

RINGKASAN TEKS BAHASA MELAYU MENGGUNAKAN PEMBELAJARAN MENDALAM

AMIRA SYAFIKA BINTI ABD RAHIM

PROF. MADYA DR. NAZLIA BINTI OMAR

*Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia, 43600 UKM Bangi,
Selangor Darul Ehsan, Malaysia*

ABSTRAK

Ringkasan teks dalam Bahasa Melayu menggunakan pembelajaran mendalam merupakan teknologi yang canggih di mana ia dapat melatih model komputer untuk memahami dan meringkaskan teks Bahasa Melayu yang lebih panjang. Apabila sistem ini diberikan teks Bahasa Melayu yang panjang, ia akan menggunakan pengetahuan yang telah dipelajari untuk menghasilkan ringkasan teks yang lebih singkat dengan memuatkan idea-idea utama dan maklumat penting dari teks asal. Dalam konteks Bahasa Melayu, kaedah ringkasan teks yang sedia ada sering gagal memberikan ringkasan yang berkualiti tinggi dan menyeluruh. Tambahan pula, masalah antara muka yang kurang mesra pengguna serta kekurangan pilihan untuk mengawal panjang ringkasan mengurangkan kegunaan dan aksesibiliti alat-alat ini. Matlamat utama projek ini adalah untuk membangunkan sistem yang kukuh dengan menggunakan teknik pembelajaran mendalam untuk menjalankan ringkasan teks yang bersifat abstraktif. Kaedah abstraktif membolehkan alat ini menghasilkan ringkasan yang padat dan jeleket yang merangkumi inti kandungan teks asal. Selain meningkatkan kualiti ringkasan, alat ini akan dilengkapi antara muka yang mesra pengguna serta kelebihan untuk memilih Panjang ringkasan yang membolehkan pengguna menyesuaikan panjang ringkasan yang dihasilkan mengikut keperluan mereka. Untuk menghasilkan ringkasan teks Bahasa Melayu menggunakan pembelajaran mendalam, langkah-langkah yang dilakukan adalah mengumpulkan dataset teks yang sesuai, praolah teks, pilih model pembelajaran mendalam, latih model dengan data tersebut, nilai prestasi model, dan selaraskan jika perlu untuk menghasilkan ringkasan teks Bahasa Melayu yang berkualiti. Hasil daripada projek ini adalah penyelesaian inovatif dan efektif untuk ringkasan teks dalam Bahasa Melayu yang bersifat abstraktif. Sistem ini membolehkan pengguna untuk mendapatkan ringkasan berkualiti yang dapat disesuaikan dengan pilihan mereka. Kesimpulannya, projek ini diharapkan dapat memberikan manfaat yang signifikan kepada masyarakat dengan menyediakan alat yang canggih untuk merangkum teks dalam Bahasa Melayu, serta memajukan bidang pemrosesan bahasa semulajadi untuk bahasa ini.

Kata kunci: Ringkasan Teks, Pembelajaran mendalam, Abstraktif, Pemrosesan bahasa semulajadi

PENGENALAN

Dalam era maklumat yang penuh dengan teks panjang dan kompleks ini, kemahiran untuk menyederhanakan dan meringkaskan maklumat menjadi format yang lebih ringkas dan teratur merupakan perkara yang sangat penting. Ringkasan teks merupakan satu bidang penting dalam pemprosesan bahasa tabii (PBT), yang menggunakan teknologi komputer untuk mengautomatiskan proses pengekstrakan maklumat penting dari teks panjang. Kaedah ringkasan teks merupakan kemahiran yang penting dalam membantu kita menghadapi pelbagai jenis bahan bacaan, termasuk kertas penyelidikan panjang, rencana berita, dan kandungan dalam talian. Selain itu, ia juga turut membantu untuk mengekstrak maklumat penting, menyusunnya dalam bentuk yang lebih mudah difahami, dan membolehkan kita mendapatkan gambaran keseluruhan tanpa perlu membaca keseluruhan teks.

Terdapat dua jenis ringkasan teks yang boleh kita gunakan iaitu ringkasan ekstraktif dan ringkasan abstraktif. Ringkasan ekstraktif bertujuan untuk mencipta ringkasan dengan mengenal pasti ayat-ayat penting dari teks asal dan mengekstraknya daripada teks tanpa mengubah apa-apa. Dengan kata lain, ia mengenal pasti dan menyalin kandungan yang sedia ada dari dokumen asal untuk membentuk ringkasan. Manakala ringkasan abstraktif pula bertujuan untuk mencipta ringkasan dengan menghasilkan ayat-ayat baru daripada teks asal. Pendekatan ini tidak bergantung pada menyalin ayat-ayat secara harfiah tetapi sebaliknya mencipta ayat-ayat baru yang menyampaikan maklumat yang sama.

Permulaan ringkasan teks automatik bermula apabila Luhn(1958) memperkenalkan sistem pionir. Pendekatan Luhn adalah berdasarkan analisis kekerapan frasa. Pada tahun 1969, sistem ringkasan teks automatik muncul dengan menggunakan kaedah kata kunci konvensional yang merangkumi penilaian penting berdasarkan kekerapan, kaedah petunjuk, kaedah tajuk, dan kaedah lokasi. Sistem ini penting dalam menetapkan penilaian keutamaan ayat untuk mengeluarkan kandungan penting. Pada tahun 1995, kemunculan *Trainable Document Summarizer* merupakan langkah penting dalam pengembangan teknik ringkasan teks, di mana ia mampu menggabungkan berbagai kaedah penilaian untuk menghasilkan ayat ringkasan.

Selama bertahun-tahun, banyak usaha telah dilakukan untuk memperbaiki teknik ringkasan teks. Walau bagaimanapun untuk mencapai hasil yang optimum terutamanya dalam konteks ringkasan teks dalam Bahasa Melayu masih merupakan cabaran yang berterusan. Bahasa Melayu yang kompleks, dengan struktur linguistiknya yang unik, ungkapan idiomatik, dan nuansa budaya, mencipta cabaran tersendiri yang memerlukan pemahaman yang mendalam. Perjalanan mencapai keunggulan dalam bidang ini masih berterusan. Para penyelidik terus bekerja dengan tekun untuk mencapai matlamat meningkatkan ringkasan teks, terutamanya dalam konteks Bahasa Melayu. Cabaran yang dihadapi sepanjang perjalanan ini telah menjadi pendorong yang mendorong perkembangan berterusan dalam bidang ini. Objektif utama adalah untuk meningkatkan ringkasan teks, terutamanya untuk kandungan dalam Bahasa Melayu, dengan matlamat menyampaikan maklumat yang boleh diakses dan difahami oleh khalayak yang lebih luas.

METODOLOGI KAJIAN

Metodologi ini akan dibahagikan kepada lima peringkat utama yang akan dijalankan dalam kajian ini. Kajian ini melibatkan lima fasa utama, iaitu Fasa Permulaan Kajian, Fasa Penyediaan Data, Fasa Peringkasan Teks, Fasa Pengujian dan Fasa Pembangunan Prototaip. Melalui metodologi ini, ia dapat membantu untuk memastikan bahawa setiap perkara dan kajian yang dijalankan adalah sistematik, berfokus, dan berdasarkan prinsip-prinsip yang kukuh. Melalui perjalanan lima fasa ini, pembangunan sistem ringkasan teks Bahasa Melayu menggunakan pembelajaran mendalam dapat dijalankan secara sistematik, memastikan setiap langkah membawa sumbangan yang signifikan untuk keseluruhan proses pengembangan sistem.

Fasa Permulaan Kajian

Fasa permulaan kajian ini melibatkan beberapa perkara penting yang memberikan landasan kukuh bagi pembangunan sistem ringkasan teks Bahasa Melayu menggunakan pembelajaran mendalam. Pertama, kajian literatur merupakan langkah awal yang penting untuk menyelidiki literatur terdahulu, teknologi terkini, dan metodologi yang berkaitan dengan ringkasan teks dalam konteks Bahasa Melayu. Melalui kajian literatur, peneliti dapat memahami perkembangan terkini dalam bidang ini, mencari inspirasi dari penemuan terdahulu, dan mengidentifikasi kekosongan pengetahuan yang boleh diisi melalui penyelidikan ini.

Selain itu, pengenalan masalah adalah langkah yang penting dalam fasa ini. Ini melibatkan penentuan isu-isu atau cabaran utama yang perlu diselesaikan dalam pembangunan sistem ringkasan teks. Dengan mengenal pasti masalah-masalah ini, peneliti dapat menyusun strategi yang sesuai untuk memastikan bahawa sistem yang dibangunkan dapat menangani permasalahan tersebut dengan efektif.

Seterusnya, pengenalan objektif kajian pula membantu dalam memberikan arah yang jelas kepada penyelidikan. Objektif kajian perlu ditetapkan dengan tepat, menyenaraikan apa yang ingin dicapai dalam penyelidikan ini. Contohnya, objektif mungkin termasuk pemahaman mendalam tentang struktur kalimat dalam Bahasa Melayu, pengenalan ciri-ciri keunikan bahasa, atau penentuan kesesuaian metrik penilaian untuk menilai keberkesanan ringkasan dalam konteks Bahasa Melayu.

Terakhir, perancangan projek adalah aspek penting yang melibatkan penjadualan dan pengurusan keseluruhan penyelidikan. Ini termasuk menentukan langkah-langkah praktikal dan merancang sumber daya yang diperlukan untuk mencapai objektif kajian. Perancangan projek membantu dalam memastikan kelancaran dan kecekapan penyelidikan dari segi organisasi dan pelaksanaan. Melalui kesemua perkara yang dilaksanakan dalam fasa permulaan kajian, penyelidik dapat memastikan bahawa asas yang kukuh dan terperinci telah diletakkan sebelum memasuki fasa-fasa seterusnya dalam pembangunan sistem ringkasan teks.

