Sistem Question Answering pada Data Kesehatan Menggunakan Model pre-trained BERT

Bagas Millen Alhafidz
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
bagasmillena@student.telkomuniversity.ac.id

Ema Rachmawati
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
emarachmawati@telkomuniversity.ac.id

Prasti Eko Yunanto
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
gppras@telkomuniversity.ac.id

***Abstrak* — Setelah pandemi covid-19, kesehatan menjadi hal yang harus diperhatikan. Sebagian besar masyarakat menggunakan search engine sebagai alat untuk mencari informasi tentang kesehatan. Namun informasi yang didapatkan berupa query hasil search engine yang masih umum. Sistem Question Answering adalah sistem yang memberikan informasi sesuai informasi yang dibutuhkan oleh pengguna secara spesifik. Pada penelitian ini dibangun sistem Question Answering menggunakan metode Bidirectional Encoder Representations from Transformer (BERT). BERT merupakan sebuah pre-trained model yang menggunakan arsitektur transformer. BERT dapat menyelesaikan tugas sistem Question Answering. Dengan pre-trained model, sistem tidak perlu melakukan training model dari awal. Sistem hanya perlu menggunakan train model yang telah dibuat oleh orang lain sesuai tugas yang dibutuhkan untuk menghemat waktu dan sumber daya. Untuk mengukur performansi, digunakan metode Exact Match (EM) dan F1-score. Hasil dari penelitian ini skor terbaik yang didapat yaitu Exact Match 75% dan F1-score 76%.**

***Kata kunci*— question answering, BERT, pre-trained model, kesehatan**

I. PENDAHULUAN

Indonesia telah memasuki fase endemi Covid-19 setelah pemerintah menyatakan berakhirnya masa pandemi Covid-19 pada tanggal 21 Juni 2023 [1]. Dampak dari pandemi Covid-19 adalah adanya kepanikan, rasa was-was dan kecurigaan terhadap orang asing yang dapat menyebabkan menurunnya kesehatan masyarakat fisik dan juga kesehatan mental [2]. Adanya masa karantina dan isolasi membuat masyarakat dituntut untuk bisa beradaptasi menggunakan teknologi digital khususnya pada bidang kesehatan [3]. Dalam bidang kesehatan, penggunaan layanan kesehatan digital seperti aplikasi healthcare meningkat secara signifikan [4] yang berujung pada timbulnya tantangan untuk menyediakan informasi terpercaya dan akurat secara cepat. Pertanyaan dasar yang cukup penting seperti "Apa gejala Covid-19?", "Bagaimana pertolongan pertama pada pasien Covid-19?" sering ditanyakan oleh masyarakat. Untuk mencari informasi tentang covid-19, masyarakat menggunakan internet sebagai medianya. Hal ini dikarenakan keputusan dan pilihan yang diambil oleh masyarakat lebih banyak didasarkan pada informasi dari internet, terutama media sosial [5]. Berdasarkan kebiasaan tersebut, informasi yang diperoleh terkadang belum spesifik dengan yang diharapkan. Oleh karena itu untuk mengatasi permasalahan tersebut maka perlu Sistem Question Answering (QA) untuk mendapatkan informasi spesifik.

Sistem QA digunakan untuk memberikan informasi secara spesifik yang dibutuhkan oleh pengguna. Berbeda dengan search engine, sistem QA bertujuan untuk memberikan jawaban atas pertanyaan secara langsung daripada hanya memberikan tautan yang relevan, sehingga memberikan kemudahan dan efisiensi dalam mendapatkan informasi [6]. Pengguna hanya perlu memberikan pertanyaan, kemudian sistem akan otomatis menghasilkan jawaban yang sesuai. Sistem QA ini dapat digunakan untuk memudahkan masyarakat untuk mendapatkan jawaban seputar kesehatan dalam waktu yang singkat. Salah satu framework yang cocok untuk sistem QA adalah BERT. BERT dianggap sebagai stateof-the-art dalam tugas NLP (Natural Language Processing) [7].

Pada penelitian ini, model BERT yang digunakan adalah IndoBERT, yaitu model BERT yang di-train menggunakan dataset berbahasa Indonesia [8]. Pada beberapa penelitian, IndoBERT memiliki hasil yang unggul dibandingkan model berbasis BERT yang lain. Penelitian [9] yang membandingkan IndoBERT dengan RoBERTa pada studi kasus sistem QA terhadap dataset Buku Panduan Akademik menghasilkan F1-score 91,32 dan Exact Match 81,17 untuk IndoBERT dan F1-score 90,18 dan Exact Match 79,53 untuk RoBERTa. Penelitian lain [10] melakukan uji coba terhadap data BPS (Badan Pusat Statistik) Indonesia menggunakan model IndoBERT, distilbert, all-MiniLM-L6-V2, dan RoBERTa. Hasil dari penelitian tersebut didapat F1-score masing-masing 46,03; 35,81; 32,67; 36,63, dan masing-masing Exact Match 1,33; 0,33; 1,0; 1,0.

