

## DAFTAR PUSTAKA

- Allahyari, M., Trippe, E. D., & Gutierrez, F. (2017). *Text summarization techniques: A brief survey*. *arXiv preprint arXiv:1707.02268*. Retrieved from <https://arxiv.org/abs/1707.02268>
- Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). *Layer normalization*. *arXiv preprint arXiv:1607.06450*. Retrieved from <https://arxiv.org/abs/1607.06450>
- Desjardins, J. (2022). Data never sleeps 10.0. *Visual Capitalist*. Retrieved from <https://www.visualcapitalist.com/data-never-sleeps-10-0/>
- Edmunds, A., & Morris, A. (2000). The problem of information *overload* in business organizations: A review of the literature. *International Journal of Information Management*, 20(1), 17-28. [https://doi.org/10.1016/S0268-4012\(99\)00051-1](https://doi.org/10.1016/S0268-4012(99)00051-1)
- Etemad, A., Abidi, S. R., & Chhabra, A. (2021). Hybrid *attention*-based pointer-generator *networks* for summarizing long *texts*. *arXiv preprint arXiv:2105.01643*. Retrieved from <https://arxiv.org/abs/2105.01643>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual *learning* for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). <https://doi.org/10.1109/CVPR.2016.90>
- Itsnaini, Q. A., Hayaty, M., Putra, A. D., & Jabari, N. A. M. (2023). *Abstractive text summarization using pre-trained language model Text-to-text transfer transformer*. *arXiv preprint arXiv:2301.00001*. Retrieved from <https://arxiv.org/abs/2301.00001>
- Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing* (3rd ed.). Pearson.
- Kurniawan, K., & Louvan, S. (2018). IndoSum: A New Benchmark Dataset for Indonesian Text Summarization. 2018 International Conference on Asian Language Processing (IALP), 215-220. <https://arxiv.org/abs/1810.05334v5>
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: *Denoising* sequence-to-sequence *pre-training* for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7871-7880). <https://doi.org/10.18653/v1/2020.acl-main.703>
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511809071>
- Mitchell, T. M. (2018). *Machine learning* (2nd ed.). McGraw Hill Education.

- Moratanch, K., & Chitrakala, S. (2016). A survey on *extractive text* summarization. In *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)* (pp. 1-7). IEEE. <https://doi.org/10.1109/ICCPCT.2016.7530193>
- Pal, A., Kanupriya, Fan, L., & Igodifo, V. (2024). *Text Summarization using BERT and T5*. University College London. Retrieved from [https://anjali001.github.io/Project\\_Report.pdf](https://anjali001.github.io/Project_Report.pdf)
- Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of *machine translation*. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318). <https://doi.org/10.3115/1073083.1073135>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). *Pytorch: An imperative style, high-performance deep learning library*. *Advances in Neural Information Processing Systems*, 32, 8024-8035. Retrieved from <https://arxiv.org/abs/1912.01703>
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2019). Exploring the limits of *transfer learning* with a unified *text-to-text transformer*. *arXiv preprint arXiv:1910.10683*. Retrieved from <https://arxiv.org/abs/1910.10683>
- Radev, D. R., Hovy, E., McKeown, K., & Hovy, E. (2020). Introduction to the Special Issue on Summarization. *Computational Linguistics*, 28(4), 399-408. <https://doi.org/10.1162/089120102762671927>
- Richardson, M. (2013). Beautiful Soup Documentation. Retrieved from <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator *networks*. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1073-1083). <https://doi.org/10.18653/v1/P17-1099>
- Sposteru, M. (2022). Information *Overload* and the Digital Age. *Journal of Digital Information Management*, 20(1), 45-53. <https://doi.org/10.1111/j.1365-2575.2022.tb01702.x>
- Tobergte, D. R., & Curtis, S. (2023). Online News Consumption: Trends and Insights. *International Journal of Digital Society*, 14(2), 198-207. <https://doi.org/10.1504/IJDS.2023.100424>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention* is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998-6008). <https://arxiv.org/abs/1706.03762>

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google's *neural machine* translation system: Bridging the gap between human and *machine* translation. *arXiv preprint arXiv:1609.08144*. Retrieved from <https://arxiv.org/abs/1609.08144>

Zolotareva, E. (2020). Analysis of *attention* mechanisms in *neural networks* for *text* summarization. *Journal of Artificial Intelligence Research*, 69, 551-589. <https://doi.org/10.1613/jair.1.12317>

