

LAMPIRAN

✓ Install Packages

```
[ ] !pip install datasets
    !pip install rouge-score
    !pip install transformers==4.45.2
```



Tampilkan output tersembunyi

Lampiran 1. *Install Packages*

Lampiran 2. *Import Library (3)*

```
+ Kode + Teks
v Import Library

[ ] import transformers

[ ] print(transformers.__version__)
4.45.2

[ ] import os
import re
import json
import string
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from transformers import BertTokenizer, DistilBertTokenizer
```

Lampiran 3. *Import Library (2)*

```
[ ] def make_dataset(dataset_, filename):
    save_name = filename.split('/')[-1].split('.')[0]
    dataset_ = dataset_.sample(n=100, replace=True, random_state=1) # Get 90 random sample
    n_dataset = 0
    articles = []
    summaries = []
    len_data = []
    for i in range(len(dataset_)):
        len_ = len(dataset_['FULL'].iloc[i].split(' '))
        if len_ <= 512:
            len_data.append(int(len_))
            articles.append(dataset_['FULL'].iloc[i])
            summaries.append(dataset_['SUMMARY'].iloc[i])
            n_dataset += 1
            if n_dataset == 80: # Stop if 80
                break
    df_save = pd.DataFrame({'FULL':articles, 'SUMMARY':summaries, 'LEN': len_data})
    df_save.to_excel(f'{save_name}_CHECK.xlsx', index_label=None)
    return df_save
```

Lampiran 4. *Load Dataset (1)*

▼ Load Dataset

```
[ ] dataset_1 = pd.read_csv('/content/DATA UJI FULL.csv')
dataset_1 = dataset_1.dropna()
dataset_1 = make_dataset(dataset_1, '/content/DATA UJI FULL.csv')
len(dataset_1)
```

```
80
```

```
[ ] dataset_2 = pd.read_csv('/content/DATASET ARTIKEL POLITIK FULL.csv')
dataset_2 = dataset_2.dropna()
dataset_2 = make_dataset(dataset_2, '/content/DATASET ARTIKEL POLITIK FULL.csv')
len(dataset_2)
```

```
80
```

```
[ ] !tar -xvzf /content/drive/MyDrive/indosum.tar.gz -C /content/
```

```
indosum/
indosum/test.02.jsonl
indosum/README.txt
indosum/train.05.jsonl
indosum/train.04.jsonl
indosum/test.04.jsonl
indosum/train.01.jsonl
indosum/test.05.jsonl
indosum/test.03.jsonl
indosum/test.01.jsonl
indosum/dev.04.jsonl
indosum/dev.01.jsonl
indosum/dev.02.jsonl
indosum/train.02.jsonl
indosum/dev.03.jsonl
indosum/dev.05.jsonl
indosum/CHANGELOG.txt
indosum/train.03.jsonl
```

Lampiran 5. *Load Dataset (2)*

```
[ ] dataset_3 = []
for i in os.listdir('/content/indosum'):
    if i.endswith('.jsonl'):
        with open(f'/content/indosum/{i}') as file:
            for line in file.readlines():
                if i.split('.')[0] == 'train':
                    dataset_3.append(json.loads(line))

[ ] def flatten_paragraphs(list_paragraphs):
    list_sentences = []
    for paragraph in list_paragraphs:
        for sentence in paragraph:
            sent = ' '.join(sentence)
            list_sentences.append(sent)
    return list_sentences

[ ] def flatten_summaries(list_sentences):
    final_sentences = []
    for sentence in list_sentences:
        sent = ' '.join(sentence)
        final_sentences.append(sent)
    return final_sentences

[ ] for datum in dataset_3:
    datum['flatten_article'] = flatten_paragraphs(datum['paragraphs'])
    datum['flatten_summary'] = flatten_summaries(datum['summary'])

[ ] dataset_3 = pd.DataFrame({
    'SUMMARY': [' '.join(data['flatten_summary']) for data in dataset_3],
    'FULL': [' '.join(data['flatten_article']) for data in dataset_3]
})
```

Lampiran 6. Load Dataset (3)

```
[ ] dataset_3 = make_dataset(dataset_3, '/DATASET INDOSUM.xlsx')
    len(dataset_3)
```

```
80
```

```
[ ] df = pd.concat([dataset_1, dataset_2, dataset_3])
    df.drop(columns=['LEN'], inplace=True)
```

