# BAB II

# LANDASAN TEORI

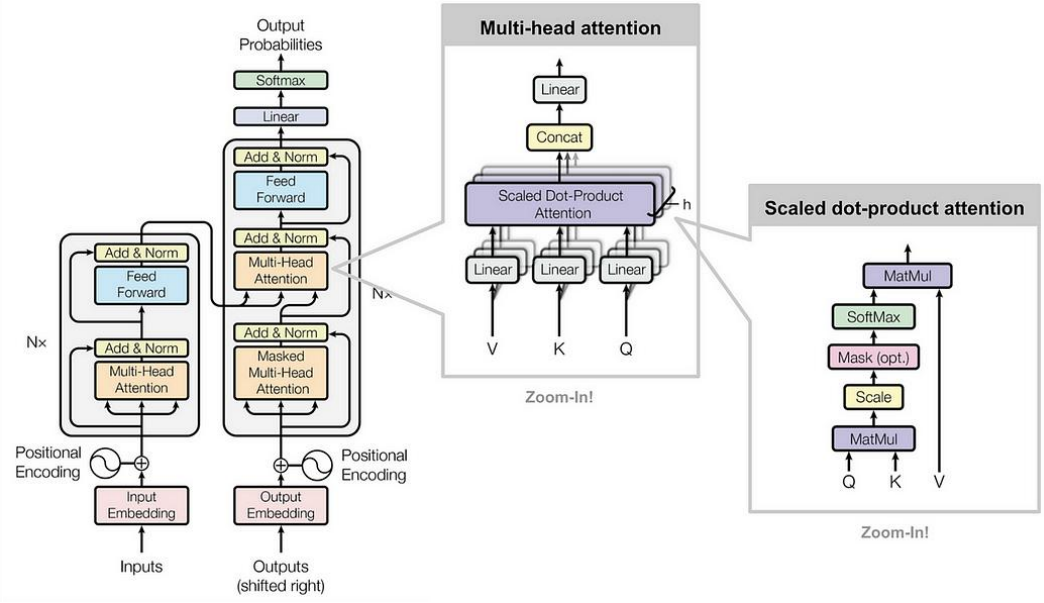
## 2.1 *Natural Language Processing*

*Natural Language Processing* (NLP) atau pemrosesan bahasa alami adalah salah satu cabang dari *Artificial Intelligence* (AI) yang berfokus pada pengembangan sistem yang dapat menerima dan memahami bahasa alami manusia (Carrasco Ramírez, 2024). Seiring perkembangannya, NLP bertujuan untuk mengonversi bahasa yang diproses oleh komputer (berbentuk bit dan *byte*) menjadi bahasa manusia yang mudah dipahami (Dong, 2024). NLP berperan sebagai dasar untuk menciptakan komunikasi antara mesin dan manusia melalui pemrosesan bahasa, baik dalam bentuk lisan maupun tulisan yang digunakan sehari-hari. NLP memiliki beragam aplikasi, seperti klasifikasi teks, analisis sentimen, penjawaban pertanyaan, dan ekstraksi informasi (Uchendu & Le, 2024).

Beberapa teknik yang sering digunakan dalam NLP antara lain *tokenisasi* (memecah teks menjadi kata atau unit terkecil), *stemming* (mengembalikan kata ke bentuk dasarnya), dan *lemmatization* (mengolah kata untuk menghasilkan bentuk yang lebih baku) (Nair, 2023). Dalam konteks pengolahan peraturan perundang-undangan, penerapan NLP sangat relevan untuk mempercepat proses pengambilan keputusan yang cepat dan akurat, terutama dalam mengklasifikasikan peraturan. Penggunaan model NLP dapat mempercepat proses ini, mengurangi ketergantungan pada analisis manual yang memakan waktu dan sumber daya (Dong, 2024).

## 2.2 *Transformers*

*Transformer*s adalah arsitektur yang dirancang untuk menangani tugas-tugas terjemahan dalam berbagai bahasa (Kamble & Kasodekar, 2024). Secara umum, *transformer*s dapat diartikan sebagai arsitektur *deep learning* baru yang mengandalkan mekanisme *self-attention*, dengan dua komponen utama yaitu *encoder* dan *decoder* (Y. Li dkk., 2024). *Self-attention* berfungsi untuk menghubungkan setiap kata dalam kalimat dengan kata lainnya, sehingga memungkinkan setiap kata untuk saling berhubungan. Arsitektur *transformers* ini dapat dilihat pada Gambar 2.1.




Gambar 2.1 Arsitektur *Transformers* (Singh & Mahmoud, 2020)

Pada tahap *encoder*, terdapat mekanisme *attention* yang memungkinkan sistem untuk memproses semua masukan yang diterima, yang kemudian dibagi menjadi kata-kata terpisah (Islam dkk., 2024). Dalam *encoder*, *self-attention* bekerja untuk menganalisis dan menentukan hubungan antar kata, sehingga setiap kata dapat dihubungkan dengan kata lainnya.

Sementara itu, pada tahap *decoder* terdapat *multi self-attention* yang mengambil *embedding* dari urutan *input* dan target untuk menentukan hubungan antara kata dalam *input* dan urutan kata dalam target (R. Zhang dkk., 2023). Secara keseluruhan, *transformers* menggunakan arsitektur *encoder-decoder*, di mana *encoder* bertugas untuk menganalisis dan memahami konteks serta struktur tata bahasa dalam data yang dilatih, sedangkan *decoder* berfungsi untuk memahami hubungan antara kata dalam data latih dan data uji.

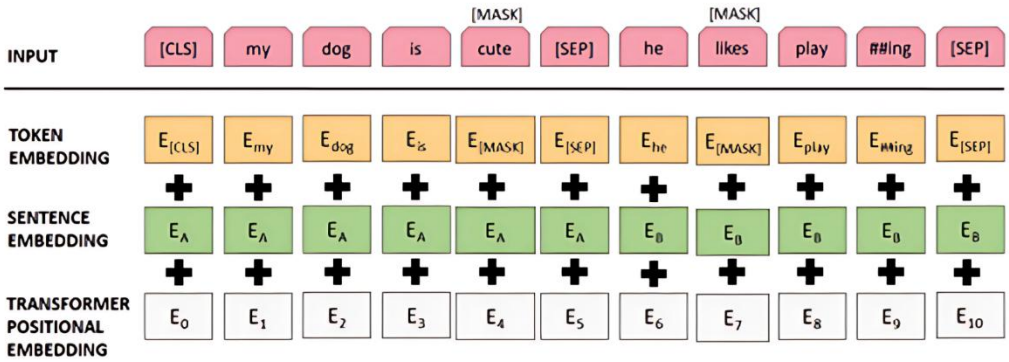
*Transformers* menjadi dasar bagi berbagai model NLP, salah satunya adalah *Bidirectional Encoder Representations from Transformers* (BERT) (Pradipta & Widodo, 2024). BERT mengadopsi pendekatan *bidirectional*, yang memungkinkan model ini untuk mempertimbangkan konteks kata baik dari sisi kiri maupun kanan. Pendekatan ini sangat penting untuk memahami makna yang lebih dalam teks-teks hukum yang sering kali memiliki makna yang ambigu.

## 2.3 BERT (*Bidirectional Encoder Representations from Transformers*)

BERT (*Bidirectional Encoder Representations from Transformers*) adalah model yang terkenal karena mengintegrasikan pelatihan dua arah pada *transformers* dengan mekanisme *attention* dalam pemodelan bahasa (Fang dkk., 2023). BERT mengubah pendekatan model bahasa sebelumnya yang hanya memandang urutan teks secara satu arah, baik kiri-kanan atau kanan-kiri. Dengan menggunakan pelatihan dua arah, BERT memungkinkan model untuk memiliki pemahaman konteks yang lebih dalam.