## **LAMPIRAN**

## Lampiran 1 Dataset

```
def format_tanggal(range):
    test_date = datetime.datetime.strptime(f"30-{range}-2023", "%d-%m-%Y")
    K = 7
    date_generated = pd.date_range(test_date, periods=K)
    return date_generated

with open('berita_detik.csv', 'a') as file:
    wr = csv.writer(file, delimiter=',')
    wr.writerow(['title', 'released', 'url', 'content'])
    count = 0
    for i in range(10,11):
        date_generate = format_tanggal(i)
        formattgl = date_generate.strftime("%m/%d/%Y")
        for tgl in formattgl:
            url = f"https://news.detik.com/indeks?date={tgl}"
            ge = req.get(url).text
            sop = bs(ge, 'lxml')
            try:
                paging = sop.find_all('div', 'pagination text-center mgt-16 mgb-16')[0].find_all('a')[-2]
                last_page = paging.text
            except:
                last_page = 1
            for page in range(1, int(last_page)+1):
                try:
                    url_ = f"https://news.detik.com/indeks/{page}?date={tgl}"
                    ge = req.get(url_).text
                    sop_ = bs(ge_, 'lxml')
                    contai = sop_.find_all('div', class_='media_text')
                    for x in contai:
                        time = x.find('div', class_='media__date').find('span')['title'].split()
                        if time[2].lower() == 'okt':
                            time[2] = 10
                        elif time[2].lower() == 'nov':
                            time[2] = 11

                        tgl_ = f"{time[1]}/{time[2]}/{time[3]} {time[4]} {time[5]}"
                        title = x.find('a').text
                        link = x.find('a')['href']
                        ge_a = req.get(f"{link}?single=1").text
                        sop_a = bs(ge_a, 'lxml')
                        content = sop_a.find_all('div', class_='detail__body-text itp_bodycontent')
                        for i in content:
                            isi = [x.text for x in i.find_all('p')]
                            ressi = ''.join(isi).replace('\n', '').replace('ADVERTISEMENT', '').replace('SCROLL TO CONTINUE WITH CONTENT', '')
                            try:
                                wr.writerow([title, tgl_, link, ressi])
                                count += 1
                                print(f'[{count}][{tgl}][{title}]')
                            except:
                                pass
            except: pass
```

### (a) Cuplikan *scraping* data

| index | summary                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | paragraphs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0     | Ketua MPR Zulkifli Hasan menyesalkan kisruh yang terjadi antara pelaku sarana transportasi online dan tradisional. Zulkifli menyarankan adanya pertemuan bersama antara pemerintah, pelaku transportasi online dan transportasi tradisional demi meredam kisruh yang masih belum terselesaikan. Zulkifli menilai aturan yang dikeluarkan pemerintah, pelaku transportasi online dan transportasi tradisional demi meredam kisruh yang masih belum terselesaikan. Zulkifli menilai aturan yang dikeluarkan pemerintah seharusnya tidak hanya membahas tarif tapi juga mekanisme yang dapat menguntungkan semua pihak, baik pelaku transportasi online maupun tradisional.                                                                                                                                                                                                                          | Ketua MPR Zulkifli Hasan menyesalkan kisruh yang terjadi antara pelaku sarana transportasi online dan tradisional. Zulkifli menyarankan adanya pertemuan bersama antara pemerintah, pelaku transportasi online dan transportasi tradisional demi meredam kisruh yang masih belum terselesaikan. Zulkifli menilai aturan yang dikeluarkan pemerintah seharusnya tidak hanya membahas tarif tapi juga mekanisme yang dapat menguntungkan semua pihak, musyawarah, duduk bareng kemudian dibuat aturan yang saling menguntungkan. Kan bisa saling melengkapi, negara lain bisa masa kita enggak bisa, " ucap Zulkifli di Gedung DPR, Senayan, Jakarta Pusat, Senin (27/3). Baca juga: Setya Novanto: Jangan Sampai Kisruh Taksi dan Ojek Online Jadi Besar Ketua Umum PAN menambahkan bahwa hal ini harus diatur karena menyangkut mata pencaharian dari masyarakat itu sendiri. Menurut dia, titik temu masih bisa dicari. Sebab, transportasi online tradisional masih memiliki pasar sendiri dan transportasi tradisional juga memiliki pasarnya sendiri. " Yang tradisional tentu akan ada pasar sendiri, yang online punya pasar sendiri. Nah itulah segera ditata kalau tidak tentu terjadi bentrokan - bentrokan. Itu namanya urusan perut, urusan rezeki tentu kalau tidak ditata dengan baik akan menimbulkan keresahan, " lanjut Zulkifli. Zulkifli menilai bahwa tidak mungkin bila keduanya bertarung bebas. Diperlukan adanya aturan yang membahas hal tersebut. " Itu mesti diatur kan kalau semua tarung bebas kan enggak mungkin. Kalau semua dibiarkan tarung bebas atau disebut pasar bebas itu tentu nanti yang tradisional akan teriak, akan menjerit, " ujar Zulkifli. Baca juga: Sebelum 1 April Sudah Ada Ketetapan Tarif Taksi Online di Jakarta |
| 1     | Cerita sekuel terbaru James Bond bocor. Menurut sumber yang terlibat dalam produksi film ini, agen rahasia 007 berhenti menjadi mata-mata Inggris demi menikah dengan perempuan yang dicintainya. " Bond berhenti menjadi agen rahasia karena jatuh cinta dan menikah dengan perempuan yang dicintainya. " tutur seorang sumber yang dekat dengan produksi seperti dikutip laman PageSix.com. Dalam film tersebut, Bond diduga menikahi Madeleine Swann yang diperankan oleh Lea Seydoux. Lea diketahui bermain sebagai gadis Bond di sekuel Spectre pada 2015 silam. Jika benar, ini merupakan sat satunya sekuel yang bercerita pernikahan James Bond sejak 1969. Sebelumnya, di sekuel On Her Majesty, James Bond menikahi Tracy Draco yang diperankan Diana Rigg. Namun, di film itu Draco terbunuh. Plot sekuel film James Bond ke - 25 bocor tak lama setelah Daniel Craig mengumumkan baka | Suara.com - Cerita sekuel terbaru James Bond bocor. Menurut sumber yang terlibat dalam produksi film ini, agen rahasia 007 berhenti menjadi mata-mata Inggris demi menikah dengan perempuan yang dicintainya. " Bond berhenti menjadi agen rahasia karena jatuh cinta dan menikah dengan perempuan yang dicintai, " tutur seorang sumber yang dekat dengan produksi seperti dikutip laman PageSix.com. Dalam film tersebut, Bond diduga menikahi Madeleine Swann yang diperankan oleh Lea Seydoux. Lea diketahui bermain sebagai gadis Bond di sekuel Spectre pada 2015 silam. Jika benar, ini merupakan sat satunya sekuel yang bercerita pernikahan James Bond sejak 1969. Sebelumnya, di sekuel On Her Majesty, James Bond menikahi Tracy Draco yang diperankan Diana Rigg. Namun, di film itu Draco terbunuh. Plot sekuel film James Bond ke - 25 bocor tak lama setelah Daniel Craig mengumumkan baka                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

