Persiapan UTS NLP (IF5153)

Pilihan ganda (kunci soal lama dan bank latihan)
dan essay desain sistem NLP dengan rumus IO–TaDa–PRIME

Dokumen ini adalah bahan latihan UTS IF5153 Advanced NLP. Isinya mengikuti cakupan catatan *UTS_NLP*, yaitu catatan 1, 2, 3, 5, 6, dan 7: pengantar dan kategori task, klasifikasi teks, word representation, Transformer dan BERT, decoder/prompting/PEFT, serta sentence comparison.

Format UTS. Pertama, **pilihan ganda** berisi teori, dalam bentuk grid True/False per pernyataan. Kedua, **essay** meminta merancang sistem NLP untuk suatu kebutuhan dan menjelaskan komponennya. Format ini sesuai dengan info yang beredar dan dengan dua soal UTS lama (IF4072 2020/2021 dan 2022/2023).

Soal lama yang dipakai. Hanya soal yang materinya ada di catatan tahun ini. Soal lama yang materinya tidak ada di catatan tahun ini tidak dimasukkan.

Kunci soal lama. Centang pada file Google Form 2020/2021 adalah jawaban seorang mahasiswa, bukan kunci, dan sebagian salah. Kunci di dokumen ini disusun ulang dari materi beserta alasannya. Tanda [!] berarti pernyataan bisa diperdebatkan; yang ditulis adalah jawaban yang paling bisa dipertahankan.

Cara memakai dokumen ini:

1. Baca bagian 1 (format dan strategi PG), lalu kerjakan bagian 2 dan 3 dengan menutup kolom kunci.
2. Hafalkan rumus desain **IO–TaDa–PRIME** di bagian 5.
3. Latih rumus itu pada essay lama (bagian 6) dan studi kasus prediksi (bagian 7).

Penjelasan yang tidak ada di catatan kuliah ditandai *(tambahan)*.

Daftar Isi

1 Format Soal dan Strategi Pilihan Ganda **4**

1.1 Dua Jenis Grid 4
1.2 Lima Pola Jebakan 4
1.3 Strategi Mengerjakan 4

2 Kunci dan Pembahasan PG Soal Lama **4**

2.1 Preprocessing untuk klasifikasi sentimen data Twitter 4
2.2 Deep learning untuk NLP 5
2.3 Fitur yang baik untuk spam filtering berbasis ML 5
2.4 Pengantar NLP 5
2.5 Proses lanjutan setelah vector space model term × document 6
2.6 Vector space model term × document 6
2.7 Mengurangi dimensi data latih klasifikasi teks 6

2.8 Metode untuk tiap task (grid mencocokkan) . . . 7
2.9 Mengukur kemiripan kata . . . 7
2.10 Sub-masalah aspect based sentiment classification (grid mencocokkan) . . . 8
2.11 Sequential labeling . . . 8
2.12 RNN untuk NLP . . . 8

3 Bank Latihan PG Materi Tahun Ini **9**
3.1 L1. Word representation . . . 9
3.2 L2. Word2Vec . . . 9
3.3 L3. Klasifikasi teks dan fitur . . . 10
3.4 L4. Transformer encoder . . . 10
3.5 L5. Multi-head attention dan decoder . . . 10
3.6 L6. BERT pre-training . . . 10
3.7 L7. BERT fine-tuning dan varian . . . 11
3.8 L8. Tokenizer . . . 11
3.9 L9. Prompting . . . 11
3.10 L10. Fine-tuning dan instruction finetuning . . . 12
3.11 L11. PEFT . . . 12
3.12 L12. Sentence comparison . . . 13
3.13 L13. SimCSE, E5, BGE-M3 . . . 13
3.14 L14. Zero-shot classification dengan NLI . . . 13
3.15 L15. Kategori task (grid mencocokkan) . . . 14

4 Pasangan Konsep yang Sering Tertukar **14**

5 Desain Sistem NLP: Rumus IO–TaDa–PRIME **15**
5.1 Sistem NLP adalah Pipeline . . . 15
5.2 Rumus Enam Langkah: IO–TaDa–PRIME . . . 15
5.3 Langkah Ta: Kata Kerja di Soal Menentukan Kategori . . . 16
5.4 Langkah P, R, dan I: Komponen per Kategori . . . 17
5.5 Langkah M: Pohon Keputusan Metode Melatih . . . 18
5.6 Langkah E: Evaluasi dan Trade-off . . . 19
5.7 Template Jawaban Essay . . . 19

6 Essay Lama, Dikerjakan dengan Rumus **19**
6.1 UTS 2020/2021: Arsitektur Hate Speech di Twitter . . . 19
6.2 UTS 2022/2023 No. 2: NER dengan Tiga Pendekatan . . . 21
6.3 UTS 2022/2023 No. 4: Arsitektur ABSA Modular (Samsung/Canon) . . . 23
6.4 UTS 2022/2023 No. 3: Information Retrieval dengan LSI . . . 24

7 Studi Kasus Prediksi **25**

7.1 Kasus A: Pemantau Opini Publik tentang Layanan Kampus 25
7.2 Kasus B: Chatbot Tanya-Jawab Peraturan Akademik 26
7.3 Kasus C: Penilaian Otomatis Jawaban Singkat 28
7.4 Kasus D: Pemrosesan Email Layanan Akademik 29

8 Ringkasan Inti **29**

1 Format Soal dan Strategi Pilihan Ganda

1.1 Dua Jenis Grid

Jenis	Bentuk	Contoh kalimat pembuka
Grid True/False	Satu kalimat pembuka diikuti 5 pernyataan; **tiap pernyataan dinilai sendiri** True atau False	"Below are correct preprocessing for sentiment classification on twitter data"
Grid mencocokkan	Daftar item (task, sub-masalah) dicocokkan ke kategori (text categorization / sequential labeling / other)	"Select correct method for each NLP task below"

Akibatnya, satu soal berisi 5–10 keputusan kecil. Yang diuji adalah **ketelitian membedakan konsep yang mirip**, bukan hafalan definisi panjang.

1.2 Lima Pola Jebakan

Pola	Contoh dari soal lama dan materi tahun ini
1. Konsep ditukar	"Text classification melabeli tiap kata" (seharusnya sequential labeling); "Q dari encoder, K dan V dari decoder" (terbalik)
2. Klaim "sama"	"Pemakaian word2vec sama dengan BERT"; "many-to-many MT sama dengan many-to-many sequence labeling". Hampir selalu False
3. Contoh salah kategori	"Bank" dijadikan contoh ambiguitas POS, padahal kedua artinya sama-sama noun (ambiguitas makna)
4. Arah efek terbalik	"Bigram mengurangi dimensi" (malah menambah); "IDF menaikkan bobot kata yang muncul di banyak dokumen" (malah menurunkan)
5. Kemampuan berlebihan	"VSM bisa menangani sinonim" (tidak bisa, itu kelemahannya); "prompting lebih efisien dari fine-tuning" (prompt diproses ulang tiap prediksi)

1.3 Strategi Mengerjakan

Empat langkah tiap grid

1. **Baca kalimat pembuka**: apa yang ditanyakan ("correct preprocessing", "reduce dimension", "next possible process").
2. **Nilai tiap pernyataan terpisah**; jangan menebak pola seperti "pasti ada dua yang False".
3. **Cari satu kata yang bisa membuatnya salah**: subjek yang ditukar, "same", "only", "always", "each word", "complete".
4. **Bila ragu, uji kebalikannya.** Contoh: "bigram mengurangi dimensi" berarti unigram lebih banyak; mustahil, karena bigram adalah pasangan kata.

2 Kunci dan Pembahasan PG Soal Lama

Sumber: UTS IF4072 2020/2021 (Google Form). Pernyataan ditulis apa adanya dalam bahasa Inggris. **T** = True, **F** = False, [!] = bisa diperdebatkan.

2.1 Preprocessing untuk klasifikasi sentimen data Twitter

Pernyataan	Kunci	Alasan
word normalization	**T**	Tweet penuh singkatan dan slang (“gk”, “bgt”)
entity masking	**T**	URL, @mention, nomor diganti token kelas agar fitur tidak terpecah
stop word elimination	**T** [!]	Lazim dipakai, **asal** kata negasi (“not”, “tidak”) tidak ikut dibuang. Contoh slide: “…the food is not recommended” → “…food recommended” membalik makna
stemming	**T**	Menyatukan variasi kata, mengurangi fitur
pos tagger	**F** [!]	POS tagging adalah tools analisis/fitur, bukan langkah membersihkan teks. Bisa dipakai untuk memilih stop word, tetapi bukan preprocessing wajib

2.2 Deep learning untuk NLP

Pernyataan	Kunci	Alasan
one hot vector can be used as input layer	**T**	Input neural network harus angka; one-hot adalah bentuk paling dasar
the input layer may include string value feature	**F**	String harus diubah menjadi angka dulu (vectorization/embedding)
word2vec transforms one hot vector into a dense vector	**T**	One-hot dikali matriks bobot hidden layer menghasilkan vektor dense (lookup table)
the usage of word2vec pretrain model is the same as bert pretrain model for text classification	**F**	Word2vec: vektor statis sebagai fitur input. BERT: model kontekstual yang di-**fine-tune** seluruhnya atau diambil embedding kontekstualnya
fine tuning in bert is an example of transfer learning method	**T**	Pengetahuan dari pretraining dipindahkan ke task baru