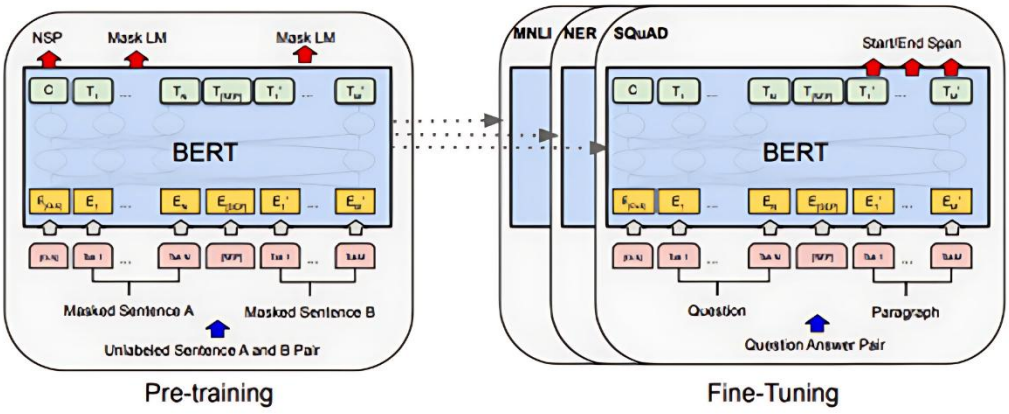
Sebelum melakukan pemrosesan, BERT memerlukan *input* khusus. Gambar 2.2 menunjukkan representasi *input* BERT, yang meliputi tiga tahap utama: token

*embedding* (menambahkan token [CLS] di awal kalimat dan [SEP] di akhir kalimat), *segment embedding* (penanda untuk membedakan kalimat yang berbeda), dan *positional embedding* (untuk menunjukkan posisi kata dalam kalimat) (Ali dkk., 2023).




Gambar 2.2 Representasi *Input* BERT (Bai dkk., 2020)

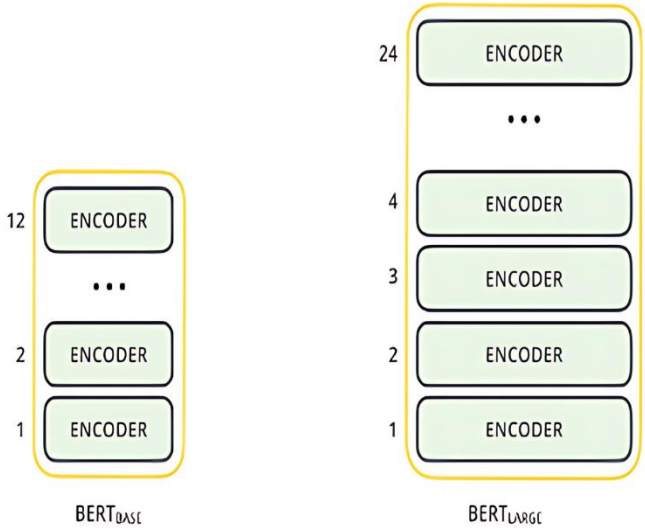
Selama pelatihan, BERT menggunakan teknik baru yang disebut *Masked Language Model* (MLM), yang secara acak menutupi 15% kata dalam kalimat dan mencoba memprediksi kata-kata tersebut (Kryeziu & Shehu, 2023). Dalam memprediksi kata yang hilang, BERT memproses kalimat secara dua arah, memperhatikan kata sebelum dan sesudahnya, untuk memperoleh makna kata sepenuhnya. Pendekatan ini efektif dalam mengatasi ambiguitas kata yang memiliki ejaan serupa namun makna berbeda. Selain itu, BERT juga menggunakan *Next Sentence Prediction* (NSP), yang membantu model memahami hubungan antar dua kalimat, sehingga model dapat memperhitungkan keterkaitan makna antar kalimat (Kryeziu & Shehu, 2023). BERT melibatkan dua tahapan penting, yaitu *pre-training* dan *fine-tuning* (Ali dkk., 2023). Gambar 2.3 menggambarkan kedua tahapan ini.




Gambar 2.3 Proses *Pre-Training* dan *Fine-Tuning* pada BERT (Joloudari dkk., 2023)

Pada tahap *pre-training*, BERT menggunakan *Masked Language Model* (MLM) untuk melakukan *masking* pada token *input* dan memprediksi kata berdasarkan konteksnya. Selain itu, *pre-training* juga mencakup *Next Sentence Prediction* (NSP) untuk membantu model memahami hubungan antar kalimat (Ali dkk., 2023).

Dalam tahap *input*, BERT menambahkan token khusus [CLS] di awal kalimat dan [SEP] di akhir kalimat untuk menandai bagian penting dalam teks. Setelah *pre-training*, tahap *fine-tuning* dilakukan dengan menambahkan *layer output* tambahan agar model BERT dapat menyelesaikan berbagai tugas (Zhao dkk., 2023). Ada dua versi arsitektur BERT yang berbeda berdasarkan ukuran dan performanya (C. Liu dkk., 2023), yang dijelaskan pada Gambar 2.4.




Gambar 2.4 Arsitektur BERT berdasarkan Ukuran (Kumar dkk., 2023)

$BERT_{BASE}$ memiliki 110 juta parameter dan terdiri dari 12 *transformers block*, 12 *attention layer*, serta 768 *hidden layers*. Sedangkan $BERT_{LARGE}$ memiliki 340 juta parameter dengan 24 *transformers block*, 16 *attention layer*, dan 1024 *hidden layers*. $BERT_{LARGE}$ umumnya menunjukkan performa yang lebih baik dibandingkan $BERT_{BASE}$. BERT memerlukan sumber daya komputasi yang tinggi, yang mendorong pengembangan varian yang lebih ringan seperti BERT-*medium*, BERT-*small*, BERT-*mini*, dan BERT-*tiny* (Gessler & Zeldes, 2022). Tabel 2.1 menunjukkan jumlah parameter yang digunakan oleh berbagai varian BERT.

