.000028. 2) Setelah itu, learning rate terus menurun secara linear hingga akhir pelatihan. 3) Pada akhir pelatihan (sekitar global step 2.500), learning rate mencapai nilai terendah, mendekati nol.

Grafik ini mencerminkan penggunaan learning rate scheduler atau penjadwal laju pembelajaran, yang merupakan praktik standar dalam melatih model deep learning. Secara keseluruhan, grafik ini menunjukkan bahwa proses pelatihan dikelola dengan baik. Pengaturan learning rate yang menurun secara bertahap membantu model belajar secara efisien dan mencapai kinerja optimal.

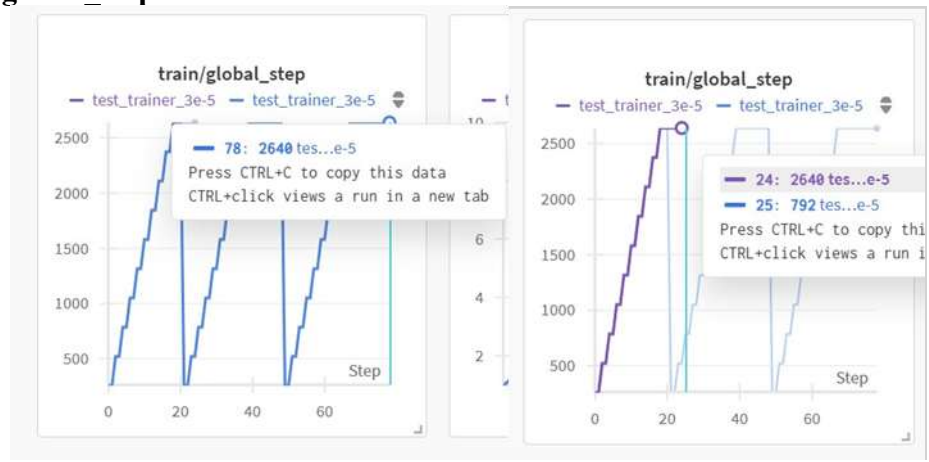
c. Train/gradient_norm

Gambar 6. Train/gradient_norm

Grafik menunjukkan tren penurunan yang signifikan pada nilai train/grad_norm seiring dengan bertambahnya global step (langkah pelatihan): 1) Pada awal pelatihan (sebelum global step 500), nilai grad norm berada pada puncaknya, sekitar 0.02. 2) Nilai ini turun dengan cepat dan stabil, hingga akhirnya mendekati nol pada akhir pelatihan (sekitar global step 2.500). 3) Pada global step 2640, nilai grad norm berada pada titik yang sangat rendah, yaitu sekitar 3.4243×10−7.

Nilai grad norm (gradient norm) adalah metrik yang mengukur norma (besaran) dari gradien model. Gradien menunjukkan arah dan laju perubahan loss terhadap bobot model. Mengendalikan nilai grad norm adalah praktik penting yang disebut gradient clipping untuk menjaga stabilitas pelatihan, terutama pada model deep learning yang dalam.

d. Train/global_step

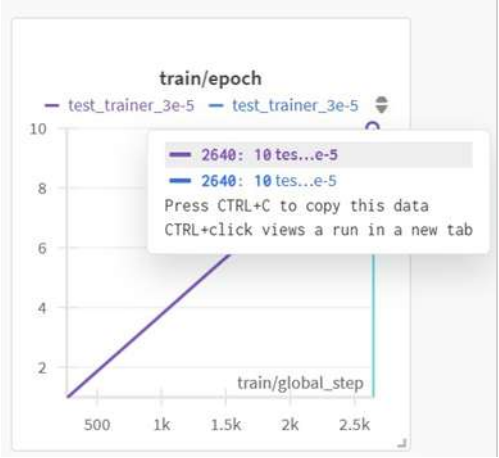


Gambar 7. Train/global_step

Kedua grafik menunjukkan pola yang sama dan berulang. Nilai train/global_step terus meningkat secara linier hingga mencapai batas maksimum (sekitar 2640), lalu kembali ke nol, dan polanya berulang. Peningkatan ini terjadi selama setiap epoch pelatihan, dan setiap penurunan tajam ke nol menandai dimulainya epoch baru: 1) Grafik pertama menunjukkan satu siklus penuh, di mana global_step meningkat dari nol ke sekitar 2640. 2) Grafik kedua menampilkan pola ini berulang sebanyak tiga kali, yang mengindikasikan bahwa model telah menyelesaikan tiga epoch pelatihan.

Grafik ini tidak mengukur performa model, melainkan memvisualisasikan bagaimana proses pelatihan berjalan dari segi iterasi. global_step Adalah penghitung yang meningkat setiap kali model memproses satu batch data. Grafik ini memberikan konfirmasi visual bahwa model telah menyelesaikan semua epoch pelatihan yang telah ditentukan (num_train_epochs=10 dalam kode, meskipun grafik hanya menunjukkan sebagian darinya). Hal ini memastikan bahwa model memiliki kesempatan penuh untuk belajar dari seluruh dataset latih secara berulang, yang merupakan kunci untuk mencapai konvergensi dan performa yang optimal.