Lampiran 7. Dataset Processing and Split

▼ Dataset Preprocessing and Splitting

```
[ ] DOT_REGEX = r"(?<!\w)(?:[A-Z][A-Za-z]{,3}|[a-z]{1,2})\."
```

```
def clean_text(text):
    # 1. Noise Removal: Menghilangkan karakter
    text = re.sub(r'^\w\s.!?()', '', text)

    # 2. Sentence Tokenization: Membagi teks menjadi kalimat
    sentences = sent_tokenize(text)

    # 3. Word Tokenization: Membagi setiap kalimat menjadi kata
    words = [word_tokenize(sentence) for sentence in sentences]

    # 4. Remove punctuation marks: Menghapus tanda baca
    words = [[word for word in sentence if word not in string.punctuation] for sentence in words]

    # 5. Convert to lowercase: Mengubah teks menjadi huruf kecil
    text = text.lower()

    # 6. Normalize whitespace: Menggabungkan kata-kata kembali menjadi kalimat dengan spasi normal
    sentences = [' '.join(sentence) for sentence in words]
    text = ' '.join(sentences)
    return text
```

```
[ ] df['FULL'] = df['FULL'].apply(clean_text)
    df['SUMMARY'] = df['SUMMARY'].apply(clean_text)
```

```
[ ] df
```

Lampiran 8. BERT-SUM (1)

```
[ ] train_data, eval_data = train_test_split(df, test_size=0.2, random_state=42)
```

▼ BERT-SUM

```
[ ] import torch
    from transformers import EncoderDecoderModel, BertTokenizer, DistilBertTokenizer, DistilBertConfig
    from datasets import Dataset
```

```
[ ] # Initialize the model
    model = EncoderDecoderModel.from_pretrained("cahya/bert-indonesian-summarization")

    # Initialize the tokenizer
    tokenizer = BertTokenizer.from_pretrained("cahya/bert-indonesian-summarization")

    # Set special tokens
    model.config.eos_token_id = tokenizer.sep_token_id
    model.config.bos_token_id = tokenizer.cls_token_id
```

Lampiran 9. *BERT-SUM* (2)

```
[ ] def preprocess_function(examples):
    inputs = examples['FULL']
    targets = examples['SUMMARY']

    # Tokenize inputs
    model_inputs = tokenizer(inputs, max_length=512, truncation=True, padding='max_length')

    # Tokenize targets
    with tokenizer.as_target_tokenizer():
        labels = tokenizer(targets, max_length=128, truncation=True, padding='max_length')

    model_inputs['labels'] = labels['input_ids']
    return model_inputs
```

```
▶ train_df = Dataset.from_pandas(train_data)
train_df = train_df.map(preprocess_function, batched=True)
eval_df = Dataset.from_pandas(eval_data)
eval_df = eval_df.map(preprocess_function, batched=True)
```

 [Tampilkan output tersembunyi](#)

```
[ ] os.environ["WANDB_DISABLED"] = "true"
```

```
[ ] from transformers import TrainingArguments
```

```
training_args = TrainingArguments(
    report_to=None,
    output_dir="./results",
    evaluation_strategy="epoch",
    learning_rate=5e-5,
    per_device_train_batch_size=8,
```

Lampiran 10. *BERT-SUM* (3)

```
[ ]     per_device_train_batch_size=8,
        per_device_eval_batch_size=8,
        num_train_epochs=100,
        weight_decay=0.01,
        save_total_limit=2,
        logging_dir="./logs",
        logging_steps=10,
        logging_first_step=True
    )
```

 [Tampilkan output tersembunyi](#)

```
▶ from transformers import Trainer
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_df,
    eval_dataset=eval_df,
    tokenizer=tokenizer,
)
```

```
trainer.train()
```

 [Tampilkan output tersembunyi](#)

```
[ ] metrics = trainer.evaluate()
    print(metrics)
```