2.3 Fitur yang baik untuk spam filtering berbasis ML

Pernyataan	Kunci	Alasan
target email address	**F**	Alamat tujuan sama untuk semua email ke pengguna itu, jadi tidak membedakan spam
complete string of email subject	**F**	Seluruh subjek sebagai satu nilai hampir selalu unik, sehingga model tidak bisa menggeneralisasi; yang baik adalah **term** di subjek
vector of term in email body	**T**	Bag of words/TF-IDF, fitur utama klasifikasi teks
datetime of email	**F** [!]	Umumnya lemah dan tidak terkait isi
character in email sender	**T** [!]	Pola karakter pengirim (angka acak, domain aneh) sering menandai spam

2.4 Pengantar NLP

Pernyataan	Kunci	Alasan
Text classification aims to classify each word in a sentence into certain label automatically	**F**	Text classification melabeli **seluruh teks**; melabeli tiap kata adalah sequential labeling
Out of vocabulary problem: an input word in the testing data is not available in the training data	**T**	Definisi OOV
Rule based technique in solving NLP task is efficient for large vocabulary	**F**	Aturan dibuat manual, sehingga mahal untuk kosakata besar
"Bank" is an ambiguity problem example in POS tagger where it can be a finance institution or a river bank	**F**	Kedua arti sama-sama **noun**, jadi tidak ada ambiguitas POS. Ini ambiguitas **makna**
Compound word in Indonesian language is an example of ambiguity for tokenization	**T**	"rumah sakit": satu token atau dua?

2.5 Proses lanjutan setelah vector space model term × document

Pernyataan	Kunci	Alasan
clustering the vector by grouping similar vector	**T**	Document clustering
finding similar word by comparing the vector	**T**	Baris matriks adalah vektor kata
classify the vector by the label of similar vector	**T**	Klasifikasi berbasis kemiripan (mis. k-nearest neighbor)
finding relevant document vector from a given query vector	**T**	Ini **information retrieval**, kegunaan klasik VSM
calculating word sense disambiguation based on vector comparison	**T** [!]	Vektor konteks dibandingkan dengan vektor tiap arti; mungkin, walau bukan kegunaan utama

2.6 Vector space model term × document

Pernyataan	Kunci	Alasan
document is represented as a vector of term score	**T**	Kolom matriks = vektor dokumen
one word is a vector of score	**T**	Baris matriks = vektor kata
boolean term score in the vector space means the number of terms in the document	**F**	Boolean = **ada (1) atau tidak (0)**; jumlah kemunculan adalah TF
it can handle synonyms	**F**	Tiap kata satu dimensi, sehingga sinonim dianggap berbeda. Inilah alasan muncul LSI/word embedding
TF × IDF score in the vector represents term weight in documents	**T**	Bobot kata pada dokumen

2.7 Mengurangi dimensi data latih klasifikasi teks

Pernyataan	Kunci	Alasan
using stop word elimination before feature extraction	**T**	Kata umum dibuang
using bi-gram token as the feature, instead of unigram (word)	**F**	Kombinasi pasangan kata jauh lebih banyak, sehingga dimensi **bertambah**
applying Singular Value Decomposition on term × document matrix	**T**	Reduksi ke ruang laten berdimensi k
adding part of speech as the feature	**F**	Menambah fitur
applying lemmatization/stemming before feature extraction	**T**	Variasi kata disatukan

2.8 Metode untuk tiap task (grid mencocokkan)

Pilihan: text categorization / sequential labeling / other.

Task	Kunci	Alasan
machine translation	other	Seq2seq (teks ke teks)
document level sentiment analysis	text categ.	Satu label untuk seluruh dokumen
author classification	text categ.	Satu label (penulis) per teks
phrase chunker	seq. label.	Tiap token diberi label bagian frasa
language detection	text categ.	Satu label bahasa per teks
named entity tagger	seq. label.	Tiap token diberi label entitas
image captioning	other	Gambar → teks (generatif)
non task chatbot	other	Generatif
keyword extraction	seq. label. [!]	Tiap token ditandai kata kunci atau bukan; bisa juga dipandang ranking (other)
information retrieval	other	Text comparison (query vs dokumen)

2.9 Mengukur kemiripan kata

Pernyataan	Kunci	Alasan
calculate the similarity from latent semantic × term matrix in SVD result	**T**	Baris U (atau US) = vektor kata di ruang laten
calculate the similarity from term × document matrix of vector space model	**T**	Baris = vektor kata
calculate the cosine distance of word embedding vector	**T**	Cara standar
calculate the similarity from term × term matrix	**T**	Matriks co-occurrence (mis. HAL)
calculate the similarity of complete vector in output layer of bert pretraining model	**T** [!]	Embedding kontekstual BERT bisa dibandingkan (kata dalam konteks); yang lazim dipakai adalah **hidden state** layer terakhir

2.10 Sub-masalah aspect based sentiment classification (grid mencocokkan)

Pilihan: text classification / sequential labeling / other.

Sub-masalah	Kunci	Alasan
to get keywords of aspect information	seq. label.	Menandai token aspek
to classify the sentiment of each aspect	text class.	Satu label sentimen per aspek
to get aspect label of each sentence	text class.	Kategori aspek per kalimat
to relate the extracted aspect with each entity found in the text	other	Relasi antar dua hal yang diekstrak (relation extraction)
to extract the entity in the text	seq. label.	NER

2.11 Sequential labeling

Pernyataan	Kunci	Alasan
the target is to label each token automatically	**T**	Definisi
it is an automatic labeling for a complete given document	**F**	Itu text classification
essay scoring is an example of sequential labeling	**F**	Satu skor per teks (klasifikasi, atau text comparison dengan kunci)
sequential labeling can be used to extract factoid answer from a given paragraph in a QA system	**T**	Token jawaban dilabeli Answer, lainnya O
ambiguity in sequential labeling can be solved by using previous token label	**T**	Contoh “bisa” setelah noun cenderung kata kerja bantu

2.12 RNN untuk NLP

Pernyataan	Kunci	Alasan
RNN can be used for multi label classification	**T**	Output akhir diberi beberapa sigmoid (satu per label)
"many to many" RNN for machine translation is the same as "many to many" RNN for sequence labeling	**F**	MT: panjang input dan output berbeda dan tidak sejajar (encoder-decoder). Sequence labeling: satu output per token, sejajar
image captioning is an example of seq2seq RNN	**F** [!]	Input berupa satu gambar, bukan sequence (pola **one-to-many**)
The output layer of RNN can be used as input for the shallow machine learning (CRF, SVM, etc)	**T**	Output RNN sebagai fitur
RNN is used to build BERT pretraining model	**F**	BERT memakai **Transformer encoder**, tanpa RNN

3 Bank Latihan PG Materi Tahun Ini

Format sama dengan soal lama, untuk materi yang belum pernah keluar. Tutup kolom kunci saat mengerjakan.

3.1 L1. Word representation

Pernyataan	Kunci	Alasan
One-hot vector captures similarity between synonyms	**F**	Semua vektor one-hot saling tegak lurus
GloVe vectors are obtained in closed form like SVD	**F**	GloVe dioptimasi dengan SGD/AdaGrad
In HAL, a word vector is the concatenation of its row in R and in L (dimension 2V)	**T**	$v(w) = [R[w,:] \,\|\| \, L[w,:]]$
LDA is a deterministic method like HAL	**F**	LDA generatif probabilistik; HAL deterministik
Word2Vec and GloVe give one fixed vector per word regardless of the sentence	**T**	Embedding statis

3.2 L2. Word2Vec

Pernyataan	Kunci	Alasan
CBOW predicts context words from the target word	**F**	Itu skip-gram; CBOW memprediksi target dari konteks
In skip-gram, the hidden layer weight matrix becomes the word vector lookup table	**T**	
Word2Vec is based on the distributional hypothesis	**T**	Kata mirip muncul di konteks mirip
$V_{king} - V_{man} + V_{woman} \approx V_{queen}$ illustrates algebraic relations	**T**	
Word2Vec gives different vectors for "bank" in different sentences	**F**	Statis; yang kontekstual adalah BERT/ELMo

3.3 L3. Klasifikasi teks dan fitur

Pernyataan	Kunci	Alasan
A stemming result may not be a valid word	**T**	*writing* → *writ*
The lemma of "writing" is "writ"	**F**	Lema = *write*
Maximum matching chooses the shortest dictionary word from the pointer	**F**	Kata **terpanjang**
IDF increases the weight of terms appearing in many documents	**F**	Makin banyak dokumen, makin **kecil** idf
CNN for text classification (Kim) uses max-over-time pooling	**T**	Tiap feature map menjadi satu nilai

3.4 L4. Transformer encoder

Pernyataan	Kunci	Alasan
Transformer removes sequential computation so training can be parallelized	**T**	Motivasi utama
Positional encoding is concatenated to the word embedding	**F**	**Dijumlahkan**
Attention scores are divided by $\sqrt{d_k}$ so the dot product does not grow too large	**T**	Scaled dot-product
A token ID carries semantic meaning	**F**	ID hanya index baris embedding matrix
Self-attention complexity per layer is $O(n^2 \cdot d)$	**T**	Kuadratik terhadap panjang sequence