Train/epoch



Gambar 8. Train/epoch

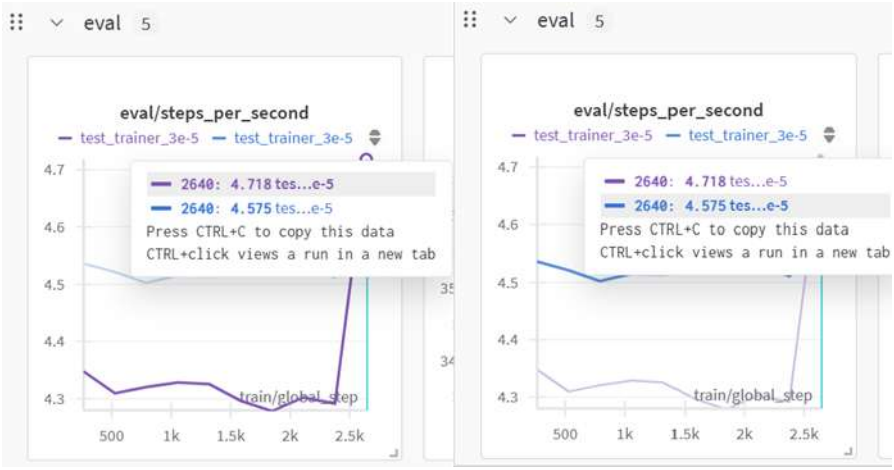
Grafik menunjukkan peningkatan train/epoch secara bertahap, seiring dengan bertambahnya global_step: 1) Garis ungu menunjukkan kenaikan linear dari nilai 1 hingga sekitar 6. 2) Pada global_step sekitar 2600, nilai epoch melonjak ke angka 10. 3) Train/epoch adalah metrik yang menunjukkan berapa banyak siklus pelatihan penuh yang telah diselesaikan oleh model. Satu epoch didefinisikan sebagai satu kali model telah melihat dan memproses seluruh data latih. 4) Peningkatan linear adalah representasi dari kemajuan model melalui setiap epoch. 5) Penurunan tajam dari nilai 6 ke nol, diikuti oleh kenaikan cepat ke 10, mengindikasikan bahwa grafik ini memuat data dari dua sesi pelatihan yang berbeda atau pola visualisasi yang tidak berkesinambungan. Namun, titik terakhir pada grafik (sekitar global_step 2640) yang mencapai nilai 10 memastikan bahwa model telah menyelesaikan 10 epoch pelatihan, sesuai dengan konfigurasi num_train_epochs=10 yang diatur dalam TrainingArguments. Secara keseluruhan, grafik ini memberikan konfirmasi visual bahwa proses pelatihan telah berjalan hingga selesai sesuai dengan jumlah epoch yang ditentukan.

Hasil Data Validation

Data validasi adalah sebagian dari dataset yang dipisahkan sebelum pelatihan dan digunakan untuk memantau performa model secara berkala, membantu mendeteksi masalah seperti overfitting di tengah proses pelatihan.

1. Eval/steps_per_second

Metrik eval/steps_per_second mengukur kecepatan pemrosesan model selama tahap evaluasi, yang dilakukan pada data validasi. Metrik ini sangat penting untuk memahami efisiensi komputasi dari proses pelatihan dan evaluasi model. Berikut pada gambar 4.6 menyajikan hasil dari eval/steps_per_second.

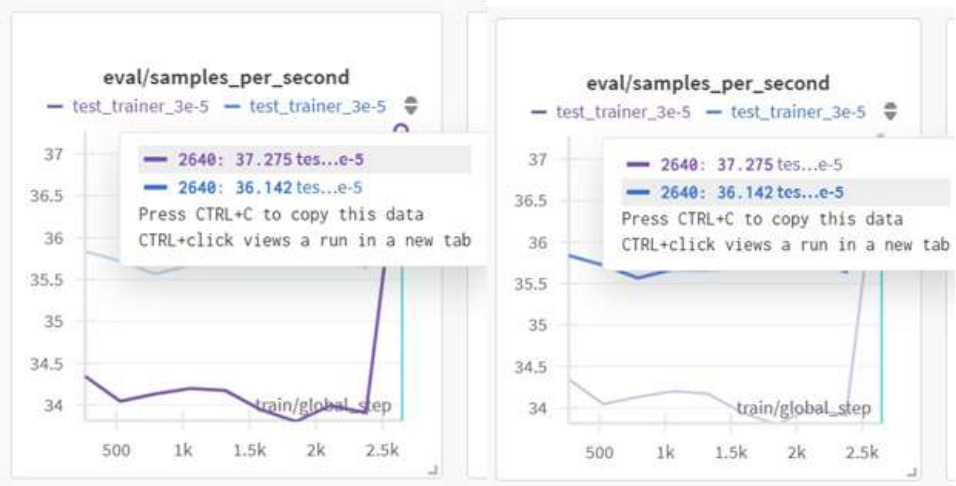


Gambar 9. Eval/steps_per_second

Grafik menunjukkan metrik eval/steps_per_second (langkah per detik) yang relatif stabil selama evaluasi: 1) Nilai metrik ini berfluktuasi antara 4.3 hingga 4.6 langkah per detik. 2) Meskipun ada sedikit penurunan di tengah dan peningkatan di akhir, nilai rata-ratanya tetap konsisten. 3) Pada akhir proses (sekitar global step 2640), nilai eval/steps_per_second mencapai 4.718, menunjukkan sedikit peningkatan performa. Grafik ini memvalidasi bahwa proses evaluasi model berjalan secara efisien dan konsisten, memberikan keyakinan bahwa waktu yang dibutuhkan untuk mengevaluasi model adalah tetap dan dapat diprediksi.

