Pertemuan 16 — UAS: Tugas Besar

Melatih *neural network*, mendiagnosisnya, lalu mempertanggungjawabkan hasilnya

Hendri Karisma

Semester Ganjil 2026/2027

Program Studi Teknik Informatika, STMIK Tazkia

Alur pertemuan hari ini

1. Tujuan dan kaitan dengan CPMK
2. Dari UTS ke UAS
3. Tiga jalur proyek dan pilihan data
4. Kebutuhan wajib dan nilai lebih
5. Protokol eksperimen dan diagnosis
6. Analisis, *cost*, dan etika
7. Repositori, laporan, dan presentasi
8. Rubrik penilaian
9. Integritas akademik, aturan, dan jadwal
10. Pertanyaan yang sering muncul

Weight dan cakupan

UAS bernilai **45%** nilai akhir, mengukur CPMK-4 dan CPMK-5, dan mencakup materi pertemuan 9–15: *linear regression*, evaluasi, dan *gradient descent*, *perceptron*, MLP dan *backpropagation*, praktik melatih jaringan, *deep neural network*, CNN, *attention* dan *transformer*, serta LLM dan *generative AI*.

Tujuan

Mengapa UAS berbentuk proyek

Pada UTS, model dilatih oleh satu panggilan `fit()` dan yang dinilai adalah cara Anda membandingkannya. Pada UAS, **proses *training* itu sendiri** menjadi bahan yang dinilai:

- Mengapa *loss* berhenti turun pada *epoch* ke-12?
- *Learning rate* berapa yang dipilih, dan apa buktinya?
- Apakah model dalam memang perlu, atau MLP dua *layer* sudah cukup?
- Berapa jam GPU yang dihabiskan, dan sepadankah hasilnya?

Yang Anda hasilkan

Satu proyek *neural network* utuh: masalah nyata, patokan non-saraf yang kuat, model saraf yang dilatih dengan *training loop* tulisan Anda sendiri, catatan eksperimen yang lengkap, evaluasi yang jujur, dan pembahasan keterbatasan, *cost*, serta etikanya.

Ringkasan

Kurva *training* yang dibaca dengan benar lebih berharga daripada angka akhir yang tidak dapat dijelaskan asal-usulnya.

Kaitan dengan CPMK

CPMK	Kemampuan yang diukur	Bukti di tugas besar
CPMK-4	*Loss function*, *gradient descent* dan SGD, *perceptron*, MLP, *backpropagation*, DNN, CNN	Uji *gradient* yang lulus; *training loop* sendiri; sapuan *learning rate*; kurva *training* yang didiagnosis; perbandingan kapasitas atau *transfer learning*
CPMK-5	*Transformer*, LLM, *generative AI*; keterbatasan dan etika; komunikasi hasil	*Attention* yang diterapkan dan diuji (jalur B); pembahasan posisi LLM terhadap masalah Anda; *computational cost* dan emisi; laporan dan presentasi

Setiap komponen rubrik menunjuk ke salah satu baris tabel ini. Bagian yang tidak punya bukti di repositori atau laporan tidak dapat diberi nilai.

Dari UTS ke UAS: yang tetap dan yang baru

Yang tetap

- Kelompok **3–4 orang**; boleh sama dengan kelompok UTS
- Laporan **8–12 halaman**, repositori kode, presentasi
- *Test data* dikunci sejak awal dan dipakai sekali
- *Metric* beralasan *cost of error*, disertai *confidence interval*
- Kartu data, `AI-USAGE.md`, penilaian sejawat

Yang baru

- *Training loop* PyTorch ditulis sendiri, dengan kurva *loss* dan *metric* per *epoch*
- Uji *gradient*: *backpropagation* dicocokkan dengan selisih hingga
- Catatan eksperimen untuk setiap percobaan, termasuk yang gagal
- *Configuration* akhir dijalankan dengan **tiga *seed***
- *Computational cost* dicatat dalam jam GPU dan estimasi emisi
- Demo inferensi dari *checkpoint* tersimpan

Jalur tabular (jalur C) memakai model terbaik UTS sebagai patokan yang harus dikalahkan, pada *test data* yang sama.

Tiga jalur proyek

Tiga jalur proyek

Jalur A — citra

Classification citra dengan CNN.