3.5 L5. Multi-head attention dan decoder

Pernyataan	Kunci	Alasan
Each head has its own W^Q, W^K, W^V	**T**	
Head outputs are averaged, then multiplied by W^O	**F**	**Dikonkatenasi**, lalu dikali W^O
Masked self-attention sets scores of future positions to $-\infty$ before softmax	**T**	Setelah softmax menjadi 0
In cross-attention, Q comes from the encoder, K and V from the decoder	**F**	**Q dari decoder, K dan V dari encoder**
During decoding, the encoder runs once while the decoder runs per time step	**T**	Autoregressive

3.6 L6. BERT pre-training

Pernyataan	Kunci	Alasan
BERT uses only the encoder part of the Transformer	**T**	
MLM masks 15% of tokens	**T**	
NSP is predicted from the output at [CLS]	**T**	
BERT position embeddings are fixed sinusoids	**F**	**Dipelajari**
BERT input embedding is the sum of token, segment, and position embeddings	**T**	

3.7 L7. BERT fine-tuning dan varian

Pernyataan	Kunci	Alasan
Fine-tuning trains the whole model plus the added classifier	**T**	
Feature extraction freezes BERT and feeds its embeddings to a separate ML model	**T**	
NER fine-tuning predicts all labels from the [CLS] output only	**F**	Label **per token**
SQuAD QA output is a start and end span	**T**	
RoBERTa removes NSP	**T**	Juga training lebih lama dengan data lebih besar

3.8 L8. Tokenizer

Pernyataan	Kunci	Alasan
BERT uses WordPiece; continuation pieces start with **##**	**T**	
GPT-2 uses SentencePiece	**F**	GPT-2 memakai **BPE**; SentencePiece dipakai Flan-T5
SentencePiece does not depend on spaces	**T**	Language-independent
Subword granularity sits between word and character	**T**	
Character-level tokenization produces shorter sequences than subword	**F**	Lebih **panjang**

3.9 L9. Prompting

Pernyataan	Kunci	Alasan
A base model after pretraining is automatically a helpful assistant	**F**	Hanya melanjutkan dokumen
Few-shot prompting updates the model weights	**F**	In-context learning, tanpa gradient update
Zero-shot-CoT adds "Let's think step by step"	**T**	
Self-consistency picks the most frequent answer from several sampled reasoning paths	**T**	
Prompt-based learning is more efficient than fine-tuning because the prompt is processed once	**F**	Prompt diproses **ulang tiap prediksi** (inefisien)

3.10 L10. Fine-tuning dan instruction finetuning

Pernyataan	Kunci	Alasan
Pretraining acts as parameter initialization for fine-tuning	**T**	
Instruction finetuning is evaluated on unseen tasks	**T**	
Fine-tuning requires as much labeled data as pretraining	**F**	Tidak butuh banyak label
Traditional models are one model per task; LLMs are one model for many tasks via prompting	**T**	
FLAN-T5 is an example of an instruction-finetuned model	**T**	

3.11 L11. PEFT

Pernyataan	Kunci	Alasan
LoRA is an additive PEFT method	**F**	**Reparameterized**
BitFit trains only the bias terms and is a selective method	**T**	
After prefix-tuning, only the prefix is stored for each task	**T**	
LoRA always adds inference latency because its matrices cannot be merged	**F**	Bisa di-**merge** ke bobot asli
QLoRA applies LoRA on a 4-bit quantized base model	**T**	

Grid mencocokkan PEFT (additive / selective / reparameterized / hybrid):

Additive	Selective	Reparameterized	Hybrid
Adapter, prefix-tuning/soft prompt, $(IA)^3$, SSF	BitFit, masking (tidak terstruktur/terstruktur)	LoRA, DyLoRA, DoRA	UniPELT, MAM Adapter

3.12 L12. Sentence comparison

Pernyataan	Kunci	Alasan
Lexical-based comparison uses sparse vectors	**T**	Bag of words
A cross-encoder can precompute and cache sentence embeddings	**F**	Itu bi-encoder
Bi-encoder + classifier uses u, v, and $\|u - v\|$ as features	**T**	
SBERT is fine-tuned on NLI and then STS	**T**	
A decision tree can directly process a cross-encoder's input	**F**	Input token mentah, hanya model neural sequence yang bisa

3.13 L13. SimCSE, E5, BGE-M3

Pernyataan	Kunci	Alasan
Unsupervised SimCSE gets positive pairs from dropout differences	**T**	Dua kali forward pass
Supervised SimCSE uses contradiction sentences as negatives	**T**	
E5 uses the prefixes "query:" and "passage:"	**T**	
BGE-M3 supports only English	**F**	Lebih dari 100 bahasa
BGE-M3 hybrid retrieval combines dense, lexical (sparse), and multi-vector scores	**T**	$s_{rank} = w_1 s_{dense} + w_2 s_{lex} + w_3 s_{mul}$

3.14 L14. Zero-shot classification dengan NLI

Pernyataan	Kunci	Alasan
The input text becomes the premise	**T**	
Each candidate label becomes a hypothesis "this text is {class}"	**T**	
The chosen class has the highest P(contradiction)	**F**	P(**entailment**) tertinggi
XLM-RoBERTa-Large-XNLI is fine-tuned on XNLI (15 languages)	**T**	
It requires labeled training data of the target classes	**F**	Zero-shot

3.15 L15. Kategori task (grid mencocokkan)

Classification	Comparison	Sequential labeling	Seq2seq
Spam filtering, emotion detection, language identification, sentimen dokumen	Plagiarism detection, short answer scoring, paraphrase identification, stance detection, information retrieval	POS tagger, named entity tagger, QA factoid (span jawaban), aspect marker pada ABSA	Machine translation, summarization, chatbot, lexical normalization

4 Pasangan Konsep yang Sering Tertukar

Pasangan	Bedanya
Stemming vs lemmatization	Stemming memotong (*writing* → *writ*, bisa bukan kata valid); lemmatization ke lema valid (*write*)
CBOW vs skip-gram	CBOW: konteks → kata target; skip-gram: kata target → konteks
Word2Vec/GloVe vs BERT	Statis (satu vektor per kata) vs kontekstual (vektor tergantung kalimat)
LSI/SVD vs GloVe	LSI closed-form dari SVD; GloVe dioptimasi dengan gradient (SGD/AdaGrad)
HAL vs LDA	HAL deterministik, co-occurrence lokal dengan bobot jarak dan arah; LDA generatif probabilistik dengan ruang topik
Positional encoding Transformer vs BERT	Transformer asli: sinus-kosinus tetap; BERT: position embedding **dipelajari**
Encoder vs decoder	Encoder melihat seluruh kalimat (bidirectional, NLU); decoder memakai **masked** self-attention (autoregressive, NLG)
Self-attention vs cross-attention	Self: Q, K, V dari sequence yang sama; cross: **Q dari decoder, K dan V dari encoder**
MLM vs NSP	MLM: tebak 15% token yang di-mask; NSP: apakah kalimat B lanjutan kalimat A (dari [CLS])
Fine-tuning vs feature extraction	Fine-tuning melatih ulang seluruh model; feature extraction membekukan model dan memakai embedding sebagai fitur
Zero-shot vs few-shot	Zero-shot hanya instruksi; few-shot (disebut **in-context learning**) menempel contoh berlabel di prompt; keduanya **tanpa gradient update**
Few-shot-CoT vs zero-shot-CoT	Few-shot-CoT: contoh berisi langkah penalaran; zero-shot-CoT: cukup "Let's think step by step"
Self-consistency vs Auto-CoT	Self-consistency: sample beberapa penalaran, ambil jawaban terbanyak; Auto-CoT: cluster pertanyaan lalu bangkitkan demonstrasi otomatis

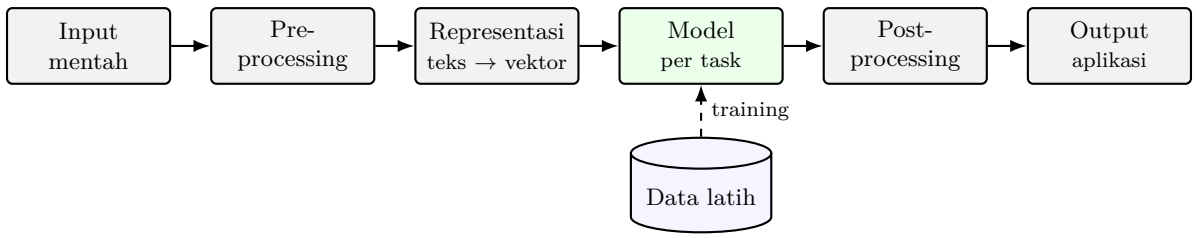
Pasangan	Bedanya
PEFT additive vs selective vs reparameterized	Additive menambah komponen baru (adapter, prefix); selective memilih parameter yang sudah ada (BitFit); reparameterized memakai low-rank yang bisa di-merge (LoRA)
LoRA vs QLoRA	QLoRA = LoRA di atas model dasar terkuantisasi 4-bit
Cross-encoder vs bi-encoder	Cross: dua kalimat satu input, akurat, mahal, tidak bisa di-precompute; bi: encode terpisah, cepat, bisa di-cache
Bi-encoder + classifier vs cross-encoder ("ML mana yang bisa")	Bi-encoder + classifier: input vektor tetap, sehingga ML klasik apa pun bisa; cross-encoder: input token mentah, hanya model neural sequence
Lexical vs semantic similarity	Leksikal: vektor sparse BoW (jarak kata); semantik: sentence embedding dense (jarak makna)
SimCSE unsupervised vs supervised	Unsupervised: positif dari dropout dua kali forward; supervised: data NLI, contradiction sebagai negatif
E5 vs BGE-M3	E5: contrastive pre-training data weakly supervised + fine-tuning dengan hard negative, prefix query/passage; BGE-M3: multilingual, dense + sparse + multi-vector, hingga 8192 token
Text classification vs sequential labeling	Satu label untuk seluruh teks vs satu label per token
Multi class vs multi label	Tepat satu kelas dari banyak kelas vs satu teks bisa punya beberapa label sekaligus