Evaluation metric yang digunakan pada peneliatian ini adalah Exact Match (EM) dan F1-score yang juga umum digunakan pada BERT untuk sistem QA [11].

II. KAJIAN TEORI

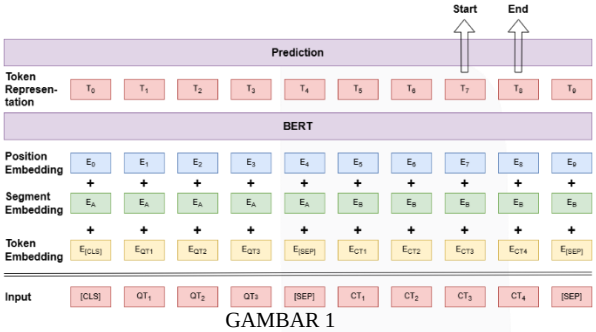
A. Sistem Question Answering

Sistem Question Answering atau Sistem Tanya Jawab adalah sistem yang bisa mengeluarkan jawaban (answer) ringkas dan tepat ketika pengguna memasukkan pertanyaan (question) [12]. Berbeda dengan search engine, Sistem QA (Question Answering) menghasilkan jawaban yang akurat daripada hanya kumpulan dokumen informasi. Ada beragam kategori pada Sistem QA berdasarkan pada tipe question, tipe answer yang diharapkan, sumber bukti diperolehnya answer,

dan model untuk mendapatkan answer. Dalam kategori model untuk mendapatkan answer, ada sub kategori yaitu raw text-based Question Answering. Pada sub kategori ini, sistem QA menghasilkan answer berdasarkan context [13].

B. BERT

BERT: Bidirectional Encoder Representations from Transformers adalah model berbasis transformer yang dibuat untuk menyelesaikan tugas NLP termasuk pada sistem QA. Salah satu karakteristiknya adalah Bidirectional, yang berarti BERT dapat memahami konteks dari kanan ke kiri dari suatu kata. BERT termasuk ke dalam contextual model yaitu menghasilkan representasi tiap kata berdasarkan kata lain pada suatu kalimat [7]. Tahapan utama pada BERT yaitu pre-training dan fine-tuning. Tahap pre-training, BERT menggunakan metode MLM (Masked Language Model) yang memprediksi output berdasarkan kata yang di-mask pada input token dan NSP (Next Sentence Prediction) yang memprediksi output berdasarkan keterkaitan dua kalimat yang bersebelahan. Tahap fine-tuning, satu atau lebih layer encoder ditambahkan di akhir untuk menyesuaikan domain yang lebih spesifik [14]. Secara default, layer di bawahnya tidak di-freeze saat finetuning, namun untuk dataset yang kecil dapat dilakukan freeze layer untuk mempercepat proses training. Namun melakukan freeze layer kurang cocok untuk sistem QA dengan closed-domain dataset [15].



GAMBAR 1
Arsitektur BERT untuk sistem QA

Pada sistem QA, input pada BERT berupa question token (QT) dan context token (CT). Input tersebut direpresentasikan dalam bentuk Token Embedding, Segment Embedding dan Position Embedding. Token Embedding adalah bentuk token yang berasal dari hasil WordPiece Tokenization dengan membagi kata menjadi sub kata. Segment Embedding adalah penanda bahwa EA merupakan kata pada question, dan EB merupakan kata pada context. Position Embedding menunjukkan posisi tiap token terhadap kalimat keseluruhan. Kemudian diproses oleh layer encoder BERT sebanyak n layer (n=12 untuk BERT-base dan n=24 untuk BERT-large). Output yang dihasilkan berupa nilai logits dari token context yang akan menjadi acuan untuk start atau end token answer [16] [17]. Nilai logits adalah nilai mentah yang belum dilakukan normalisasi. Nilai ini merupakan input dari fungsi softmax. Untuk mengubah nilai logits menjadi nilai probabilitas yang lebih mudah diinterpretasikan, dilakukan penambahan layer akhir pada BERT [18] dengan fungsi aktivasi (1).

$$softmax = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_i}} \quad (1)$$

dengan:
z_i = elemen ke-i dari vektor logits
n = banyaknya elemen pada vektor logits
e^{z_i} = hasil fungsi eksponensial dari vektor logits

Fungsi aktivasi yang umum digunakan pada sistem QA BERT adalah softmax, karena fungsi softmax dapat menghasilkan nilai probabilitas dari gabungan nilai logits pada start token dan end token.