Tabel 2.1 Parameter Varian Model BERT

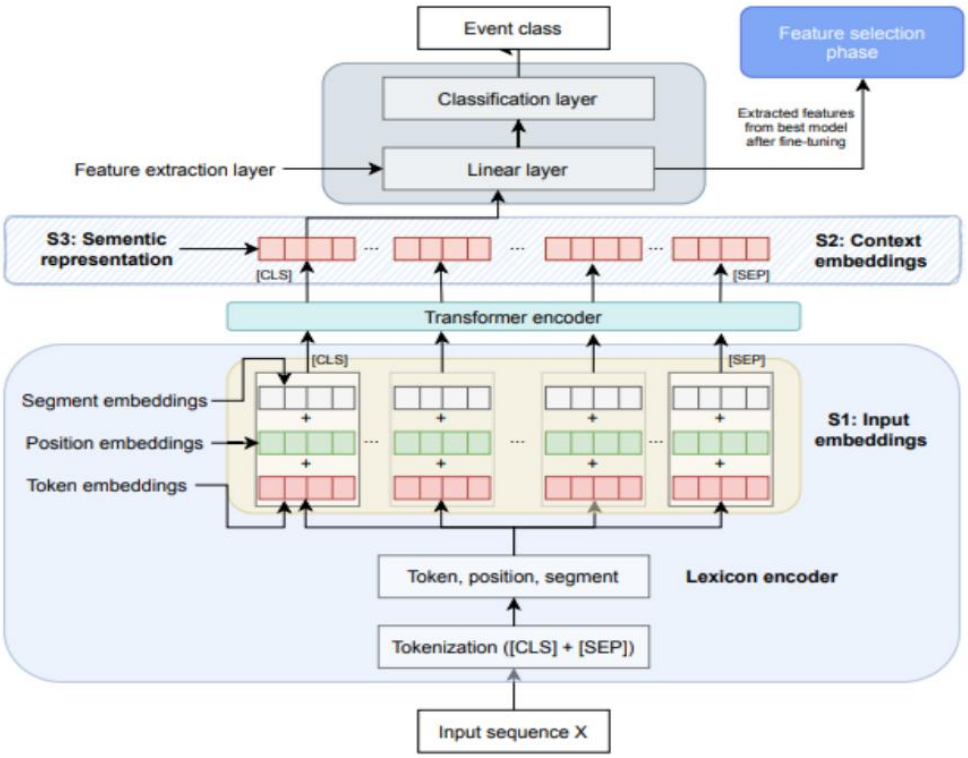
| **Varian** | **Parameter** |
|---|---|
| BERT-*base* | 110 juta |
| BERT-*large* | 340 juta |
| BERT-*medium* | 41,7 juta |
| BERT-*small* | 29,1 juta |

| | |
|---|---|
| BERT-*mini* | 11,3 juta |
| BERT-*tiny* | 4,4 juta |

Meskipun BERT sangat kuat dalam pemrosesan bahasa, kelemahan utamanya adalah kebutuhan akan sumber daya komputasi yang besar, yang membatasi penerapannya dalam banyak skenario, terutama bagi pengguna dengan sumber daya terbatas (Ali dkk., 2023). Untuk mengatasi masalah ini, varian BERT yang lebih efisien dan ringan, seperti DistilBERT, dikembangkan. BERT dapat di-*fine-tune* untuk berbagai tugas spesifik, salah satunya adalah klasifikasi peraturan perundang-undangan. Melalui *fine-tuning*, BERT dapat disesuaikan untuk menangani teks spesifik, seperti dokumen hukum yang memiliki struktur dan bahasa yang unik.

## 2.4 DistilBERT

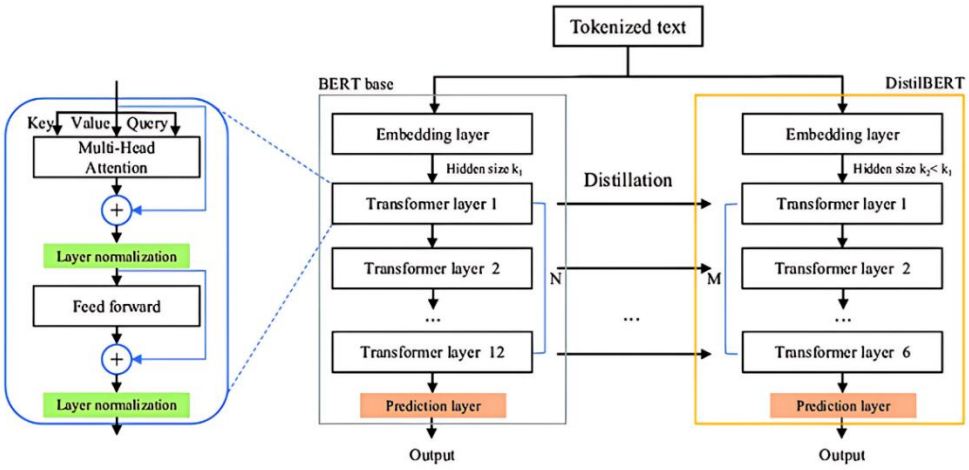
*Distillation* BERT, atau yang lebih dikenal dengan DistilBERT, dirancang untuk mengecilkan ukuran dan mempercepat pelatihan model *encoder* dua arah dari *transformers* BERT (X. Wang dkk., 2023). Dengan menggunakan teknik distilasi (penyulingan), DistilBERT berhasil mengurangi parameter model BERT hingga 40% dan meningkatkan kecepatan inferensi hingga 60% (Sanh dkk., 2019). Arsitektur model ekstraksi fitur yang menggunakan DistilBERT dapat dilihat pada Gambar 2.5.




Gambar 2.5 Model Ekstraksi Fitur Menggunakan DistilBERT (Martinez dkk., 2024)

DistilBERT menerima *input* X berupa urutan kata dari dataset. *Input* ini kemudian diubah menjadi sekumpulan vektor *embedding* (Muffo & Bertino, 2020), di mana setiap vektor dipetakan secara berurutan ke setiap kata (S1: *Input Embeddings*). DistilBERT menggunakan *transformers encoder* untuk mempelajari informasi kontekstual dari setiap kata, dengan mekanisme *self-attention* untuk menghasilkan *embedding* kontekstual (S2: *Context Embeddings*). *Embedding* kontekstual yang diekstraksi untuk setiap kata kemudian digabungkan menjadi satu vektor untuk mewakili informasi semantik (S3: *Semantic Representation*). S3 ini menjadi input untuk lapisan yang sepenuhnya terhubung, yang menghasilkan vektor berukuran d, di mana d adalah jumlah neuron.

Setelah itu, lapisan klasifikasi ditempatkan di akhir model ekstraksi fitur untuk menyempurnakan DistilBERT yang telah dilatih pada tugas tertentu dan memprediksi kelas *event* yang sesuai untuk setiap urutan yang dimasukkan (Oh dkk., 2023). Model ekstraksi fitur ini berada pada lapisan *feature extraction*, yang mencakup lapisan linier untuk mengaktifkan secara linier *input* dari lapisan sebelumnya, dan keluaran dari fungsi ini adalah *input* dan bias. Setelah melalui lapisan linier, dilakukan tahap klasifikasi yang bertujuan untuk mengklasifikasikan setiap *input* yang dihasilkan dari lapisan linier. Fokus utama dari arsitektur DistilBERT ditunjukkan pada Gambar 2.6.




Gambar 2.6 Arsitektur DistilBERT (Martinez dkk., 2024)

Setiap komponen menggunakan pengetahuan distilasi untuk mengurangi parameter model dasar BERT. Proses ini membuat DistilBERT berjalan 60% lebih cepat dan membutuhkan 40% lebih sedikit parameter dibandingkan dengan BERT, namun tetap mempertahankan lebih dari 95% kinerja BERT (Ding dkk., 2023). Dengan distilasi dapat memperkirakan distribusi keluaran BERT menggunakan model yang lebih ringkas, yaitu DistilBERT, yang terdiri dari enam lapisan

*transformers*. Dengan 66 juta parameter yang dapat dilatih, DistilBERT memiliki performa yang lebih baik dibandingkan model dasar BERT yang memiliki 110 juta parameter (Ansell dkk., 2023). Pelatihan DistilBERT menggunakan akumulasi gradien dan ukuran *batch* 16. Untuk efisiensi lebih lanjut, metode ini menggabungkan gradien dari beberapa *batch mini* sebelum memodifikasi parameter (J. Li dkk., 2023). Pendekatan pelatihan ini menghilangkan *Next Sentence Prediction* (NSP) dan tujuan pembelajaran penyematan segmen, yang menyederhanakan proses pelatihan dan menjadikan DistilBERT pengganti BERT yang efisien dan produktif untuk berbagai aplikasi NLP.