5 Desain Sistem NLP: Rumus IO–TaDa–PRIME

5.1 Sistem NLP adalah Pipeline

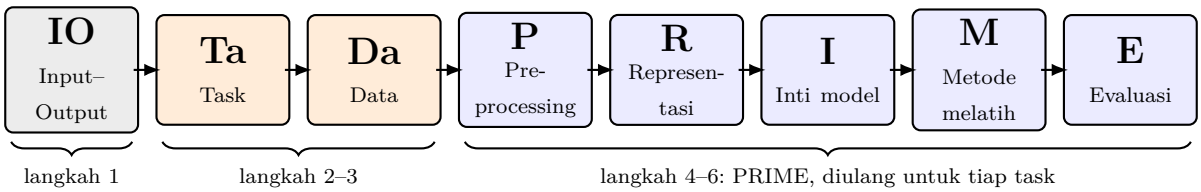
Sebuah **sistem NLP** menerima input bahasa alami (teks, suara, dokumen) dan menghasilkan output yang berguna bagi pemakai (label, ringkasan, jawaban, dashboard). Sistem itu jarang berupa satu model saja. Biasanya ia adalah **pipeline**: rangkaian komponen yang masing-masing menerima output komponen sebelumnya dan mengubahnya ke bentuk berikutnya.

Analogi: pabrik roti. Bahan mentah (teks) dibersihkan dan ditakar (preprocessing), dijadikan adonan yang bisa diolah mesin (representasi angka), dipanggang (model), lalu dikemas sesuai pesanan (post-processing dan tampilan).



5.2 Rumus Enam Langkah: IO–TaDa–PRIME

Rumus ini sejajar dengan enam langkah desain arsitektur di AAT. Bedanya, langkah "pilih komponen" di sini dipecah menjadi lima huruf **PRIME**, karena tiap task NLP selalu melewati lima keputusan yang sama.



Cara menghafal: "keluar-masuk dulu, TaDa!, baru PRIME"

1. **IO (Input–Output):** tulis **satu contoh** input dan output yang konkret.
2. **Ta (Task):** pecah kebutuhan menjadi beberapa task, petakan tiap task ke **4 kategori** (classification, comparison, sequential labeling, seq2seq).
3. **Da (Data):** ada label atau tidak, berapa banyak, bahasa formal atau informal, satu bahasa atau banyak.
4. **P (Preprocessing) dan R (Representasi):** bagaimana teks dibersihkan, lalu diubah menjadi angka.
5. **I (Inti model) dan M (Metode melatih):** model apa untuk kategori itu, dan bagaimana model belajar (ditentukan oleh Da).
6. **E (Evaluasi):** metrik, error analysis, dan trade-off.

Huruf	Langkah	Pertanyaan yang dijawab	Yang ditulis di kertas
IO	Input–Output	Apa yang masuk dan apa yang keluar?	1 contoh konkret
Ta	Task	Task NLP apa saja, dan masuk kategori apa?	Tabel task + kategori
Da	Data	Ada label? Berapa? Gaya bahasa? Berapa bahasa?	Sumber data, cara memberi label
P	Preprocessing	Teks perlu dibersihkan/dipotong bagaimana?	Langkah sesuai jenis model
R	Representasi	Teks diubah menjadi angka bagaimana?	BoW/TF-IDF, embedding, BERT, sentence embedding
I	Inti model	Model apa untuk tiap task?	Mengikuti kategori (5.4)
M	Metode melatih	Bagaimana model belajar task ini?	Pohon keputusan (5.5)
E	Evaluasi	Bagaimana tahu sistem bagus, dan apa kelemahannya?	Metrik, error analysis, trade-off

5.3 Langkah Ta: Kata Kerja di Soal Menentukan Kategori

Kata kerja di soal	Kategori	Pola input → output
"mengelompokkan, mendeteksi, menyaring, menentukan jenis/tingkat/sentimen"	**Classification**	Satu teks → satu label
"mencari yang mirip, mencocokkan, mendeteksi duplikat/plagiat, menilai jawaban terhadap kunci"	**Comparison**	Dua teks → label/skor
"mengambil nama/tanggal/lokasi/produk, menandai bagian kalimat, mencari target/aspek"	**Sequential labeling**	Tiap token → label token
"menjawab pertanyaan dari paragraf (jawaban ada di teks)"	**Sequential labeling** (span)	Token jawaban dilabeli
"meringkas, menerjemahkan, membalas, menulis ulang, menormalkan teks"	**Seq2seq**	Teks → teks baru
"menghubungkan target dengan opininya"	**Other** (relasi), atau classification atas pasangan	(teks, target) → ya/tidak

5.4 Langkah P, R, dan I: Komponen per Kategori

Aturan P (preprocessing) yang mudah diingat: *model leksikal dibersihkan banyak, Transformer dibersihkan sedikit.*

- **Model leksikal** (BoW/TF-IDF + NB/SVM): tokenizer, lowercase, normalisasi, entity masking, stop word removal (kecuali negasi), stemming/lemmatization.
- **Transformer** (BERT, LLM): tokenizer subword **bawaan model** (WordPiece, BPE, SentencePiece) + normalisasi/masking bila teks informal. Stop word dan stemming tidak dilakukan, karena konteks dan urutan kata dipakai model.
- **Teks media sosial**: selalu tambah **word/lexical normalization** dan **masking URL**. @mention dipertahankan bila bisa menjadi target.
- **Dokumen panjang**: **sentence splitter** dan pemotongan per paragraf.

Komponen P	Input → Output	Apa yang dikerjakan	Kapan perlu
Sentence splitter	Dokumen → kalimat	Memutuskan apakah ".", "?", "!" akhir kalimat ("etc." bukan)	Dokumen panjang
Tokenizer / word segmentation	Kalimat → token	Memecah kalimat; bahasa tanpa spasi butuh word segmentation (Maximum Matching)	Selalu
Word normalization	"gk bgt" → "tidak banget"	Singkatan, slang, typo ke bentuk baku	Teks informal
Entity masking	URL, nomor → `<URL>`, `<PHONE>`	Nilai unik diganti token kelas	Spam, tweet
Stop word elimination	Token → token tanpa kata umum	Membuang kata yang tidak membedakan kelas	Model leksikal; jangan buang negasi
Stemming / lemmatization	*writing* → *writ* / *write*	Menyatukan variasi bentuk kata	Model leksikal, data sedikit
Subword tokenizer	Kalimat → token sub-kata → ID	Kata jarang dipecah (`capital`, `##ization`); ID untuk lookup embedding	Wajib untuk Transformer

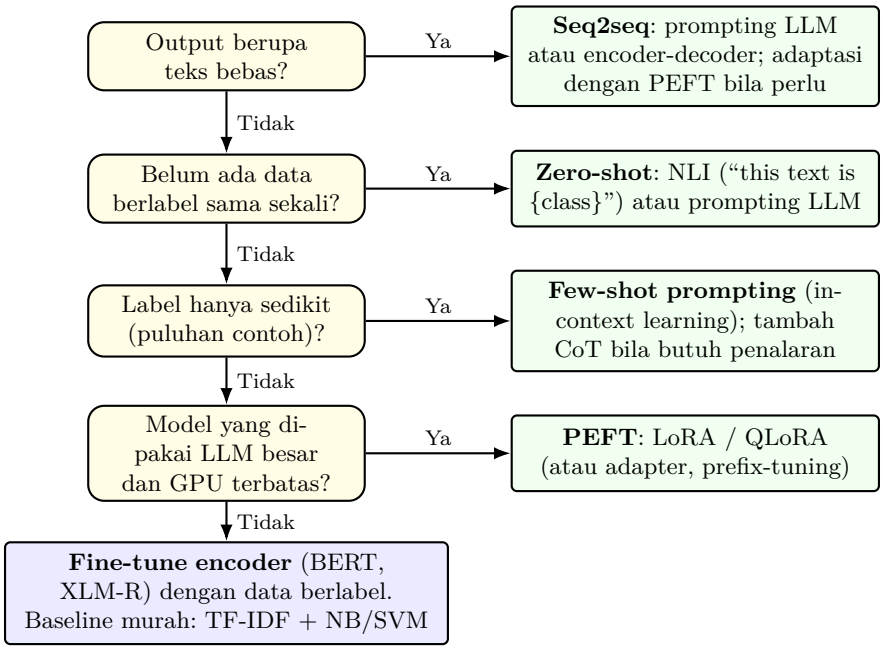
Komponen R	Input → Output	Kelebihan / kekurangan
Bag of words / TF-IDF	Dokumen → vektor sparse	Cepat dan mudah dijelaskan; sinonim dianggap berbeda, urutan hilang
Word embedding statis (Word2Vec, GloVe)	Kata → vektor dense tetap	Menangkap kemiripan makna; satu kata satu vektor ("bisa" racun = "bisa" mampu)
Contextual embedding (BERT)	Token dalam kalimat → vektor tergantung konteks	Paling kaya makna; komputasi lebih besar
Sentence embedding (SBERT, E5, BGE-M3)	Kalimat → satu vektor dense	Dasar pencarian dan perbandingan semantik

Langkah I: model default per kategori. Tulis satu pilihan utama, lalu sebut alternatifnya.