C. IndoBERT

IndoBERT adalah model transformer berbasis BERT dengan konfigurasi bawaan menggunakan BERT-Base uncased yang dimodifikasi dengan 512 training token per batch. IndoBERT ditrain menggunakan vocabulary WordPiece Indonesia sebesar 31.923 vocabulary dengan total training IndoBERT yaitu lebih dari 220 juta kata, yang merupakan gabungan dari 3 sumber utama yaitu: (1) Wikipedia Indonesia sebanyak 74 juta kata; (2) artikel berita dari Kompas, Tempo dan liputan6 sebanyak 55 juta kata; dan (3) Web Corpus Indonesia sebanyak 90 juta kata. Data tersebut ditrain dengan batch size 128, learning rate 1e-4, adam optimizer, dan linear scheduler dan kemudian model ditrain dengan 180 epoch selama dua bulan penuh [8].

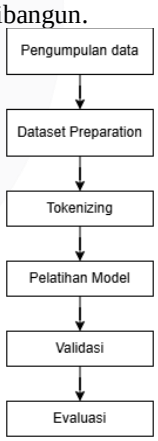
D. Website Halodoc

Website Halodoc merupakan website berbahasa Indonesia yang berfungsi untuk perantara konsultasi kesehatan antara pasien dan dokter secara 24 jam. Di website Halodoc juga terdapat artikel tentang kesehatan yang ditulis oleh ahli dalam bidang kesehatan. Pada tahun 2021, Halodoc memiliki 27 juta pengguna aktif bulanan, dengan rating 4,9 dari 177.037 pengguna di play store dan app store [19].

III. METODE

A. Diagram Sistem

Pada penelitian ini, sistem dibangun untuk mendapatkan answer yang sesuai. Gambar 2. menunjukkan diagram blok gambaran sistem yang dibangun.



GAMBAR 2
Diagram blok sistem yang dibangun

B. Implementasi Sistem

a. Pengumpulan Dataset

Tahap pertama adalah pengumpulan data. Dataset Sistem Question Answering terdiri dari context, question, dan answer. Context berisi teks paragraf yang menjadi acuan dari question dan answer. Question berisi teks pertanyaan yang terkait dengan context, sedangkan answer berisi teks jawaban yang terdapat pada context.

Context berasal dari website artikel kesehatan halodoc yang diperoleh dengan cara scrapping. Scrapping dilakukan secara manual dengan menggunakan halodoc api dan custom algoritma python yang hasilnya disimpan dalam format json. Data context yang masih mentah kemudian dilakukan pembersihan data secara manual dengan menghilangkan beberapa kata yang tidak memiliki makna seperti “k...”, “..”, “...”, “ ”, dan sebagainya. Data Context yang berhasil diperoleh ada sebanyak 200 context. Setiap satu context berisi teks satu halaman artikel pada website halodoc. Maka untuk membuat dataset sebanyak 200 context, dibutuhkan 200 halaman artikel.

Setelah data context dibersihkan, setiap 1 context akan memiliki kurang lebih 10 pasang question dan answer. Question dan answer di-generate menggunakan api Chat-GPT dengan prompt “Saya memiliki sebuah artikel x (x adalah context), Tolong berikan saya 10 pertanyaan berdasarkan artikel tersebut dan juga jawaban atas pertanyaan tersebut.”.

Question dan answer hasil dari generate chat-GPT kemudian dibersihkan dengan menghilangkan question dan answer jika salah satunya terdapat teks string kosong (“ ”) dan mengedit jika ada kalimat yang terpotong.

b. Dataset Preparation

Context beserta pasangan question dan answer diformat ke dalam bentuk anotasi dataset SQuAD. Dataset SQuAD (Stanford Question Answering Dataset) merupakan standar format dataset untuk sistem Question Answering pada BERT.

Format dataset pada Gambar 3. memiliki detail sebagai berikut; data berisi 200 context beserta question dan answer, article_id berisi nilai integer dari id artikel, paragraphs berisi context beserta question dan answer, context berisi teks context yang telah di-scrap dan dibersihkan, qas berisi 10 pasang question dan answer, question berisi teks question yang telah di-generate oleh chat-GPT, answer berisi data text dan answer_start. Text berisi teks answer yang telah di-generate oleh chat-GPT, answer_start berisi nilai integer dari posisi teks answer terhadap teks context. Id berisi nilai integer dari id qas, is_impossible berisi nilai boolean dari status question. True jika question tidak terkait dengan context dan False jika question terkait dengan context. Pada dataset ini, semua label is_impossible diisi dengan False karena question pasti terkait dengan context.


```
{"data": [
    {
        "article_id": 0,
        "paragraphs": [
            {
                "context": "",
                "qas": [
                    {
                        "question": "",
                        "answer": [
                            {
                                "text": "",
                                "answer_start": 0
                            }
                        ],
                        "id": 0,
                        "is_impossible": False
                    }
                ]
            }
        ]
    }
]}
```