- CNN yang dirancang sendiri
- *Transfer learning* dari model terlatih
- Patokan: MLP atas piksel, atau model klasik atas *feature* sederhana

Materi utama: P11–P13.

Jalur B — teks

Classification teks berbahasa Indonesia.

- *Attention* ditulis dan diuji sendiri
- *Fine-tuning transformer* terlatih
- Patokan: TF-IDF dengan *logistic regression*

Materi utama: P11, P14–P15.

Jalur C — tabular

Melanjutkan *dataset* UTS.

- MLP dari nol dengan NumPy
- MLP PyTorch dengan *regularization*
- Patokan: model terbaik UTS pada pembagian yang sama

Materi utama: P9–P12.

Ringkasan

Ketiga jalur dinilai dengan rubrik yang sama. Jalur C tampak paling ringan, dan karena itu dituntut analisis yang lebih tajam: kapan *neural network* tidak sepadan dengan *cost*-nya pada data tabular.

Contoh masalah dan data

Jalur	Masalah	Sumber data	Catatan
A	Penyakit daun padi atau tanaman pangan	*Dataset* citra daun terbuka (UCI, Kaggle, Mendeley Data)	Citra diambil di lahan atau di laboratorium? Ini menentukan *generalization*
A	Aksara Jawa tulisan tangan	*Dataset* hanacaraka terbuka	Banyak citra dari satu penulis: bagi per penulis bila datanya ada
A	Jenis sampah untuk pemilahan	TrashNet atau padanannya	Kelas tidak seimbang; *augmentation* wajib dibahas
B	Sentimen atau emosi teks	SmSA dan EmoT (IndoNLU)	Bandingkan dengan hasil acuan IndoNLU
B	Ujaran kebencian dan kata kasar	Korpus Twitter berbahasa Indonesia yang terbuka	Isu etika dan bias anotator wajib dibahas
C	Lanjutan masalah UTS	*Dataset* UTS kelompok Anda	Minimal 2.000 baris agar MLP layak dicoba

MNIST, Fashion-MNIST, dan CIFAR-10 sudah dipakai di praktikum, sehingga **tidak diterima** sebagai data utama. Boleh dipakai untuk menguji kode sebelum data utama siap.

Syarat data dan komputasi

Data

- Data terbuka dengan lisensi yang mengizinkan pemakaian pendidikan, atau data mitra dengan izin tertulis
- Tidak memuat wajah atau identitas orang yang dapat dikenali
- Kartu data seperti UTS, ditambah resolusi citra atau panjang teks
- Satu *dataset* paling banyak dua kelompok, dengan pertanyaan berbeda

Komputasi

- Semua *training* harus dapat dijalankan di GPU gratis Google Colab atau Kaggle
- Anggaran wajar: total sekitar 10 jam GPU per kelompok untuk seluruh eksperimen
- Model terlatih yang dipakai berukuran paling besar sekitar ukuran *base* (puluhan sampai ratusan juta parameter)
- Simpan *checkpoint* secara berkala; sesi Colab dapat terputus sewaktu-waktu

Ringkasan

Anggaran komputasi adalah bagian dari soal. Kelompok yang mencapai hasil setara dengan jam GPU lebih sedikit menunjukkan pemahaman yang lebih baik tentang apa yang sebenarnya bekerja.

Kebutuhan wajib

Kebutuhan wajib untuk semua jalur (W1–W10)

W1 **Perumusan masalah dan kartu data**: *T*, *P*, *E*, satuan observasi, cara membagi data.

W2 **Patokan non-saraf yang kuat**, disetel dengan cara UTS.

W3 **Uji *gradient***: *backpropagation* satu jaringan kecil dicocokkan dengan selisih hingga.

W4 **Arsitektur beralasan**: diagram, jumlah parameter, dan alasan tiap pilihan (aktivasi, *normalization*, ukuran).

W5 ***Training loop* sendiri** di PyTorch: *optimizer*, *scheduler*, *early stopping*, *training curve* dan validasi per *epoch*.

W6 **Eksperimen tercatat**: sapuan *learning rate* minimal tiga nilai, minimal satu teknik *regularization*, dan satu perbandingan kapasitas atau *transfer learning*.