Kategori	Model (klasik → modern)	Bentuk output
Classification	TF-IDF + Naive Bayes/SVM → CNN/LSTM → **BERT + classifier di [CLS]** → zero-shot NLI / prompting LLM	Label + probabilitas
Sequential labeling	Fitur per token + classifier (perlu fitur label sebelumnya) → RNN → **BERT token classification**	Label per token (mis. B-PER, I-PER, O); QA: start/end span
Comparison	Cosine BoW → **bi-encoder + cosine** → bi-encoder + classifier $(u, v, \lvert u - v \rvert)$ → **cross-encoder**	Skor kemiripan atau label (entail, duplicate, relevan)
Seq2seq	Encoder-decoder Transformer (T5) → **decoder LLM + prompting**	Teks baru

5.5 Langkah M: Pohon Keputusan Metode Melatih

Jawab pertanyaan dari atas ke bawah; berhenti di jawaban "Ya" pertama.



Aturan tambahan:

- **Banyak bahasa** → model multilingual (XLM-R, BGE-M3, XLM-R-XNLI).
- **Kelas timpang** → class-weighted loss dan metrik F1 (seperti Lab Work 4).
- **Membandingkan query dengan sangat banyak dokumen** → bi-encoder (vektor dokumen dihitung sekali) lalu cross-encoder untuk re-rank top-k.
- **Harus ringan saat inference** → model kecil (DistilBERT) atau kuantisasi.

5.6 Langkah E: Evaluasi dan Trade-off

Task	Metrik
Classification	Akurasi; **F1** (lebih adil saat kelas timpang)
Sequential labeling	F1 per entitas/label
QA ekstraktif	**EM** (exact match) dan F1
Machine translation	**BLEU**
Comparison	Akurasi/F1 untuk label; korelasi dengan skor manusia untuk skor *(tambahan)*

Selalu tambahkan **error analysis**: kumpulkan contoh yang salah, kelompokkan penyebabnya (sarkasme, slang, negasi, kata baru), lalu perbaiki data atau model. Tutup dengan **trade-off**: apa yang dikorbankan oleh pilihan model (akurasi vs biaya, zero-shot vs fine-tuning, error yang menumpuk di pipeline bertahap).

5.7 Template Jawaban Essay

Yang ditulis di lembar jawaban

1. **IO**: satu kalimat tujuan + satu contoh input dan output.
2. **Ta**: tabel task dan kategorinya.
3. **Gambar pipeline**: kotak dan panah dari input sampai output; tulis kategori di tiap kotak.
4. **Tabel komponen** (inilah "tiap komponen ngapain"):

Komponen	Input → Output	Fungsi	Metode + alasan
…	…	…	…

5. **Da + M**: sumber data, contoh isi training data, dan cara melatih.
6. **E**: metrik, error analysis, trade-off.

6 Essay Lama, Dikerjakan dengan Rumus

Tiga essay desain dan satu essay teknik dari UTS lama yang materinya ada di catatan tahun ini. Setiap essay desain dikerjakan dengan urutan IO–TaDa–PRIME.

6.1 UTS 2020/2021: Arsitektur Hate Speech di Twitter

Soal

Gambarkan arsitektur modul NLP (termasuk detail preprocessing) untuk aplikasi yang mengambil informasi hate speech dari Twitter. Informasinya mencakup **target** hate speech (orang atau organisasi) dan **jenisnya** (extreme hate, dislike, not hate). Jelaskan tiap modul menurut jenis metodenya, seperti text classification, sequence labeling, dan lain-lain.

IO Input–Output

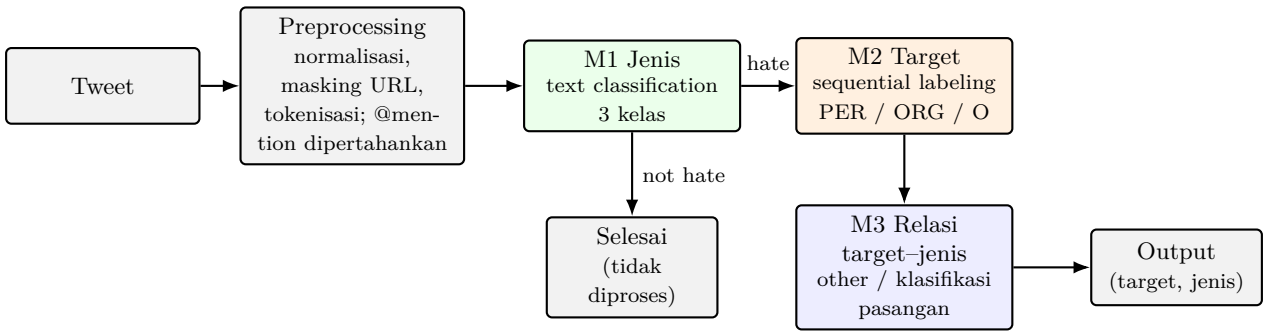
Input: tweet "Dasar @PartaiX tukang bohong, muak liatnya https://t.co/abc". Output: target = PartaiX (organisasi), jenis = dislike.

Ta Task dan kategori

Task	Kategori	Kenapa perlu
M1. Menentukan jenis hate speech	Text classification (3 kelas)	Tingkat kebencian berlaku untuk seluruh tweet
M2. Mengambil target (orang/organisasi)	Sequential labeling (seperti NER)	Target adalah potongan kata di dalam tweet
M3. Memastikan entitas mana yang diserang	Other (relasi), dibuat sebagai klasifikasi pasangan	Satu tweet bisa menyebut beberapa entitas, tidak semuanya sasaran

Da Data

Tweet berbahasa Indonesia informal. Label yang dibutuhkan: M1 berupa tweet berlabel 3 kelas; M2 berupa label per token (PER, ORG, O); M3 berupa pasangan (tweet, entitas) berlabel sasaran atau bukan.



PRIME Komponen

Komponen	Input → Output	Fungsi	Metode + alasan
Preprocessing	Tweet mentah → token	**Word normalization** ("liatnya" → "lihatnya"), **masking URL**, tokenisasi. **@mention tidak dibuang** karena bisa menjadi target; stop word tidak dibuang agar negasi tetap ada	Tokenizer subword bawaan model
M1 Jenis	Tweet → {extreme hate, dislike, not hate}	Menentukan tingkat kebencian seluruh tweet; "not hate" tidak diproses lanjut	BERT + classifier di [CLS], fine-tuning
M2 Target	Token → PER/ORG/O	Menandai orang atau organisasi yang disebut	BERT token classification
M3 Relasi	(tweet, entitas) → sasaran/bukan	Memastikan entitas mana yang benar-benar diserang	BERT sentence pair: `[CLS] tweet [SEP] entitas [SEP]`

M (metode melatih): bila data berlabel sudah ada, fine-tuning ketiganya. Bila belum, M1 bisa memakai zero-shot NLI atau prompting LLM sambil mengumpulkan label.

E Evaluasi dan trade-off

F1 macro untuk M1 (kelas "extreme hate" biasanya jarang), F1 per entitas untuk M2. Error analysis pada sarkasme dan kata kasar yang dipakai bercanda. Trade-off: pipeline bertahap menumpuk error (target yang gagal diekstrak di M2 tidak pernah dinilai di M3).

6.2 UTS 2022/2023 No. 2: NER dengan Tiga Pendekatan

Soal

(a) Gambar dan jelaskan arsitektur proses **training** sistem NER dengan machine learning (algoritma decision tree). Jelaskan peran setiap komponen (termasuk preproses dan fitur beserta alasannya), dan beri contoh output tiap komponen untuk kalimat "Joko Widodo mengunjungi Institut Teknologi Bandung".
(b) Untuk kalimat yang sama, gambar dan jelaskan topologi RNN dengan input hasil word2vec untuk NER.
(c) Untuk kalimat yang sama, gambar dan jelaskan topologi BERT dengan fine-tuning untuk NER.

IO Input–Output

Input: "Joko Widodo mengunjungi Institut Teknologi Bandung". Output: Joko/B-PER, Widodo/I-PER, mengunjungi/O, Institut/B-ORG, Teknologi/I-ORG, Bandung/I-ORG. Skema **BIO** *(tambahan)*: B = awal entitas, I = lanjutan entitas, O = bukan entitas.