*Pre-training* pada model DistilBERT merujuk pada proses pelatihan awal yang dilakukan pada data yang sangat besar dan umum untuk membangun representasi bahasa dasar. Dalam *pre-training*, model dilatih menggunakan tugas-tugas seperti *Masked Language Modeling* (MLM) dimana sebagian kata dalam kalimat disembunyikan, dan model diminta untuk memprediksi kata yang hilang tersebut. Proses ini memungkinkan model untuk mempelajari pola bahasa umum yang ada dalam data teks, tanpa terikat pada tugas spesifik (Kryeziu & Shehu, 2023). *Fine-tuning* adalah langkah lanjutan yang dilakukan setelah *pre-training*, di mana model yang telah dilatih pada data umum kemudian disesuaikan (*fine-tuned*) untuk tugas atau domain spesifik. Melalui *fine-tuning*, model dapat disesuaikan agar lebih efektif dalam memahami terminologi dan struktur, meningkatkan kinerjanya dalam tugas-tugas spesifik (Radiya-Dixit, 2020).

Keunggulan DistilBERT dalam efisiensi komputasi menjadikannya pilihan yang sangat baik untuk tugas NLP yang memerlukan pemrosesan teks dalam jumlah

besar, seperti pengolahan dokumen hukum khususnya peraturan perundang-undangan (Oh dkk., 2023). Dalam penelitian ini, DistilBERT digunakan untuk memperoleh akurasi yang tinggi pada evaluasi, namun tetap hemat dalam penggunaan sumber daya selama proses *fine-tuning*. Hal ini menjadikan DistilBERT pilihan yang ideal untuk aplikasi yang memerlukan performa tinggi namun tetap efisien dalam penggunaan memori dan waktu komputasi.

## 2.5 Optimasi

Optimasi merupakan suatu algoritma yang dapat meningkatkan efisiensi proses pembelajaran model algoritma dengan meningkatkan akurasinya (Sihotang dkk., 2023). Pada NLP, optimasi bertujuan untuk meningkatkan kinerja model, perbaikan terhadap kecepatan eksekusi serta penggunaan dataset dan sumber daya yang lebih efisien. Optimasi membentuk model yang paling akurat dengan mencari parameter optimal dan merubah parameter sedemikian mungkin untuk meminimalisir *loss* atau *cost function* (Park dkk., 2020). Terdapat 3 (tiga) tindakan optimasi pada penelitian ini sebagai berikut.

*Adaptive Moment Estimation* yang disebut Adam merupakan algoritma optimalisasi hasil kombinasi antara *Root Mean Square Propagation* (RMSProp) dan *Stochastic Gradient Descent* (SGD) (Y. Wang dkk., 2021). Sebelum menjelaskan detail tentang Adam, *Stochastic Gradient Descent* (SGD) merupakan salah satu jenis dari metode *optimizer gradient descent* yang bekerja dengan meminimalisir *error* melalui pertimbangan *error* dari setiap poin datanya pada setiap iterasinya. Adagrad adalah algoritma modifikasi dari *Stochastic Gradient Descent* (SGD) yang mengoptimalkan berdasarkan gradien, menyesuaikan

*learning rate* untuk setiap parameter. Adadelta merupakan perkembangan dari Adagrad yang menyesuaikan *learning rate* berdasarkan jendela pergerakan gradien daripada mengakumulasi jumlah gradien kuadrat sepanjang waktu. *Root Mean Square Propagation* (RMSProp) juga merupakan perkembangan dari algoritma Adagrad yang dibuat dengan performa lebih baik dalam pengaturan *non-konveks* dengan mengganti akumulasi gradien menjadi *wighted moving average* (Maiya dkk., 2021).

Adam menghitung tingkat pembelajaran adaptif untuk setiap parameter berdasarkan perkiraan momen pertama dan kedua dari gradien. Adam menggabungkan kelebihan dari Adagrad yang efektif untuk gradien yang jarang (*sparse*) dan RMSProp yang baik dalam pengaturan *online* dan *non-stasioner*. Algoritma ini menggabungkan metode SGD dari RMSProp untuk menyesuaikan *learning rate* dan *weighted moving average* dengan memanfaatkan momentum. Dalam implementasi modelnya, Adam memiliki kemampuan untuk secara otomatis memperbarui bobot dan *learning rate* (Nanni dkk., 2021). AdamW (*Adaptive Moment Estimation with Weight Decay*) merupakan pengembangan dari *optimizer* Adam dengan menambahkan regularisasi *weight decay* secara eksplisit. AdamW memodifikasi Adam dengan memisahkan *weight decay* dari proses *update* parameter. Ini mengatasi beberapa kekurangan Adam dalam mengelola regularisasi *weight decay*. Persamaan AdamW dituliskan pada Persamaan 2.1 sebagai berikut.

$$\theta_{t+1} = \theta_t - \eta \cdot \left( \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} + \lambda \cdot \theta_t \right) \tag{2.1}$$

Keterangan notasi:

$\theta_t$ = nilai awal parameter

$\widehat{m}_t$ = momentum gradien

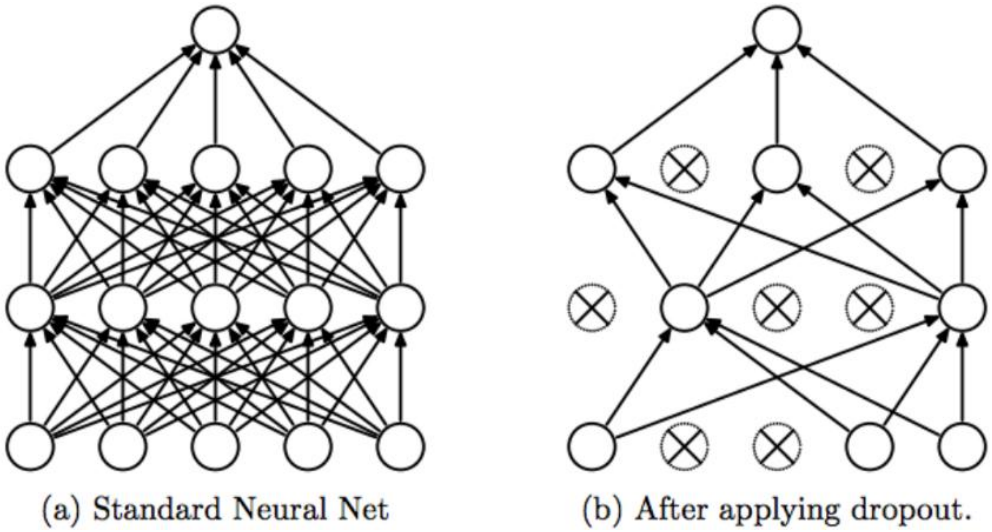
$\hat{v}_t$ = momentum gradien kuadrat

$\epsilon$ = nilai stabilisasi

$\lambda = weight\ decay$

$\eta = learning\ rate$

*Dropout* merupakan teknik regulasi yang sangat efektif untuk menyederhanakan kompleksitas jaringan saraf untuk menghindari *overfitting* (Shunk, 2022). Cara kerja *dropout* yaitu memutuskan beberapa neuron penghubung, oleh karena itu neuron sebelumnya harus mencari neuron lain untuk dapat melanjutkan ke lapisan terakhir, neuron yang akan dihilangkan akan dipilih secara acak oleh sistem dan bobotnya tidak diperbarui dalam proses pelatihan. Ini bertujuan untuk mengurangi pembelajaran yang saling bergantung di setiap neuron. Lapisan *dropout* diimplementasikan pada masing-masing lapisan *fully connected* yang merupakan konfigurasi paling umum. Lapisan *dropout* juga dapat diimplementasikan pada lapisan konvolusi operasi ReLu (Pansambal & Nandgaokar, 2023). *Dropout layer* diilustrasikan pada Gambar 2.7.