W7 **Evaluasi akhir**: tiga *seed*, rata-rata $\pm$ simpangan baku, *confidence interval* pada *test data*, dibandingkan dengan patokan.

W8 **Ablasi dan analisis *error***: satu komponen dilepas untuk diukur sumbangannya; contoh kesalahan dianalisis.

W9 **Keterbatasan, *cost*, dan etika**: daftar periksa etika data dan *model card* (Pertemuan 12), serta posisi LLM terhadap masalah Anda.

W10 **Reproduktibilitas**: *seed*, `requirements.txt`, *checkpoint*, dan *notebook* inferensi yang berjalan.

Bila salah satu kebutuhan wajib tidak dipenuhi, nilai maksimum komponen rubrik yang terkait dibatasi setengahnya.

Kebutuhan khusus per jalur

Jalur	Wajib	Yang dibahas di laporan
A — citra	CNN rancangan sendiri ($\geq$ 2 blok konvolusi), satu model *transfer learning* (misalnya ResNet-18 atau MobileNet), *data augmentation*	Tabel dimensi *output* tiap *layer* dan jumlah parameternya; perbandingan dilatih dari nol dan dari model terlatih; contoh citra yang salah
B — teks	*Scaled dot-product attention* ditulis sendiri dan dicocokkan dengan fungsi PyTorch; *fine-tuning* satu *transformer* terlatih (misalnya IndoBERT)	Pengaruh *tokenization* dan panjang maksimum teks; perbandingan dengan TF-IDF; contoh teks yang salah dan pola bahasanya
C — tabular	MLP dengan *forward* dan *backward* ditulis dengan NumPy, lalu MLP PyTorch dengan *regularization*; model terbaik UTS dilatih ulang pada pembagian yang sama	Kapan MLP mengalahkan model UTS dan kapan kalah; *training cost* dan *inference cost* keduanya; sensitivitas terhadap *scaling*

Model terlatih apa pun yang dipakai wajib disebut nama, versi, dan sumbernya, beserta lisensinya.

Uji *gradient* (W3): membuktikan *backpropagation* Anda benar

Untuk setiap parameter θ_j pada jaringan kecil, bandingkan *gradient* hasil *backpropagation* g_j dengan hampiran selisih hingga terpusat:

$$\tilde{g}_j = \frac{L(\theta + \varepsilon e_j) - L(\theta - \varepsilon e_j)}{2\varepsilon}, \qquad \varepsilon = 10^{-5}$$

Lalu hitung *error* relatifnya:

$$r = \frac{\|g - \tilde{g}\|}{\|g\| + \|\tilde{g}\|}$$

Dengan *precision* `float64`, nilai $r < 10^{-6}$ menandakan implementasi benar; $r > 10^{-3}$ hampir pasti menandakan *error* pada *derivative*.

Cara mengerjakannya

- Jaringan kecil: misalnya 4 *input*, 5 unit tersembunyi, 3 kelas
- Jalur C: uji *backward* NumPy Anda sendiri
- Jalur A dan B: uji satu *layer* buatan sendiri, atau bandingkan `torch.autograd` dengan selisih hingga lewat `torch.autograd.gradcheck`
- Laporkan *r* per *layer* dalam satu tabel kecil

Uji ini menyambung langsung ke Pertemuan 10: aturan rantai yang Anda hitung dengan tangan kini diperiksa oleh mesin.

Nilai lebih (tidak wajib, maksimal +10%)

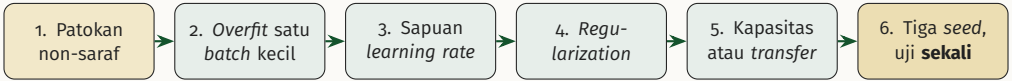
- Visualisasi filter, *feature map*, atau peta perhatian (Grad-CAM untuk citra, *weight attention* untuk teks)
- Pembanding LLM: *zero-shot* atau *few-shot prompting* pada *test data* yang sama, dengan catatan risiko kontaminasi data
- Kalibrasi peluang *output* model
- Kurva belajar: kinerja terhadap jumlah *training data*
- *Model compression*: kuantisasi atau distilasi, dengan ukuran dan *inference time* sebelum dan sesudah
- Aplikasi demo sederhana (misalnya Gradio) yang memuat *checkpoint*

Ringkasan

Nilai lebih hanya dihitung bila W1–W10 sudah terpenuhi. Aplikasi demo yang menarik tidak menutup kurva *training* yang tidak dibahas.