### (b) Cuplikan data berita *online* berbahasa Indonesia

## Lampiran 2 Pra-pemrosesan data (*data preprocessing*)

```
# Filtering (Removing non-alphabet characters)
def filter_text(text):
    text = re.sub(r'^a-zA-Z\s', '', text)
    return text

# Apply filtering
train_df["filtered_paragraphs"] = train_df["paragraphs"].apply(filter_text)
train_df["filtered_summary"] = train_df["summary"].apply(filter_text)

valid_df["filtered_paragraphs"] = valid_df["paragraphs"].apply(filter_text)
valid_df["filtered_summary"] = valid_df["summary"].apply(filter_text)

test_df["filtered_paragraphs"] = test_df["paragraphs"].apply(filter_text)
test_df["filtered_summary"] = test_df["summary"].apply(filter_text)

# Save filtered data
train_df.to_csv('train_01_filtered.csv', index=False)
valid_df.to_csv('valid_01_filtered.csv', index=False)
test_df.to_csv('test_01_filtered.csv', index=False)

# Display filtered data
print(train_df.head())
```

(a) Source code *filtering*

```
# Stopword and Numbers Removal
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()

def remove_stopwords(text):
    return stopword.remove(text)

# Apply stopword removal
train_df["no_stopwords_paragraphs"] = train_df["filtered_paragraphs"].apply(remove_stopwords)
train_df["no_stopwords_summary"] = train_df["filtered_summary"].apply(remove_stopwords)

valid_df["no_stopwords_paragraphs"] = valid_df["filtered_paragraphs"].apply(remove_stopwords)
valid_df["no_stopwords_summary"] = valid_df["filtered_summary"].apply(remove_stopwords)

test_df["no_stopwords_paragraphs"] = test_df["filtered_paragraphs"].apply(remove_stopwords)
test_df["no_stopwords_summary"] = test_df["filtered_summary"].apply(remove_stopwords)

# Save data without stopwords
train_df.to_csv('train01_no_stopwords.csv', index=False)
valid_df.to_csv('valid01_no_stopwords.csv', index=False)
test_df.to_csv('test01_no_stopwords.csv', index=False)

# Display data without stopwords
print(train_df.head())
```

(b) *Stopwords* and numbers removal

```

# Case Folding
def case_folding(text):
    return text.lower()

# Apply case folding
train_df["case_folded_paragraphs"] = train_df["no_stopwords_paragraphs"].apply(case_folding)
train_df["case_folded_summary"] = train_df["no_stopwords_summary"].apply(case_folding)

valid_df["case_folded_paragraphs"] = valid_df["no_stopwords_paragraphs"].apply(case_folding)
valid_df["case_folded_summary"] = valid_df["no_stopwords_summary"].apply(case_folding)

test_df["case_folded_paragraphs"] = test_df["no_stopwords_paragraphs"].apply(case_folding)
test_df["case_folded_summary"] = test_df["no_stopwords_summary"].apply(case_folding)

# Save case folded data
train_df.to_csv('train01_case_folded.csv', index=False)
valid_df.to_csv('valid01_case_folded.csv', index=False)
test_df.to_csv('test01_case_folded.csv', index=False)

# Display case folded data
print(train_df.head())