Lampiran 11. *BERT-SUM* (4)

```
[ ] from transformers import BertTokenizer, EncoderDecoderModel

checkpoint_dir = "/content/results/checkpoint-2400"

# Load tokenizer and model from the checkpoint directory
tokenizer_bert = BertTokenizer.from_pretrained(checkpoint_dir)
model_bert = EncoderDecoderModel.from_pretrained(checkpoint_dir)

def preprocess_input(text, tokenizer):
    return tokenizer(text, return_tensors="pt", max_length=512, truncation=True, padding="max_length")
def generate_summary(inputs, model, tokenizer):
    summary_ids = model.generate(
        input_ids=inputs['input_ids'],
        attention_mask=inputs['attention_mask'],
        max_length=150,
        num_beams=4,
        early_stopping=True
    )
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return summary

[ ]

def summary_result(x):
    inputs = preprocess_input(x, tokenizer_bert)
    summary = generate_summary(inputs, model_bert, tokenizer_bert)
    return summary
```

Lampiran 12. *BERT-SUM* (5)

```
[ ] dataset_1['BERT_RESULT'] = dataset_1['FULL'].apply(summary_result)
dataset_2['BERT_RESULT'] = dataset_2['FULL'].apply(summary_result)
dataset_3['BERT_RESULT'] = dataset_3['FULL'].apply(summary_result)

▼ DistilBERT

[ ] # Initialize the model
model = EncoderDecoderModel.from_pretrained("Shobhank-iiitdwd/Distil_BERT_summary")

# Initialize the tokenizer
tokenizer = DistilBertTokenizer.from_pretrained("Shobhank-iiitdwd/Distil_BERT_summary")

# Set special tokens
model.config.eos_token_id = tokenizer.sep_token_id
model.config.bos_token_id = tokenizer.cls_token_id
```

Lampiran 13. *DISTILBERT* (1)

```
[ ] def preprocess_function(examples):
    inputs = examples['FULL']
    targets = examples['SUMMARY']

    # Tokenize inputs
    model_inputs = tokenizer(inputs, max_length=512, truncation=True, padding='max_length')

    # Tokenize targets
    with tokenizer.as_target_tokenizer():
        labels = tokenizer(targets, max_length=128, truncation=True, padding='max_length')

    model_inputs['labels'] = labels['input_ids']
    return model_inputs

train_df = Dataset.from_pandas(train_data)
train_df = train_df.map(preprocess_function, batched=True)

eval_df = Dataset.from_pandas(eval_data)
eval_df = eval_df.map(preprocess_function, batched=True)
```

 [Tampilkan output tersembunyi](#)

Lampiran 14. *DISTILBERT* (2)

```

▶ from transformers import TrainingArguments

training_args = TrainingArguments(
    report_to=None,
    output_dir="./results_distil",
    evaluation_strategy="epoch",
    learning_rate=5e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=100,
    weight_decay=0.01,
    save_total_limit=2,
    logging_dir="./logs",
    logging_steps=10,
    logging_first_step=True
)

Tampilkan output tersembunyi

[ ] os.environ["WANDB_DISABLED"] = "true"

[ ] from transformers import Trainer

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_df,
    eval_dataset=eval_df,
    tokenizer=tokenizer,
)

```

Lampiran 15. *DISTILBERT* (3)

```

[ ] )
trainer.train()

Tampilkan output tersembunyi

▶ from transformers import BertTokenizer, EncoderDecoderModel

checkpoint_dir = "/content/results_distil/checkpoint-2400"

# Load tokenizer and model from the checkpoint directory
tokenizer = BertTokenizer.from_pretrained(checkpoint_dir)
model = EncoderDecoderModel.from_pretrained(checkpoint_dir)
model.to('cuda')

Tampilkan output tersembunyi

[ ] def preprocess_input(text, tokenizer):
    return tokenizer(text, return_tensors="pt", max_length=512, truncation=True, padding="max_length")
def generate_summary(inputs, model, tokenizer):
    summary_ids = model.generate(
        input_ids=inputs['input_ids'].to('cuda'),

        attention_mask=inputs['attention_mask'].to('cuda'),
        max_length=150,
        num_beams=4,
        early_stopping=True
    )
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
    return summary

```

Lampiran 16. *DISTILBERT* (4)

```
[ ] dataset_1['DISTILBERT_RESULT'] = dataset_1['FULL'].apply(summary_result)
dataset_2['DISTILBERT_RESULT'] = dataset_2['FULL'].apply(summary_result)
dataset_3['DISTILBERT_RESULT'] = dataset_3['FULL'].apply(summary_result)
```

```
[ ] df_30 = df.head(30)
df_30.to_csv('result_30.csv', index=False)
```