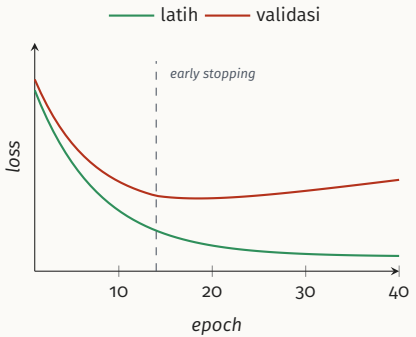
Protokol eksperimen dan diagnosis

Urutan eksperimen yang dianjurkan



- **Langkah 2** adalah pemeriksaan termurah. Jaringan yang tidak sanggup menghafal 32 contoh hampir pasti punya *error* di kode, *label*, atau *learning rate*.
- **Langkah 3**: coba skala logaritmik, misalnya 10^{-4}, 10^{-3}, 10^{-2}, dengan *epoch* pendek.
- **Langkah 4–5**: ubah satu hal setiap kali, lalu catat. Semua pemilihan memakai *validation data*.
- **Langkah 6**: *configuration* akhir dilatih ulang dengan tiga *seed*, lalu dievaluasi pada *test data* untuk pertama dan terakhir kali.

Membaca kurva *training*



Ilustrasi bentuk kurva, bukan hasil nyata.

Gejala	Dugaan	Tindakan
Loss latih tidak turun	*learning rate* salah, *error* kode	langkah 2, sapuan *learning rate*
Loss melonjak atau `NaN`	*learning rate* terlalu besar, *gradient* meledak	turunkan, *gradient clipping*
Latih turun, validasi naik	*overfitting*	*regularization*, *augmentation*, *early stopping*
Keduanya tinggi dan datar	*underfitting*	tambah kapasitas, latih lebih lama
Validasi lebih baik dari latih	*dropout* aktif saat latih, atau kebocoran	periksa `model.eval()` dan pembagian data

Setiap kurva di laporan disertai satu paragraf diagnosis: gejala apa yang terlihat, dugaan penyebabnya, dan tindakan yang Anda ambil sesudahnya.

Catatan eksperimen dan tabel hasil yang diminta

Catatan eksperimen (`hasil/eksperimen.csv`), satu baris per percobaan, termasuk yang gagal:

ID	Perubahan	LR	Val F1	Menit GPU
E01	patokan CNN	10^{-3}	…	…
E02	LR lebih besar	10^{-2}	…	…
E03	+ *augmentation*	10^{-3}	…	…
E04	+ *dropout* 0,3	10^{-3}	…	…
E05	ResNet-18 terlatih	10^{-4}	…	…

Kolom tambahan: *seed*, *epoch* terbaik, ukuran *batch*, catatan.

Tabel hasil akhir pada *test data*:

Model	F1 (3 *seed*)	CI 95%	Param.
Patokan non-saraf	…	…	…
Model saraf rancangan	…± …	…	…
Model terlatih	…± …	…	…
Ablasi (tanpa X)	…± …	…	…

Tambahkan kolom *training time* dan *inference time* per 1000 contoh. Untuk *regression*, ganti F1 dengan RMSE.

Ringkasan

Percobaan yang gagal tetap dicatat. Deretan percobaan yang gagal dan dijelaskan dengan baik menunjukkan pemahaman, dan membuktikan bahwa angka akhir tidak dipilih secara kebetulan.

Analisis, *cost*, dan etika

Analisis yang diminta (W8)

Ablasi

Lepaskan satu komponen dari *configuration* akhir, latih ulang dengan tiga *seed*, dan ukur selisihnya. Contoh komponen: *augmentation*, *batch normalization*, *residual connection*, *weight* terlatih, atau satu kelompok *feature*. Selisih yang lebih kecil dari simpangan baku antar-*seed* belum dapat diklaim sebagai sumbangan.

Analisis *error*

Ambil minimal sepuluh contoh yang salah ditebak, kelompokkan menurut pola (citra buram, teks bernada sarkas, kelas yang mirip), dan tunjukkan *confusion matrix*.