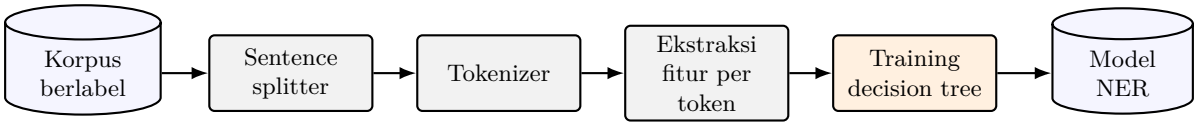
Ta Task

Satu task: NER, kategori **sequential labeling**.

Da Data

Korpus kalimat yang setiap tokennya sudah diberi label BIO.

(a) Decision tree: arsitektur training



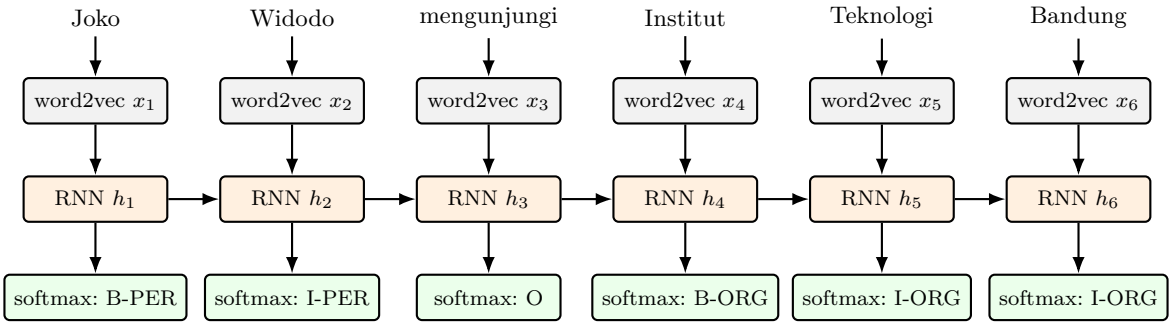
Komponen	Peran	Contoh output
Korpus berlabel	Data latih: label per token	Joko/B-PER Widodo/I-PER mengunjungi/O Institut/B-ORG …
Sentence splitter	Memisah kalimat	1 kalimat
Tokenizer	Memecah kalimat menjadi token	[Joko, Widodo, mengunjungi, Institut, Teknologi, Bandung]
Ekstraksi fitur	Mengubah tiap token menjadi vektor fitur, karena decision tree tidak bisa membaca teks	Tabel fitur di bawah
Training decision tree	Mempelajari aturan if-then dari fitur ke label	Mis. "jika kapital dan label sebelumnya B-ORG maka I-ORG"

Preprocessing yang sengaja tidak dilakukan: lowercasing (huruf kapital adalah petunjuk kuat nama), stemming, dan stop word removal (kata seperti "mengunjungi" adalah konteks penting).

Fitur per token	Alasan dipilih	Contoh untuk "Bandung"
Kata itu sendiri (lowercase)	Kata tertentu sering menjadi entitas	"bandung"
Diawali huruf kapital?	Nama diri ditulis kapital	ya
Kata sebelum dan sesudah	Konteks ("mengunjungi X" → X tempat/organisasi)	sebelum "Teknologi"
Label token sebelumnya	Decision tree tidak punya memori; fitur ini membedakan awal dan lanjutan entitas	I-ORG
Ada di gazetteer (daftar nama)	Daftar nama orang/kota/instansi	ada di daftar kota
Awalan/akhiran	"meng-" cenderung kata kerja, bukan entitas	–

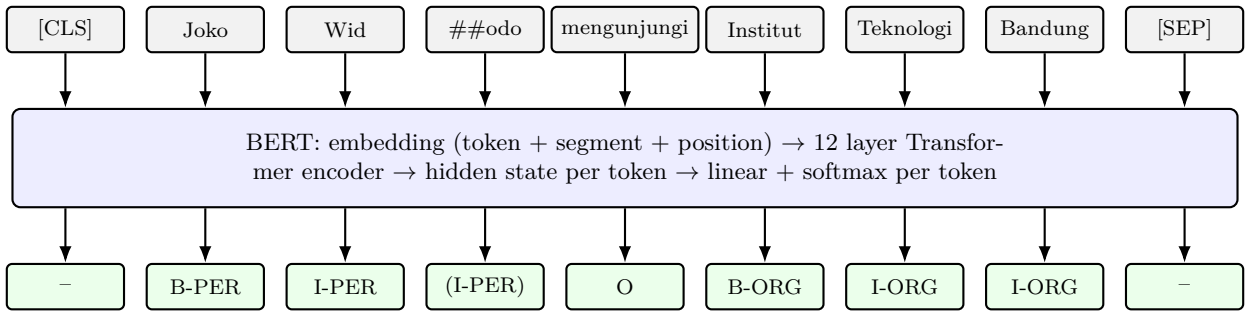
Tanpa fitur label sebelumnya, "Bandung" bisa diberi label lokasi (kota). Dengan fitur itu, ia dikenali sebagai lanjutan nama organisasi (I-ORG).

(b) RNN dengan input word2vec



- Pola **many-to-many sejajar**: satu output per token input.
- **Word2vec** mengubah tiap kata menjadi vektor dense (lookup table; bisa dibekukan atau ikut dilatih).
- **Hidden state** h_t membawa informasi token sebelumnya, sehingga fitur "label sebelumnya" tidak perlu dibuat manual.
- Di tiap posisi, softmax memilih label dari {B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC, O}.
- Varian bidirectional *(tambahan)* juga membaca dari kanan, sehingga label B-ORG pada "Institut" ikut dipengaruhi "Teknologi Bandung" di sebelah kanannya.

(c) BERT dengan fine-tuning



- Kalimat dipecah dengan **WordPiece** ("Widodo" bisa menjadi "Wid" + "##odo"), diapit [CLS] dan [SEP].

- Label diambil dari **hidden state tiap token**, bukan dari [CLS] seperti pada klasifikasi teks.
- Untuk kata yang terpecah, label biasanya hanya dihitung pada sub-kata pertama *(tambahan)*.
- **Fine-tuning**: seluruh model (BERT + linear layer) dilatih ulang dengan data NER berlabel (contoh benchmark di catatan: CoNLL-2003).
- Kelebihan dibanding (a) dan (b): representasi kontekstual dua arah dari pretraining besar, tanpa rekayasa fitur manual.

E Evaluasi

F1 per entitas. Error analysis pada nama yang ambigu ("Bandung" sebagai kota atau bagian nama institusi) dan nama yang tidak pernah muncul di data latih.

6.3 UTS 2022/2023 No. 4: Arsitektur ABSA Modular (Samsung/Canon)

Soal

Gambarkan dan jelaskan arsitektur sistem aspect based sentiment analysis berbasis machine learning yang modular (minimal mencakup aspect classification dan sentiment classification). Sistem harus dapat menangani teks panjang seperti:

"I bought a Samsung camera and my friend bought a Canon camera yesterday. The photos from my Samy are not that great, and the battery life is short too. My friend was very happy with the camera and loves its picture quality."

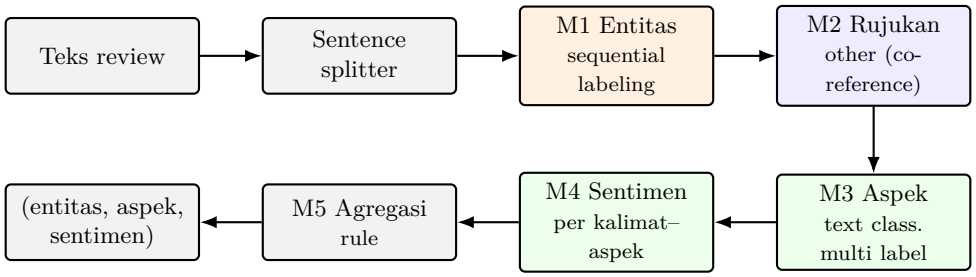
Lengkapi setiap komponen dengan penjelasan teknik dan fitur. Untuk setiap komponen yang memakai machine learning, tunjukkan contoh isi training data. Output: Samsung negatif untuk kualitas foto dan negatif untuk masa hidup baterai; Canon positif untuk kualitas foto.

IO Input–Output

Input: teks review di atas. Output: (Samsung, kualitas foto, negatif), (Samsung, baterai, negatif), (Canon, kualitas foto, positif).