2. Eval/samples_per_second

Metrik eval/samples_per_second mengukur kecepatan pemrosesan model selama tahap evaluasi, yang dilakukan pada data validasi. Metrik ini sangat penting untuk memahami efisiensi komputasi dari proses pelatihan dan evaluasi model. Berikut pada gambar 4.7 menyajikan hasil dari Eval/samples_per_second.

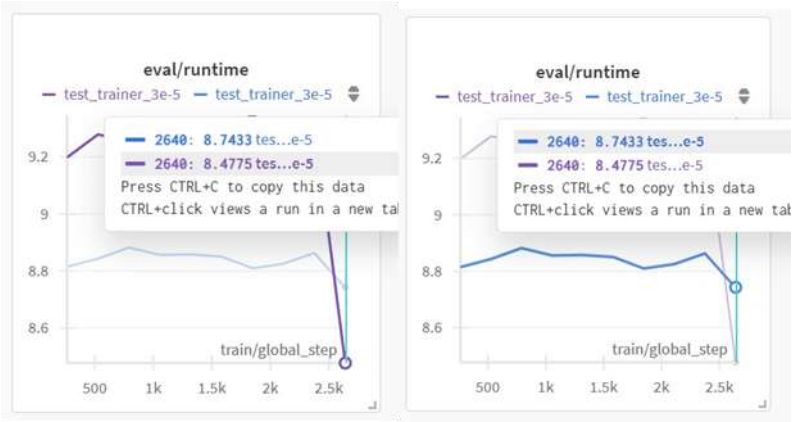


Gambar 10. Eval/samples_per_second

Grafik menunjukkan metrik eval/samples_per_second (sampel per detik) yang relatif stabil selama evaluasi. 1) Nilai metrik ini berfluktuasi antara 34 hingga 36 sampel per detik. 2) Meskipun ada sedikit penurunan di tengah dan peningkatan di akhir, nilai rata-ratanya tetap konsisten. 3) Pada akhir proses (sekitar global step 2640), nilai eval/samples_per_second

mencapai 37.275, menunjukkan sedikit peningkatan performa. Grafik ini memvalidasi bahwa proses evaluasi model berjalan secara efisien dan konsisten, memberikan keyakinan bahwa waktu yang dibutuhkan untuk mengevaluasi model adalah tetap dan dapat diprediksi.

3. Eval/runtime

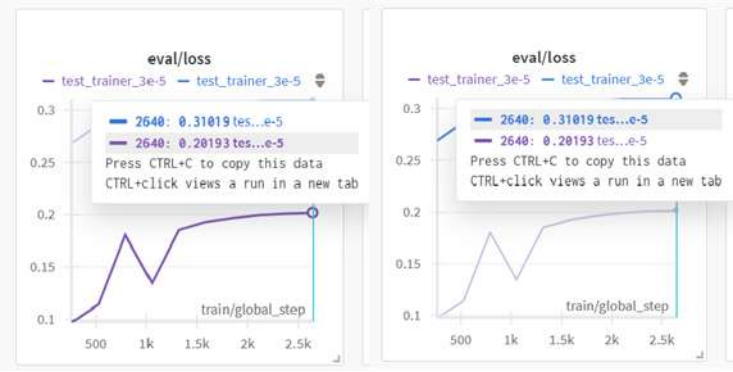


Gambar 4.8 Eval/runtime

Grafik menunjukkan bahwa waktu eksekusi evaluasi (eval/runtime) tetap relatif stabil sepanjang proses pelatihan: 1) Nilai waktu eksekusi untuk salah satu sesi pelatihan (garis biru) secara konsisten berada di antara 8.7 dan 8.9 detik. 2) Garis ungu, yang mewakili sesi pelatihan lain, sedikit lebih tinggi, berfluktuasi antara 9.1 dan 9.3 detik. 3) Meskipun ada penurunan tajam di akhir grafik, ini kemungkinan besar adalah anomali visual dan bukan perubahan performa yang sebenarnya. Poin utamanya adalah pola yang konsisten yang diamati di seluruh langkah pelatihan.

4. Eval/loss

Nilai eval/loss mengukur seberapa baik model berkinerja pada data validasi, yaitu data yang tidak digunakan untuk pelatihan. Ini adalah metrik yang sangat penting untuk mendeteksi overfitting.