Lampiran 17. *ROUGE* Score (1)

▼ Rouge Score

```
from rouge_score import rouge_scorer

def compute_rouge_scores(original, generated):
    scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)
    scores = scorer.score(original, generated)
    return scores

dataset_1['BERT_ROUGE'] = dataset_1.apply(lambda row: compute_rouge_scores(row['SUMMARY'], row['BERT_RESULT']), axis=1)
dataset_1['DISTILBERT_ROUGE'] = dataset_1.apply(lambda row: compute_rouge_scores(row['SUMMARY'], row['DISTILBERT_RESULT']), axis=1)

dataset_2['BERT_ROUGE'] = dataset_2.apply(lambda row: compute_rouge_scores(row['SUMMARY'], row['BERT_RESULT']), axis=1)
dataset_2['DISTILBERT_ROUGE'] = dataset_2.apply(lambda row: compute_rouge_scores(row['SUMMARY'], row['DISTILBERT_RESULT']), axis=1)

dataset_3['BERT_ROUGE'] = dataset_3.apply(lambda row: compute_rouge_scores(row['SUMMARY'], row['BERT_RESULT']), axis=1)
dataset_3['DISTILBERT_ROUGE'] = dataset_3.apply(lambda row: compute_rouge_scores(row['SUMMARY'], row['DISTILBERT_RESULT']), axis=1)

[ ] def average_rouge_scores(df, column):
    rouge1_precision = []
    rouge1_recall = []
    rouge1_fmeasure = []
    rouge2_precision = []
    rouge2_recall = []
    rouge2_fmeasure = []
    rougeL_precision = []
    rougeL_recall = []
    rougeL_fmeasure = []
```

Lampiran 18. *ROUGE* Score (2)

```
for score in df[column]:
    rouge1_precision.append(score['rouge1'].precision)
    rouge1_recall.append(score['rouge1'].recall)
    rouge1_fmeasure.append(score['rouge1'].fmeasure)
    rouge2_precision.append(score['rouge2'].precision)
    rouge2_recall.append(score['rouge2'].recall)
    rouge2_fmeasure.append(score['rouge2'].fmeasure)
    rougeL_precision.append(score['rougeL'].precision)
    rougeL_recall.append(score['rougeL'].recall)
    rougeL_fmeasure.append(score['rougeL'].fmeasure)

return {
    'rouge1': {
        'precision': sum(rouge1_precision) / len(rouge1_precision),
        'recall': sum(rouge1_recall) / len(rouge1_recall),
        'fmeasure': sum(rouge1_fmeasure) / len(rouge1_fmeasure),
    },
    'rouge2': {
        'precision': sum(rouge2_precision) / len(rouge2_precision),
        'recall': sum(rouge2_recall) / len(rouge2_recall),
        'fmeasure': sum(rouge2_fmeasure) / len(rouge2_fmeasure),
    },
    'rougeL': {
        'precision': sum(rougeL_precision) / len(rougeL_precision),
        'recall': sum(rougeL_recall) / len(rougeL_recall),
        'fmeasure': sum(rougeL_fmeasure) / len(rougeL_fmeasure),
    }
}

[ ] dataset_1_avg_bert_rouge = average_rouge_scores(dataset_1, 'BERT_ROUGE')
dataset_1_avg_dbert_rouge = average_rouge_scores(dataset_1, 'DISTILBERT_ROUGE')
```

Lampiran 19. *ROUGE* Score (3)

```

dataset_1_avg_bert_rouge = average_rouge_scores(dataset_1, 'BERT_ROUGE')
dataset_1_avg_dbert_rouge = average_rouge_scores(dataset_1, 'DISTILBERT_ROUGE')

dataset_2_avg_bert_rouge = average_rouge_scores(dataset_2, 'BERT_ROUGE')
dataset_2_avg_dbert_rouge = average_rouge_scores(dataset_2, 'DISTILBERT_ROUGE')

dataset_3_avg_bert_rouge = average_rouge_scores(dataset_3, 'BERT_ROUGE')
dataset_3_avg_dbert_rouge = average_rouge_scores(dataset_3, 'DISTILBERT_ROUGE')