```

(c) Case-folding

```

# Normalisasi
normalization_dict = {
    "yg": "yang",
    "tdk": "tidak",
    "kpd": "kepada",
    "dgn": "dengan",
    "udah": "sudah",
    "gak": "tidak",
    "selamat": "selamat",
    "karier": "karir",
    "enggak": "tidak",
    "istirahat": "istirahat",
    "Bimtek": "bimbingan teknis",
    "Pemprov": "pemerintah provinsi",
    "SDM": "sumber daya manusia",
    "Plt.": "pelaksana tugas",
    "ASN": "aparatur sipil negara",

    # Tambahkan lebih banyak pasangan kata untuk normalisasi sesuai kebutuhan
}

def normalize_text(text):
    for word, replacement in normalization_dict.items():
        text = re.sub(r'\b{}\b'.format(word), replacement, text)
    return text

# Apply normalization
train_df["normalized_paragraphs"] = train_df["case_folded_paragraphs"].apply(normalize_text)
train_df["normalized_summary"] = train_df["case_folded_summary"].apply(normalize_text)

valid_df["normalized_paragraphs"] = valid_df["case_folded_paragraphs"].apply(normalize_text)
valid_df["normalized_summary"] = valid_df["case_folded_summary"].apply(normalize_text)

test_df["normalized_paragraphs"] = test_df["case_folded_paragraphs"].apply(normalize_text)
test_df["normalized_summary"] = test_df["case_folded_summary"].apply(normalize_text)

# Save normalized data
train_df.to_csv('train01_normalized.csv', index=False)
valid_df.to_csv('valid01_normalized.csv', index=False)
test_df.to_csv('test01_normalized.csv', index=False)

# Display normalized data
print(train_df.head())

```

(d) normalisasi

```

# Tokenizing
def tokenize_text(text):
    return word_tokenize(text)

# Apply tokenization
train_df["tokenized_paragraphs"] = train_df["normalized_paragraphs"].apply(tokenize_text)
train_df["tokenized_summary"] = train_df["normalized_summary"].apply(tokenize_text)

valid_df["tokenized_paragraphs"] = valid_df["normalized_paragraphs"].apply(tokenize_text)
valid_df["tokenized_summary"] = valid_df["normalized_summary"].apply(tokenize_text)

test_df["tokenized_paragraphs"] = test_df["normalized_paragraphs"].apply(tokenize_text)
test_df["tokenized_summary"] = test_df["normalized_summary"].apply(tokenize_text)

# Save tokenized data
train_df.to_csv('train01_tokenized.csv', index=False)
valid_df.to_csv('valid01_tokenized.csv', index=False)
test_df.to_csv('test01_tokenized.csv', index=False)

# Display tokenized data
print(train_df.head())

```

(e) *tokenizing*

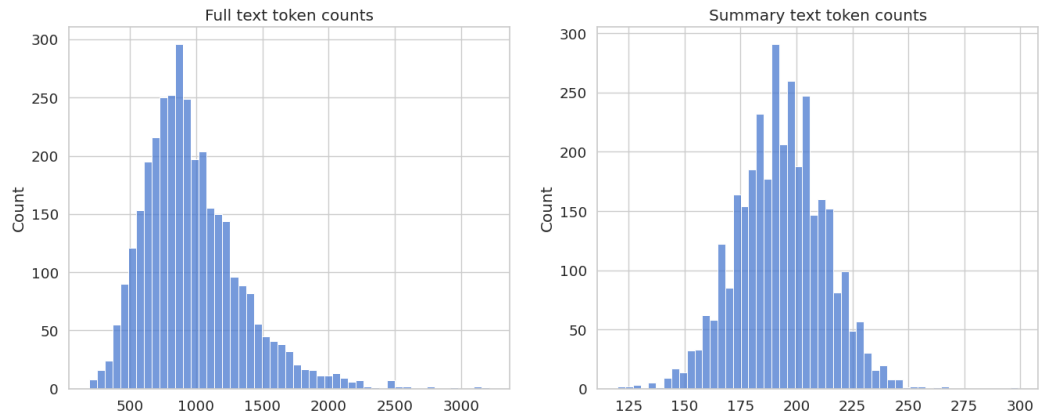
### Lampiran 3 Distribusi data

```
[ ] fig, (ax1,ax2) = plt.subplots(1,2)

sns.histplot(text_token_counts,ax=ax1)
ax1.set_title("Full text token counts")

sns.histplot(summary_token_counts,ax=ax2)
ax2.set_title("Summary text token counts")
```