Posisi LLM terhadap masalah Anda (W9)

Satu subbab yang menjawab: dapatkah model fondasi atau LLM menyelesaikan masalah ini? Dengan *cost*, *latency*, dan risiko apa? Kapan model kecil yang Anda latih lebih tepat? Jawaban boleh berupa argumen dengan rujukan; eksperimen LLM sungguhan masuk nilai lebih.

Keterbatasan

Data apa yang tidak terwakili, kelompok mana yang mungkin dirugikan oleh kesalahan model, dan dalam kondisi apa model ini sebaiknya tidak dipakai.

Computational cost dan estimasi emisi

Catat menit GPU setiap percobaan, lalu jumlahkan. Estimasi energi dan emisi:

$$E\,[\text{kWh}] = \frac{P_{\text{GPU}}\,[\text{W}] \times t\,[\text{jam}] \times \text{PUE}}{1000}$$

$$\text{CO}_2\text{e}\,[\text{kg}] = E \times f_{grid}\,[\text{kg CO}_2\text{e/kWh}]$$

- P_{GPU}: daya rancangan GPU dari lembar spesifikasi resmi
- PUE: efisiensi pusat data; pakai angka yang dirujuk, atau 1,5 sebagai asumsi yang dinyatakan
- f_{grid}: faktor emisi jaringan listrik lokasi pusat data, dari sumber resmi yang dicantumkan

Mengapa diminta

Pertemuan 15 membahas jejak karbon LLM. Proyek kecil Anda memakai rumus yang sama dengan skala yang jauh lebih kecil, sehingga Anda dapat membayangkan *cost* model yang dilatih ribuan GPU selama berminggu-minggu.

Ringkasan

Yang dinilai adalah model perhitungan dan asumsinya. Angka mutlaknya kasar, dan itu wajar selama asumsinya ditulis.

Repositori, laporan, dan presentasi

Struktur repositori

`uas-ml-kelompok-XX/`	
`README.md`	cara jalan, pembagian kerja
`requirements.txt`	versi paket disematkan
`data/README.md`	cara unduh, lisensi
`data/split_uji.csv`	indeks *test data*
`src/model.py`	arsitektur
`src/latih.py`	*training loop*
`src/uji_gradien.py`	W3
`configs/`	satu *file* per eksperimen
`notebooks/`	eksplorasi, analisis
`notebooks/inferensi.ipynb`	demo dari *checkpoint*
`hasil/eksperimen.csv`	catatan eksperimen
`hasil/kurva/`	kurva tiap eksperimen
`laporan/laporan.pdf`	8–12 halaman
`AI-USAGE.md`	catatan alat bantu AI

- *Checkpoint* akhir tidak di-*commit*. Unggah ke Google Drive atau Hugging Face, lalu cantumkan tautan dan *checksum*-nya.
- `inferensi.ipynb` memuat *checkpoint* dan menghasilkan ulang angka *test data* dalam waktu kurang dari 10 menit di Colab.
- `latih.py` dapat dijalankan dengan satu *file configuration* untuk mengulang eksperimen mana pun.
- Riwayat *commit* memperlihatkan kontribusi setiap anggota.

Templat laporan (8–12 halaman, lampiran di luar hitungan)

1. **Ringkasan** — masalah, model terbaik, angka utama dengan CI
2. **Perumusan masalah dan data** — *T*, *P*, *E*; kartu data; pembagian
3. **Patokan** — model non-saraf dan cara *tuning*-nya
4. **Rancangan model** — diagram, jumlah parameter, alasan; uji *gradient*
5. ***Training*** — *optimizer*, *scheduler*, *regularization*, anggaran GPU
6. **Eksperimen** — ringkasan catatan eksperimen dan kurva terpilih beserta diagnosisnya
7. **Hasil** — tabel akhir tiga *seed*, perbandingan dengan patokan
8. **Analisis** — ablasi, analisis *error*, posisi LLM
9. **Keterbatasan, *cost*, dan etika**
10. **Kesimpulan**
11. **Lampiran** — catatan eksperimen lengkap, daftar periksa etika, *model card*, pembagian kerja, `AI-USAGE.md`

Ringkasan

Tulis analisis. “Kami memakai Adam” tidak menambah nilai; “Adam dengan $\eta = 10^{-3}$ dipilih karena 10^{-2} membuat *loss* melonjak pada *epoch* ketiga (E02), terlihat pada Gambar 4” menambah nilai.