GAMBAR 3
Format dataset Question Answering

Pada label answer_start merupakan proses labeling yang cukup sulit, karena teks kalimat answer yang didapatkan dari chat-GPT banyak yang outputnya tidak sama persis dengan kalimat yang ada di context. Akibatnya, saat dilakukan pencarian answer terhadap context, menghasilkan nilai -1, yang artinya answer tidak ada pada context.


```
{"context": "Ini Besar Kalori Lontong Sayur dan Cara
Sehat Mengonsumsinya “Jumlah kalori seporsi lontong
sayur sekitar 357 kkal. Namun, jumlah tersebut . . .,
 "qas":
[{"question": "Berapa jumlah kalori dalam satu porsi
lontong sayur?",
 "answer":
[{"text": "Jumlah kalori dalam satu porsi lontong sayur
adalah sekitar 357 kkal.",
 "answer_start": -1}]
```


GAMBAR 4
Contoh answer yang tidak ada pada context

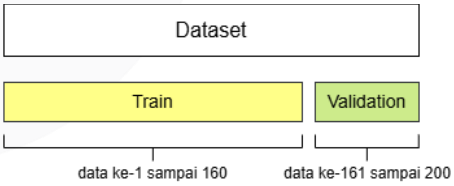
Sedangkan seharusnya saat dilakukan pencarian answer terhadap context, akan menghasilkan nilai dari indeks posisi kata pertamanya jika teks answer sama persis dengan kalimat yang ada dalam context.


```
{"context": "Ini Besar Kalori Lontong Sayur dan Cara
Sehat Mengonsumsinya “Jumlah kalori seporsi lontong
sayur sekitar 357 kkal. Namun, jumlah tersebut . . .,
 "qas":
[{"question": "Berapa jumlah kalori dalam satu porsi
lontong sayur?",
 "answer":
[{"text": "357 kkal",
 "answer_start": 15}]
```


GAMBAR 5
Contoh answer yang ada pada context

Oleh karena itu, untuk mendapatkan nilai answer_start, dilakukan pengeditan teks answer secara manual untuk membuat teks answer menjadi sama persis dengan teks yang ada dalam context.

Dataset sebanyak 200 data dibagi menjadi train dan validation dengan perbandingan masing-masing 80% dan 20%. Pembagian dataset dilakukan secara manual dengan rincian; data ke-1 sampai 160 merupakan data train dan data ke161 sampai 200 merupakan data validation.

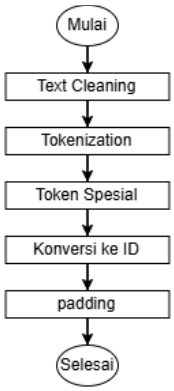


GAMBAR 6
Split dataset menjadi data train dan validation

Format dataset diubah yang sebelumnya disusun berdasarkan id context menjadi dataset yang disusun berdasarkan id question sehingga data train yang sebelumnya berjumlah 160 menjadi 1601 data, dan data validation yang sebelumnya berjumlah 40 menjadi 396 data.

c. Tokenizing

Data train dan validation diproses dengan mengubah teks question dan context menjadi token dengan alur pada Gambar 7.



GAMBAR 7
Alur Tokenizing

i. Text Cleaning

Teks dibersihkan dengan mengubah input menjadi lowercase seperti contoh input "`Nama saya Bagas`" akan diubah menjadi "`nama saya bagas`".

ii. Tokenization

Pada model BERT, tokenization yang digunakan adalah jenis WordPiece. Tokenization WordPiece dilakukan dengan membagi kata menjadi subkata, seperti contoh teks "`Makanan apa yang kamu suka? Saya suka makan makanan manis.`" akan menjadi token `[makan, ##an, apa, yang, kamu, suka, ?, saya, suka, makan, makan, ##an, manis, .]`. Tanda `##` menunjukan bahwa kata tersebut merupakan subkata dari kata.

iii. Token Spesial

BERT memiliki token spesial yaitu [CLS] (Classification), [SEP] (Separator) dan [PAD] (Padding). [CLS] terletak di awal token, [SEP] berfungsi untuk memisahkan token question dan token answer dan [PAD] berfungsi untuk menyamakan panjang teks pada setiap batch.

iv. Konversi ke ID

Token diubah menjadi id integer yang mewakili kata tersebut, seperti contoh teks "`Makanan apa yang kamu suka? Saya suka makan makanan manis.`" menjadi token `[[CLS], makan, ##an, apa, yang, kamu, suka, ?, [SEP], saya, suka, makan, makan, ##an, manis, ., [SEP]]`, dan kemudian diubah menjadi id token `[3, 3005, 4000, 2064, 1497, 3162, 4346, 35, 4, 1731, 4346, 2387, 3005, 3005, 4000, 7025, 18, 4]`.

