

Bab 13 — Capstone AI-First: Dari Ide sampai Demo yang Bisa Dipertanggungjawabkan

Cara membaca bab ini

Bab 13 adalah jembatan dari belajar konsep menuju membangun proyek. Setelah pembaca mengenal AI, data, matematika, machine learning, deep learning, generative AI, dan reinforcement learning, pertanyaan terakhirnya adalah: bagaimana semua itu disatukan menjadi satu capstone yang rapi, realistis, dan layak dipresentasikan?

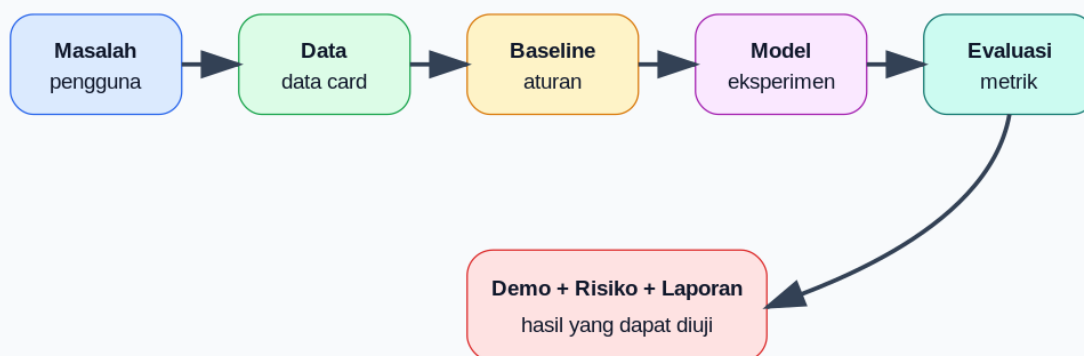
Capstone bukan sekadar “membuat model”. Capstone adalah latihan berpikir end-to-end:

masalah nyata → batasan → data → baseline → model → evaluasi → demo → risiko → laporan

Cara membacanya: proyek AI yang baik dimulai dari masalah, bukan dari algoritma favorit. Data dan model berada di tengah, bukan di awal. Evaluasi, demo, risiko, dan laporan adalah bagian inti, bukan aksesori di akhir.

Janji bab: setelah bab ini, pembaca mampu merancang dan mengerjakan proyek akhir AI-first tanpa robotics: memilih masalah, membuat data card, membangun baseline, mengevaluasi model, menambahkan komponen generative/RAG/RL bila relevan, membuat demo, menulis risk register, dan mempresentasikan hasil dengan jujur.

Peta Capstone AI-First



Capstone yang baik bukan hanya model; ia adalah alur keputusan yang dapat dipertanggungjawabkan.

Peta capstone AI-first

Subbab 1 — Cerita pembuka: capstone bukan lomba model tercanggih

Banyak pemula memulai proyek akhir dengan kalimat: “Saya mau pakai Transformer”, “Saya mau pakai DQN”, atau “Saya mau pakai LLM”. Kalimat itu terdengar modern, tetapi sering berbahaya. Dalam proyek nyata, algoritma adalah alat. Alat dipilih setelah masalah dipahami.

Bayangkan tim kecil di Karawang ingin membantu UMKM memutuskan promo mingguan. Mereka punya data transaksi sederhana: hari, jenis produk, harga, diskon, jumlah terjual, dan catatan komplain. Tim bisa tergoda langsung memakai deep learning. Tetapi pertanyaan pertama seharusnya:

- keputusan apa yang ingin dibantu?
- siapa pengguna akhirnya?
- apa konsekuensi jika rekomendasi salah?
- data apa yang tersedia dan apa yang belum tersedia?
- baseline sederhana apa yang harus dikalahkan?

Capstone yang bagus tidak selalu memakai model tercanggih. Capstone yang bagus menunjukkan alur berpikir yang benar. Jika baseline rule-based sudah cukup, jelaskan itu. Jika logistic regression mengalahkan neural network kecil, terima hasilnya. Jika data tidak cukup untuk RL penuh, gunakan bandit atau simulasi kecil sebagai pembelajaran, bukan klaim produksi.

Tes cepat subbab 1

1. Mengapa capstone tidak boleh dimulai dari algoritma favorit?
2. Apa contoh konsekuensi salah rekomendasi pada proyek UMKM?
3. Mengapa baseline sederhana wajib dibuat?

Subbab 2 — Definisi capstone AI-first

Capstone AI-first dalam buku ini adalah proyek yang menggabungkan minimal empat lapisan:

Lapisan	Pertanyaan	Contoh deliverable
Problem framing	masalah apa yang diselesaikan?	problem statement, stakeholder map
Data	bukti apa yang dipakai?	data card, EDA ringkas
Model	metode apa yang diuji?	baseline, model utama, ablation
Evaluasi & risiko	apakah hasil dapat dipercaya?	metrik, error analysis, risk register

Capstone disebut AI-first karena fokusnya adalah siklus AI dari data sampai keputusan. Robotics fisik, deployment produksi besar, cloud MLOps penuh, dan integrasi perangkat keras ditunda. Namun etika, evaluasi, dan reproducibility tetap wajib.