Fasa Penyediaan Data

Terdapat dua komponen yang penting dalam fasa penyediaan data ini iaitu pengumpulan data dan pra-pemrosesan data. Pengumpulan data melibatkan pemilihan dan pengumpulan sumber data yang relevan. Selanjutnya, pra-pemrosesan data melibatkan langkah-langkah untuk mempersiapkan data sebelum digunakan untuk melatih model. Pra-pemrosesan data bertujuan untuk menyediakan data yang konsisten, relevan, dan siap untuk digunakan oleh model pembelajaran mendalam.

a. Sumber Data

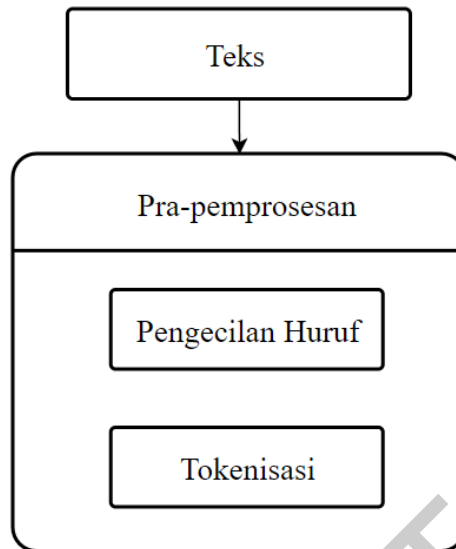
Dalam pembangunan model peringkasan teks, pemilihan dataset yang sesuai adalah kritikal untuk memastikan keberkesanan dan ketepatan model. Salah satu dataset yang digunakan dalam kajian ini adalah Peringkasan Teks Berita *CNN/DailyMail*. Dataset *CNN/DailyMail* adalah dataset dalam Bahasa Inggeris yang mengandungi lebih daripada 300,000 artikel berita unik yang ditulis oleh wartawan di *CNN* dan *Daily Mail*. Versi semasa dataset ini menyokong kedua-dua peringkasan ekstraktif dan abstraktif, walaupun versi asalnya dicipta untuk pembacaan mesin dan pemahaman serta menjawab soalan secara abstraktif.

Dataset ini mengandungi beberapa medan data penting seperti id, artikel, dan sorotan. Medan id adalah rentetan yang mengandungi *SHAI hash* dalam format heksadesimal dari URL di mana cerita tersebut diperoleh. Medan artikel adalah rentetan yang mengandungi badan artikel berita, sementara medan sorotan mengandungi sorotan artikel seperti yang ditulis oleh penulis artikel.

Dataset *CNN/DailyMail* dibahagikan kepada tiga bahagian: latihan (*train*), validasi (*validation*), dan ujian (*test*). Bahagian latihan mengandungi 287,113 contoh, bahagian validasi mengandungi 13,368 contoh, dan bahagian ujian mengandungi 11,490 contoh. Dalam kajian ini, dataset *CNN/DailyMail* digunakan sebagai sumber utama untuk melatih model *PEGASUS*. Model ini dilatih untuk menghasilkan ringkasan abstraktif, di mana ia bukan sahaja mengekstrak bahagian penting dari artikel, tetapi juga menulis semula teks untuk memberikan ringkasan yang lebih kohesif dan bermakna. Dengan menggunakan dataset yang besar dan pelbagai ini, model dapat belajar untuk menangkap pelbagai corak dan konteks dalam teks, yang penting untuk menghasilkan ringkasan berkualiti tinggi. Pembahagian dataset kepada bahagian latihan, validasi, dan ujian membolehkan penilaian yang komprehensif terhadap prestasi model.

b. Pra-pemrosesan Data

Rajah 1 memberikan ilustrasi mengenai proses pra-pemrosesan yang dicadangkan untuk penelitian ini. Rajah tersebut menjelaskan langkah-langkah yang harus diikuti oleh data latihan dan data ujian.



Rajah 1 Proses Pra-pemprosesan

Sebelum teks daripada set data diproses menggunakan pemprosesan Bahasa Tabii, teks Bahasa Melayu ini akan melalui fasa pra-pemprosesan terlebih dahulu. Pertama sekali, pengecilan huruf dalam teks akan berlaku dimana ia akan mengubah teks kepada huruf kecil agar lebih mudah diproses.

Seterusnya, tokenisasi akan dilakukan dimana ia akan memecah teks kepada unit-unit kecil yang dikenali sebagai token. Token boleh menjadi perkataan, frasa, atau karakter, bergantung kepada tahap yang diinginkan. Sebagai contoh, ayat "Saya suka makan nasi goreng" boleh dipecah kepada token seperti ["Saya", "suka", "makan", "nasi", "goreng"]. Secara keseluruhannya, langkah-langkah ini membantu dalam memproses dan memahami teks secara lebih efektif dalam pelbagai tugas pemprosesan bahasa semula jadi. Jadual 1 berikut merupakan contoh bagi proses yang dibincangkan di atas:

Jadual 1 Langkah Pra-Pemprosesan Teks

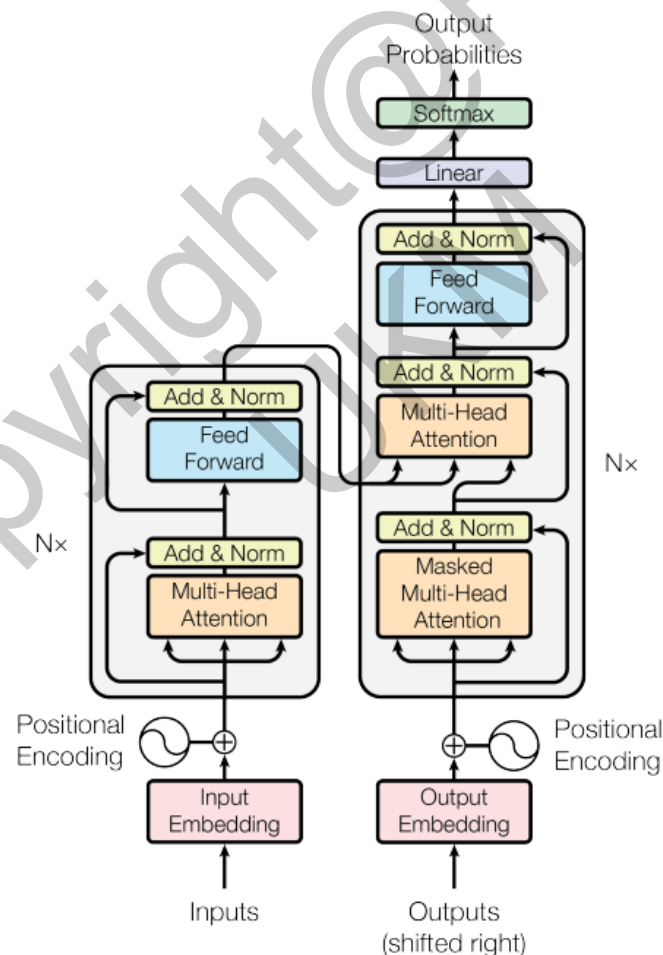
Teks Asli	Langkah Pra-Pemprosesan Teks
Teks Setelah Pengecilan Huruf	"Beberapa anak-anak sedang bermain di taman pada hari yang cerah. Mereka tertawa dan senang bermain bola."
Teks Setelah Tokenisasi	["beberapa", "anak-anak", "sedang", "bermain", "di", "taman", "pada", "hari", "yang", "cerah", "mereka", "tertawa", "dan", "senang", "bermain", "bola"]

Fasa Peringkasan Teks

Fasa peringkasan teks ini melibatkan penggunaan model-model pembelajaran mendalam yang canggih seperti BART, PEGASUS, dan T5. Model-model ini menggunakan seni bina transformer yang telah terbukti berkesan dalam tugas pemprosesan bahasa semula jadi, termasuk ringkasan teks. Setiap model ini mempunyai keunikan dan kelebihan tersendiri dalam menangani tugas ringkasan teks, yang akan diterangkan dengan lebih terperinci dalam subseksyen berikut.

a. Model *BART*

Model *BART* (Transformer Bipermulaan dan Autoregresif) memegang peranan penting dalam dunia pembelajaran mesin, khususnya dalam tugas pemprosesan bahasa semula jadi. Model ini terkenal kerana seni bina yang canggih, yang terdiri daripada beberapa komponen utama dan langkah-langkah yang teratur. Rajah 2 berikut merupakan seni bina model transformer yang digunakan oleh *BART* iaitu jenis rangkaian neural yang digunakan dalam pemprosesan bahasa semula jadi.



Rajah 2 Seni Bina Model *BART*

Rajah 2 yang dipaparkan di atas menggambarkan seni bina model *Transformer* yang digunakan oleh model *BART*, satu jenis model pembelajaran mendalam yang amat berjaya

dalam pelbagai tugas pemprosesan bahasa semula jadi, termasuk ringkasan teks. Model *Transformer* terkenal kerana keupayaannya mengendalikan data berurutan tanpa memerlukan rangkaian neural berulang.

Proses bermula dengan teks input yang perlu diringkaskan. Teks ini pertama kali diubah kepada bentuk nombor melalui proses yang dipanggil tokenisasi dan penyiratan. Setiap perkataan atau sub-perkataan dalam teks dipetakan kepada vektor unik dalam ruang dimensi tinggi iaitu penyiratan input (*input embedding*). Kedudukan setiap perkataan dalam jujukan juga penting, jadi pengekodan kedudukan (*positional encoding*) ditambahkan kepada penyiratan input untuk mengekalkan urutan perkataan.