(a) source code distribusi data



(b) Hasil distribusi data

## Lampiran 4 *Train model*

```

class NewsSummaryDataset(Dataset):
    def __init__(self, data: pd.DataFrame, tokenizer: T5Tokenizer, text_max_token_len: 512, summary_max_token_len: 128):
        self.tokenizer = tokenizer
        self.data = data
        self.text_max_token_len = text_max_token_len
        self.summary_max_token_len = summary_max_token_len

    def __len__(self):
        return len(self.data)

    def __getitem__(self, index: int):
        data_row = self.data.iloc[index]
        text = data_row["text"]

        text_encoding = self.tokenizer(
            text,
            max_length=self.text_max_token_len,
            padding="max_length",
            truncation=True,
            return_attention_mask=True,
            return_tensors="pt"
        )

        summary_encoding = self.tokenizer(
            data_row["summary"],
            max_length=self.summary_max_token_len,
            padding="max_length",
            truncation=True,
            return_attention_mask=True,
            return_tensors="pt"
        )

class NewsSummaryDataModule(pl.LightningDataModule):
    def __init__(self, train_df: pd.DataFrame, valid_df: pd.DataFrame, test_df: pd.DataFrame, tokenizer: T5Tokenizer, batch_size: int = 8, text_max_token_len: int = 512, summary_max_token_len: int = 128):
        super().__init__()
        self.train_df = train_df
        self.valid_df = valid_df
        self.test_df = test_df
        self.batch_size = batch_size
        self.tokenizer = tokenizer
        self.text_max_token_len = text_max_token_len
        self.summary_max_token_len = summary_max_token_len

    def setup(self, stage=None):
        self.train_dataset = NewsSummaryDataset(
            self.train_df,
            self.tokenizer,
            self.text_max_token_len,
            self.summary_max_token_len
        )
        self.valid_dataset = NewsSummaryDataset(
            self.valid_df,
            self.tokenizer,
            self.text_max_token_len,
            self.summary_max_token_len
        )
        self.test_dataset = NewsSummaryDataset(
            self.test_df,
            self.tokenizer,
            self.text_max_token_len,
            self.summary_max_token_len
        )

    def train_dataloader(self):
        return DataLoader(
            self.train_dataset,
            batch_size=self.batch_size,
            shuffle=True,
            num_workers=2,
            persistent_workers=True
        )

    def val_dataloader(self):
        return DataLoader(
            self.valid_dataset,
            batch_size=self.batch_size,
            shuffle=False,
            num_workers=2,
            persistent_workers=True
        )

    def test_dataloader(self):
        return DataLoader(
            self.test_dataset,
            batch_size=self.batch_size,
            shuffle=False,
            num_workers=2,
            persistent_workers=True
        )

    def prepare_data(self):
        labels = summary_encoding["input_ids"]
        labels[labels == 0] = -100

    def training_step(self, batch, batch_idx):
        return dict(
            text=text,
            summary=data_row["summary"],
            text_input_ids=text_encoding["input_ids"].flatten(),
            text_attention_mask=text_encoding["attention_mask"].flatten(),
            labels=labels.flatten(),
            labels_attention_mask=summary_encoding["attention_mask"].flatten()
        )

```

```

class NewsSummaryModel(pl.LightningModule):
    def __init__(self):
        super().__init__()
        self.model = T5ForConditionalGeneration.from_pretrained("t5-base", return_dict=True)

    def forward(self, input_ids, attention_mask, decoder_attention_mask, labels=None):
        output = self.model(
            input_ids,
            attention_mask=attention_mask,
            labels=labels,
            decoder_attention_mask=decoder_attention_mask
        )
        return output.loss, output.logits

    def training_step(self, batch, batch_idx):
        input_ids = batch["text_input_ids"]
        attention_mask = batch["text_attention_mask"]
        labels = batch["labels"]
        labels_attention_mask = batch["labels_attention_mask"]

        loss, outputs = self(
            input_ids=input_ids,
            attention_mask=attention_mask,
            decoder_attention_mask=labels_attention_mask,
            labels=labels
        )
        self.log("train_loss", loss, prog_bar=True, logger=True)
        return loss

model = NewsSummaryModel()

import torch
torch.cuda.is_available()

checkpoint_callback = ModelCheckpoint(
    dirpath="checkpoints",
    filename="best-checkpoint",
    save_top_k=1,
    verbose=True,
    monitor="val_loss",
    mode="min"
)

logger = TensorBoardLogger("lightning_loss", name="news-summary")

trainer = pl.Trainer(
    logger=logger,
    callbacks=[checkpoint_callback],
    max_epochs=N_EPOCHS,
    accelerator='auto',
    enable_progress_bar=True
)

tokenizer = T5Tokenizer.from_pretrained("t5-base")

N_EPOCHS = 7
BATCH_SIZE = 16

data_module = NewsSummaryDataModule(
    train_df=train_df,
    valid_df=valid_df,
    test_df=test_df,
    tokenizer=tokenizer,
    batch_size=BATCH_SIZE,
    text_max_token_len=512,
    summary_max_token_len=128
)

trainer.fit(model, datamodule=data_module)

trainer.test(model, datamodule=data_module)

trained_model = NewsSummaryModel.load_from_checkpoint(trainer.checkpoint_callback.best_model_path)

trained_model.freeze()