Dua presentasi: Pertemuan 15 dan Pertemuan 16

Pertemuan 15 — presentasi proyek

Sesuai RPS: masalah, data, metode, hasil sementara, dan keterbatasan.

- 10 menit per kelompok, termasuk tanya jawab
- Dinilai sebagai tugas (komponen “tugas dan kuis”), terpisah dari nilai UAS
- Umpan balik dosen dan kelompok lain dipakai untuk memperbaiki proyek
- *Test data* belum boleh dibuka

Pertemuan 16 — presentasi akhir (UAS)

Masalah, rancangan, eksperimen	6 menit
Hasil, analisis, keterbatasan	6 menit
Demo inferensi dari *checkpoint*	3 menit
Tanya jawab	6 menit

Sesi UAS berlangsung 100 menit; bila kelompok melebihi kapasitas, sebagian dijadwalkan pada sesi tambahan di minggu yang sama.

Ringkasan

Pada presentasi akhir, tampilkan perubahan sejak Pertemuan 15 dan bagaimana umpan balik ditindaklanjuti. Setiap anggota wajib berbicara dan menjawab minimal satu pertanyaan.

Rubrik penilaian

Rubrik penilaian

Komponen	*Weight*	Nilai penuh bila
Rancangan model dan data	15%	Masalah dan kartu data jelas; patokan kuat; arsitektur beralasan; uji *gradient* lulus (W1–W4)
Training dan diagnosis	20%	*Training loop* sendiri; sapuan *learning rate*; *regularization*; setiap kurva disertai diagnosis; eksperimen tercatat (W5, W6)
Evaluasi dan pembanding	20%	*Test data* dipakai sekali; tiga *seed*; CI; perbandingan adil dengan patokan; reproduktibel (W7, W10)
Analisis, keterbatasan, dan etika	20%	Ablasi bermakna; analisis *error* berpola; *cost* dan emisi; daftar periksa etika dan *model card*; posisi LLM (W8, W9)
Laporan	10%	Templat diikuti, 8–12 halaman, gambar terbaca, analitis
Presentasi dan demo	15%	Demo inferensi berhasil; tepat waktu; semua anggota menjawab dengan paham
Nilai lebih	+10%	*Visualisasi, pembanding LLM, kalibrasi, kurva belajar, compression, aplikasi demo*

Komponen mengikuti RPS: rancangan, *training*, evaluasi, analisis, laporan, ditambah presentasi. Nilai akhir UAS dibatasi 100. UAS bernilai 45% dari nilai mata kuliah.

Tingkat capaian untuk tiga komponen terbesar

Komponen	Sangat baik (85–100)	Cukup (60–84)	Kurang (<60)
Training dan diagnosis (20%)	Urutan eksperimen runtut; setiap keputusan merujuk kurva atau ID eksperimen; percobaan gagal dijelaskan	Kurva ada, tetapi diagnosis umum dan keputusan tidak dirujuk ke bukti	Memakai `Trainer` atau `fit()` siap pakai tanpa *loop* sendiri; kurva tidak dibahas
Evaluasi (20%)	Tiga *seed*, CI, *test data* terbukti dipakai sekali, patokan disetel setara	Hasil satu *seed*, atau patokan memakai nilai bawaan	*Test data* dipakai untuk memilih model, atau tidak ada patokan; komponen maksimal setengah
Analisis (20%)	Ablasi dengan tiga *seed*; *error* dikelompokkan menurut pola; *cost* dan posisi LLM dibahas dengan rujukan	Ablasi ada tanpa pengulangan; analisis *error* berupa daftar contoh	Hanya mengulang angka tabel; tidak ada keterbatasan

Komponen lain dinilai dengan pola yang sama. Nilai individu disesuaikan dengan aturan UTS: tanya jawab, riwayat *commit*, dan penilaian sejawat, maksimal ± 15 poin.