Langkah seterusnya melibatkan komponen teras *Transformer*, mekanisme perhatian pelbagai kepala (*multi-head attention*). Dalam konteks ringkasan, mekanisme ini membolehkan model memberi tumpuan kepada bahagian yang berbeza dalam teks input semasa meramalkan setiap perkataan ringkasan. Ini dicapai dengan mengira skor perhatian yang menentukan sejauh mana tumpuan diberikan kepada setiap input semasa menghasilkan setiap perkataan output.

Setiap blok perhatian pelbagai kepala (*multi-head attention*), diikuti dengan lapisan tambah & norma (*add & norm*) yang mengaplikasikan sambungan residual dan penormalan lapisan untuk memudahkan latihan dan menstabilkan proses pembelajaran. Output daripada gabungan ini kemudian melalui rangkaian neural suapan ke depan (*feed forward*) yang lebih memproses maklumat tersebut. Rangkaian suapan ke depan (*feed forward*) ini terdiri daripada lapisan sepenuhnya berhubung yang mengaplikasikan transformasi bukan linear ke data. " Nx " merujuk kepada hakikat bahawa lapisan-lapisan ini iaitu perhatian pelbagai kepala (*multi-head attention*), tambah & norma (*add & norm*), dan suapan ke depan (*feed forward*) diulangi sebanyak N kali dalam satu tumpukan. Pengulangan ini membolehkan model mempelajari corak dan hubungan yang kompleks dalam data.

Selepas pengulangan terakhir lapisan-lapisan ini dalam pengekod (bahagian kiri gambar), maklumat tersebut diteruskan kepada penyahkod (bahagian kanan), yang mempunyai struktur yang serupa tetapi melibatkan lapisan perhatian pelbagai kepala yang ditutupi (*masked multi-head attention*) tambahan. Lapisan ini memastikan bahawa ramalan untuk perkataan tertentu hanya boleh bergantung kepada perkataan yang dihasilkan sebelumnya, menghalang model daripada "menipu" dengan melihat perkataan masa depan dalam output.

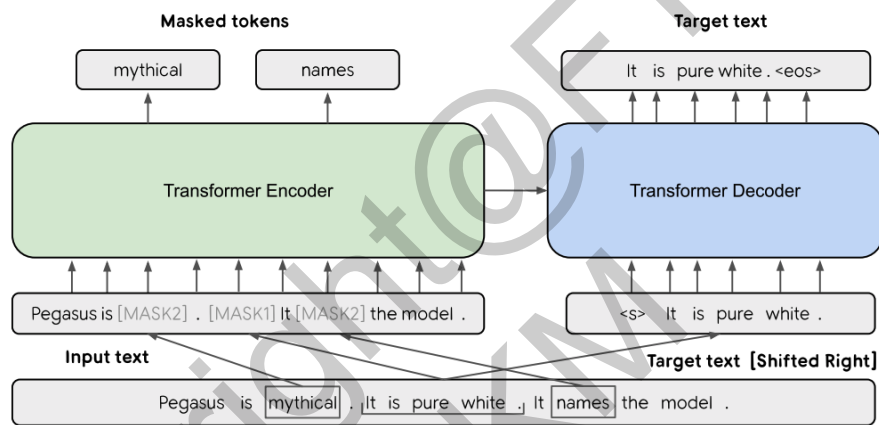
Output penyiratan kemudian dilalui melalui lapisan linear, yang menyaksikan representasi berdimensi tinggi ke saiz kosa kata. Lapisan *softmax* kemudian mengubahnya kepada kebarangkalian, yang mewakili ramalan model untuk perkataan seterusnya dalam ringkasan.

Secara ringkasnya, teks input disiratkan dan dikodkan, diproses melalui pelbagai lapisan perhatian dan rangkaian suapan ke depan (*feed-forward*) dan kemudian dikodkan

menjadi output ringkasan, dengan setiap komponen memainkan peranan tertentu dalam menangkap nuansa bahasa dan hakikat teks asal. Keupayaan Transformer mengendalikan dependensi jarak jauh dan kebolehpnyelariannya menjadikan ia sangat berkesan untuk tugas ringkasan teks.

b. Model PEGASUS

Model *PEGASUS* adalah model pembelajaran mendalam yang dibangunkan oleh *Google Research* untuk tugas-tugas penjanaaan teks, terutamanya ringkasan teks secara abstraktif. *PEGASUS* menggunakan seni bina transformer, yang terdiri daripada dua komponen utama: *Transformer Encoder* dan *Transformer Decoder*. *PEGASUS* dilatih dengan teknik yang unik yang dikenali sebagai "Gap Sentence Generation" (*GSG*), di mana beberapa ayat dalam dokumen asal dipilih dan dihapuskan, kemudian model dilatih untuk meramalkan ayat-ayat yang dipadamkan tersebut berdasarkan konteks yang tinggal. Rajah 3 di bawah merupakan seni bina model *PEGASUS* yang melibatkan beberapa komponen utama.



Rajah 3 Seni Bina Model PEGASUS

Pertama, penyediaan input dan pengkodan dilakukan dengan memasukkan teks input yang mengandungi ayat-ayat yang telah dipadamkan secara strategik, dikenali sebagai masked tokens. Sebagai contoh, dalam ayat "*PEGASUS is mythical. It is pure white. It names the model.*" beberapa kata telah diganti dengan token *[MASK1]* dan *[MASK2]*. Pengekod (*Transformer Encoder*) menerima teks ini dan bertanggungjawab untuk memahami konteks keseluruhan teks serta meramalkan token yang telah dipadamkan. Proses pengkodan melibatkan penyandian setiap token dalam teks input menjadi representasi vektor berdimensi tinggi yang menangkap makna semantik dan hubungan konteks antara token.

Dalam proses pengkodan, *Transformer Encoder* menerima teks yang mengandungi token yang dipadamkan dan menyandikan keseluruhan teks ke dalam representasi vektor. Representasi ini merangkum konteks penuh dari teks input, yang membantu model memahami hubungan antara token yang ada dan token yang dipadamkan. Teknik *self-attention* digunakan dalam pengkod (*encoder*) untuk membolehkan setiap token memperhatikan semua token lain dalam teks, membantu memahami konteks yang lebih luas dan menangkap hubungan yang kompleks antara token.

Selepas proses pengekodan, representasi konteks yang dihasilkan oleh pengekod (*encoder*) dihantar ke *Transformer Decoder*. Penyahkod (*decoder*) menggunakan maklumat ini untuk menghasilkan teks sasaran satu token pada satu masa. Sebagai contoh, jika teks sasaran adalah "*It is pure white*" *Decoder* memulakan proses dengan token khas "<s>" yang menandakan permulaan urutan. Berdasarkan maklumat dari pengekod (*encoder*), penyahkod (*decoder*) menghasilkan token pertama "*It*" kemudian token kedua "*is*" dan seterusnya sehingga seluruh ayat dihasilkan dan token akhir "<eos>" dicapai. Teknik perhatian (*attention*) digunakan antara pengekod (*encoder*) dan penyahkod (*decoder*), membolehkan penyahkod (*decoder*) memberi perhatian khusus pada bahagian tertentu dari teks input semasa menghasilkan setiap token.

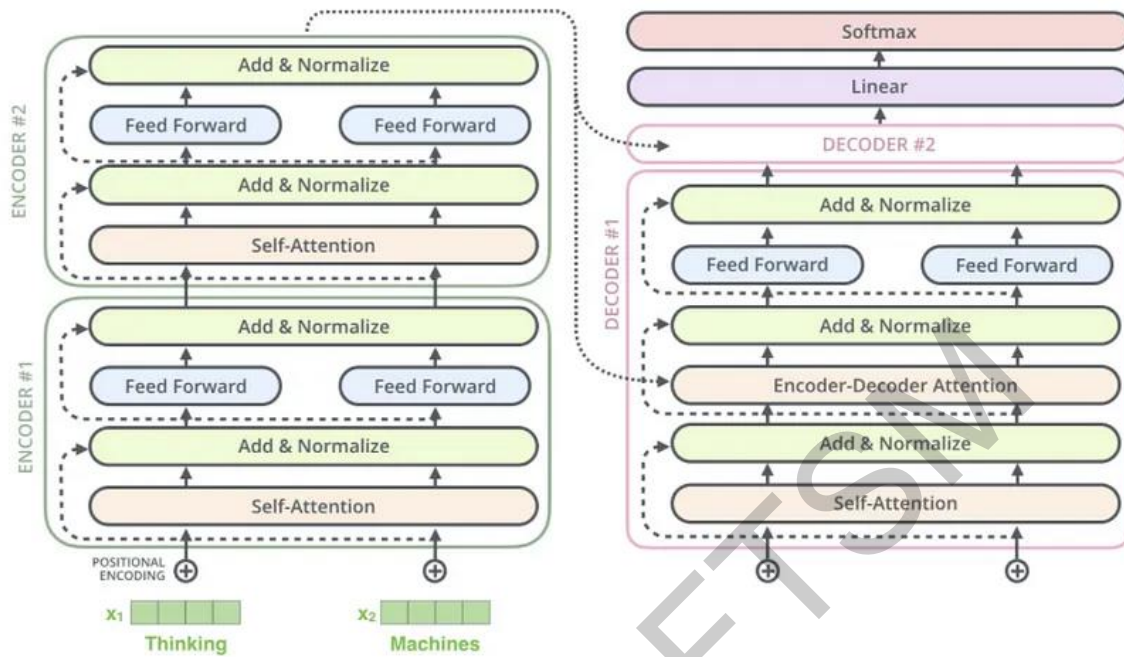
PEGASUS dilatih dengan teknik *Gap Sentence Generation (GSG)*, di mana beberapa ayat dalam dokumen asal dipilih dan dipadamkan, kemudian model dilatih untuk meramalkan ayat-ayat yang dipadamkan tersebut. Semasa latihan, model melihat konteks yang tinggal dan cuba menghasilkan ayat yang telah dipadamkan seolah-olah ia adalah ringkasan dari teks yang tinggal. Ini membolehkan model belajar untuk memahami makna dan hubungan dalam teks secara mendalam dan menghasilkan ringkasan yang tepat dan bermakna.