```

```

def summarize_text(model, tokenizer, text):
    model.eval()
    with torch.no_grad():
        # Encode text
        text_encoding = tokenizer(
            text,
            max_length=512,
            padding="max_length",
            truncation=True,
            return_attention_mask=True,
            return_tensors="pt"
        )

        # Move tensors to the same device as model
        device = model.device
        input_ids = text_encoding["input_ids"].to(device)
        attention_mask = text_encoding["attention_mask"].to(device)

        # Generate summary
        generated_ids = model.model.generate(
            input_ids=input_ids,
            attention_mask=attention_mask,
            max_length=150,
            num_beams=2,
            repetition_penalty=2.5,
            length_penalty=1.0,
            early_stopping=True
        )

        preds = [tokenizer.decode(gen_id, skip_special_tokens=True, clean_up_tokenization_spaces=True) for gen_id in generated_ids]
        return " ".join(preds)

sample_row = test_df.iloc[0]
text = sample_row["text"]
model_summary = summarize_text(model, tokenizer, text)
print("Original Text:", text)
print("Generated Summary:", model_summary)
print("Actual Summary:", sample_row["summary"])

```

## Lampiran 5. Evaluasi model

## Lampiran 6. Evaluasi model

```
def summarize_text(model, tokenizer, text):
    model.eval()
    with torch.no_grad():
        # Encode text
        text_encoding = tokenizer(
            text,
            max_length=512,
            padding="max_length",
            truncation=True,
            return_attention_mask=True,
            return_tensors="pt"
        )

        # Move tensors to the same device as model
        device = model.device
        input_ids = text_encoding["input_ids"].to(device)
        attention_mask = text_encoding["attention_mask"].to(device)

        # Generate summary
        generated_ids = model.generate(
            input_ids=input_ids,
            attention_mask=attention_mask,
            max_length=150,
            num_beams=2,
            repetition_penalty=2.5,
            length_penalty=1.0,
            early_stopping=True
        )

        preds = [tokenizer.decode(gen_id, skip_special_tokens=True, clean_up_tokenization_spaces=True) for gen_id in generated_ids]
        return " ".join(preds)
```

```
# Fungsi untuk mengevaluasi model
def evaluate_model(model, tokenizer, test_df):
    model.eval()
    summaries = []
    references = []

    # Untuk menampilkan contoh ringkasan
    display_examples = True
    example_count = 0

    for index, row in test_df.iterrows():
        text = row["text"]
        summary = row["summary"]

        generated_summary = summarize_text(model, tokenizer, text)
        summaries.append(generated_summary)
        references.append(summary)

        if display_examples and example_count < 2: # Menampilkan hingga 2 contoh
            print(f"Original: {text}")
            print(f"Reference Summary: {summary}")
            print(f"Generated Summary: {generated_summary}")
            example_count += 1

    scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)
    scores = [scorer.score(ref, gen) for ref, gen in zip(references, summaries)]
```

```
avg_scores = {
    "rouge1": {
        "precision": np.mean([score["rouge1"].precision for score in scores]),
        "recall": np.mean([score["rouge1"].recall for score in scores]),
        "fmeasure": np.mean([score["rouge1"].fmeasure for score in scores])
    },
    "rouge2": {
        "precision": np.mean([score["rouge2"].precision for score in scores]),
        "recall": np.mean([score["rouge2"].recall for score in scores]),
        "fmeasure": np.mean([score["rouge2"].fmeasure for score in scores])
    },
    "rougeL": {
        "precision": np.mean([score["rougeL"].precision for score in scores]),
        "recall": np.mean([score["rougeL"].recall for score in scores]),
        "fmeasure": np.mean([score["rougeL"].fmeasure for score in scores])
    }
}
```

```

# Calculate BLEU score
bleu = corpus_bleu([ref.split() for ref in references], [gen.split() for gen in summaries])

print(f"Average ROUGE-1: Precision: {avg_scores['rouge1']['precision']}, Recall: {avg_scores['rouge1']['recall']}, F1: {avg_scores['rouge1']['fmeasure']}")
print(f"Average ROUGE-2: Precision: {avg_scores['rouge2']['precision']}, Recall: {avg_scores['rouge2']['recall']}, F1: {avg_scores['rouge2']['fmeasure']}")
print(f"Average ROUGE-L: Precision: {avg_scores['rougeL']['precision']}, Recall: {avg_scores['rougeL']['recall']}, F1: {avg_scores['rougeL']['fmeasure']}")
print(f"Average BLEU Score: {bleu}")

# Menggunakan model terbaik yang disimpan
best_model_path = "checkpoints/best-checkpoint.ckpt"
model = NewsSummaryModel.load_from_checkpoint(best_model_path)
tokenizer = T5Tokenizer.from_pretrained("t5-base")