Ta Task dan kategori

Task	Kategori	Kenapa perlu
M1. Ekstraksi entitas produk	Sequential labeling	Menemukan "Samsung", "Canon", "Samy"
M2. Resolusi rujukan	Other (reference resolution)	"my Samy" = Samsung; "the camera", "its" = Canon
M3. Klasifikasi aspek per kalimat	Text classification **multi label**	Satu kalimat bisa membahas dua aspek
M4. Sentimen per (kalimat, aspek)	Text classification	Sentimen bisa berbeda per aspek
M5. Agregasi	Rule	Menggabungkan entitas, aspek, sentimen



PRIME **Komponen, fitur, dan contoh training data**

Komponen	Teknik dan fitur	Contoh training data
Sentence splitter	Rule/ML; memecah teks panjang menjadi 3 kalimat	–
M1 Entitas	Sequential labeling. Fitur: kapital, gazetteer merek, kata sekitar, label sebelumnya (versi BERT: token classification)	I/O bought/O a/O Samsung/B-PROD camera/I-PROD
M2 Rujukan	Other. Fitur: kemiripan string ("Samy" – "Samsung"), kepemilikan ("my" vs "my friend"), jarak penyebutan	("my Samy", "Samsung") = sama; ("the camera" di kalimat 3, "Canon") = sama
M3 Aspek	Text classification multi label (satu sigmoid per aspek). Fitur: TF-IDF kalimat; versi BERT: [CLS] + sigmoid	"The photos …not that great, and the battery life is short too" → {kualitas foto, baterai}
M4 Sentimen	Text classification per (kalimat, aspek). Fitur: TF-IDF + kata opini di sekitar aspek; **negasi "not" dipertahankan**. Versi BERT: `[CLS] kalimat [SEP] aspek [SEP]`	("…photos …not that great …", kualitas foto) → negatif
M5 Agregasi	Rule: gabungkan hasil per kalimat	–

Penelusuran per kalimat:

- Kalimat 1: Samsung dan Canon disebut; belum ada aspek beropini.
- Kalimat 2: "my Samy" = Samsung; aspek {kualitas foto, baterai}; keduanya negatif ("not that great", "short").
- Kalimat 3: "the camera"/"its" = Canon (kamera milik teman); aspek {kualitas foto}; positif ("loves").

E **Evaluasi dan trade-off**

F1 per komponen dan F1 untuk tuple akhir (entitas, aspek, sentimen). Bagian tersulit adalah M2. Alternatif tahun ini: **prompting LLM** yang langsung mengekstrak tuple dari seluruh teks. Lebih sederhana, tetapi lebih mahal per review dan hasilnya sensitif terhadap wording prompt.

6.4 UTS 2022/2023 No. 3: Information Retrieval dengan LSI

Soal

Jelaskan teknik information retrieval menggunakan Latent Semantic Indexing, termasuk penjelasan matriks yang dihasilkan dari teknik LSI tersebut serta teknik untuk membandingkan query input dengan setiap dokumen yang telah dikenai LSI.

Lima langkah LSI

1. **Bangun matriks term × document** A (n term × p dokumen), isinya TF atau TF-IDF.
2. **SVD:** $A_{n\times p} = U_{n\times n}\, S_{n\times p}\, V^T_{p\times p}$.
   - U: kolomnya eigenvector AA^T; **baris U = vektor term** di ruang laten.
   - S: diagonal berisi **singular value** (akar eigenvalue), terurut menurun; kekuatan tiap dimensi laten.
   - V: kolomnya eigenvector A^TA; **baris V = vektor dokumen** di ruang laten.
3. **Reduksi dimensi:** ambil k singular value terbesar (U_k, S_k, V_k). Dimensi kecil yang membawa noise dibuang, dan kata yang sering muncul bersama (mis. sinonim) masuk ke dimensi laten

yang sama.

4. **Petakan query:** query dijadikan vektor term Q (bobot sama seperti A), lalu $q = Q^T U_k S_k^{-1}$. Dokumen baru dipetakan dengan cara yang sama: $d = D^T U_k S_k^{-1}$.
5. **Bandingkan:** hitung **cosine similarity** antara q dan **baris** V_k (pemetaan di langkah 4 menaruh q di koordinat yang sama dengan baris V_k), lalu urutkan dari skor tertinggi.

Kelebihan dibanding VSM biasa: dokumen bisa relevan walau tidak memuat kata query persis (sinonim tertangani), karena perbandingan dilakukan di ruang makna laten.

7 Studi Kasus Prediksi

Empat kasus dengan pola yang sama seperti essay lama, memakai materi tahun ini (BERT, prompting, sentence comparison).

7.1 Kasus A: Pemantau Opini Publik tentang Layanan Kampus

Soal

Kampus ingin memantau keluhan dan pujian mahasiswa tentang layanan kampus (wifi, perpustakaan, kantin, akademik) dari media sosial. Output berupa dashboard: sentimen per aspek layanan per minggu, dan ringkasan keluhan terpenting. Rancang sistem NLP-nya dan jelaskan fungsi tiap komponen.

IO Input–Output

Input: “Wifi perpus lemot bgt dr pagi, tp kantinnya enak sih”. Output: (wifi, negatif), (kantin, positif), lokasi perpustakaan; lalu diagregasi ke dashboard.

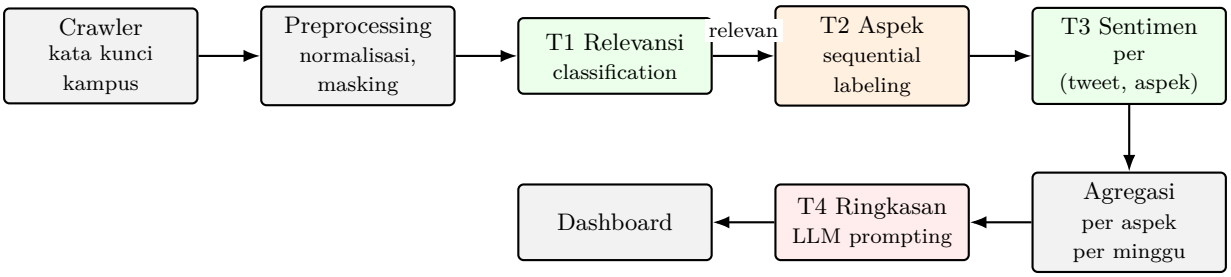
Ta Task dan kategori

Task	Kategori	Kenapa perlu
T1. Filter relevansi (membahas layanan kampus?)	Classification (biner)	Hasil crawling berisi banyak tweet tidak relevan
T2. Ekstraksi aspek (wifi, kantin, …)	Sequential labeling	Satu tweet bisa membahas beberapa aspek
T3. Sentimen per aspek	Classification	Sentimen bisa berbeda per aspek
T4. Ringkasan keluhan mingguan	Seq2seq	Admin tidak sempat membaca ribuan tweet

T2 dan T3 bersama-sama adalah **ABSA**: sequential labeling untuk menandai aspect marker dan sentiment marker, lalu text classification untuk nilai sentimen tiap aspek.

Da Data

Tweet informal (singkatan, typo, campur bahasa). Label belum ada: anotasi sekitar 2.000–3.000 tweet (relevansi, span aspek, sentimen). Sebelum label siap, T1 memakai zero-shot.



PRIME Komponen

Komponen	Input → Output	Fungsi	Metode + alasan
Crawler	Kata kunci → tweet mentah	Mengambil tweet secara berkala	API/scraper *(tambahan)*
Preprocessing	Tweet → tweet bersih	Normalisasi ("bgt" → "banget"), masking URL/@mention, buang retweet; stop word **tidak** dibuang	Teks informal; negasi penting
Tokenizer	Kalimat → token sub-kata + ID	Memetakan ke vocabulary model	Bawaan model (WordPiece/SentencePiece)
T1 Relevansi	Tweet → relevan/tidak	Membuang tweet di luar topik	Awal: zero-shot NLI; setelah ada label: fine-tuning BERT
T2 Aspek	Token → B-ASP/I-ASP/O	Menandai kata aspek, lalu dipetakan ke kategori (wifi → infrastruktur IT)	BERT token classification
T3 Sentimen	(tweet, aspek) → pos/neg/net	Sentimen terhadap aspek tertentu	BERT sentence pair `[CLS] tweet [SEP] aspek [SEP]`
Agregasi	Hasil per tweet → statistik	Persentase sentimen per aspek; buang prediksi berprobabilitas rendah	Post-processing
T4 Ringkasan	Keluhan negatif → 3 poin	Merangkum keluhan utama	Prompting LLM: instruction, context (tweet), output indicator (3 poin)

E Evaluasi dan trade-off

F1 macro untuk T1 dan T3 (kelas timpang, pakai class-weighted loss); F1 per aspek untuk T2. Error analysis: sarkasme ("mantap banget wifinya, loading 5 menit"), negasi, slang baru. Trade-off: zero-shot cepat dipasang tetapi kurang akurat untuk slang lokal; pipeline bertahap menumpuk error. Alternatif: satu LLM dengan prompt ABSA, lebih sederhana tetapi mahal per tweet dan sensitif terhadap wording.

7.2 Kasus B: Chatbot Tanya-Jawab Peraturan Akademik

Soal

Mahasiswa sering bertanya hal yang jawabannya ada di dokumen peraturan akademik (PDF ratusan halaman), mis. "Berapa SKS maksimal kalau IP saya 3,2?". Rancang chatbot yang menjawab berdasarkan dokumen itu.