Gambar 2.7 Ilustrasi *Dropout* (Salehin & Kang, 2023)

*Multi-Task Learning* adalah pendekatan dalam *machine learning* di mana satu model dilatih untuk menyelesaikan beberapa tugas yang berbeda, tetapi saling berkaitan, secara bersamaan (Y. Zhang & Yang, 2018). Tujuan utamanya adalah untuk meningkatkan kinerja dan efisiensi model dengan memanfaatkan pengetahuan yang diperoleh dari satu tugas untuk membantu tugas lainnya. Alih-alih membuat model terpisah untuk setiap tugas, MTL memungkinkan berbagi representasi dan parameter antar tugas, yang pada akhirnya dapat meningkatkan performa keseluruhan model. Konsep dasar dari MTL adalah bahwa beberapa tugas dapat saling memperkuat satu sama lain ketika dilatih bersama, terutama jika tugas-tugas tersebut saling berkaitan atau berasal dari domain yang sama.

MTL membantu model menghindari *overfitting*, karena model belajar dari data tambahan yang berasal dari tugas-tugas lain. Selain itu, MTL dapat meningkatkan generalisasi model dengan menangkap fitur bersama (*shared features*) yang berguna untuk semua tugas (Brasoveanu dkk., 2020).

Dalam konteks pemrosesan bahasa alami (*Natural Language Processing*/NLP), MTL banyak diterapkan pada arsitektur *transformer-based models* seperti BERT, DistilBERT, dan lainnya, di mana token [CLS] yang menyimpan informasi representasi kalimat digunakan sebagai *input* ke berbagai *task-specific heads* untuk menyelesaikan masing-masing tugas klasifikasi atau prediksi.

## 2.6 Peraturan Perundang-Undangan di Indonesia

Di Indonesia, sumber hukum umumnya dibagi menjadi beberapa kategori, seperti konstitusi, peraturan perundang-undangan, peradilan, dan hukum adat. Dalam penelitian ini, data hukum yang digunakan berasal dari peraturan perundang-undangan (Mahardika, 2023). Dalam hierarki hukum negara, jenis-jenis peraturan yang ada mencakup Undang-Undang Dasar, TAP MPR, Undang-Undang, Peraturan Presiden, UUDRT, Peraturan Pemerintah, Keputusan Presiden, Instruksi Presiden, Peraturan Menteri, Peraturan Badan Negara, dan Peraturan Daerah (Simanjuntak, 2019). Dokumen peraturan perundang-undangan di Indonesia memiliki karakteristik yang berbeda dengan dokumen hukum di negara lain. Peraturan di Indonesia cenderung menggunakan bahasa yang sangat formal dan rumit, dengan format yang bervariasi antara satu peraturan dengan lainnya (Firma dkk., 2019).

Selain itu, peraturan-peraturan tersebut juga sering mengalami perubahan status, seperti pembaruan atau pembatalan, yang perlu dipantau dengan teliti (Pardede, 2023). Oleh karena itu, model NLP yang digunakan untuk memproses dokumen peraturan perundang-undangan di Indonesia harus dapat menangani perbedaan format dan kompleksitas bahasa hukum tersebut. Pada dasarnya,

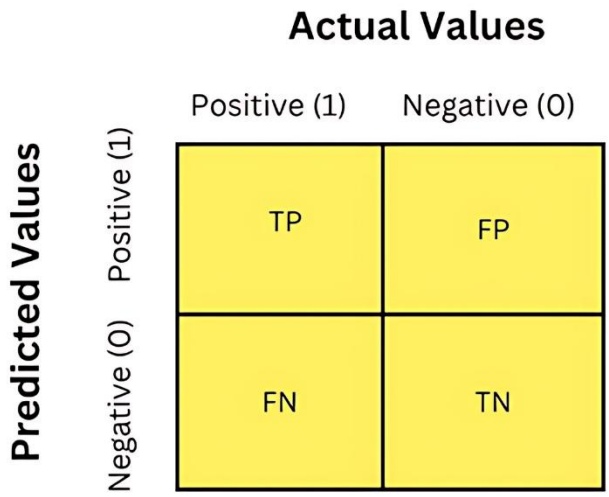
pengolahan peraturan perundang-undangan bertujuan untuk mengklasifikasikan status hukum suatu peraturan, apakah masih berlaku, sudah tidak berlaku, atau mengalami perubahan status. Model yang digunakan harus mampu mengidentifikasi teks yang menunjukkan status hukum peraturan dan mengklasifikasikannya dengan akurat.

Selain klasifikasi status hukum, penting juga untuk mengklasifikasikan jenis peraturan yang ada dalam peraturan perundang-undangan Indonesia. Jenis peraturan ini meliputi berbagai macam bentuk hukum yang dikeluarkan oleh lembaga negara dan memiliki kedudukan yang berbeda dalam hierarki hukum. Penting bagi model untuk tidak hanya mengenali perubahan status peraturan, tetapi juga untuk secara tepat mengidentifikasi jenis peraturan yang dimaksud. Hal ini berguna dalam konteks klasifikasi hukum yang lebih tepat, karena setiap jenis peraturan memiliki dampak dan implikasi yang berbeda dalam sistem hukum Indonesia. Pemahaman terhadap jenis-jenis peraturan ini akan meningkatkan akurasi model dalam memproses dan mengklasifikasikan dokumen hukum sesuai dengan konteks yang berlaku.

## 2.7 *Confusion Matrix*

*Confusion matrix* merupakan metode untuk mengevaluasi kinerja model dalam masalah klasifikasi, yang mengukur tingkat akurasi model dengan *output* berupa dua kelas atau lebih, untuk menentukan hasil yang benar atau salah (Riehl dkk., 2023). *Confusion matrix* terdiri dari empat kategori, yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN), yang masing-masing

memiliki dua nilai, yaitu nilai aktual dan nilai prediksi (Dadalto dkk., 2023). *Confusion matrix* dapat dilihat pada Gambar 2.8.




Gambar 2.8 *Confusion Matrix* (Riehl dkk., 2023)

*True Positive* (TP) adalah data aktual yang bernilai benar atau 1 dan prediksi yang juga bernilai positif. *False Positive* (FP) adalah data yang seharusnya salah atau 0, tetapi diprediksi positif. *True Negative* (TN) adalah data yang bernilai salah atau 0 dan prediksi yang bernilai negatif. *False Negative* (FN) adalah data yang seharusnya benar atau 1, namun diprediksi negatif.