```

```

# Menggunakan model terbaik yang disimpan
best_model_path = "checkpoints/best-checkpoint.ckpt"
model = NewsSummaryModel.load_from_checkpoint(best_model_path)
tokenizer = T5Tokenizer.from_pretrained("t5-base")

print("Validation Evaluation:")
evaluate_model(model, tokenizer, valid_df)

print("\nTest Evaluation:")
evaluate_model(model, tokenizer, test_df)

```

## Lampiran 6. Daftar hadir dan berita acara seminar hasil



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

---

DAFTAR HADIR SEMINAR HASIL

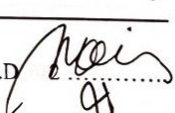
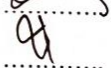
Nama/Stambuk : 1. Kisana Adzan Sitorus D121201106

Judul Skripsi/T.A : **“Implementasi Text-To Text Transfer Transformer Berbasis Abstraksi dan Ekstraksi dalam Text Summarization Berita Online “**

Hari/Tanggal : Jumat , 14 Juni 2024

Jam : 14.30 Wita – Selesai

Tempat : Ruang Lab. CBS Departemen Teknik Informatika Gowa

| No. | Jabatan         | Nama Dosen                                  | Tanda Tangan                                                                                  |
|-----|-----------------|---------------------------------------------|-----------------------------------------------------------------------------------------------|
| L.  | Pembimbing I    | 1. Elly Warni,ST.,M.T                       | 1. ....  |
| II. | Anggota Penguji | 2. Novy Nur R.A Mokobombang,ST.,Ms.,TM.Ph.D | 2. ....  |
|     |                 | 3. Anugrayani Bustamin,ST.,M.T              | 3. ....  |

PANITIA UJIAN

Ketua,

  
Elly Warni,ST.,M.T



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
**UNIVERSITAS HASANUDDIN**  
**FAKULTAS TEKNIK**  
**DEPARTEMEN TEKNIK INFORMATIKA**

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

Nomor : 776/UN4.7.7/TD.06/2024  
Lamp : -  
Hal : Penerbitan Surat Penugasan Panitia/Penguji  
Seminar Hasil Strata Satu (S1)

Kepada Yth :

Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Universitas Hasanuddin

Di-

Gowa

Dengan hormat,

Berdasarkan Persetujuan Pembimbing Mahasiswa, Bersama ini diusulkan susunan Panitia/Penguji Seminar Hasil Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik tersebut di bawah ini :

Nama / Stambuk : Kisana Adzan Sitorus D121201106  
Judul TA : Implementasi Text-To Text Transfer Transformer Berbasis  
Abstraksi dan Ekstraksi dalam Text  
Summarization Berita Online

Dengan ini kami sampaikan Susunan Panitia Seminar Hasil Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua : 1. Elly Warni, ST., M.T  
Anggota : 2. Novy Nur R.A Mekobombang, ST., Ms.TM. Ph. D.  
: 3. Anugrayani Bustamin, ST., M.T.

Untuk dapat diterbitkan surat penugasannya

Demikian penyampaian kami, atas perhatian dan kerjasamanya diucapkan terima kasih.

Gowa, 11 Juni 2024  
Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu, ST, MT., M.Bus.Sys., IPM, ASEAN.Eng  
Nip.19750716 200212 1 004

Tembusan :  
1. Arsip

waktu : 14.30  
tempat : Lab CBS





KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
**UNIVERSITAS HASANUDDIN**  
**FAKULTAS TEKNIK**  
Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan  
☎ +62811 4420 909, E-mail: [teknik@unhas.ac.id](mailto:teknik@unhas.ac.id) , <https://eng.unhas.ac.id>

SURAT PENUGASAN  
No. 13411/UN4.7.1/TD.06/2024

Dari : Dekan Fakultas Teknik Universitas Hasanuddin  
Kepada : Mereka yang tercantum namanya dibawah ini  
Isi : 1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin Nomor :  
**29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin** , dengan ini menugaskan Saudara sebagai **PENGUJI/PANITIA SEMINAR HASIL** Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :  
Pembimbing I/ Ketua : 1. Elly Warni, ST, M T  
Anggota : 2. Novy Nur R.A Mokobombang, ST., Ms. TM. Ph. D  
: 3. Anugrayani Bustamin, ST., M.T.