v. Konversi ke ID

Setiap token memiliki panjang yang berbeda. Untuk menyamakan panjang token digunakan padding, yaitu menambahkan angka 0 (nol) di awal token (untuk padding left) atau di akhir token (untuk padding right). Pada kasus ini padding yang digunakan adalah padding right, sehingga angka 0 ditambahkan di akhir token seperti contoh teks 1: "`Siapa nama kamu? Nama saya adalah Bagas.`" dan teks 2: "`Makanan apa yang kamu suka? Saya suka makan makanan manis.`" diubah menjadi id token 1: `[3, 3476, 1966, 3162, 35, 4, 1966, 1731, 1581, 1663, 18, 4, 0, 0, 0, 0, 0, 0]` dan id token 2: `[3, 3005, 4000, 2064, 1497, 3162, 4346, 35, 4, 1731, 4346, 2387, 3005, 3005, 4000, 7025, 18, 4]`.

d. Pelatihan Model

Proses pelatihan model Question Answering dilakukan pada google colab dengan GPU T4. Pelatihan model menggunakan pre-trained indoBERT dengan model checkpoint indolem/indobert-base-uncased yang kemudian fine-tuning menggunakan data train dan validation yang telah diubah menjadi sequence token. Data train ditraining oleh framework BERT dan dilakukan validasi oleh data validation. Model BERT menerima input dataset berbentuk sequence token yang kemudian diproses oleh encoder BERT. Fine-tuning dilakukan dengan hyperparameter learning rate 1e-5, batch size 16 dan 32, training epoch 100, 3 dan 10. Learning rate dapat mempengaruhi kestabilan proses training. Learning rate 1e-5 menghasilkan skor akurasi yang baik terutama untuk dataset yang kecil [20]. Batch size yang umum digunakan pada BERT yaitu 16 dan 32 [7]. Batch size yang terlalu kecil dapat membuat proses training menjadi lebih lama tetapi lebih stabil, sedangkan batch size yang besar akan memakan lebih banyak resource. Oleh karena itu pada beberapa skenario akan digunakan batch size yang berbeda untuk memperlihatkan pengaruh batch size pada skor. Training epoch yang besar dapat membuat skor akurasi semakin baik, tetapi membutuhkan training yang lebih lama [20]. Epoch yang digunakan pada BERT umumnya antara 2 sampai 4 [7], maka pada salah satu skenario dipilih epoch 3 sebagai nilai tengah dari rekomendasi. Pada skenario yang lain, epoch 100 digunakan untuk mengetahui hasil pengujian model terhadap epoch yang sangat tinggi. Selain itu, epoch 10 juga digunakan pada skenario lain untuk mengetahui hasil pengujian pada epoch yang lebih tinggi dari rekomendasi, namun lebih rendah dari epoch tertinggi (100).

Output yang dihasilkan oleh model berupa nilai loss, start logits dan end logits. Nilai loss didapat hasil dari perhitungan jumlah CrossEntropyLoss untuk posisi start index dan end index. Nilai start logits dan end logits merepresentasikan kemungkinan posisi start index dan end index dari answer pada context.

e. Validasi

Tahap validasi dilakukan menggunakan data validation berupa sequence token dan diproses ke dalam model yang telah dibangun. Banyaknya kandidat answer yang akan dihasilkan diatur dengan parameter n_best_size 20, dan panjang teks answer diatur dengan parameter max_answer_length 30.

Setiap token yang dimasukkan ke dalam model BERT akan menghasilkan nilai start logits dan end logits. Nilai start logits dan end logits diubah menjadi skor start probabilitas dan end probabilitas dengan fungsi softmax. Skor start probabilitas sebanyak 20 data dipilih yang nilainya paling besar untuk menghasilkan answer terbaik, begitu juga dengan end probabilitas. Masing-masing nilai probabilitas start yang terpilih akan menjadi acuan untuk start index yang kemudian akan menentukan kata pertama dari answer. Begitu pula dengan end probabilitas, nilai terpilih akan menjadi acuan

untuk end index yang akan menentukan kata terakhir dari answer.

Nilai start probabilitas dijumlahkan dengan end probabilitas dibagi dua, sehingga menghasilkan nilai score. Nilai score ini diurutkan lagi dan kemudian dipilih satu data dengan nilai terbesar yang akan menjadi answer terbaik.