Keunikan dan kehebatan *PEGASUS* terletak pada penggunaan teknik yang inovatif dalam pemadaman dan penghasilan ayat untuk melatih modelnya. Berbeza dengan pendekatan tradisional yang mungkin hanya mempelajari perwakilan token individu tanpa memahami konteks yang lebih luas, pendekatan *GSG* membolehkan *PEGASUS* memahami dan meramalkan konteks yang lebih kompleks, menjadikannya sangat efektif untuk tugas ringkasan teks. Keupayaan ini membolehkan *PEGASUS* menghasilkan ringkasan yang bukan sahaja tepat dari segi fakta tetapi juga alih bahasa, memberikan ringkasan yang lebih bermakna dan mudah difahami.

Kesimpulannya, *PEGASUS* adalah model transformer yang sangat canggih yang direka khas untuk tugas-tugas ringkasan teks secara abstraktif. Dengan teknik latihan *Gap Sentence Generation*, ia mampu menangkap hubungan konteks yang mendalam dalam teks dan menghasilkan ringkasan yang padat dan bermakna. Ini menjadikannya alat yang sangat berguna dalam pemrosesan bahasa semulajadi, terutamanya untuk aplikasi seperti pembuatan ringkasan dokumen, pembuatan kandungan, dan lain-lain.

c. Model T5

Model *T5 (Text-to-Text Transfer Transformer)* adalah salah satu model transformer yang sangat canggih dan serba guna, direka oleh *Google Research* untuk menangani pelbagai tugas pemrosesan bahasa semula jadi (NLP) dalam format teks-ke-teks. Seni bina *T5* adalah berdasarkan seni bina transformer standard yang terdiri daripada beberapa lapisan pengekod (*encoder*) dan penyahkod (*decoder*). Rajah 4 berikut adalah penjelasan terperinci tentang seni bina model *T5* seperti yang ditunjukkan dalam gambar:



Rajah 4 Seni Bina Model T5

Pertama, sebelum memasuki pengekod pertama, teks input diberikan melalui penyandian kedudukan (*positional encoding*) yang membantu model memahami urutan token dalam teks kerana transformer tidak mempunyai kesedaran urutan secara bawaan. Penyandian ini ditambahkan kepada token input (x_1, x_2, \dots). Kemudian, pengekod (*encoder*) menerima teks ini dan memulakan proses dengan lapisan perhatian sendiri (*self-attention*), di mana setiap token dalam input boleh memperhatikan semua token lain untuk membantu model memahami konteks global teks. Selepas perhatian sendiri, hasilnya ditambahkan kembali ke input asal (*skip connection*) dan kemudian dinormalisasi untuk membantu stabiliti dan kecekapan pelatihan. Data kemudian melalui lapisan *feed forward* yang menggunakan dua lapisan sepenuhnya terhubung dengan aktivasi bukan linear di antara mereka, dan sekali lagi melalui proses tambah dan normalisasi sebelum diteruskan ke lapisan seterusnya. Struktur ini diulang untuk beberapa lapisan pengekod, membolehkan model menangkap maklumat kontekstual yang lebih dalam dan kompleks.

Setelah pengkodan selesai, representasi konteks yang dihasilkan oleh pengkod dihantar ke penyahkod (*decoder*). Penyahkod juga bermula dengan lapisan perhatian sendiri, tetapi perhatian sendiri di penyahkod hanya boleh memperhatikan token sebelumnya (*causal or masked self-attention*) untuk mengekalkan kausalitas dalam generasi teks. Hasil dari perhatian sendiri ditambahkan ke input asal dan dinormalisasi. Kemudian, lapisan perhatian pengkod-penyahkod membolehkan penyahkod memperhatikan output dari semua lapisan pengkod, membantu penyahkod menggunakan konteks yang telah difahami oleh pengkod. Selepas itu, hasil dari perhatian pengkod-penyahkod juga ditambahkan ke input asal dan dinormalisasi, diikuti oleh lapisan *feed forward* yang diikuti oleh proses tambah dan normalisasi. Struktur ini diulang untuk beberapa lapisan penyahkod, membolehkan model menghasilkan teks yang lebih baik berdasarkan konteks penuh dari pengkod. Selepas melalui

semua lapisan penyahkod, hasil akhir dari penyahkod diteruskan ke lapisan linear dan kemudian ke lapisan *softmax* yang menghasilkan kebarangkalian untuk setiap token dalam vokabulari, menentukan token yang akan dihasilkan seterusnya dalam urutan output.

Proses pelatihan model T5 menggunakan teknik seperti *Gap Sentence Generation (GSG)*, di mana beberapa ayat dalam dokumen asal dipilih dan dipadamkan, kemudian model dilatih untuk meramalkan ayat-ayat yang dipadamkan tersebut berdasarkan konteks yang tinggal. Selain itu, T5 menggunakan format teks-ke-teks untuk semua tugas NLP, termasuk penterjemahan, ringkasan, soalan-jawapan, dan lain-lain, yang membolehkan model dilatih dalam cara yang seragam dan serba guna. Keunikan dan kehebatan model T5 terletak pada pendekatan "*text-to-text*" yang seragam, membolehkan penggunaan model yang sama untuk pelbagai tugas NLP tanpa perlu mengubah seni bina asasnya. Dengan memanfaatkan seni bina transformer yang telah terbukti keberkesannya dalam memahami konteks teks, T5 mampu menangkap hubungan konteks yang mendalam dalam teks dan menghasilkan output yang bermakna dan relevan untuk pelbagai aplikasi dalam pemrosesan bahasa semula jadi. Dengan struktur ini, model T5 mampu menangani pelbagai tugas NLP dengan cara yang sangat efisien dan serba guna, menjadikannya salah satu model yang paling kuat dan fleksibel dalam pemrosesan bahasa semula jadi.

Fasa Pengujian

Dalam fasa pengujian, prestasi model ringkasan teks bahasa Melayu dinilai menggunakan metrik *ROUGE (Recall-Oriented Understudy for Gisting Evaluation)*. Metrik *ROUGE* adalah alat yang biasa digunakan untuk menilai kualiti ringkasan teks dengan membandingkan ringkasan yang dihasilkan oleh model dengan ringkasan rujukan (*ground truth*). Terdapat beberapa variasi metrik *ROUGE* yang digunakan dalam penilaian ini, termasuk *ROUGE-1*, *ROUGE-2*, dan *ROUGE-L*.

ROUGE-1 mengukur persamaan antara n-gram satu perkataan dalam ringkasan model dan ringkasan rujukan. *ROUGE-2* pula mengukur persamaan antara n-gram dua perkataan dalam ringkasan model dan ringkasan rujukan, manakala *ROUGE-L* mengukur persamaan berdasarkan urutan terpanjang n-gram yang sama (*Longest Common Subsequence - LCS*) dalam ringkasan model dan ringkasan rujukan.

Setiap metrik *ROUGE* merangkumi tiga komponen utama: *Precision (p)*, *Recall (r)*, dan *F1-score (f)*. *Precision* adalah peratusan n-gram yang dijana oleh model yang juga terdapat dalam ringkasan rujukan. *Recall* pula adalah peratusan n-gram dalam ringkasan rujukan yang juga dijana oleh model. *F1-score* adalah purata harmonik antara *precision* dan *recall*.

$$\text{Precision (p)} = \frac{|ngram_{model} \cap ngram_{rujukan}|}{|ngram_{model}|}$$

$$\text{Recall (r)} = \frac{|ngram_{model} \cap ngram_{rujukan}|}{|ngram_{rujukan}|}$$

$$\text{F1 - Score (f)} = \frac{2 \times p \times r}{p + r}$$

Rajah 5 Formula bagi *ROUGE-1* dan *ROUGE-2*

Rajah 5 di atas menunjukkan formula yang digunakan oleh *ROUGE-1* dan *ROUGE-2*. Untuk *ROUGE-1*, *precision* merujuk kepada peratusan n-gram satu perkataan yang dijana oleh model yang terdapat dalam ringkasan rujukan, *recall* merujuk kepada peratusan n-gram satu perkataan dalam ringkasan rujukan yang dijana oleh model, dan *F1-score* adalah purata harmonik antara *precision* dan *recall* untuk n-gram satu perkataan.

Bagi *ROUGE-2*, *precision* merujuk kepada peratusan n-gram dua perkataan yang dijana oleh model yang terdapat dalam ringkasan rujukan, *recall* merujuk kepada peratusan n-gram dua perkataan dalam ringkasan rujukan yang dijana oleh model, dan *F1-score* adalah purata harmonik antara *precision* dan *recall* untuk n-gram dua perkataan.

$$\text{Precision (p)} = \frac{LCS(model, rujukan)}{|model|}$$

$$\text{Recall (r)} = \frac{LCS(model, rujukan)}{|rujukan|}$$

$$\text{F1 - Score (f)} = \frac{2 \times p \times r}{p + r}$$

Rajah 6 Formula bagi *ROUGE-L*

Rajah 6 di atas menunjukkan formula yang digunakan untuk *ROUGE-L*. Untuk *ROUGE-L*, *precision* merujuk kepada peratusan urutan terpanjang n-gram yang sama (*LCS*) yang dijana oleh model yang terdapat dalam ringkasan rujukan, *recall* merujuk kepada peratusan urutan terpanjang n-gram yang sama (*LCS*) dalam ringkasan rujukan yang dijana oleh model, dan *F1-score* adalah purata harmonik antara *precision* dan *recall* untuk urutan terpanjang n-gram yang sama (*LCS*).