Setelah memperoleh nilai dari keempat kategori *confusion matrix* (Mielniczuk & Wawrzeńczyk, 2024), hasil perhitungannya menghasilkan metrik seperti akurasi, *precision, recall, dan F1-score* (Owusu-Adjei dkk., 2023). Nilai-nilai ini digunakan untuk menilai kinerja model dan menunjukkan bahwa penggunaan DistilBERT dapat menghasilkan akurasi yang lebih baik. Akurasi mencerminkan sejauh mana model dapat mengklasifikasikan data dengan benar. *Precision* mengukur ketepatan prediksi model terhadap data yang diminta. *Recall* menggambarkan seberapa banyak kelas positif yang dapat diprediksi dengan tepat. *F1-score* adalah nilai rata-

rata harmonis antara *precision* dan *recall*. Berikut adalah rumus untuk menghitung nilai akurasi, *precision, recall, dan F1-score* pada Persamaan 2.2 – 2.5 (Riehl dkk., 2023).

$$Accuration = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{2.5}$$

## 2.8 Penelitian Terkait

Dalam beberapa tahun terakhir, banyak penelitian yang fokus pada pengembangan model-model berbasis *Artificial intelligence* (AI) untuk mengatasi tantangan dalam klasifikasi peraturan perundang-undangan. Peraturan perundang-undangan adalah teks yang sering kali rumit, dengan struktur yang kaku dan bahasa yang sangat teknis. Oleh karena itu, penelitian dalam pengolahan peraturan perundang-undangan sangat membutuhkan teknologi yang dapat menangani kompleksitas teks tersebut, mengklasifikasikan status hukum, mengklasifikasikan jenis peraturan, serta memberikan kemudahan akses dan pemahaman bagi para pengguna, baik itu praktisi hukum, pemerintah, maupun masyarakat umum.

Penelitian terkait dengan penggunaan model NLP dalam pengolahan dokumen hukum Indonesia, khususnya dalam klasifikasi peraturan perundang-undangan, masih sangat terbatas. Oleh karena itu, penelitian ini untuk mengoptimalkan model DistilBERT yang telah di-*fine-tuned* pada data peraturan perundang-undangan Indonesia dengan meningkatkan efisiensi dan akurasi dalam mengklasifikasikan status hukum dan jenis suatu peraturan. Oleh karena itu, dilakukan penyusunan *state of the art* yang berkaitan dengan penelitian ini serta mengevaluasi kesesuaian penerapan penelitian tersebut.

Tabel 2.2 *State Of The Art* Penelitian Terkait

| No. | Penulis, Tahun Terbit | Judul | Metode | Hasil |
|---|---|---|---|---|
| 1. | (Wehnert dkk., 2022) | *Applying BERT Embeddings to Predict Legal Textual Entailment* | Sentence-BERT, LEGAL-BERT | Penelitian ini mengeksplorasi tiga pendekatan untuk mengklasifikasikan keterlibatan tekstual hukum menggunakan penyematan BERT. Yang pertama menggabungkan penyematan Sentence-BERT dengan jaringan saraf grafik, sedangkan yang kedua menggunakan LEGAL-BERT, disesuaikan untuk klasifikasi terkait. Pendekatan ketiga menggunakan *encoder* KERMIT dengan BERT untuk menyematkan pohon *parse* sintaksis. Penelitian ini menyoroti bahwa LEGAL-BERT mengungguli pendekatan berbasis grafik, menunjukkan efektivitas model khusus domain dalam konteks kompetisi COLIEE pada versi bahasa Inggris dari Kode Sipil Jepang. |

| | | | | |
|---|---|---|---|---|
| | | | | Selain itu, kinerja terbaik arsitektur KERMITBERT pada data COLIEE 2021 mencapai akurasi 58, dengan akurasi rata-rata 51,03. |
| 2. | (Hegel dkk., 2021) | *The Law of Large Documents: Understanding the Structure of Legal Contracts Using Visual Cues* | BERT | Penelitian ini mengeksplorasi bagaimana isyarat visual seperti tata letak, gaya, dan penempatan teks secara signifikan meningkatkan pemahaman dokumen hukum yang panjang. Ini menyoroti bahwa strategi segmentasi tradisional sering mengabaikan elemen struktural ini, yang mengarah pada pengurangan akurasi dalam tugas-tugas seperti segmentasi dokumen, ekstraksi entitas, dan klasifikasi atribut. Dengan memasukkan isyarat visual melalui metode visi komputer, penelitian ini menunjukkan peningkatan kinerja pada tugas pemahaman dokumen panjang, melampaui metode yang ada pada Dataset Pemahaman Kontrak *Atticus*. |
| 3. | (Chaudhary dkk., 2020) | *TopicBERT for Energy Efficient* | TopicBERT | Penelitian ini mengeksplorasi TopicBERT menjadi kerangka kerja terpadu yang dirancang untuk mengoptimalkan biaya komputasi |

| | | | | |
|---|---|---|---|---|
| | | *Document Classification* | | penyempurnaan untuk klasifikasi dokumen. Dengan mengintegrasikan pembelajaran pelengkap dari model topik dan bahasa, ini secara signifikan mengurangi jumlah operasi *self-attention*, yang merupakan hambatan kinerja utama di BERT. Pendekatan ini menghasilkan Model TopicBERT mencapai percepatan 1,4x dalam penyetelan untuk tugas klasifikasi dokumen, yang berarti pengurangan biaya komputasi sekitar 40%. Model ini juga menghasilkan penurunan emisi karbon sekitar 40% sambil mempertahankan kinerja 99,9% di lima kumpulan data yang berbeda. |
| 4. | (Fragkogiannis dkk., 2023) | *Context-Aware Classification of Legal Document Pages* | BERT | Penelitian ini menyajikan metode untuk klasifikasi halaman dokumen hukum berdasarkan *context-aware* dengan meningkatkan *input* dengan token tambahan yang membawa informasi berurutan dari halaman sebelumnya. Pendekatan ini memungkinkan penggunaan model *transformer pre-training* seperti BERT, mengatasi keterbatasan panjang |

| | | | | |
|---|---|---|---|---|
| | | | | *input*. Menggunakan kumpulan data hukum bahasa Inggris dan Portugis menunjukkan peningkatan kinerja yang signifikan dalam klasifikasi halaman dokumen dibandingkan dengan pengaturan *non-looping* dan metode *context-aware* lainnya. Metode *context-aware* yang diusulkan, yang menggunakan model BERT berulang, mencapai kinerja tertinggi pada empat kelas dalam hal *F1-score*, dengan rata-rata makro 3,49 dan *F1-score* rata-rata tertimbang 0,84, mengungguli model canggih sebelumnya pada dataset US *Appellate Briefs*. |
| 5. | (Lu dkk., 2021) | *A Sentence-level Hierarchical BERT Model for Document Classification* | BERT Hierarchical (HBM) | Penelitian ini memperkenalkan Model BERT Hierarchical (HBM), yang dirancang untuk klasifikasi dokumen, terutama efektif dengan data berlabel terbatas (50 hingga 200 contoh). HBM berfokus pada pembelajaran fitur tingkat kalimat, meningkatkan kinerja dalam mengklasifikasikan dokumen panjang dibandingkan dengan metode sebelumnya. Eksperimen evaluasi menunjukkan keunggulannya dalam |

| | | | | |
|---|---|---|---|---|
| | | *with Limited Labelled Data* | | akurasi, dan model ini juga mengidentifikasi kalimat menonjol yang berfungsi sebagai penjelasan yang berguna untuk pelabelan dokumen. |
| 6. | (B. Li & Wang, 2023) | *Design of intelligent legal text analysis and information retrieval system based on BERT model* | BERT | Penelitian ini menyimpulkan bahwa model Bleem, yang didasarkan pada implikasi teks dan menggunakan lapisan BERT untuk pengkodean, mencapai akurasi 96% dalam ekstraksi elemen dokumen, mengungguli model lain. Ini menunjukkan efektivitas model dalam mengidentifikasi elemen semantik dalam dokumen hukum. Selanjutnya, penelitian merancang sistem analisis cerdas untuk dokumen hukum, yang terdiri dari *crawling* dokumen, analisis membaca, dan alat analisis, untuk membantu peneliti hukum dalam memperoleh informasi ringkasan dan memahami dokumen hukum. Sistem ini berhasil diterapkan pada tata kelola risiko publik yang komprehensif, menunjukkan implikasi praktis dari temuan penelitian. |