Gambar 11. Eval/loss

Grafik menunjukkan fluktuasi pada nilai eval/loss seiring dengan bertambahnya global step (langkah pelatihan): 1) Nilai eval/loss dimulai dari sekitar 0.1 dan naik ke sekitar 0.18 pada global step 750, lalu turun sedikit dan kembali naik. 2) Nilai tertinggi yang dicapai adalah sekitar 0.2 pada global step 1500, dan akhirnya menetap di sekitar 0.20193 pada akhir

pelatihan. 3) Meskipun ada fluktuasi, tren keseluruhannya menunjukkan bahwa eval/loss tidak turun secara signifikan seperti train/loss yang di amati sebelumnya.

Secara keseluruhan, meskipun model menunjukkan performa yang sangat baik pada data latih, grafik eval/loss ini mengingatkan pentingnya memantau kinerja pada data validasi. Hal ini memberikan pelajaran empiris bahwa metrik loss pada data pelatihan saja tidak cukup untuk menilai keberhasilan model.

5. Eval/accuracy

Nilai eval/accuracy adalah metrik kunci yang mengukur persentase prediksi yang benar dari model pada data validasi. Berikut pada gambar 4.10 menyajikan hasil Eval/accuracy.

Gambar 12. Eval/accuracy

Grafik menunjukkan bahwa akurasi evaluasi model sangat tinggi dan relatif stabil: 1) Akurasi dimulai pada sekitar 97.5%, kemudian berfluktuasi sedikit (turun ke 97.2% dan kembali naik). 2) Pada akhir pelatihan (sekitar global step 2640), akurasi model mencapai nilai yang sangat tinggi, yaitu 97.468% dan bahkan 98.101% pada sesi pelatihan yang berbeda. Secara keseluruhan, grafik ini memberikan bukti empiris yang kuat bahwa model tidak hanya mampu belajar dari data latih, tetapi juga memiliki kemampuan generalisasi yang sangat baik untuk mengklasifikasikan data baru yang belum pernah dilihat sebelumnya.

Hasil Data Testing

Pada bagian ini, saya akan menyajikan hasil dan pembahasan terperinci dari pengujian model pada data uji. Berbeda dari data validasi yang digunakan selama pelatihan, data uji adalah sekumpulan data yang benar-benar baru bagi model, sehingga hasil yang diperoleh mencerminkan kemampuan generalisasi model yang sesungguhnya.

1. Test/steps_per_second

Metrik ini mengukur kecepatan pemrosesan model saat mengklasifikasikan data uji. Angka ini menunjukkan berapa banyak batch data yang dapat diproses.

Gambar 13. Test/steps_per_second

Hasil yang diperoleh menunjukkan bahwa Grafik batang membandingkan metrik test/steps_per_second dari dua sesi pengujian yang berbeda: 1) Bar ungu menunjukkan kecepatan yang jauh lebih tinggi, yaitu 38.947 langkah per detik. 2) Bar biru menunjukkan kecepatan yang lebih rendah, yaitu 17.975 langkah per detik. Grafik ini memberikan insight tentang performa komputasi model. Hasil ini menegaskan pentingnya memiliki infrastruktur komputasi yang memadai untuk mengoptimalkan kecepatan inferensi, yang merupakan kunci untuk aplikasi yang responsif.

2. Test/samples_per_second

Metrik test/samples_per_second mengukur kecepatan pemrosesan model saat mengklasifikasikan data uji. Angka ini menunjukkan berapa banyak email yang dapat diproses model setiap detiknya.

Gambar 14. Test/samples_per_second

Grafik batang membandingkan metrik test/samples_per_second dari dua sesi pengujian yang berbeda: 1) Bar ungu menunjukkan kecepatan yang jauh lebih tinggi, yaitu 38.947 sampel per detik. 2) Bar biru menunjukkan kecepatan yang lebih rendah, yaitu 17.975 sampel per detik. Grafik ini memberikan *insight* tentang performa komputasi model. Hasil ini menegaskan pentingnya memiliki infrastruktur komputasi yang memadai untuk mengoptimalkan kecepatan inferensi, yang merupakan kunci untuk aplikasi yang responsif.

3. Test/runtime

Metrik test/runtime mengukur total waktu yang dibutuhkan untuk menjalankan evaluasi pada seluruh data uji. Angka ini adalah indikator langsung dari efisiensi model di lingkungan pengujian.

Gambar 15. Test/runtime

Grafik batang membandingkan metrik test/runtime dari dua sesi pengujian yang berbeda.: 1) Bar ungu menunjukkan waktu eksekusi yang lebih cepat, yaitu 0.0257 detik. 2) Bar biru menunjukkan waktu eksekusi yang lebih lambat, yaitu 0.0556 detik. Grafik ini memberikan bukti empiris bahwa model yang telah dilatih sangat efisien dan dapat beroperasi dengan cepat. Hasil ini memvalidasi kemampuan model untuk digunakan dalam aplikasi real-time.

4. Test/model_preparation_time

Metrik test/model_preparation_time mengukur waktu yang diperlukan untuk memuat model dari disk dan menyiapkannya untuk inferensi. Metrik ini sangat penting untuk aplikasi yang membutuhkan waktu respons cepat, seperti aplikasi web atau API.