```

```

[ ] def print_rouge_scores(avg_scores, model_name):
    print(f"Average ROUGE Scores for {model_name}:")
    for rouge, scores in avg_scores.items():
        print(f"\n{rouge.upper():}")
        print(f"Precision: {scores['precision']:.4f}")
        print(f"Recall: {scores['recall']:.4f}")
        print(f"F1 Score: {scores['fmeasure']:.4f}")

```

```

▶ print('Dataset 1 Evaluation')
print()
print_rouge_scores(dataset_1_avg_bert_rouge, "BERT")
print()
print_rouge_scores(dataset_1_avg_dbert_rouge, "DISTILBERT")

```

 Tampilkan output tersembunyi

Lampiran 20. *ROUGE* Score (4)

```

[ ] print('Dataset 2 Evaluation')
print()
print_rouge_scores(dataset_2_avg_bert_rouge, "BERT")
print()
print_rouge_scores(dataset_2_avg_dbert_rouge, "DISTILBERT")

```

 Tampilkan output tersembunyi

```

▶ print('Dataset 3 Evaluation')
print()
print_rouge_scores(dataset_3_avg_bert_rouge, "BERT")
print()
print_rouge_scores(dataset_3_avg_dbert_rouge, "DISTILBERT")

```

 Tampilkan output tersembunyi

[] Mulai coding atau buat kode dengan AI.

Lampiran 21. Jumlah Token Dataset

	A	B	C
		FULL	LEN
1			
2	0	Jakarta, CNN Indonesia - - Salah satu 'anak' Marvel juga menjadi 'korban' penembakan Las Vegas. Netflix dan Marvel memutuskan untuk tid	414
3	1	BANDUNG, JUARA.net - Salah satu legenda Persib, Dede Iskandar, menyarankan agar manajemen Persib secepatnya mencari pelatih baru per	200
4	2	Rimanews - Penyidik Polda Metro Jaya mengungkapkan lima tersangka dugaan upaya makar berencana menggelar aksi unjuk rasa pada lima kota	247
5	3	Pernahkah kamu mengalami sariawan? Sebagian besar orang tentunya pernah mengalami gangguan mulut yang satu ini. Meski tampak sepele, nan	467
6	4	Jakarta, CNN Indonesia - - Persib Bandung bersiap menghadapi Persija Jakarta pada pertandingan lanjutan Liga 1 Indonesia 2017 di Stadion Gel	308
7	5	Jakarta (ANTARA News) - Sebagai sebuah film spin-off pertama dari seri Star Wars, Rogue One memperkenalkan pada khalayak sederet kar	470
8	6	Suara.com - Apakah Anda berpikir tentang pindah lokasi pekerjaan yang jauh dari tempat tinggal Anda saat ini? Tentunya banyak alasan mengapa	462
9	7	Jakarta, CNN Indonesia - - Kriminolog Universitas Indonesia Josias Simon mengatakan penggunaan alat kejut listrik atau stun gun sangat efektif ur	329
10	8	JUARA.NET - Petenis putra Spanyol, Rafael Nadal, meragukan dirinya bisa kembali ke turnamen musim depan setelah dia membatalkan sebuah l	213
11	9	Dua legiun asing PSM Makassar, Marc Klok dan Wiljan Phlim, memastikan tetap bertahan di Juku Eja pada musim depan, kendati mendapat tav	412
12	10	Presiden Joko Widodo turut memberikan perhatian kepada fenomena persekusi yang saat ini muncul. Ia meminta persekusi atau main hakim sendir	430
13	11	Merdeka.com - Peneliti Pusat Kajian Anti korupsi Universitas Gadjah Mada Zaenal Arifin Mochtar mengatakan Pemerintah, DPR, serta Komisi F	286
14	12	Kebakaran hebat terjadi di sebuah mal di kota Davao, Filipina. Pejabat setempat pada Minggu (24/12) mengatakan sedikitnya 37 orang tewas	234
15	13	Suara.com - Pemerintah Indonesia siap bekerja sama dengan pemerintah Inggris, tidak terganggu reshuffle kabinet negara itu, khususnya dalam rar	302
16	14	JAKARTA- Libur panjang Imlek, sebagian masyarakat memanfaatkannya untuk mudik. Di Terminal Kalideres, Jakarta Barat sejak Kamis (15/	219
17	15	Direktorat Gratifikasi KPK sudah menerima laporan 2 kuda berjenis Sandelwood dari Presiden Joko Widodo. Kuda yang didapatkan Jokowi pac	170
18	16	Jakarta, CNN Indonesia - - Isyana Sarasvati resmi merilis album ke-duanya pada Jumat (1/9) ini. Album yang diberi judul 'Paradox' itu mela	336
19	17	BANDUNG, JUARA.net - Tim voli putra Jakarta BNI Taplus dipastikan menempati peringkat ketiga setelah mengalahkan Jakarta Pertamina pad	435
20	18	Jakarta, CNN Indonesia - - Partai Keadilan Sejahtera (PKS) memberikan sinyal kuat akan turut mengusung Saifullah Yusuf atau Gus Ipul pada P	344
21	19	BPOM menyatakan 4 produk mi instan asal Korea, positif mengandung babi. Yaitu Samyang (U-Dong), Samyang (Kimchi), Nongshim, dan	476
22	20	KORFA SFI ATAN - Mamoo, dibarengi mengalami kecelakaan mobil pada Rabu (13/9/2017) waktu Korea Selatan. Menuut agensi E	171