| | | | | |
|---|---|---|---|---|
| 7. | (Si & Roberts, 2021) | *Hierarchical Transformer Networks for Longitudinal Clinical Document Classification* | BERT | Penelitian ini menggunakan Jaringan *Transformer* Hierarkis pada tiga tingkat *encoder* berbasis *Transformer* untuk memodelkan dependensi jangka panjang dalam catatan klinis untuk prediksi tingkat pasien. Level pertama menggunakan model BERT *pre-training*, sedangkan level kedua dan ketiga terdiri dari *encoder* 2-*layer*. Arsitektur ini memungkinkan urutan *input* yang lebih panjang daripada model BERT tradisional, meningkatkan klasifikasi dokumen klinis. Hasil eksperimental pada dataset MIMIC-III menunjukkan bahwa model hierarkis ini mengungguli jaringan saraf hierarkis canggih sebelumnya. |
| 8. | (Aumiller dkk., 2021) | *Structural Text Segmentation of Legal Documents* | Sistem segmentasi yang dibangun di | Penelitian ini mengusulkan sistem segmentasi untuk dokumen hukum yang memprediksi koherensi topikal di seluruh segmen teks berurutan yang mencakup beberapa paragraf. Ini membahas keterbatasan metode tradisional yang sering mengabaikan konteks dan koherensi. Dengan memanfaatkan jaringan transformator dan membingkai segmentasi teks |

| | | | | |
|---|---|---|---|---|
| | | | atas jaringan transformator | struktural sebagai deteksi perubahan topikal, sistem melakukan klasifikasi independen untuk penyetelan yang efisien. Model ini dilatih pada kumpulan data baru dari sekitar 74.000 dokumen *Terms-of-Service*, menunjukkan peningkatan kinerja yang signifikan dibandingkan metode dasar. |
| 9. | (AL-Qurishi, 2023) | *Leveraging BERT Language Model for Arabic Long Document Classification* | BERT, RoBERT dan ToBERT | Penelitian ini mengusulkan dua model berbasis BERT yang dirancang khusus untuk mengklasifikasikan dokumen Arab yang panjang, mengatasi tantangan komputasi yang ditimbulkan oleh panjangnya. Model-model ini membagi dokumen menjadi kalimat dan menggunakan algoritma pencocokan kesamaan berbasis BERT untuk mengidentifikasi kalimat relevansi tinggi untuk klasifikasi. Penelitian ini menyempurnakan *Longformer* dan RoBERT untuk perbandingan, dengan model mereka mengungguli keduanya dalam hal akurasi, |

| | | | | |
|---|---|---|---|---|
| | | | | mencapai *F1-Score* makro 98,4 pada kumpulan data berita Arab, menunjukkan klasifikasi dokumen panjang yang efektif. |
| 10. | (Mamakas dkk., 2022) | *Processing Long Legal Documents with Pre-trained Transformers: Modding LegalBERT and Longformer* | LegalBERT | Penelitian ini mengeksplorasi modifikasi *Longformer*, yang dimulai dengan hangat dari LegalBERT, untuk memproses teks hukum yang lebih panjang (hingga 8.192 sub-kata) dan memperkenalkan representasi TF-IDF untuk LegalBERT. *Longformer* yang dimodifikasi mengungguli model canggih sebelumnya dalam tugas klasifikasi dokumen panjang dalam tolok ukur *LexGlue*. Sementara pendekatan TF-IDF efisien secara komputasi, dengan mengorbankan beberapa kinerja. Penelitian ini menyoroti pertukaran antara intensitas sumber daya dan akurasi klasifikasi, yang pada akhirnya menetapkan LegalBERT *Longformer* sebagai teknologi terbaru untuk pemrosesan dokumen hukum yang panjang. |

| | | | | |
|---|---|---|---|---|
| 11. | (Vatsal dkk., 2023) | *Classification of US Supreme Court Cases using BERT-Based Techniques* | BERT | Penelitian ini mengeksplorasi teknik berbasis BERT untuk mengklasifikasikan keputusan Mahkamah Agung AS dari *Database* Mahkamah Agung (SCDB). Ini mengatasi tantangan yang ditimbulkan oleh dokumen panjang, menggunakan metode seperti analisis potongan, ringkasan dokumen, dan pendekatan ansambel. Penelitian ini mencapai akurasi 80% untuk klasifikasi luas (15 kategori) dan 60% untuk klasifikasi berbutir halus (279 kategori), menandai peningkatan 8% dan 28% dibandingkan hasil canggih sebelumnya. Berbagai model BERT, termasuk LegalBERT, digunakan dalam percobaan. |
| 12. | (Darji dkk., 2023) | *German BERT Model for Legal Named Entity Recognition* | Jerman BERT | Penelitian ini menyajikan model Jerman BERT yang disesuaikan khusus untuk *Named Entity Recognition* (NER). Ini mengatasi kurangnya model berbasis transformator untuk domain hukum dalam bahasa Jerman dengan pelatihan tentang kumpulan data Pengakuan Badan Hukum. Model ini disempurnakan pada GPU Nvidia GeForce |

| | | | | |
|---|---|---|---|---|
| | | | | RTX selama 7 zaman, mencapai kinerja yang unggul dibandingkan dengan model BiLSTM-CRF+. Penelitian telah membuat model tersedia untuk umum melalui *Hugging Face*, mendukung kerangka kerja *PyTorch* dan *TensorFlow*. |
| 13. | (Clavié & Alphonsus, 2021) | *The Unreasonable Effectiveness of the Baseline: Discussing SVMs in Legal Text Classification* | LegalBERT dan Caselaw-BERT | Penelitian ini membahas kinerja yang kuat dari pengklasifikasi *Support Vector Machine* (SVM) dalam klasifikasi teks hukum, menunjukkan bahwa mereka mencapai hasil yang kompetitif dibandingkan dengan model pembelajaran mendalam seperti BERT. Ini menyoroti bahwa keuntungan kinerja dari pendekatan berbasis BERT secara signifikan lebih kecil di domain hukum daripada dalam tugas umum. Penelitian ini mengusulkan tiga hipotesis untuk menjelaskan temuan ini, menekankan perlunya eksplorasi lebih lanjut dari model BERT dan integrasi basis pengetahuan hukum untuk meningkatkan kinerja dalam tugas NLP hukum. |

| | | | | |
|---|---|---|---|---|
| 14. | (Shah dkk., 2023) | *An Automated Text Document Classification Framework using BERT* | BERT | Penelitian ini menyajikan kerangka klasifikasi teks otomatis menggunakan BERT, arsitektur *deep learning*. Ini memproses data teks dengan menghapus *stop-words* dan karakter khusus, meningkatkan kinerja klasifikasi. Kerangka kerja dilatih pada dataset email UCI dan dataset BBC News, mencapai akurasi maksimum 91,4%. Efektivitas dievaluasi menggunakan metrik seperti akurasi, *precision*, dan *recall*, menunjukkan ketahanan dan penerapannya untuk organisasi yang membutuhkan solusi klasifikasi teks yang efisien. |
| 15. | (AL-Qurishi dkk., 2022) | *AraLegal-BERT: A pretrained language model for Arabic Legal text* | AraLegal-BERT | Penelitian ini menggunakan AraLegal-BERT sebagai model berbasis transformator *encoder* dua arah yang secara khusus dilatih sebelumnya untuk teks hukum Arab. Ini dikembangkan untuk meningkatkan aplikasi pemrosesan bahasa alami dalam domain hukum, mencapai akurasi yang unggul dalam tugas-tugas seperti klasifikasi teks hukum, *Named Entity Recognition*, dan ekstraksi kata kunci dibandingkan |

| | | | | |
|---|---|---|---|---|
| | | | | dengan model BERT standar. Dilatih pada dataset sekitar 4,5 GB, AraLegal-BERT menunjukkan peningkatan kinerja yang signifikan, menjadikannya alat yang berharga untuk menganalisis dokumen hukum dan yurisprudensi Arab. |