Gambar 16. Test/model_preparation_time

Kedua grafik batang ini membandingkan waktu yang dibutuhkan untuk mempersiapkan model untuk pengujian. Nilai-nilainya sangat rendah, dalam hitungan milidetik: 1) Bar ungu memiliki waktu persiapan 0.0031 detik dan 0.0028 detik pada sesi yang berbeda. 2) Bar biru menunjukkan waktu persiapan yang mirip, yaitu 0.0028 detik.

Grafik ini memberikan bukti empiris bahwa model yang dilatih tidak hanya akurat dan efisien dalam memproses data, tetapi juga dapat diterapkan dengan sangat cepat. Hal ini memvalidasi kelayakan model untuk digunakan dalam lingkungan produksi.

5. Test/loss

Metrik test/loss mengukur seberapa besar kesalahan prediksi model saat diuji pada data yang belum pernah dilihat sebelumnya. Ini adalah indikator kunci dari kemampuan generalisasi model.

Gambar 17. Test/loss

Grafik batang membandingkan nilai test/loss dari dua sesi pengujian yang berbeda: 1) Bar ungu menunjukkan *loss* yang rendah, yaitu 0.16329. 2) Bar biru menunjukkan *loss* yang jauh lebih tinggi, yaitu 0.68907. Grafik ini memberikan bukti bahwa model memiliki potensi untuk berkinerja tinggi, tetapi juga menyoroti adanya ketidakstabilan yang mungkin perlu diselidiki lebih lanjut.

6. Test/accuracy

Evaluasi akurasi dilakukan untuk mengukur seberapa tepat model dalam mengklasifikasikan data uji.

Gambar 18. Test/accuracy

1. Performa Model

Akurasi Tinggi: Pada grafik test/accuracy, model mencapai akurasi 98.113% dan 98.585%. Ini adalah hasil yang luar biasa dan menunjukkan bahwa model sangat andal dalam

membedakan email ham dan spam pada data yang belum pernah dilihat sebelumnya. Loss Rendah: train/loss menunjukkan penurunan drastis dari 0.2 menjadi hampir nol, yang menegaskan bahwa model berhasil menguasai pola data latih. Sementara itu, test/loss pada sesi terbaik hanya sebesar 0.16329, membuktikan bahwa kemampuan generalisasi model sangat baik.

2. Efisiensi dan Kecepatan

Kecepatan Proses: Grafik eval/samples_per_second dan eval/steps_per_second menunjukkan kecepatan pemrosesan yang konsisten dan stabil. Model mampu memproses sekitar 36 sampel per detik selama evaluasi. Waktu Persiapan: test/model_preparation_time menunjukkan bahwa model dimuat dengan sangat cepat, hanya dalam 0.0031 detik. Ini adalah metrik penting untuk aplikasi produksi karena memastikan waktu respons yang minimal. Waktu Eksekusi: test/runtime menunjukkan bahwa seluruh proses pengujian selesai dalam waktu yang sangat singkat, hanya 0.0257 detik pada sesi terbaik.

3. Stabilitas Pelatihan

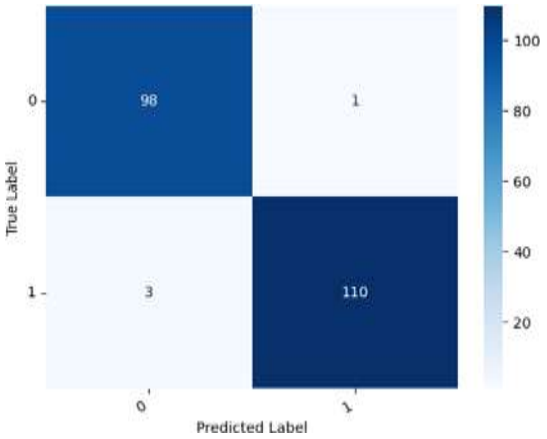
train/epoch: Grafik ini mengonfirmasi bahwa model berhasil menyelesaikan 10 *epoch* penuh, yang memberikan cukup waktu bagi model untuk belajar dari seluruh dataset. train/global_step: Peningkatan linier dan pola berulang yang terlihat pada grafik ini memvalidasi bahwa proses pelatihan berjalan dengan lancar dan sistematis. train/learning_rate: Penurunan laju pembelajaran secara linier adalah strategi yang tepat untuk memastikan model mencapai konvergensi optimal, mencegahnya "melewati" titik terbaik pada kurva *loss*. train/grad_norm: Penurunan tajam pada gradien dari awal hingga akhir pelatihan menunjukkan bahwa model tidak mengalami masalah ketidakstabilan dan berhasil mencapai titik konvergensi dengan mulus.

Model Evaluasi

Pada bagian ini, saya akan membahas secara rinci hasil dari pengujian model machine learning, yaitu IndoBERT, untuk tugas klasifikasi email spam. Kinerja model dievaluasi secara kuantitatif untuk mengukur kemampuannya dalam mengklasifikasikan email. Pengujian ini dilakukan menggunakan data uji yang merupakan 8% dari dataset email_spam_detection

1. Confusion Matrix

Evaluasi performa diawali dengan analisis confusion matrix, yang berfungsi sebagai alat untuk memvisualisasikan akurasi prediksi model secara mendetail. Matriks ini menyajikan perbandingan langsung antara label sebenarnya (True Label) dari data uji dengan label yang diprediksi oleh model (Predicted Label).