IO Input–Output

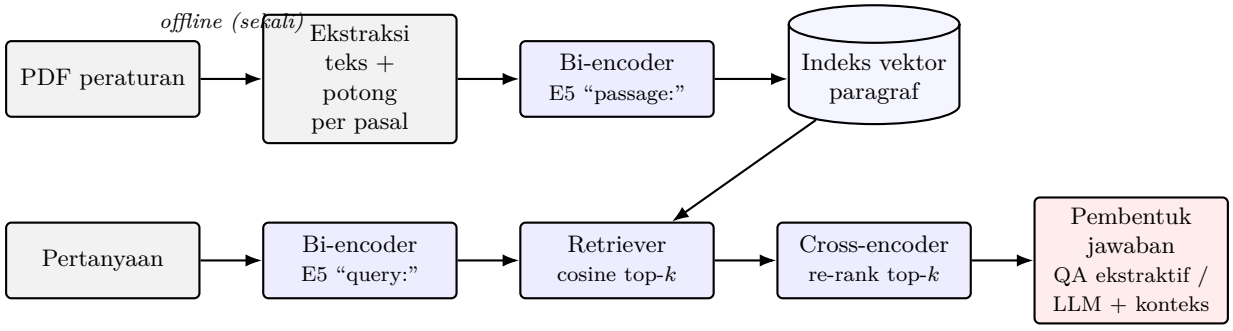
Input: pertanyaan di atas. Output: “Maksimal 24 SKS (Pasal 12 ayat 3)” beserta kutipan pasalnya (angka hanya contoh).

Ta Task dan kategori

Task	Kategori	Fungsi
Mencari paragraf yang relevan	Comparison (information retrieval)	Ratusan halaman → beberapa paragraf
Membentuk jawaban	Sequential labeling (QA ekstraktif) **atau** seq2seq (LLM)	Menghasilkan jawaban
(Opsional) intent: peraturan/di luar topik	Classification	Menolak pertanyaan di luar cakupan

Da Data

Dokumen peraturan (tanpa label). Untuk evaluasi: kumpulan pertanyaan uji beserta jawaban dan pasalnya.



PRIME Komponen

Komponen	Input → Output	Fungsi	Metode + alasan
Ekstraksi + pemotongan	PDF → paragraf per pasal	Mengambil teks (OCR bila scan), memotong dengan sentence splitter	Panjang input model terbatas (BERT 512 token *(tambahan)*)
Bi-encoder (indeks)	Paragraf → vektor dense	Embedding semua paragraf **sekali saja**, disimpan	E5 (prefix “passage:”) atau BGE-M3 (multilingual, hingga 8192 token, hybrid)
Bi-encoder (query)	Pertanyaan → vektor	Embedding pertanyaan saat ditanya	Model sama, prefix “query:”
Retriever	Vektor query + indeks → top-k	Mengambil k paragraf dengan cosine tertinggi	Cepat karena vektor paragraf sudah ada
Re-ranker	(q, paragraf) $\times k$ → urutan baru	Menilai ulang kandidat lebih teliti	**Cross-encoder**: akurat, mahal, tetapi hanya untuk k kandidat
Pembentuk jawaban	Pertanyaan + paragraf → jawaban	(1) **QA ekstraktif**: BERT (SQuAD) menandai span jawaban; (2) **LLM prompting**: instruction “jawab hanya dari konteks”, context (paragraf), input (pertanyaan), output indicator (format + nomor pasal)	Ekstraktif aman tetapi kaku; LLM luwes tetapi bisa mengarang

Pola "cari dokumen dulu, lalu LLM menjawab dari dokumen itu" dikenal sebagai RAG (Retrieval-Augmented Generation) *(tambahan)*; semua komponennya ada di materi.

E Evaluasi dan trade-off

Retriever: apakah paragraf benar ada di top-k *(tambahan)*; jawaban: EM dan F1. Error analysis: parafrase jauh ("jatah SKS" vs "beban studi maksimum"), pertanyaan yang butuh dua pasal. Trade-off: bi-encoder cepat tetapi kadang salah urut (maka ditambah cross-encoder); LLM butuh GPU/API dan bisa mengarang.

7.3 Kasus C: Penilaian Otomatis Jawaban Singkat

Soal

Dosen ingin sistem yang menilai jawaban singkat mahasiswa (1–3 kalimat) dengan membandingkannya ke kunci jawaban, memberi skor 0–5.

IO Input–Output

Input: (jawaban mahasiswa, kunci jawaban). Output: skor 0–5.

Ta Task

Satu task **comparison** (short answer scoring adalah contoh text comparison di catatan pengantar).

Da Data

Jawaban semester lalu yang sudah dinilai dosen: pasangan (jawaban, kunci, skor).

PRIME Pilihan model untuk langkah I

Pendekatan	Cara kerja	Kelebihan	Kekurangan
Cosine leksikal (BoW/TF-IDF)	Hitung kata yang sama	Sederhana	Parafrase yang benar mendapat skor rendah
Bi-encoder + cosine (SBERT)	Encode terpisah, cosine, petakan ke skor	Menangkap makna, cepat	Kurang peka detail ("bukan", angka)
Cross-encoder regresi	`[CLS] jawaban [SEP] kunci [SEP]` → skor, di-fine-tune seperti STS-B	**Paling akurat**	Lebih mahal, tetapi pasangannya sedikit (1 kunci per jawaban), jadi layak
Pemeriksaan NLI	Kunci = premise, jawaban = hypothesis	Mendeteksi jawaban yang **bertentangan** walau kata-katanya mirip	Butuh model NLI multilingual

Rancangan yang dipilih: (P) normalisasi typo tanpa membuang stop word; (I) cross-encoder XLM-R; (M) fine-tuning pada data penilaian dosen; plus pemeriksaan NLI: bila "contradiction" tinggi, skor dibatasi rendah. Jawaban dengan keyakinan rendah ditandai untuk diperiksa dosen *(tambahan)*.

E **Evaluasi**

Korelasi skor sistem dengan skor dosen *(tambahan)*; akurasi skor yang dibulatkan; error analysis pada jawaban benar yang memakai istilah lain.

7.4 Kasus D: Pemrosesan Email Layanan Akademik

Soal

Bagian akademik menerima ratusan email per hari (izin, permohonan surat, keluhan nilai). Rancang sistem yang meneruskan email ke unit yang tepat, mengambil data penting, dan membuat ringkasan untuk petugas.

IO **Input–Output**

Input: email "Yth. Bapak/Ibu, saya Budi (NIM 13523xxx) mohon surat aktif kuliah untuk beasiswa, tenggat Jumat." Output: unit = kemahasiswaan, prioritas = tinggi, data = {nama: Budi, NIM, tenggat: Jumat}, ringkasan satu kalimat.

Ta **Task dan kategori**

Task	Kategori	Komponen dan fungsi
Routing ke unit	Classification multi kelas	BERT fine-tuned pada email lama berlabel unit
Prioritas	Classification	Menandai email mendesak (tenggat dekat)
Ekstraksi data	Sequential labeling (NER)	Mengambil nama, NIM, mata kuliah, tanggal untuk mengisi formulir
Ringkasan	Seq2seq	LLM meringkas email panjang menjadi 1–2 kalimat
Deteksi duplikat (opsional)	Comparison	Bi-encoder + cosine untuk email yang dikirim berulang

Da **Data**

Email lama yang sudah diteruskan ke unit sudah menjadi **label gratis** untuk routing. NER butuh anotasi (mulai dari sedikit data). Ringkasan memakai prompting LLM zero-shot karena tidak ada contoh ringkasan.

PRIME **Pipeline**

Email → preprocessing (buang signature dan kutipan balasan, masking nomor/URL) → tokenizer → [routing, prioritas, NER berjalan paralel] → ringkasan → antrean unit tujuan.

E **Evaluasi dan trade-off**

F1 routing per unit; F1 per entitas NER; penilaian manusia untuk ringkasan *(tambahan)*. Email salah rute dikoreksi petugas, dan koreksinya dipakai sebagai data latih baru. Ringkasan LLM bisa menghilangkan detail, sehingga email asli tetap ditampilkan.

8 Ringkasan Inti

Bawa ke ruang ujian (di kepala)

- **PG = grid True/False dan grid mencocokkan.** Nilai tiap pernyataan sendiri. Lima jebakan: konsep ditukar, klaim "sama", contoh salah kategori, arah efek terbalik, kemampuan berlebihan.
- **Essay = gambar pipeline untuk kasus nyata.** Rumusnya **IO** → **TaDa** → **PRIME**:

  - **IO**: satu contoh input dan output;
  - **Ta**: task → 4 kategori (kata kerja di soal menentukan kategori);
  - **Da**: label ada/berapa, bahasa formal/informal;
  - **P**: leksikal dibersihkan banyak, Transformer sedikit (jangan buang negasi);
  - **R**: TF-IDF / embedding statis / BERT / sentence embedding;
  - **I**: classification → BERT + [CLS]; sequential labeling → BERT per token; comparison → bi-encoder lalu cross-encoder; seq2seq → LLM/encoder-decoder;
  - **M**: teks bebas → seq2seq; tanpa label → zero-shot; label sedikit → few-shot; LLM + GPU terbatas → LoRA/QLoRA; selain itu fine-tune encoder;
  - **E**: F1 (kelas timpang), EM/F1 (QA), BLEU (MT), error analysis, trade-off.
- **Kasus favorit soal lama**: opini/sentimen berbasis target atau aspek (hate speech + target, ABSA Samsung/Canon), dan satu task dengan beberapa pendekatan (decision tree, RNN + word2vec, BERT). Tahun ini siapkan juga versi prompting LLM dan PEFT.
- **Selalu tulis** kategori di tiap kotak pipeline, contoh output tiap komponen, dan contoh isi training data.