Penggunaan BERT dan model berbasis *transformers* lainnya untuk pengolahan teks hukum menunjukkan hasil yang sangat menjanjikan, terbukti dengan sejumlah penelitian yang berhasil mengoptimalkan kinerja model pada tugas klasifikasi teks hukum seperti yang tercantum dalam Tabel 2.2. Berdasarkan *state of the art* yang ada, penelitian ini mengusung ide penelitian optimasi model *fine-tuned* DistilBERT untuk klasifikasi status dan jenis peraturan perundang-undangan. Tabel 2.3 memperlihatkan perbandingan capaian yang dapat diharapkan dari penelitian ini dengan capaian yang telah diraih oleh penelitian sebelumnya. Perbandingan tersebut mencakup beberapa aspek utama, antara lain metodologi yang digunakan, hasil yang diperoleh, serta kontribusi penelitian terhadap pengembangan ilmu pengetahuan. Selain itu, penelitian ini juga membandingkan keterbatasan yang dihadapi pada penelitian sebelumnya, serta mengidentifikasi peluang-peluang baru yang dapat dijadikan fokus untuk penelitian lebih lanjut. Capaian yang diharapkan dari penelitian ini dapat memberikan pemahaman yang lebih mendalam dan solusi yang lebih efektif dibandingkan dengan penelitian terdahulu.

Relevansi penelitian ini terkait dengan penelitian terdahulu terletak pada penerapan model berbasis *transformers*, khususnya DistilBERT, dalam konteks klasifikasi teks hukum. Sebelumnya, penelitian-penelitian yang ada sudah berhasil menunjukkan potensi besar BERT dan model *transformer* lainnya dalam memproses dan memahami teks hukum, namun kebanyakan terbatas pada pengolahan bahasa alami secara umum tanpa fokus mendalam pada peraturan perundang-undangan. Penelitian ini berupaya untuk mengisi celah tersebut dengan mengoptimalkan model *fine-tuned* DistilBERT untuk tugas yang lebih spesifik,

yaitu klasifikasi peraturan perundang-undangan. Dengan demikian, penelitian ini tidak hanya meneruskan tren yang ada, tetapi juga menawarkan inovasi dalam meningkatkan akurasi dan relevansi klasifikasi peraturan hukum, serta memberikan sumbangan pada pengembangan aplikasi teknologi dalam sektor hukum. Dengan fokus pada peraturan perundang-undangan, penelitian ini berpotensi memperkaya penelitian sebelumnya dengan memperkenalkan pendekatan yang lebih terfokus dan kontekstual dalam pengolahan teks hukum.

Tabel 2.3 Metrik Penelitian

| No. | Penulis, Tahun Terbit | Judul | Pendekatan Model | | | Strategi Pelatihan | | Parameter Uji | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ***Traditional Machine*** | ***Deep Learning*** | ***Transformer*** | ***Pre-Training*** | ***Fine-Tuning*** | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1-Score*** |
| 1. | (Wehnert dkk., 2022) | *Applying BERT Embeddings to Predict Legal Textual Entailment* | - | - | ✓ | ✓ | ✓ | ✓ | - | - | ✓ |
| 2. | (Hegel dkk., 2021) | *The Law of Large Documents: Understanding the* | - | - | ✓ | ✓ | ✓ | ✓ | - | ✓ | - |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *Structure of Legal Contracts Using Visual Cues* | | | | | | | | | |
| 3. | (Chaudhary dkk., 2020) | *TopicBERT for Energy Efficient Document Classification* | - | ✓ | - | ✓ | ✓ | ✓ | - | - | ✓ |
| 4. | (Fragkogiannis dkk., 2023) | *Context-Aware Classification of Legal Document Pages* | - | - | ✓ | ✓ | ✓ | - | - | - | ✓ |
| 5. | (Lu dkk., 2021) | *A Sentence-level Hierarchical BERT Model for Document Classification with Limited Labelled Data* | - | - | ✓ | ✓ | ✓ | ✓ | - | - | ✓ |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 6. | (B. Li & Wang, 2023) | *Design of intelligent legal text analysis and information retrieval system based on BERT model* | - | - | ✓ | ✓ | ✓ | ✓ | - | - | ✓ |
| 7. | (Si & Roberts, 2021) | *Hierarchical Transformer Networks for Longitudinal Clinical Document Classification.* | - | - | ✓ | ✓ | ✓ | ✓ | - | ✓ | - |
| 8. | (Aumiller dkk., 2021) | *Structural Text Segmentation of Legal Documents* | - | - | ✓ | ✓ | ✓ | ✓ | - | - | - |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 9. | (AL-Qurishi, 2023) | *Leveraging BERT Language Model for Arabic Long Document Classification* | - | - | ✓ | ✓ | ✓ | ✓ | - | - | ✓ |
| 10. | (Mamakas dkk., 2022) | *Processing Long Legal Documents with Pre-trained Transformers: Modding LegalBERT and Longformer* | - | - | ✓ | ✓ | ✓ | ✓ | - | - | - |
| 11. | (Vatsal dkk., 2023) | *Classification of US Supreme Court Cases using BERT-Based Techniques* | - | - | ✓ | ✓ | ✓ | ✓ | - | - | - |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 12. | (Darji dkk., 2023) | *German BERT Model for Legal Named Entity Recognition* | - | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ |
| 13. | (Clavié & Alphonsus, 2021) | *The Unreasonable Effectiveness of the Baseline: Discussing SVMs in Legal Text Classification* | ✓ | - | - | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 14. | (Shah dkk., 2023) | *An Automated Text Document Classification Framework using BERT* | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| 15. | (AL-Qurishi dkk., 2022) | *AraLegal-BERT: A pretrained language* | - | - | ✓ | ✓ | ✓ | ✓ | - | - | - |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *model for Arabic Legal text* | | | | | | | | | |
| 16. | Penelitian Ini | Optimasi Model *Fine-Tuned* DistilBERT untuk Klasifikasi Status dan Jenis Peraturan Perundang-Undangan | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Relevansi penelitian ini terletak pada kemampuan untuk mengisi celah yang ada pada penelitian-penelitian sebelumnya. Tabel 2.3 menunjukkan bahwa meskipun banyak penelitian telah berhasil mengembangkan model untuk klasifikasi dokumen hukum menggunakan BERT atau varian lainnya, tidak ada penelitian yang secara menyeluruh mengoptimasi parameter model untuk meningkatkan performa secara signifikan. Penelitian ini berfokus pada penerapan optimasi parameter, yang dapat membawa kemajuan besar dalam kinerja model, suatu pendekatan yang belum banyak diterapkan pada penelitian sebelumnya.

Meskipun model berbasis DistilBERT telah terbukti efisien di berbagai domain, penerapannya dalam klasifikasi dokumen hukum masih minim. Oleh karena itu, penelitian ini menawarkan kebaruan dengan mengintegrasikan DistilBERT dalam dokumen hukum, yang berpotensi memperbaiki akurasi dan efisiensi klasifikasi.