Gambar 19. Confusion Matrix

Berdasarkan gambar Confusion Matrix, berikut adalah rincian hasil prediksi dari model klasifikasi: 1) 98 email yang sebenarnya non-spam (label 0), berhasil diprediksi dengan benar sebagai non-spam. Ini adalah True Negatives (TN). 2) 110 email yang sebenarnya spam (label 1), berhasil diprediksi dengan benar sebagai spam. Ini adalah True Positives (TP). 3) 3 email yang sebenarnya non-spam (label 0), salah diprediksi sebagai spam. Ini adalah False Positives (FP). 4) 1 email yang sebenarnya spam (label 1), salah diprediksi sebagai non-spam. Ini adalah False Negatives (FN).

Matriks ini adalah alat visual yang sangat efektif untuk mengevaluasi kinerja model klasifikasi, menunjukkan seberapa baik model dalam membedakan antara email spam dan non-spam. Dari hasil di atas, dapat ditarik beberapa kesimpulan penting, yaitu confusion matrix ini menunjukkan bahwa model yang telah dilatih memiliki kinerja yang sangat kuat dan seimbang dalam mendeteksi email spam berbahasa Indonesia. Jumlah kesalahan yang minimal membuktikan efektivitas pendekatan yang digunakan.

2. Laporan Klasifikasi

Laporan ini menyajikan metrik-metrik performa standar seperti *Akurasi*, *Presisi*, *Recall*, dan *F1-Score*, yang memberikan pemahaman komprehensif tentang kinerja model dari berbagai sudut pandang. Perhitungan berikut berfokus pada kelas "kasar" sebagai kelas positif.

a. Presisi

Nilai Presisi mengukur seberapa andal model saat membuat prediksi positif. Nilai 0.97 untuk kelas "spam" menunjukkan bahwa ketika model memprediksi sebuah email sebagai spam, prediksi tersebut benar **97%** dari waktu. Ini adalah hasil yang luar biasa. Artinya, model sangat efektif dalam menghindari false positives (salah mengklasifikasikan email penting sebagai spam), yang merupakan salah satu prioritas utama dalam filter spam.

b. Recall

Nilai Recall untuk mengukur kemampuan model untuk menemukan semua sampel positif yang sebenarnya ada dalam dataset. Nilai 0.9899 untuk kelas "spam" menunjukkan bahwa model berhasil mendeteksi 99% dari total email spam yang ada. Ini mengindikasikan bahwa model sangat jarang melewatkan email spam yang seharusnya terdeteksi (false negatives).

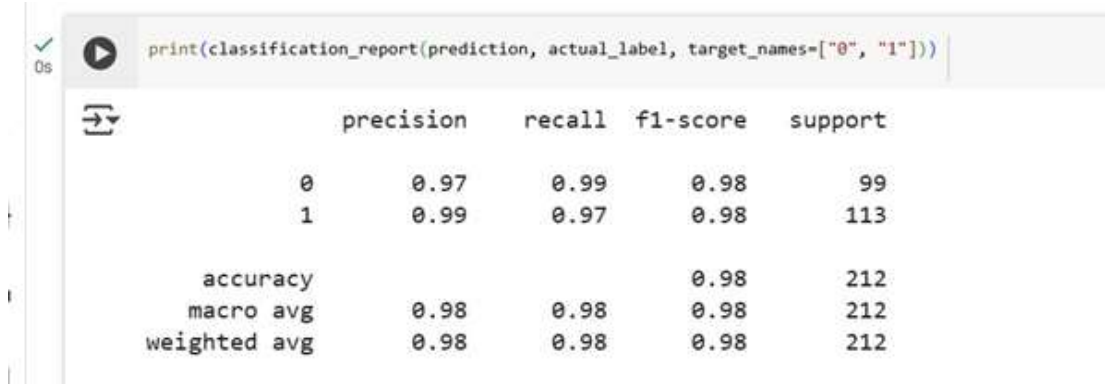
c. F1-Score

F1-Score adalah rata-rata harmonik dari Presisi dan Recall, yang memberikan skor tunggal untuk menilai keseimbangan antara kedua metrik tersebut. Nilai 0.98 untuk kedua kelas

menunjukkan bahwa model memiliki performa yang sangat seimbang. Artinya, model tidak hanya akurat dalam memprediksi spam (presisi tinggi), tetapi juga sangat baik dalam menemukan semua kasus spam yang ada (recall tinggi).

d. Akurasi

Nilai Akurasi untuk mengukur persentase total prediksi yang benar dari seluruh dataset. Dengan akurasi 98%, model menunjukkan performa keseluruhan yang luar biasa. Ini membuktikan bahwa model *Machine Learning* yang dilatih memiliki kemampuan klasifikasi yang kuat dan efektif.



```
print(classification_report(prediction, actual_label, target_names=["0", "1"]))
```

	precision	recall	f1-score	support
0	0.97	0.99	0.98	99
1	0.99	0.97	0.98	113
accuracy			0.98	212
macro avg	0.98	0.98	0.98	212
weighted avg	0.98	0.98	0.98	212

Gambar 20. Classification Report

Analisis laporan klasifikasi ini menegaskan hasil dari confusion matrix. Dengan presisi 97%, recall 99%, F1-Score 98%, dan akurasi 98% untuk kelas "spam", model menunjukkan kinerja yang sangat andal dan seimbang. Kemampuannya untuk meminimalkan kesalahan, baik false positives maupun false negatives, menjadikan model ini solusi yang sangat efektif untuk tugas deteksi spam.