Lampiran 22. Result

result - Excel

File Home Insert Page Layout Formulas Data Review View Developer Help Tell me what you want to do

Clipboard Font Alignment Number Conditional Formatting Styles Cell Styles Insert Delete Cell

C3 : aleix espargaro, pembalap aprilia racing, baru - baru ini mengungkapkan keprihatinannya terhadap tenaga yang terlalu besar di k

	A	B	C	D
1	FULL	SUMMARY	BERT_RESULT	DISTILBERT_RESULT
	<p>mengumumkan bercerai setelah menikah selama nyaris dua tahun. kabar itu dikonfirmasi oleh jiyeon t-ara melalui kuasa hukumnya, law firm taesung. diberitakan korea joongang daily, minggu(6/10), pasangan yang resmi menikah pada akhir 2022 itu memutuskan berpisah karena ada perbedaan yang tidak dapat diatasi. advertisement scroll to continue with content"kedua belah pihak tidak dapat mengatasi perbedaan mereka, lalu mereka mengajukan mediasi perceraian di pengadilan keluarga seoul setelah menyetujui perceraian," ujar choi yu-na dari law firm taesung."kami dengan tulus meminta anda untuk tidak membuat laporan provokatif maupun komentar spekulatif," lanjutnya. jiyeon kemudian ikut buka suara tentang kabar perceraian itu. ia semula meminta maaf soal berita perpisahan dirinya dengan mantan suami yang muncul mendadak tersebut. ia kemudian mengonfirmasi sudah mengajukan perantara dari pengadilan, sehingga langkah perceraian itu tidak diikuti dengan tuntutan hukum terhadap satu sama lain. jiyeon mengakhiri surat terbuka itu dengan berjanji akan menjadi seseorang yang lebih baik lagi pada masa mendatang."saya ingin meminta maaf karena menyapa kalian dengan berita yang tidak menyenangkan ini. kami senekat mungkin melanjutkan</p>	<p>jiyeon t - ara dan atlet bisbol hwang jae mengumumkan bercerai setelah hampir dua tahun menikah. kabar ini dikonfirmasi oleh jiyeon t-ara melalui kuasa hukumnya, law firm taesung. mereka memutuskan berpisah karena perbedaan yang tak dapat diatasi dan telah menjalani mediasi perceraian di pengadilan keluarga seoul. jiyeon meminta maaf atas berita perceraian ini dan mengonfirmasi bahwa mereka akan melanjutkan mediasi perceraian tanpa tuntutan hukum. perceraian terjadi beberapa bulan sebelum pernikahan mereka berusia 2 tahun. jiyeon bergabung dengan t-ara sejak 2009 dan aktif dalam dunia akting. sedangkan hwang jae-gyun adalah atlet bisbol profesional korea selatan sejak 2007. mereka menikah pada desember 2022 setelah mengumumkan hubungan mereka pada februari tahun itu. perceraian mereka dikonfirmasi setelah rumor tak sedap beredar, yang semakin meningkat ketika hwang jae-gyun ditemukan di bar hingga larut malam. meski telah mengajukan mediasi cerai, jiyeon berjanji untuk menjadi pribadi yang lebih baik di masa depan dan berharap pengertian dari publik. rumor perceraian mereka belum terbukti sejak</p>	<p>jiyeon t - ara dan hwang jae - gyun mengumumkan bercerai setelah hampir dua tahun menikah. kabar ini dikonfirmasi oleh jiyeon o - ara melalui kuasa hukumnya, law firm taesung. mereka memutuskan berpisah karena perbedaan yang tak dapat diatasi dan telah menjalani mediasi perceraian di pengadilan keluarga seoul. jiyeon meminta maaf atas berita perceraian ini dan mengonfirmasi bahwa mereka akan melanjutkan mediasi bercerai tanpa tuntutan hukum. perceraian terjadi beberapa bulan sebelum pernikahan mereka berusia 2 tahun. jiaru bergabung dengan t - e sejak 2009 dan aktif dalam dunia akting. sedangkan hwang jaeni adalah atlet bisbol profesional korea selatan sejak 2007. mereka menikah pada desember 2022.</p>	<p>jiyeon t - ara dan atlet bisbol hwang jae - gyun mengumumkan bercerai setelah hampir dua tahun menikah. kabar ini dikonfirmasi oleh jiyeon's memutuskan berpisah karena perbedaan yang tak dapat diatasi. kemudian meminta maaf atas berita perpisahan dirinya dengan, sementara spekulatif "" kedua belah pihak tidak membuat laporan provokatif maupun"</p>