Ketentuan dalam memutuskan bahwa answer yang dihasilkan adalah valid yaitu jika end index lebih besar daripada start index; start index atau end index masuk ke dalam token question; answer tidak memiliki panjang nol; dan answer tidak melebihi batas max_answer_length.

f. Evaluasi

Exact Match adalah metode pengukuran yang bersifat true atau false. Metric akan menilai benar jika setiap kata pada hasil prediksi answer sama persis dengan kata pada ground-truth answer, selain daripada itu dianggap salah.

$$Exact\ Match = \frac{Total\ answer\ yang\ Exact\ Match}{Total\ answer} \quad (2)$$

F1-score adalah rata- rata harmonik dari precision dan recall. Pengukuran F1-score ditentukan oleh beberapa hal yaitu; True Positive (TP) adalah kelas positif yang diprediksi positif; True Negative (TN) adalah kelas negatif yang diprediksi negatif; False Positive (FP) adalah kelas negatif yang diprediksi positif; dan False Negative (FN) adalah kelas positif yang diprediksi negatif [21].

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (5)$$

Evaluasi dilakukan menggunakan skor Exact Match, F1-score, akurasi, presisi dan recall. Exact Match diperoleh dengan menghitung total answer prediksi yang sama persis dengan answer ground truth terhadap total answer yang dihasilkan model. Contoh answer yang dianggap Exact Match adalah jika answer prediksi: “nama saya bagas” dan ground truth: “nama saya bagas”. Contoh answer yang tidak Exact Match adalah jika answer prediksi: “bagas” dan ground truth: “nama saya bagas”. F1-score diperoleh dengan menghitung rata-rata harmonik dari precision dan recall.

Akurasi pada Sistem Question Answering sama dengan nilai Exact Match karena skor akurasi diperoleh dari total answer yang benar dibandingkan dengan total answer yang dihasilkan model. Answer dianggap benar jika answer prediksi sama persis terhadap answer pada ground truth.

Presisi diperoleh dengan menghitung total True Positive dibandingkan dengan total True Positive ditambah total False Positive. Nilai True Positive pada Sistem Question Answering adalah prediksi answer yang sama persis dengan ground truth. Sedangkan False Positive adalah answer prediksi yang tidak sama persis dengan answer ground truth.

Recall diperoleh dengan menghitung total True Positive dibandingkan dengan total True Positive ditambah total False Negative. Nilai False Negative pada Sistem Question Answering adalah prediksi answer yang menghasilkan string kosong (“”).

IV. HASIL DAN PEMBAHASAN

Percobaan dilakukan pada dataset validation sebanyak tiga skenario. Tiap skenario dilakukan running kode program tanpa henti sampai seluruh epoch tercapai. Skor hasil dari percobaan dapat dilihat pada Tabel 1.

TABEL 1
HASIL PERCOBAAN MODEL

Skenario	Batch Size	Learning rate	Epoch	EM	F1	Akurasi	Precision	Recall
1	16	1e-5	100	0.75	0.76	0.75	0.78	0.75
2	16	1e-5	3	0.78	0.69	0.78	0.61	0.78
3	16	1e-5	10	0.75	0.74	0.75	0.73	0.75

Pada skenario pertama, diperoleh hasil presisi 0,78, artinya prediksi answer yang cocok dengan ground truth sebesar 78% dari keseluruhan hasil answer yang dinyatakan benar. Hasil recall 0,75 artinya prediksi answer yang cocok dengan ground truth sebesar 75% dari hasil answer yang benar termasuk answer yang kosong. Hasil F1- score 0,76 yang artinya harmonik rata-rata dari presisi dan recall adalah 76%. Hasil Exact Match 0,75 artinya answer yang sama persis dengan ground truth adalah 75%. Hasil akurasi 0,75 artinya persentase total prediksi yang benar dibandingkan dengan total prediksi yang dibuat oleh model adalah 75%.

Pada skenario kedua, diperoleh hasil presisi 0.61 artinya prediksi answer yang cocok dengan ground truth sebesar 61% dari keseluruhan hasil answer yang dinyatakan benar. Hasil recall 0,78 artinya prediksi answer yang cocok dengan ground truth sebesar 78% dari hasil answer yang benar termasuk answer yang kosong. Hasil F1- score 0,69 yang artinya harmonik rata-rata dari presisi dan recall adalah 69%. Hasil Exact Match 0,78 artinya answer yang sama persis dengan ground truth adalah 78%. Hasil akurasi 0,78 artinya persentase total prediksi yang benar dibandingkan dengan total prediksi yang dibuat oleh model adalah 78%.