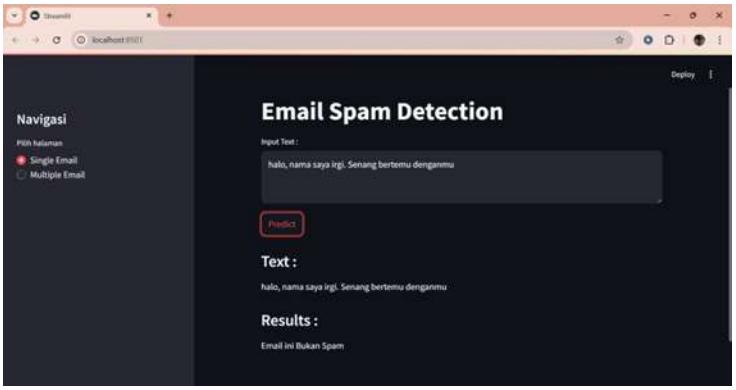
Streamlit

Pada bagian ini, saya akan membahas implementasi antarmuka pengguna (UI) yang telah dibuat untuk model klasifikasi email spam. Antarmuka ini dikembangkan menggunakan Streamlit, sebuah *framework* Python yang ideal untuk membuat aplikasi *web* interaktif dengan cepat, memungkinkan pengguna untuk berinteraksi langsung dengan model tanpa perlu memiliki keahlian teknis yang mendalam.

Tujuan utama aplikasi ini adalah membantu pengguna mengidentifikasi apakah suatu email adalah spam atau bukan. Ini sangat berguna untuk menyaring email berbahaya, promosi yang tidak diinginkan, atau pesan palsu (phishing). Dengan otomatisasi, pengguna tidak perlu lagi membaca setiap email secara manual untuk menentukan keasliannya.

1. Single Email

Aplikasi ini mendemonstrasikan proses kerja secara internal. Proses yang terjadi setelah tombol "Predict" ditekan.

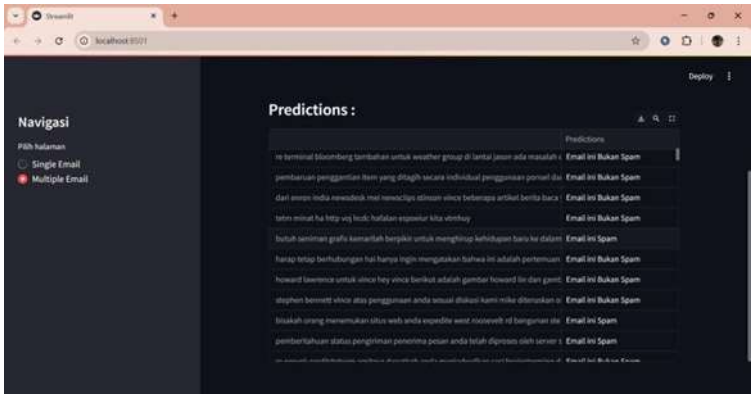


Gambar 21. email bukan spam

Aplikasi ini berhasil mengklasifikasikan email yang bersih dengan benar, menunjukkan bahwa model yang digunakan berfungsi secara efektif untuk tugas deteksi spam sederhana.

2. Multiple Email

Selanjutnya, Aplikasi ini mendemonstrasikan kemampuan untuk memproses data dalam jumlah besar, yang merupakan salah satu keunggulan model machine learning.



Gambar 22. Hasil Prediksi

Aplikasi ini berhasil menunjukkan bahwa ia tidak hanya mampu mengidentifikasi spam pada satu email, tetapi juga dapat memproses dan mengklasifikasikan data dalam skala yang lebih besar, membuatnya menjadi alat yang kuat untuk analisis dan penyaringan email secara otomatis.

KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan, dapat disimpulkan bahwa penelitian ini berhasil membangun dan melatih model klasifikasi teks berbasis IndoBERT untuk mendeteksi email spam berbahasa Indonesia secara efektif. Pemanfaatan arsitektur transformer memungkinkan model memahami konteks dan pola bahasa yang kompleks yang tidak dapat ditangkap secara optimal oleh metode lain, dengan tahapan pengembangan yang meliputi pra-pemrosesan data secara ketat berupa pembersihan teks dan tokenisasi, pembagian dataset, serta pelatihan model menggunakan framework Hugging Face Trainer, sehingga membuktikan bahwa IndoBERT merupakan fondasi yang efektif untuk tugas klasifikasi teks dalam bahasa Indonesia.