Sheet1

Ready Accessibility: Good to go

Type here to search

Lampiran 23. Hasil Ringkasan Peringkask

1	Link Artikel	Hasil Ringkasan Manusia
2	https://www.cnnindonesia.com/olahraga/20241008101102-142-1152773/skuad-final-bahrain-vs-indonesia-wonderkid-villarreal-u-17-masuk	<p>Skuad final Bahrain vs Indonesia: Wonderkid Villarreal U-17, Sayed Ahmed Alwadaei, Terpilih Timnas Bahrain telah mengumumkan skuad final 28 pemain untuk melawan Timnas Indonesia dalam Kualifikasi Piala Dunia 2026. Sayed Ahmed Alwadaei, wonderkid Villarreal U-17, berhasil masuk ke skuad tersebut. Pelatih timnas Bahrain, Dragan Talajic, memilih pemain-pemain ini untuk pertandingan Grup C melawan Indonesia dan Arab Saudi.</p> <p>Alwadaei adalah pemain termuda dalam skuad Bahrain, yang bermain untuk tim Villarreal U-17 di Spanyol. Talajic meminta Alwadaei untuk ikut serta demi memberikan pengalaman bermain bagi pemain muda ini. Di samping itu, pemain lain dalam skuad termasuk dua naturalisasi dari Nigeria, yakni Vincent Emmanuel dan Moses Atede, serta pemain veteran seperti Sayed Dhiya Saeed, Kamil Al Aswad, dan Ismail Abdullatif.</p> <p>Daftar pemain timnas Bahrain untuk pertandingan melawan Indonesia termasuk kiper, bek, pemain tengah, dan penyerang. Hal ini menandai persiapan yang serius dari timnas Bahrain untuk pertandingan kualifikasi ini.</p>



DAFTAR RIWAYAT HIDUP



Elma Margaretha, penulis skripsi berjudul *“Peringkasan Teks Otomatis Menggunakan Metode Bidirectional Encoder Representations from Transformers (BERT) pada Artikel Berita Politik Berbahasa Indonesia”*, lahir di Tigabinanga pada 15 Maret 2001. Ia merupakan putri pertama dari pasangan Bapak Lit Daniel Sebayang dan Ibu Mardiana Br Sitepu. Penulis beragama Katolik, dan saat ini kedua orang tua penulis berdomisili di Desa

Tigabinanga, Kecamatan Tigabinanga, Kabupaten Karo, Sumatera Utara. Pendidikan formalnya dimulai di SD Sint Yosep Tigabinanga, yang diselesaikan pada tahun 2013. Kemudian, penulis melanjutkan ke SMP Negeri 1 Tigabinanga dan lulus pada tahun 2016. Pendidikan menengah atas ditempuh di SMA Negeri 1 Kabanjahe, dengan jurusan Ilmu Pengetahuan Alam, dan berhasil lulus pada tahun 2019. Saat ini, penulis sedang menempuh pendidikan tinggi di Universitas Pendidikan Ganesha, pada Program Studi Ilmu Komputer, Jurusan Teknik Informatika.