Untuk menguji bagi mahasiswa tersebut dibawah ini :

Nama/NIM : Kisana Adzan Sitorus D121201106  
Program Studi : Teknik Informatika  
Judul thesis/Skripsi : Implementasi Text-To Text Transfer Transformer Berbasis Abstraksi dan Ekstraksi dalam Text Summarization Berita Online  
2. Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)  
3. Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.  
4. Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa  
Pada tanggal 11 Juni 2024  
a.n. Dekan,  
Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Unhas



Dr. Amil Ahmad Ilham, ST., M.IT  
NIP. 197310101998021001

Tembusan :  
1. Dekan Fak. Teknik Unhas  
2. Ketua Departemen Teknik Informatika FT-UH  
3. Mahasiswa yang bersangkutan



• Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSR E  
• UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1  
"Informasi Elektronik dan/atau Dokumen Elektronik dan/atau hasil cetaknya merupakan alat bukti hukum yang sah"



## Lampiran 7. Daftar hadir dan berita acara ujian skripsi



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN  
RISET DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
**DEPARTEMEN TEKNIK INFORMATIKA**  
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

**DAFTAR HADIR UJIAN SKRIPSI MAHASISWA  
FAKULTAS TEKNIK UNHAS**

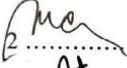
Nama/Stambuk : 1. Kisana Adzan Sitorus D121201106

Judul Skripsi/T.A : **“Implementasi Text-To Text Transfer Transformer Berbasis Abstraksi dan Ekstraksi dalam Text Summarization Berita Online “**

Hari/Tanggal : Senin, 22 Juli 2024

Jam : 09.00 Wita – Selesai

Tempat : Ruang Lab. CBS Departemen Teknik Informatika Gowa

| No. | Jabatan         | Nama Dosen                                     | Tanda Tangan                                                                                  |
|-----|-----------------|------------------------------------------------|-----------------------------------------------------------------------------------------------|
| L.  | Pembimbing I    | 1. Elly Warni, ST., M.T                        | 1. ....  |
| II. | Anggota Penguji | 2. Novy Nur R.A Mokobombang, ST., Ms., TM.Ph.D | 2. ....  |
|     |                 | 3. Anugrayani Bustamin, ST., M.T               | 3. ....  |

## PANITIA UJIAN

Ketua,  
  
Elly Warni, ST., M.T



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS HASANUDDIN  
FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK INFORMATIKA**

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa  
<http://eng.unhas.ac.id/informatika>, Email : [informatika@unhas.ac.id](mailto:informatika@unhas.ac.id)

Gowa, 18 Juli 2024

Nomor : 973/UN4.7.7.1/TD.06/2024  
Lamp : -  
Hal : Usulan Susunan Panitia/Penguji Ujian Sarjana

Yth. : Bapak Wakil Dekan Bidang Akademik dan Kemahasiswaan  
Fakultas Teknik Unhas  
Di  
Gowa

Dalam rangka penyelesaian studi pada Departemen Teknik Informatika Fakultas Teknik Unhas, bersama ini kami usulkan susunan Panitia/Penguji Ujian Sarjana Program Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas nama :

Pembimbing / Ketua : 1. Elly Warni, ST., M.T  
Penguji / Anggota : 2. Novy Nur R A Mokobombang, ST., Ms.TM., Ph.D  
3. Ir. Anugrayani Bustamin, ST., M.T

Untuk Bertugas sebagai Penguji/ Penanggap Ujian Sarjana bagi Mahasiswa :

Nama : Kisana Adzan Sitorus  
Stambuk : D121 20 1106

Dengan Judul Skripsi :

“ Implementasi Text-to-Text Transfer Transformer berbasis Abstraksi dan Ekstraksi  
dalam Text Summarization Berita Online “

Pada :  
Hari/Tanggal : Senin, 22 Juli 2024  
Jam : 09.00 Wita - Selesai  
Tempat : Ruang Sidang Lab. CBS

Demikian penyampaian kami, atas perhatiannya diucapkan terimakasih.

Ketua Departemen Tek.Informatika,



Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng  
Nip.197507016 200212 1 004

Tembusan :  
1. Arsip

Hari / Tanggal : Senin, 22 Juli 2024  
Waktu : 09.00 - Selesai  
Tempat : Lab CBS



Lampiran 8. Lembar perbaikan skripsi

### LEMBAR PERBAIKAN SKRIPSI

**“IMPLEMENTASI *TEXT-TO-TEXT TRANSFER TRANSFORMER*  
BERBASIS ABSTRAKSI DAN EKSTRAKSI DALAM TEXT  
SUMMARIZATION BERITA ONLINE”**

**OLEH:**

**KISANA ADZAN SITORUS  
D121 20 1106**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 22 Juli 2024.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari  
penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

|         | Nama                                           | Tanda Tangan                                                                          |
|---------|------------------------------------------------|---------------------------------------------------------------------------------------|
| Ketua   | Elly Warni, S.T., M.T                          |  |
| Anggota | Novy Nur R.A Mokobombang, S.T., Ms.TM.<br>Ph.D |  |
| Anggota | Anugrayani Bustamin, S.T., M.T                 |  |

Persetujuan Perbaikan oleh pembimbing:

| Pembimbing | Nama                   | Tanda Tangan                                                                          |
|------------|------------------------|---------------------------------------------------------------------------------------|
| I          | Elly Warni, S.T., M.T. |  |