Pada skenario ketiga, diperoleh hasil presisi 0.73 artinya prediksi answer yang cocok dengan ground truth sebesar 73% dari keseluruhan hasil answer yang dinyatakan benar. Hasil recall 0,75 artinya prediksi answer yang cocok dengan ground truth sebesar 75% dari hasil answer yang benar termasuk answer yang kosong. Hasil F1- score 0,74 yang artinya harmonik rata-rata dari presisi dan recall adalah 74%. Hasil Exact Match 0,75 artinya answer yang sama persis dengan ground truth adalah 75%. Hasil akurasi 0,75 artinya persentase total prediksi yang benar dibandingkan dengan total prediksi yang dibuat oleh model adalah 75%.

A. Hasil Percobaan

Pada tiap skenario, hasil answer yang dihasilkan saat prediksi dapat dilihat pada Tabel 2.

TABEL 2
HASIL PREDIKSI ANSWER

Skenario	Context	Question	Ground truth	Prediksi Answer	Skor
1	Inilah 5 Cara Mudah dan Ampuh Mengatasi Kulit Kusam “Ada sejumlah cara mudah dan ampuh untuk mengatasi kulit kusam. Salah satunya adalah dengan membersihkan wajah secara	Apa yang dapat dilakukan untuk mengatasi kulit kusam?	Bersihkan Wajah secara Lembut.	**best answer:** adalah dengan membersihkan wajah secara	-
2				**best answer:** menghindari penggunaan scrub yang keras.” Halodoc, Jakarta – Kulit kusam menjadi salah satu masalah kulit yang umum terjadi dan dapat menyerang	0.0294

	lembut dengan menghindari ...			**worst answer:** penggunaan	0.0239
3	Spesialis Kulit Dermatitis, jerawat, kesehatan dan alergi kulit, infeksi jamur, herpes, bekas luka			**best answer:** membersihkan wajah secara lembut dengan menghindari	0.0569
				worst answer: sejumlah cara mudah dan ampuh untuk mengatasi kulit kusam. Salah satunya adalah dengan membersihkan wajah secara lembut dengan menghindari	0.0451

Percobaan pada setiap skenario menghasilkan beberapa contoh hasil answer dari prediksi dengan context dan question yang sama. Skor yang ditampilkan adalah skor probabilitas. Skor probabilitas diperoleh dari nilai logits yang diubah menjadi nilai probabilitas menggunakan fungsi aktivasi softmax. Nilai logits diperoleh dari layer terakhir pada BERT yang menunjukkan seberapa cocok suatu token menjadi kandidat answer. Pada skenario pertama, pengujian hanya menampilkan best answer tanpa menghasilkan skor probabilitas, dikarenakan keterbatasan resource saat pengujian.

Pada skenario kedua didapatkan hasil best answer dengan skor probabilitas sebesar 0,0294 dan worst answer dengan skor probabilitas 0,0239. Pada skenario ketiga didapatkan hasil best answer dengan skor probabilitas sebesar 0,0569 dan worst answer dengan skor probabilitas 0,0451. Skor probabilitas diperoleh dengan memproses nilai logits dengan fungsi aktivasi softmax. Pada skenario pertama, diperoleh hasil presisi 0,78, artinya prediksi

B. Analisis

Pada skenario satu dan dua, digunakan parameter batch_size yang sama yaitu 16 namun dengan jumlah epoch yang berbeda (100 dan 3). Skenario dua dengan epoch yang jauh lebih sedikit (3) menghasilkan skor Exact Match, akurasi dan recall sedikit lebih besar daripada skenario dengan perbedaan 3%. Akan tetapi skor F1 dan presisi skenario satu jauh lebih rendah daripada skenario dua dengan perbedaan masing-masing 7% dan 17%. Artinya semakin banyak epoch akan membuat model menghasilkan prediksi answer benar lebih baik, yang ditunjukkan dengan nilai presisi yang lebih tinggi. Skenario ketiga menggunakan parameter batch_size 32 dengan epoch sebanyak 10. Skor Exact Match, akurasi dan recall skenario ketiga sama dengan skenario pertama, tetapi skor F1 dan presisi yang sedikit lebih rendah dari skenario pertama dengan perbedaan masing-masing 2% dan 5%. Hasil skornya tidak berbeda jauh jika dibandingkan dengan skenario pertama yang menggunakan 100 epoch. Artinya dengan menambah batch_size akan menaikan skor model secara signifikan walaupun dengan epoch yang rendah.

V. KESIMPULAN

Hasil dari penelitian ini menunjukkan model skenario pertama dengan parameter batch_size 16 dan epoch 100 memberikan hasil skor terbaik yaitu Exact Match 75%, F1 76%, akurasi 75%, presisi 78% dan recall 75 %. Namun dengan menambah batch_size menjadi 32, bisa menurunkan jumlah epoch menjadi 10 untuk mengurangi resource pada proses training dan validation tetapi juga menghasilkan skor yang mendekati skenario terbaik. Untuk penelitian selanjutnya disarankan untuk menyusun dataset yang lebih baik dengan membagi kalimat context yang sangat panjang menjadi beberapa context terpisah supaya context tidak telalu panjang dan answer yang dihasilkan model sesuai dengan bahasa manusia.