Prestasi model diukur dengan menghitung nilai-nilai metrik ini untuk setiap ringkasan yang dijana dan membandingkannya dengan ringkasan rujukan yang sepadan. Hasilnya akan memberikan gambaran yang jelas mengenai sejauh mana model mampu menghasilkan

ringkasan yang berkualiti dan sejauh mana ia bersamaan dengan ringkasan rujukan.

Rumusan-rumusan ini memberikan ukuran kuantitatif bagi pertindihan antara ringkasan yang dihasilkan mesin dengan ringkasan rujukan. Penting untuk diingat bahawa walaupun ROUGE adalah metrik yang berharga untuk menilai kualiti ringkasan, ia mempunyai kelemahan, seperti tidak sepenuhnya menggambarkan makna semantik atau kohesi ringkasan dan hanya bergantung pada pertindihan n-gram. Dalam pemilihan metrik ini, skor evaluasi diberikan sebagai angka, di mana skor yang lebih tinggi menunjukkan prestasi yang lebih baik. Pemilihan metrik dapat bergantung pada keperluan dan karakteristik spesifik dari tugas ringkasan yang dihadapi. Metrik ini sering digunakan untuk memberikan gambaran yang lebih komprehensif tentang kualiti ringkasan yang dihasilkan oleh model. Evaluasi yang baik membantu mengidentifikasi kelemahan model dan membimbing penambahbaikan atau pengembangan lebih lanjut.

Fasa Pembangunan Prototaip

Fasa Pembangunan Prototaip sistem ringkasan teks melibatkan beberapa langkah penting untuk memastikan sistem berfungsi dengan baik sebelum pelaksanaan skala penuh. Prototaip ini dibangunkan menggunakan teknologi seperti Python, TensorFlow, dan Flask, dengan Google Colab digunakan untuk latihan model dan Visual Studio Code untuk pembangunan aplikasi web.

Selepas penilaian prestasi model dalam tugas ringkasan teks dilakukan, keputusan penilaian akan dicatatkan dan digunakan untuk memilih model yang terbaik. Setelah model terbaik dipilih, aplikasi web dibangunkan menggunakan Flask dan Ngrok. Flask menyediakan rangka kerja yang fleksibel untuk membina antaramuka pengguna yang interaktif manakala Ngrok digunakan untuk menguji aplikasi web secara langsung di pelayan tempatan dengan menyediakan URL awam sementara. Ini membolehkan ujian dan demonstrasi aplikasi dilakukan dengan mudah tanpa perlu menyebarkan aplikasi ke pelayan produksi. Rajah 7 di bawah menunjukkan antara muka yang dihasilkan untuk Ringkasan Teks Bahasa Melayu.

Ringkasan Teks Bahasa Melayu

Rajah 7 Antara Muka Sistem Ringkasan Teks

Jika dilihat pada rajah 7 di atas, antara muka bagi sistem ringkasan teks ini telah direka dengan fokus kepada kemudahan penggunaan dan interaksi yang intuitif. Antara muka ini membolehkan pengguna memasukkan teks, memuat naik dokumen, menentukan panjang ringkasan, dan melihat hasil ringkasan dengan mudah. Pengguna mempunyai pilihan untuk menentukan panjang ringkasan teks melalui *toggle switch* yang membolehkan mereka memilih antara ringkasan pendek dan panjang. *Toggle switch* ini direka untuk memudahkan pengguna beralih antara pilihan-pilihan tersebut dengan hanya satu klik.

Teks boleh dimasukkan secara manual dalam kotak teks yang disediakan, di mana pengguna boleh menyalin dan menampal teks yang ingin diringkaskan. Kotak teks ini cukup besar untuk menampung teks yang panjang, memberikan pengguna ruang yang mencukupi untuk memasukkan teks mereka. Hasil ringkasan akan dipaparkan dalam kotak teks di sebelah kanan yang hanya boleh dibaca, memastikan pengguna tidak boleh mengubah hasil ringkasan yang dijana oleh sistem. Ini memastikan integriti hasil ringkasan yang dihasilkan oleh sistem. Di bawah setiap kotak teks, kiraan perkataan akan dipaparkan untuk membantu pengguna mengetahui jumlah perkataan dalam teks asal dan ringkasan. Kiraan perkataan ini dikira secara automatik setiap kali pengguna memasukkan teks atau apabila hasil ringkasan dijana.

Pengguna juga boleh memuat naik fail dalam format PDF atau DOCX melalui butang muat naik fail yang disediakan di bawah kotak teks untuk memasukkan teks. Ini memudahkan pengguna yang mempunyai dokumen dalam format yang disokong untuk mendapatkan ringkasan tanpa perlu menyalin dan menampal teks secara manual. Butang muat naik fail ini diintegrasikan dengan mekanisme pengecekan fail yang memastikan hanya fail yang mempunyai ekstensi yang dibenarkan (PDF atau DOCX) sahaja yang boleh dimuat naik. Setelah memasukkan teks atau memuat naik dokumen, pengguna boleh menekan butang "Ringkaskan" untuk menjana ringkasan. Butang ini berwarna merah untuk menarik perhatian

dan memudahkan pengguna mengenal pasti tindakan utama yang perlu dilakukan. Tekanan pada butang ini akan mencetuskan fungsi ringkasan yang menghantar teks input ke model pembelajaran mendalam yang telah dilatih untuk menghasilkan ringkasan.

Butang "Kosongkan" disediakan untuk mengosongkan semua medan input, berwarna kelabu untuk menunjukkan ia adalah pilihan sekunder. Butang ini membolehkan pengguna memulakan proses baru dengan mudah tanpa perlu menghapuskan teks secara manual. Apabila ditekan, semua medan input akan dikosongkan, termasuk kotak teks input, kotak teks ringkasan, dan kiraan perkataan. Warna latar belakang putih memberikan rupa yang bersih dan profesional, yang membantu dalam memastikan antaramuka pengguna kelihatan kemas dan teratur. Penggunaan warna merah pada butang "Ringkaskan" menarik perhatian kepada tindakan utama yang perlu dilakukan oleh pengguna, manakala warna kelabu pada butang "Kosongkan" menunjukkan bahawa ia adalah pilihan sekunder.

Penggunaan fon yang konsisten dan saiz teks yang sesuai memastikan teks mudah dibaca, yang sangat penting untuk pengalaman pengguna yang baik. Antara muka ini dibangunkan menggunakan HTML, CSS, dan JavaScript, dengan bantuan framework Bootstrap untuk memastikan ia responsif dan boleh digunakan pada pelbagai peranti, termasuk komputer meja dan peranti mudah alih. Bootstrap membantu dalam memastikan bahawa antara muka akan disesuaikan dengan baik pada skrin pelbagai saiz, memberikan pengalaman pengguna yang konsisten tidak kira peranti yang digunakan. Dengan reka bentuk antara muka yang intuitif dan mudah digunakan, sistem ini diharapkan dapat memberikan pengalaman pengguna yang baik dan memudahkan proses mendapatkan ringkasan teks.

Dalam keseluruhan fasa ini, prototaip yang dihasilkan menggabungkan model pembelajaran mendalam untuk menghasilkan ringkasan teks dengan antaramuka web yang membolehkan pengguna berinteraksi dengan sistem. Penilaian prestasi model menggunakan metrik ROUGE memastikan bahawa model yang dibangunkan berfungsi dengan baik dan menghasilkan ringkasan yang berkualiti tinggi. Dengan pembangunan prototaip ini, sistem ringkasan teks dapat diuji dan ditambah baik sebelum pelaksanaan skala penuh, memastikan keberkesanan dan kecekapan sistem dalam memenuhi keperluan pengguna.

KEPUTUSAN DAN PERBINCANGAN

Keputusan pengujian adalah bahagian penting dalam proses penilaian sistem, memberikan gambaran jelas mengenai prestasi dan kefungsi sistem yang diuji. Dalam bahagian ini, keputusan pengujian untuk kedua-dua Pengesahan Model dan Pengujian Kotak Hitam akan dibincangkan. Setiap bahagian akan menyajikan hasil pengujian yang diperoleh serta analisis terhadap prestasi sistem berdasarkan data yang telah diuji.

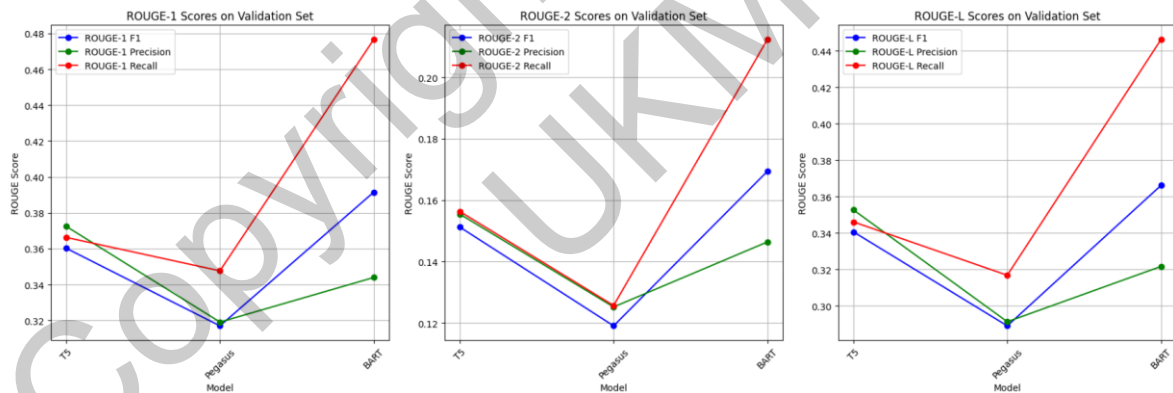
Keputusan Pengujian Pengesahan Model

Pengujian Pengesahan Model melibatkan penilaian prestasi model-model dalam menghasilkan ringkasan teks Bahasa Melayu. Dalam pengujian ini, algoritma pengoptimuman AdamW telah digunakan dan model dinilai dengan tiga model yang berbeza iaitu *BART*, *PEGASUS* dan *T5*.