Model yang dihasilkan menunjukkan akurasi dan performa yang sangat tinggi, ditunjukkan melalui nilai presisi, recall, dan F1-score yang unggul dalam mengklasifikasikan email sebagai "spam" atau "ham", serta tingkat kesalahan klasifikasi yang sangat minimal berdasarkan matriks konfusi. Selain itu, penelitian ini menegaskan bahwa pra-pemrosesan data yang baik, meliputi pembersihan teks, tokenisasi, dan padding, serta pengaturan parameter pelatihan seperti learning rate sebesar 3e−5 dan jumlah epoch sebanyak 10, memiliki pengaruh yang signifikan terhadap kinerja model, karena memungkinkan proses pembelajaran berlangsung secara efisien, stabil, dan mencapai konvergensi yang optimal tanpa mengalami overfitting maupun underfitting, sehingga kombinasi aspek-aspek tersebut menjadi kunci dalam mencapai performa deteksi email spam yang optimal.

DAFTAR PUSTAKA

Akraman, R., Candiwan, C., & Priyadi, Y. (2018). Pengukuran Kesadaran Keamanan Informasi Dan Privasi Pada Pengguna Smartphone Android Di Indonesia. *JURNAL SISTEM INFORMASI BISNIS*, *8*(2). https://doi.org/10.21456/vol8iss2pp1-8

Assiroj, P., Kurnia, A., & Alam, S. (2023). The performance of Naïve Bayes, support vector machine, and logistic regression on Indonesia immigration sentiment analysis. *Bulletin of Electrical Engineering and Informatics*, *12*(6). https://doi.org/10.11591/eei.v12i6.5688

Friska Aditia Indriyani, Ahmad Fauzi, & Sutan Faisal. (2023). Analisis sentimen aplikasi tiktok menggunakan algoritma naïve bayes dan support vector machine. *TEKNOSAINS : Jurnal Sains, Teknologi Dan Informatika*, *10*(2). https://doi.org/10.37373/tekno.v10i2.419

Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., & Alegre, E. (2023). A review of spam email detection: analysis of spammer strategies and the dataset shift problem. *Artificial Intelligence Review*, *56*(2). https://doi.org/10.1007/s10462-022-10195-4

Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., & Alegre, E. (2025). Spam email classification based on cybersecurity potential risk using natural language processing. *Knowledge-Based Systems*, *310*. https://doi.org/10.1016/j.knosys.2024.112939

Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. In *IEEE Access* (Vol. 7). https://doi.org/10.1109/ACCESS.2019.2954791

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. In *Information (Switzerland)* (Vol. 10, Number 4). https://doi.org/10.3390/info10040150

Kurniawan, N. D., Ferdian, P. R., & Hidayati, N. (2025). Analisis Sentimen Algoritma Naïve Bayes, Support Vector Machine, dan Random Forest Pada Ulasan Aplikasi Ajaib. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *11*(1). https://doi.org/10.25077/teknosi.v11i1.2025.87-97

Martani, B. A. C., & Budi Setiawan, E. (2022). Naïve Bayes-Support Vector Machine Combined BERT to Classified Big Five Personality on Twitter. *Jurnal RESTI*, *6*(6). https://doi.org/10.29207/resti.v6i6.4378

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2022). Deep Learning-Based Text Classification. In *ACM Computing Surveys* (Vol. 54, Number 3).

https://doi.org/10.1145/3439726

Perwira, R. I., Permadi, V. A., Purnamasari, D. I., & Agusdin, R. P. (2025). Domain-Specific Fine-Tuning of IndoBERT for Aspect-Based Sentiment Analysis in Indonesian Travel User-Generated Content. *Journal of Information Systems Engineering and Business Intelligence*, *11*(1). https://doi.org/10.20473/jisebi.11.1.30-40

Putri, L. I., Ananta, G. P., & Syafa'at, I. (2024). Is the Problem Based Learning Using Media Puzzle Effective on Students' Mathematical Connection Ability? *Al Ibtida: Jurnal Pendidikan Guru MI*, *11*(2). https://doi.org/10.24235/al.ibtida.snj.v11i2.15048

Subowo, E. (2024). Implementasi Pembelajaran Mendalam dalam Klasifikasi Sentimen Ulasan Aplikasi: Evaluasi Model BERT, LSTM, dan CNN. *Jurnal Surya Informatika*, *14*(2). https://doi.org/10.48144/suryainformatika.v14i2.1973

Talaat, A. S. (2023). Sentiment analysis classification system using hybrid BERT models. *Journal of Big Data*, *10*(1). https://doi.org/10.1186/s40537-023-00781-w

Yefferson, D. Y., Lawijaya, V., & Girsang, A. S. (2024). Hybrid model: IndoBERT and long short-term memory for detecting Indonesian hoax news. *IAES International Journal of Artificial Intelligence*, *13*(2). https://doi.org/10.11591/ijai.v13.i2.pp1913-1924

Yulianti, E., & Nissa, N. K. (2024). ABSA of Indonesian customer reviews using IndoBERT: single-sentence and sentence-pair classification approaches. *Bulletin of Electrical Engineering and Informatics*, *13*(5). https://doi.org/10.11591/eei.v13i5.8032

Zhao, L., Gao, W., & Fang, J. (2024). Optimizing Large Language Models on Multi-Core CPUs: A Case Study of the BERT Model. *Applied Sciences (Switzerland)*, *14*(6). https://doi.org/10.3390/app14062364