Integritas akademik, aturan, dan jadwal

Integritas akademik

Alat bantu AI dan LLM

- Boleh dipakai untuk menjelaskan *error*, mencari dokumentasi, atau merapikan kode.
- Tuliskan di `AI-USAGE.md`: alat apa, untuk bagian mana, dan apa yang Anda periksa atau ubah.
- Uji *gradient*, *training loop*, dan *attention* (jalur B) harus dapat dijelaskan baris demi baris oleh setiap anggota.
- Bila LLM dipakai sebagai **model pembanding** di dalam eksperimen, itu bagian dari metode: catat nama model, versi, tanggal, dan *prompt*-nya.

Dinilai nol

- Menyalin *notebook* kelompok lain, Kaggle, atau Hugging Face tanpa menyebut sumber
- Mengarang angka, kurva, atau catatan eksperimen
- Menyembunyikan pemakaian *test data* untuk memilih model

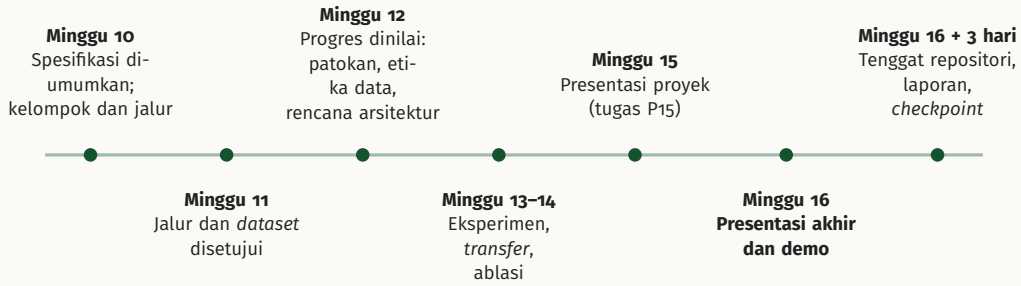
Data dan model pihak lain

Model terlatih dan *dataset* wajib disebut sumber dan lisensinya. Jangan mengirim data yang memuat identitas orang ke layanan LLM pihak ketiga.

Aturan main

- Kelompok 3–4 orang; boleh sama dengan kelompok UTS atau dibentuk ulang, paling lambat Minggu 10.
- Jalur, masalah, dan *dataset* disetujui dosen paling lambat akhir Minggu 11.
- Python 3 dan PyTorch. *Library* tingkat tinggi (misalnya Hugging Face `transformers`) boleh dipakai untuk memuat model, sedangkan *training loop* ditulis sendiri.
- Bahasa laporan: Indonesia. Istilah asing dicetak miring.
- Keterlambatan: dikurangi 10 poin per hari, maksimal 3 hari. Setelah itu tidak diterima.
- Repositori dibekukan dengan *tag* `uas-final`; perubahan sesudah tenggat tidak dinilai.
- Angka yang tidak dapat dihasilkan ulang dari *checkpoint* dan `inferensi.ipynb` dianggap tidak ada.
- Konflik kelompok dilaporkan sebelum Minggu 14. Kelompok yang berhalangan presentasi karena alasan sah menghubungi dosen sebelum hari presentasi.

Jadwal



Progres Minggu 12 (isi penyerahannya mengikuti tugas Pertemuan 12) dan presentasi proyek Minggu 15 dinilai sebagai tugas dan masuk komponen “tugas dan kuis” (20%), terpisah dari nilai UAS. *Test data* baru dibuka setelah presentasi Minggu 15, untuk *configuration* akhir saja.

Checklist pengumpulan

- ☐ Tautan repositori dapat diakses dosen
- ☐ `README.md`, `requirements.txt`, kartu data
- ☐ Hasil uji *gradient* per *layer*
- ☐ `hasil/eksperimen.csv` dan kurva tiap eksperimen
- ☐ Tautan *checkpoint* beserta *checksum*
- ☐ `inferensi.ipynb` lulus *Restart and Run All*
- ☐ Tabel akhir tiga *seed* dengan CI dan patokan
- ☐ Ablasi, analisis *error*, estimasi *cost* dan emisi
- ☐ Daftar periksa etika dan *model card*
- ☐ Laporan PDF 8–12 halaman
- ☐ `AI-USAGE.md` dan formulir penilaian sejawat
- ☐ *Slide* presentasi akhir
- ☐ *Tag* `uas-final` dibuat sebelum tenggat

Pertanyaan yang sering muncul

Pertanyaan yang sering muncul (1)