Metrik *ROUGE* digunakan sebagai penanda aras untuk menilai kualiti ringkasan yang dihasilkan oleh model. Keputusan pengujian ini akan menunjukkan bagaimana perubahan dalam kadar pembelajaran mempengaruhi ketepatan dan keberkesanan model dalam menghasilkan ringkasan teks. Jadual 2 dan rajah 8 di bawah menunjukkan perbandingan prestasi tiga model (*BART*, *PEGASUS*, dan *T5*) berdasarkan metrik *ROUGE* pada set validasi. Metrik yang digunakan termasuk *ROUGE-1*, *ROUGE-2*, dan *ROUGE-L*, dengan masing-masing metrik menilai *F1 Score*, *Precision*, dan *Recall*.

Jadual 2 Keputusan pengujian bagi Validasi

Model	Metrik	Validasi		
		<i>ROUGE-1</i>	<i>ROUGE-2</i>	<i>ROUGE-L</i>
<i>BART</i>	<i>F1 Score</i>	0.3915	0.1694	0.3661
	<i>Precision</i>	0.3439	0.1464	0.3216
	<i>Recall</i>	0.4766	0.2123	0.4463
<i>PEGASUS</i>	<i>F1 Score</i>	0.3170	0.1191	0.2889
	<i>Precision</i>	0.3191	0.1252	0.2912
	<i>Recall</i>	0.3476	0.1257	0.3167
<i>T5</i>	<i>F1 Score</i>	0.3602	0.1512	0.3405
	<i>Precision</i>	0.3725	0.1555	0.3526
	<i>Recall</i>	0.3663	0.1563	0.3460



Rajah 8 Perbandingan di antara tiga jenis model untuk Validasi

Model *BART* menunjukkan prestasi terbaik dengan nilai *F1 Score* tertinggi pada semua metrik, iaitu 0.3915 untuk *ROUGE-1*, 0.1694 untuk *ROUGE-2*, dan 0.3661 untuk *ROUGE-L*. Ini menunjukkan bahawa *BART* paling baik dalam mengenal pasti dan merumuskan perkataan-perkataan penting dari teks rujukan. *Precision BART* juga tinggi, menunjukkan bahawa kebanyakan perkataan yang dihasilkan oleh model adalah relevan dan tepat. Nilai *Recall BART* yang tinggi menunjukkan bahawa model ini mampu menangkap sebahagian besar elemen penting dari teks rujukan.

Model *PEGASUS* menunjukkan prestasi sederhana dengan *F1 Score* 0.3170 untuk *ROUGE-1*, 0.1191 untuk *ROUGE-2*, dan 0.2889 untuk *ROUGE-L*. Walaupun *Precision*

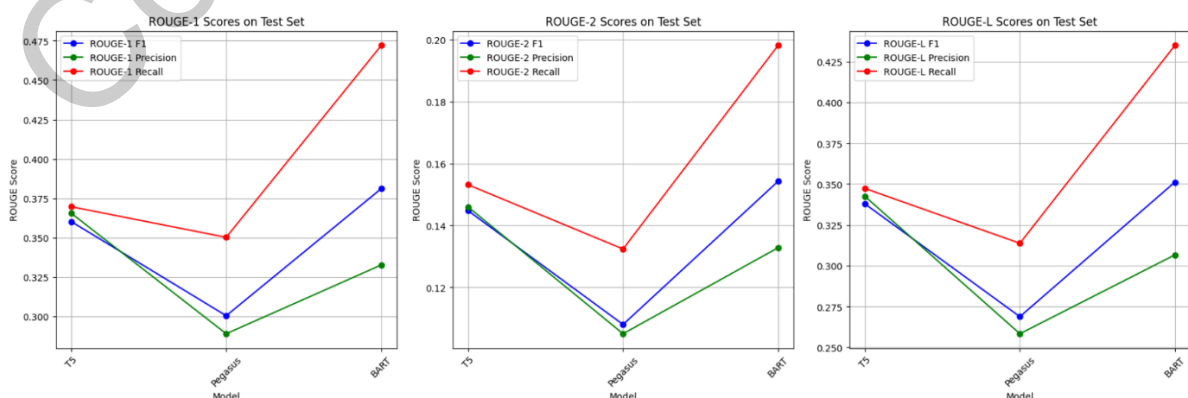
PEGASUS adalah lebih rendah berbanding *BART*, model ini masih mampu menghasilkan ringkasan yang relevan tetapi tidak sebaik *BART* dari segi ketepatan dan kebolehan menangkap elemen penting.

Model *T5* menunjukkan prestasi yang baik tetapi masih di bawah *BART*. *F1 Score T5* adalah 0.3602 untuk *ROUGE-1*, 0.1512 untuk *ROUGE-2*, dan 0.3405 untuk *ROUGE-L*. *Precision T5* adalah yang tertinggi untuk *ROUGE-1* dan *ROUGE-2*, menunjukkan bahawa perkataan-perkataan yang dihasilkan oleh *T5* adalah sangat relevan, namun model ini kurang baik dalam menangkap keseluruhan elemen penting berbanding *BART*, seperti yang ditunjukkan oleh nilai *Recall* yang lebih rendah.

Jadual 3 dan rajah 9 di bawah membandingkan prestasi tiga model (*BART*, *PEGASUS*, dan *T5*) berdasarkan metrik *ROUGE* pada set ujian. Metrik yang digunakan termasuk *ROUGE-1*, *ROUGE-2*, dan *ROUGE-L*, dengan masing-masing metrik menilai *F1 Score*, *Precision*, dan *Recall*.

Jadual 3 Keputusan pengujian bagi Ujian

Model	Metrik	Ujian		
		<i>ROUGE-1</i>	<i>ROUGE-2</i>	<i>ROUGE-L</i>
<i>BART</i>	<i>F1 Score</i>	0.3811	0.1542	0.3512
	<i>Precision</i>	0.3326	0.1328	0.3066
	<i>Recall</i>	0.4720	0.1981	0.4349
<i>PEGASUS</i>	<i>F1 Score</i>	0.3004	0.1080	0.2688
	<i>Precision</i>	0.2890	0.1050	0.2583
	<i>Recall</i>	0.3502	0.1324	0.3137
<i>T5</i>	<i>F1 Score</i>	0.3602	0.1448	0.3379
	<i>Precision</i>	0.3655	0.1460	0.3425
	<i>Recall</i>	0.3696	0.1531	0.3473



Rajah 9 Perbandingan di antara tiga jenis model untuk Ujian

Model *BART* menunjukkan prestasi terbaik secara keseluruhan. Untuk *ROUGE-1*, *BART* mencatatkan *F1 Score* tertinggi iaitu 0.3811, *Precision* 0.3326, dan *Recall* tertinggi 0.4720,

menunjukkan keupayaannya mengenal pasti dan merumuskan perkataan-perkataan penting dari teks rujukan dengan sangat baik. Berbanding dengan *BART*, model *T5* mempunyai *F1 Score* 0.3602, dengan *Precision* tertinggi untuk *ROUGE-1* (0.3655) tetapi *Recall* yang lebih rendah (0.3696). *PEGASUS*, sebaliknya, menunjukkan prestasi terendah dengan *F1 Score* 0.3004, *Precision* 0.2890, dan *Recall* 0.3502.

Untuk *ROUGE-2*, *BART* sekali lagi mengungguli dengan *F1 Score* tertinggi (0.1542), *Precision* 0.1328, dan *Recall* tertinggi 0.1981, menandakan keupayaannya mengenal pasti dan merumuskan frasa dua perkataan yang penting dengan lebih baik. *T5* mencatatkan *F1 Score* 0.1448, *Precision* tertinggi (0.1460), dan *Recall* 0.1531, sementara *PEGASUS* menunjukkan prestasi terendah dengan *F1 Score* 0.1080, *Precision* 0.1050, dan *Recall* 0.1324.

Dalam metrik *ROUGE-L*, *BART* terus menunjukkan prestasi unggul dengan *F1 Score* tertinggi (0.3512), *Precision* 0.3066, dan *Recall* tertinggi (0.4349), yang menandakan keupayaannya mengekalkan struktur keseluruhan teks dengan baik. *T5* mencatatkan *F1 Score* 0.3379, *Precision* tertinggi (0.3425), dan *Recall* 0.3473, tetapi masih di bawah prestasi *BART*. *PEGASUS* sekali lagi menunjukkan prestasi terendah dengan *F1 Score* 0.2688, *Precision* 0.2583, dan *Recall* 0.3137.

Secara keseluruhannya, *BART* adalah model yang paling berkesan untuk tugas ringkasan teks dalam Bahasa Melayu berdasarkan nilai *F1 Score*, *Precision*, dan *Recall* yang tertinggi pada semua metrik *ROUGE* dalam set ujian dan set validasi. Ini menunjukkan bahawa *BART* mempunyai keseimbangan yang baik antara ketepatan dan kebolehan menangkap elemen penting, menghasilkan ringkasan teks yang berkualiti tinggi. Model *T5* menunjukkan prestasi yang baik tetapi tidak sebaik *BART*, sementara *PEGASUS* menunjukkan prestasi yang paling rendah di antara ketiga-tiga model. Keputusan ini menguatkan lagi bahawa *BART* adalah model yang terbaik untuk digunakan dalam aplikasi ringkasan teks berdasarkan data ujian yang dinilai.