REFERENSI

[1] K. A. Rizqo, "Resmi! Jokowi Umumkan Status Pandemi COVID-19 Dicabut," detikNews, 21 Juni 2023. [Online]. Available: https://news.detik.com/berita/d-6784804/resmi-jokowi-umumkan-status-pandemi-covid-19-dicabut. [Accessed 6 Januari 2025].

[2] L. N. Hakim, "DAMPAK PANDEMI WABAH CORONAVIRUS DISEASE (COVID) 19 DAN LOCKDOWN TERHADAP KESEHATAN MENTAL: KAJIAN PSIKOLOGI DAN AGAMA," *Kajian*, vol. 25, no. 2, pp. 161-177, 2020.

[3] A. Sixsmith, B. R. Horst, D. Simeonov and A. Mihailidis, "Older People's Use of Digital Technology During the COVID-19 Pandemic," *Bulletin of Science, Technology & Society*, vol. 42, no. 1-2, pp. 19-24, 2022.

[4] D. Vargo, L. Zhu, B. Benwell and Z. Yan, "Digital technology use during COVID-19 pandemic: A rapid review," *Human Behavior and Emerging Technologies*, vol. 3, no. 3, pp. 13-24, 2020.

[5] J. Moudy and R. A. Syakurah, "Pengetahuan terkait Usaha Pencegahan Coronavirus Disease (COVID-19) di Indonesia," *HIGEIA JOURNAL OF PUBLIC HEALTH RESEARCH AND DEVELOPMENT*, vol. 4, no. 2, pp. 333-346, 2020.

[6] F. Zhu, W. Lei, J. Zheng, S. Poria and T.-S. Chua, "Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering," *arXiv*, 2021.

[7] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018.

[8] F. Koto, A. Rahimi, J. H. Lau and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," in *COLING 2020 - The 28th International Conference on Computational Linguistics*, 2020.

[9] F. Dartiko, M. Yusa, A. Erlansari and S. A. Basha, "Comparative Analysis of Transformer-Based Method In A Question Answering System for Campus Orientation Guides," *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 8, no. 1, pp. 126-143, 2024.

[10] S. P. Widodo, "Comparative Analysis of Retriever and Reader for Open Domain Questions Answering

on BPS Knowledge in Indonesian," *Proceedings of The International Conference on Data Science and Official Statistics*, vol. 2023, no. 1, pp. 337-343, 2023.

[11] J. Bulian, C. Buck, W. Gajewski, B. Boerschinger and T. Schuster, "Tomayto, Tomahto. Beyond Token-level Answer Equivalence for Question Answering Evaluation," 2022.

[12] R. Alqifari, "Question Answering Systems Approaches and Challenges," in *Proceedings of the Student Research Workshop Associated with RANLP 2019*, 2019.

[13] B. S. Bhatt and H. A. Pandya, "Question Answering Survey: Directions, Challenges, Datasets, Evaluation," 2021.

[14] A. Rogers, O. Kovalera and A. Rumshisky, "A Primer in BERTology: What We Know About How BERT Works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842-866, 2020.

[15] D. Grießhaber, J. Maucher and N. T. Vu, "Fine-tuning BERT for Low-Resource Natural Language Understanding," in *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*, Barcelona, 2020.

[16] C. Qu, L. Yang, M. Qiu, W. B. Croft, Y. Zhang and M. Iyyer, "BERT with History Answer Embedding for Conversational Question Answering," in *SIGIR'19: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[17] B. v. Aken, B. Winter, A. Löser and F. A. Gers, "How Does BERT Answer Questions?: A Layer-Wise Analysis of Transformer Representations," in *CIKM '19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[18] D. Jurafsky and J. H. Martin, "Logistic Regression," in *Speech and Language Processing*, Pearson, 2024, pp. 81-104.

[19] B. Aldita and L. Alfansi, "Intention to Use Halodoc E-Health Services in Indonesia," *Frontiers in Business and Economics*, vol. 2, no. 2, pp. 117-125, 2023.

[20] R. Yunanto, E. P. Wibowo and R. Rianto, "A BERT MODEL TO DETECT PROVOCATIVE HOAX," *Journal of Engineering Science and Technology*, vol. 18, no. 5, pp. 2281-2297, 2023.

[21] J. A. Alzubi, R. Jain, A. Singh, P. Parwekar and M. Gupta, "COBERT: COVID-19 Question Answering System Using BERT," *Arabian Journal for Science and Engineering*, vol. 48, pp. 11003-110013, 2023.