- **Model saraf kami kalah dari patokan. Apakah gagal?** Tidak. Pada data tabular dan data kecil hal itu sering terjadi. Jelaskan mengapa dengan bukti dari kurva, kurva belajar, atau ablasi; temuan seperti ini dinilai sama baiknya.
- **Boleh memakai Keras atau PyTorch Lightning?** Untuk memuat model dan *layer*, boleh. *Training loop* (*forward*, *loss*, *backward*, langkah *optimizer*, validasi per *epoch*) tetap ditulis sendiri di PyTorch.
- **Boleh memakai *dataset* yang sama dengan UTS?** Wajib untuk jalur C, boleh untuk jalur lain bila datanya cocok. Pembagian *test data* UTS dipakai ulang.
- **Kuota GPU Colab habis.** Bagi eksperimen di antara akun anggota, pakai Kaggle, perkecil resolusi citra atau panjang teks, dan simpan *checkpoint* sesering mungkin. Tuliskan kendala ini di keterbatasan.
- **Bolehkah memakai model yang lebih besar dari ukuran *base*?** Hanya dengan persetujuan dosen, dan *cost*-nya wajib dibandingkan dengan model yang lebih kecil.

Pertanyaan yang sering muncul (2)

- **Jalur B wajib memakai IndoBERT?** Tidak. *Transformer* terlatih lain yang mendukung bahasa Indonesia boleh dipakai, misalnya model multibahasa, selama ukurannya wajar dan lisensinya jelas.
- **Bagaimana bila LLM *zero-shot* mengalahkan model kami?** Itu temuan yang sah. Bahas *cost*-nya per seribu *prediction*, *latency*-nya, risiko kontaminasi data, dan apakah data boleh dikirim ke layanan pihak ketiga.
- **Hasil tiga *seed* kami berbeda jauh.** Laporkan apa adanya. Simpangan baku yang besar adalah temuan tentang kestabilan *training*, dan perlu dibahas di diagnosis.
- **Apakah generasi teks atau gambar boleh menjadi proyek?** Hanya dengan persetujuan dosen, dan harus disertai *evaluation metric* kuantitatif yang dapat diulang. *Classification* tetap dianjurkan karena evaluasinya lebih jujur dalam waktu yang tersedia.
- **Satu anggota tidak aktif.** Laporkan sebelum Minggu 14; pembagian kerja disesuaikan dan penilaian sejawat mencatatnya.

Pertanyaan yang akan muncul saat presentasi

1. Tunjukkan kurva yang paling membingungkan Anda. Apa diagnosisnya sekarang?
2. Mengapa *learning rate* itu? Apa yang terjadi pada nilai sepuluh kali lebih besar?
3. Berapa jumlah parameter model Anda, dan dari *layer* mana sebagian besarnya?
4. Komponen mana yang paling berpengaruh menurut ablasi, dan seberapa yakin Anda?
5. Jelaskan satu langkah *backward* pada *training loop* Anda.
6. Mengapa model terlatih membantu, atau mengapa tidak membantu, pada data Anda?
7. Bila diberi anggaran GPU sepuluh kali lipat, apa yang Anda ubah lebih dahulu?
8. Kapan sebaiknya masalah ini diselesaikan dengan LLM, dan kapan tidak?

Ringkasan

Daftar ini sengaja dibagikan sekarang. Jawaban yang merujuk ID eksperimen dan gambar di laporan selalu lebih meyakinkan daripada jawaban dari ingatan.

Menutup semester

Paruh pertama: model klasik yang dilatih dengan rumus atau pencarian serakah, dan cara membandingkannya dengan jujur.

Paruh kedua: model yang dilatih dengan menuruni *error surface*, dari *perceptron* sampai *transformer*, dan cara membaca proses *training*-nya.

Keduanya bertemu pada satu kebiasaan: setiap klaim tentang model harus disertai protokol yang dapat diulang orang lain.

Bekal ke depan

Definisi Mitchell dari Pertemuan 1 tetap berlaku untuk LLM terbesar sekalipun: sebutkan T, P, dan E, lalu ukur. Model berganti setiap tahun; kebiasaan mengukur dengan jujur tetap terpakai.

Selamat mengerjakan

Baca kurvanya, catat eksperimennya, laporkan apa adanya.