Keputusan Pengujian Kotak Hitam

Pengujian Kotak Hitam bertujuan untuk menilai kefungsiian dan kebolehgunaan aplikasi dari perspektif pengguna akhir tanpa melihat struktur dalaman atau kod sumber sistem. Dalam konteks projek ini, pengujian kotak hitam dilakukan untuk menilai pelbagai fungsi utama laman web ringkasan teks Bahasa Melayu yang dibangunkan menggunakan Flask. Aspek yang diuji termasuk pilihan panjang ringkasan, kotak input teks, pengiraan perkataan, fungsi muat naik fail, butang operasi (Ringkaskan dan Kosongkan), serta paparan hasil ringkasan. Keputusan pengujian yang ditunjukkan pada jadual 4 dibawah akan memberikan gambaran tentang sejauh mana aplikasi berfungsi mengikut spesifikasi yang ditetapkan dan memenuhi keperluan pengguna akhir.

Jadual 4 Keputusan pengujian kotak hitam

ID Fungsi	ID Pengujian	ID Prosedur Pengujian	Jangkaan Keputusan	Status
-----------	--------------	-----------------------	--------------------	--------

F-01	P-001	PU-001	Pengguna dapat memilih panjang ringkasan (Pendek/Panjang) dan sistem mengubah panjang ringkasan dengan betul.	Lulus
F-02	P-002	PU-002	Pengguna dapat memasukkan teks ke dalam kotak input dan teks tersebut dipaparkan dengan betul.	Lulus
F-03	P-003	PU-003	Pengguna dapat melihat kiraan perkataan dikemas kini dengan betul ketika memasukkan teks.	Lulus
F-04	P-004	PU-004	Pengguna dapat memuat naik fail <i>PDF</i> atau <i>DOCX</i> dan sistem memproses fail tersebut dengan betul.	Lulus
F-05	P-005	PU-005	Pengguna dapat menekan butang 'Ringkaskan' dan melihat ringkasan teks dipaparkan dengan betul dalam kotak output.	Lulus
F-06	P-006	PU-006	Pengguna dapat menekan butang 'Kosongkan' dan semua kotak input serta output dikosongkan.	Lulus
F-07	P-007	PU-007	Pengguna dapat melihat ringkasan yang dihasilkan dipaparkan dengan betul dalam kotak output.	Lulus

Hasil pengujian menunjukkan bahawa sistem mampu beroperasi dengan baik dan memenuhi keperluan pengguna akhir. Sebarang isu atau kekurangan yang ditemui semasa pengujian telah dikenal pasti dan diperbaiki, memastikan aplikasi yang dibangunkan adalah stabil, boleh dipercayai, dan memberikan pengalaman pengguna yang positif. Dengan pengujian kotak hitam ini, keyakinan terhadap kefunisian dan kebolegunaan aplikasi dapat dipastikan sebelum ia dilancarkan untuk kegunaan umum.

Cadangan Penambahbaikan

Terdapat beberapa masalah teknikal yang dihadapi dalam proses melatih model untuk meringkaskan teks Bahasa Melayu. Pertama, kekurangan dataset Bahasa Melayu yang besar dan berkualiti tinggi adalah satu cabaran utama. Untuk mengatasi masalah ini, langkah yang boleh diambil adalah dengan menggunakan dataset Bahasa Inggeris yang lebih besar untuk melatih model. Selain itu, penggunaan teknik transfer learning dapat membantu di mana

model yang telah dilatih dengan dataset Bahasa Inggeris akan disesuaikan dengan data Bahasa Melayu. Teknik data augmentation juga boleh digunakan untuk memperluas dataset Bahasa Melayu yang sedia ada dengan menghasilkan variasi teks yang berbeza.

Kedua, kualiti ringkasan yang dihasilkan oleh model adalah rendah, yang menyebabkan ringkasan tidak tepat. Untuk memperbaiki kualiti ringkasan, model perlu dilatih semula dengan menggunakan teknik pengoptimuman yang lebih baik dan hyperparameter tuning. Teknik fine-tuning pada model yang telah dilatih dengan dataset Bahasa Inggeris menggunakan subset dataset Bahasa Melayu juga boleh membantu menyesuaikan model dengan lebih baik. Evaluasi dan validasi secara berkala menggunakan metrik-metrik seperti ROUGE perlu dilakukan untuk menilai dan memperbaiki kualiti ringkasan yang dihasilkan.

Ketiga, penggunaan dataset Bahasa Inggeris yang terlalu besar menyebabkan masa latihan yang lama dan kekurangan RAM, yang akhirnya menyebabkan sistem terhempas. Untuk mengatasi masalah ini, teknik pengoptimuman memori seperti gradient checkpointing, mixed precision training, dan model quantization boleh digunakan untuk mengurangkan penggunaan memori semasa latihan. Selain itu, memilih subset dataset yang representatif daripada dataset yang lebih besar dapat mengurangkan masa latihan dan penggunaan memori tanpa menjejaskan kualiti model. Menggunakan infrastruktur komputasi yang lebih kuat seperti GPU dan TPU yang lebih canggih, atau perkhidmatan cloud computing juga dapat meningkatkan kapasiti komputasi.

Akhirnya, model menghasilkan ringkasan teks Bahasa Melayu yang tidak tepat kerana model lebih banyak digunakan untuk menghasilkan teks dalam Bahasa Inggeris. Penyelesaiannya adalah dengan menukar teks Bahasa Melayu ke Bahasa Inggeris sebelum proses ringkasan, dan selepas ringkasan dihasilkan, teks tersebut akan ditukar kembali ke Bahasa Melayu. Menggunakan perkhidmatan penterjemahan yang tepat dan berkualiti tinggi untuk proses ini adalah penting untuk memastikan ketepatan penterjemahan. Selain itu, melatih model multibahasa yang mampu memahami dan menghasilkan ringkasan dalam pelbagai bahasa termasuk Bahasa Melayu dapat membantu mengatasi masalah ini tanpa perlu menukar bahasa teks asal. Dengan mengimplementasikan cadangan-cadangan ini, diharapkan prestasi model untuk meringkaskan teks Bahasa Melayu dapat ditingkatkan, mengatasi kekangan-kekangan yang dihadapi, dan menghasilkan ringkasan teks yang lebih berkualiti dan tepat.

KESIMPULAN

Secara keseluruhannya, projek kajian mengenai Ringkasan Teks Bahasa Melayu Menggunakan Pembelajaran Mendalam telah mencapai objektifnya dengan jayanya. Kajian ini telah membangunkan model pembelajaran mendalam yang berupaya meringkaskan teks dalam Bahasa Melayu dengan ketepatan yang tinggi. Sepanjang proses pembangunan, beberapa kekangan teknikal telah dikenalpasti dan diatasi, termasuk kekurangan dataset Bahasa Melayu yang besar, kualiti ringkasan yang dihasilkan oleh model, serta cabaran-cabaran berkaitan dengan sumber komputasi.

Walaupun begitu, masih terdapat ruang untuk penambahbaikan. Cadangan masa hadapan termasuk pengumpulan dataset Bahasa Melayu yang lebih besar dan berkualiti, penggunaan teknik-teknik pengoptimuman untuk meningkatkan kualiti ringkasan, serta penerapan teknik pengoptimuman memori dan komputasi untuk mengatasi masalah kekurangan sumber. Di samping itu, pembangunan antara muka pengguna yang lebih mesra dan interaktif akan meningkatkan kegunaan dan keberkesanan alat ringkasan teks ini.

Secara keseluruhan, projek ini telah memberikan sumbangan yang penting dalam bidang ringkasan teks Bahasa Melayu, menunjukkan potensi penggunaan model pembelajaran mendalam untuk tugas-tugas PBT. Dengan meneruskan usaha penambahbaikan dan pengembangan, adalah diharapkan alat ringkasan teks ini akan terus memberikan manfaat kepada pengguna, memudahkan proses pengekstrakan maklumat penting dari teks yang panjang dan kompleks. Projek ini juga membuka jalan untuk kajian-kajian lanjut dalam bidang ringkasan teks dan PBT, terutamanya dalam konteks Bahasa Melayu.

Kekuatan Sistem

Kekuatan sistem ini mampu menghasilkan ringkasan yang padat dan menyeluruh dengan mengekstrak idea-idea utama dari teks asal. Keupayaan ini penting untuk memudahkan pengguna mendapatkan maklumat penting dengan cepat tanpa perlu membaca keseluruhan teks. Selain itu, integrasi antara muka pengguna yang mesra dan pilihan untuk menyesuaikan panjang ringkasan menambah nilai kepada sistem ini, menjadikannya lebih fleksibel dan mudah digunakan. Sistem ini juga menunjukkan potensi besar dalam memajukan bidang pemprosesan bahasa semulajadi (PBT) untuk Bahasa Melayu, menyediakan alat yang canggih untuk tujuan akademik, profesional, dan harian.

Kelemahan Sistem

Sistem ini juga mempunyai kelemahan yang perlu diberi perhatian. Kekurangan dataset Bahasa Melayu yang besar dan berkualiti tinggi merupakan cabaran utama yang menjejaskan kualiti dan ketepatan ringkasan yang dihasilkan. Selain itu, masa latihan model yang lama dan penggunaan memori yang tinggi akibat penggunaan dataset Bahasa Inggeris yang besar boleh menyebabkan masalah sistem terhempas dan meningkatkan kos komputasi. Kualiti ringkasan yang dihasilkan juga masih memerlukan penambahbaikan, terutama dalam memastikan ringkasan yang dihasilkan tepat dan relevan. Walaupun teknik transfer learning dan data

augmentation dapat membantu, hasilnya tidak selalu konsisten dan bergantung kepada kualiti data yang digunakan. Tambahan pula, model yang dilatih lebih banyak dengan dataset Bahasa Inggeris mungkin menghadapi kesukaran dalam menghasilkan ringkasan yang tepat dalam Bahasa Melayu, memerlukan langkah-langkah tambahan seperti terjemahan teks yang berpotensi mengurangkan ketepatan.

PENGHARGAAN

Syukur Alhamdulillah dan setinggi-tinggi kesyukuran dipanjatkan kehadiran ilahi kerana dengan izin kurnianya dapat saya menyempurnakan tugas projek ini dengan jayanya.

Dengan kesempatan yang ada ini, saya amat berbesar hati untuk menghadiahkan jutaan terima kasih kepada Assoc. Prof. Dr. Nazlia Omar, selaku penyelia kerana telah meletakkan sepenuh kepercayaan beliau kepada saya untuk melunaskan tugas yang diberikan.

Saya juga ingin mengucapkan ribuan terima kasih kepada semua pihak yang tidak putus-putus dalam usaha membantu menyempurnakan tugas ini terutamanya kepada penyelia saya Assoc. Prof. Dr. Nazlia Omar atas budi bicara beliau dalam memberi tunjuk ajar sepanjang masa tugas ini dijalankan. Selain itu, saya turut berterima kasih kepada rakan-rakan seperjuangan saya kerana telah banyak menghulurkan bantuan dan kerjasama bagi merealisasikan usaha menyempurnakan tugas ini dengan jayanya.

Ucapan ini juga ditujukan kepada semua pihak yang telah terlibat dalam menjayakan tugas ini sama ada secara langsung atau tidak langsung. Segala bantuan yang telah mereka hulurkan amatlah saya hargai kerana tanpa bantuan dan sokongan mereka semua tugas ini mungkin tidak dapat dilaksanakan dengan baik.

RUJUKAN

- Alias, S., Mohammad, S. K., Keng Hoon, G., & Tien Ping, T. (2016). A Malay text corpus analysis for sentence compression using pattern-growth method. *Jurnal Teknologi*, 78(8). <https://doi.org/10.11113/jt.v78.7413>
- Alias, S. (2021). Malay TextSummarization Dataset. <https://doi.org/10.17632/r54zh37mc7.1>
- Babar, S. A., & Patil, P. D. (2015). Improving performance of text summarization. *Procedia Computer Science*, 46, 354–363. <https://doi.org/10.1016/j.procs.2015.02.031>
- Barnett, M. (2021, January 5). *A history of machine summarisation*. RSS. <https://www.genei.io/blog/a-history-of-machine-summarisation>
- Bonthu, H. (2022, July 26). Rule-based sentiment analysis in Python. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/06/rule-based-sentiment-analysis-in-python/#:~:text=POS%20tagging%20essential%20to%20preserve,an%20article%20on%20Sentiment%20Analysis%E2%80%9D>.

- Bopaiah, Y. (2023, August 17). *Unveiling the power of Rouge Metrics in NLP*. Medium. <https://pub.aimind.so/unveiling-the-power-of-rouge-metrics-in-nlp-b6d3f96d3363>
- Chadrick. (2022, January 12). *Paper summary: "Bart: Denoising Sequence-to-sequence pre-training for natural language generation..."*. Medium. <https://medium.com/mllearning-ai/paper-summary-bart-denoising-sequence-to-sequence-pre-training-for-natural-language-generation-69e41dfbb7fe>
- Chiusano, F. (2022, January 19). Two minutes NLP-learn the Rouge metric by examples. Medium. <https://medium.com/nlplanet/two-minutes-nlp-learn-the-rouge-metric-by-examples-f179cc285499>
- CNN-dailymail news text summarization*. Kaggle. (2021). <https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail/data>
- Dagdagan, J. (2021, February 20). Fake news detection: Text pre-processing. Medium. <https://jon-dagdagan.medium.com/fake-news-detection-pre-processing-text-d9648a2854e5>
- Deery, M. (2023, August 30). *The flask web framework: A beginner's guide*. CareerFoundry. <https://careerfoundry.com/en/blog/web-development/what-is-flask/>
- Delovski, B. (2023, May 8). *How to summarize text using machine learning models*. Edlitera. <https://www.edlitera.com/blog/posts/text-summarization-nlp-how-to>
- Dyouri, A. (2022, December 21). How to make a web application using flask in python 3. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3>
- freeCodeCamp.org. (2017, January 26). *An intro to Rouge, and how to use it to evaluate summaries*. <https://www.freecodecamp.org/news/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840/>
- Georgian. (2023, August 4). *The Practical Guide to LLMs: Flan-T5*. Medium. <https://medium.com/georgian-impact-blog/the-practical-guide-to-llms-flan-t56d26cc5f14c0>
- Goularte, F. B., Nassar, S. M., Fileto, R., & Saggion, H. (2019). A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Systems with Applications*, 115, 264–275. <https://doi.org/10.1016/j.eswa.2018.07.047>
- Gupta, C., Jain, A., & Joshi, N. (2018). Fuzzy logic in natural language processing – a closer view. *Procedia Computer Science*, 132, 1375–1384. <https://doi.org/10.1016/j.procs.2018.05.052>
- Harsha, A. (2022, December 20). Understanding part-of-speech tagging in NLP: Techniques and Applications - Shiksha Online. shiksha. <https://www.shiksha.com/online-courses/articles/pos-tagging-in-nlp/#3>
- Institute, I. @ T. (2022, May 15). *Notes on abstractive summarization: Pegasusxsum and T5*. Medium. <https://medium.com/artificialis/notes-on-abstractive-summarization-pegasusxsum-and-t5-497109d72029>
- Kirti, A. (2023, April 17). *Rule based approach in NLP*. GeeksforGeeks. <https://www.geeksforgeeks.org/rule-based-approach-in-nlp/>
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for

- natural language generation, translation, and comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.703>
- Lutkevich, B., & Schmelzer, R. (2023, August 17). *What is GPT-3? everything you need to know - techtarget*. Enterprise AI. <https://www.techtarget.com/searchenterpriseai/definition/GPT-3>
- Malançon, B. R., & Hazimah, I. (2017, April). *Which extractive summarization method for Malay texts*. Semantic Scholar. <https://www.semanticscholar.org/paper/Which-Extractive-Summarization-Method-For-Malay-Ranaivo-Malan%C3%A7on-Hazimah/169c0f48a0c9ee2bf44c81f05e436d68007eb717>
- Mayo, M. (2022, November 7). *Approaches to text summarization: An overview*. KDnuggets. <https://www.kdnuggets.com/2019/01/approaches-textsummarization-overview.html>
- Mazumdar, S. (2023, April 4). *Exploring the extractive method of text summarization*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2023/03/exploring-the-extractive-method-of-text-summarization/#:~:text=Extractive%20summarization%20techniques%20select%20and,of%20the%20original%20text%20intact.>
- Nepal, P. (2022, October 16). *Text summarization*. Medium. <https://medium.com/@prabinnepal/text-summarization-a78119037548>
- Pai, A. (2023, May 15). *Comprehensive guide to text summarization using Deep Learning in python*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-textsummarization-using-deep-learning-python/#h-introduction>
- Payong, A. (2023, May). *Introduction to extractive and abstractive summarization techniques*. Paperspace Blog. <https://blog.paperspace.com/extractive-and-abstractive-summarization-techniques/>
- Payong, A. (2023, September 22). *Bart model for text summarization*. Paperspace Blog. <https://blog.paperspace.com/bart-model-for-text-summarization-part1/>
- Raval, P. (2023, October 12). *Transformers bart model explained for text summarization*. ProjectPro. <https://www.projectpro.io/article/transformers-bart-model-explained/553>
- Roy, A. (2020, August 8). *Understanding automatic text summarization-2: Abstractive Methods*. Medium. <https://towardsdatascience.com/understanding-automatic-text-summarization-2-abstractive-methods-7099fa8656fe>
- Santhosh, S. (2023, April 16). *Understanding Bleu and Rouge score for NLP evaluation*. Medium. <https://medium.com/@sthanikamsanthosh1994/understanding-bleu-and-rougescore-for-nlp-evaluation-1ab334ecadb#:~:text=While%20BLEU%20score%20is%20primarily,the%20reference%20translations%20or%20summaries.>
- Shiraly, K. (2023, August 15). *Bart text summarization vs. GPT-3 vs. Bert: An in-depth comparison*. Width.ai. <https://www.width.ai/post/bart-text-summarization#:~:text=To%20learn%20to%20summarize%20at,summary%20for%20the%20use%20case.>

- Singh, D. (2022, August 22). Text classification of news articles. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/12/text-classification-of-news-articles/>
- T. I. N. (2021, March 1). *Abstractive text summarization with Deep Learning*. Times Internet: Everything. Everyday. <https://timesinternet.in/blog/abstractive-text-summarization-with-deep-learning/>
- Taneja, A. (2023, August 6). *Demystifying multihead attention in the Transformer neural network architecture – with code*. LinkedIn. <https://www.linkedin.com/pulse/demystifying-multihead-attention-transformer-neural-network-taneja>
- Travis Status. (2020, June 14). Googletrans. PyPI. <https://pypi.org/project/googletrans/#:~:text=Googletrans%20is%20a%20free%20and,refer%20to%20the%20API%20Documentation>.
- Tripathi, U. (2023, June 4). PDF summarization using pegasus. Medium. <https://medium.com/@udbhav9696/pdf-summarization-using-pegasus-18d04f7b9b9a>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June 12). *Attention is all you need*. arXiv.org. <https://arxiv.org/abs/1706.03762>
- Zamin, N., & Ghani, A. (2010). *A hybrid approach for Malay text summarizer*. Ministry of Education - Saudi Arabia. <https://ksascholar.dri.sa/en/publications/a-hybrid-approach-for-malay-text-summarizer-2>
- Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2019, December 18). *Pegasus: Pre-training with extracted gap-sentences for abstractive summarization*. arXiv.org. <https://arxiv.org/abs/1912.08777v2>

Amira Syafika Binti Abd Rahim (A187237)
PROF. MADYA DR. NAZLIA BINTI OMAR
 Fakulti Teknologi & Sains Maklumat
 Universiti Kebangsaan Malaysia