Format minimum capstone:

```
repo/
  README.md
  data_card.md
  model_card.md
  risk_register.md
  report.md
  src/
  notebooks/
  outputs/
```

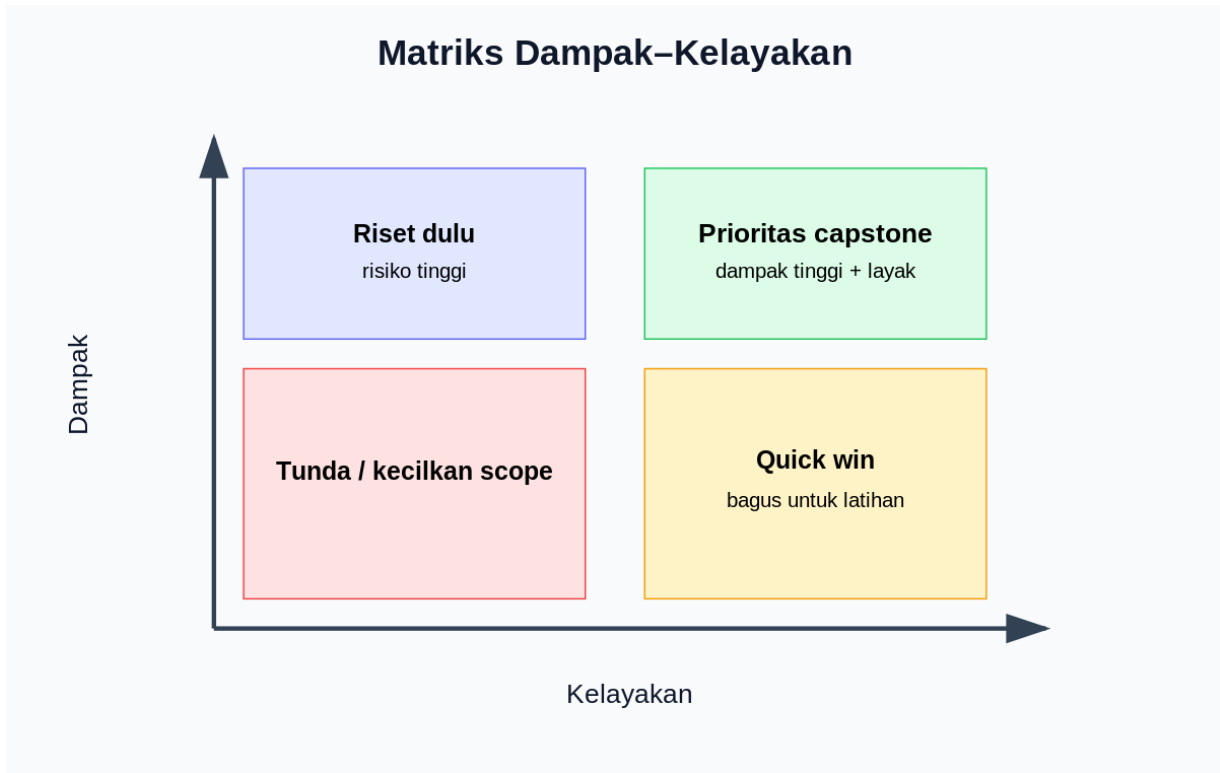
Cara membacanya: capstone bukan hanya notebook tunggal. Ia adalah paket kecil yang bisa dibaca orang lain: ada petunjuk run, catatan data, catatan model, risiko, dan output.

Tes cepat subbab 2

1. Apa bedanya capstone AI-first dan demo model biasa?
2. Mengapa data card dan model card penting?
3. Apa risiko jika proyek hanya berupa notebook tanpa README?

Subbab 3 — Memilih masalah: pakai matriks dampak-kelayakan

Masalah capstone harus cukup penting, tetapi masih bisa dikerjakan. Gunakan matriks dampak-kelayakan.



Matriks dampak-kelayakan

Skor sederhana:

$$\text{Skor prioritas} = \text{dampak} \times \text{kelayakan} \times \text{ketersediaan data} - \text{risiko}$$

Cara membacanya: masalah bernilai tinggi jika dampaknya besar, layak dikerjakan, dan datanya tersedia. Risiko mengurangi skor. Risiko bukan alasan otomatis menolak proyek, tetapi harus membuat tim lebih hati-hati.

Contoh kandidat proyek:

Ide	Dampak	Kelayakan	Data	Risiko	Catatan
Prediksi stok UMKM	tinggi	sedang	sedang	sedang	bagus untuk tabular ML
Chatbot diagnosis medis	tinggi	rendah	rendah	tinggi	tidak cocok untuk pemula
Rekomendasi promo	sedang	tinggi	sedang	sedang	cocok untuk bandit/simulasi
Klasifikasi keluhan pelanggan	sedang	tinggi	tinggi	rendah	cocok untuk NLP ringan
RL trading uang nyata	tinggi	rendah	sedang	tinggi	hindari untuk capstone pemula

Masalah yang baik biasanya punya kalimat seperti ini:

Untuk [pengguna], ketika [situasi], sistem membantu [keputusan] dengan [batasan], dan keberhasilan diukur memakai [metrik]

Cara membacanya: kalimat ini memaksa kita menyebut pengguna, konteks, keputusan, batasan, dan metrik. Jika salah satu kosong, problem framing belum matang.

Contoh:

Untuk pemilik warung kopi, ketika menentukan promo mingguan, sistem membantu memilih promo yang menyeimbangkan laba dan stok,

dengan batas diskon maksimum 20%, dan keberhasilan diukur memakai laba simulasi, stockout rate, serta jumlah komplain.

Tes cepat subbab 3

1. Buat problem statement dengan template di atas.
2. Mengapa risiko mengurangi skor prioritas?
3. Apa contoh proyek berdampak tinggi tetapi tidak layak untuk capstone pemula?

Subbab 4 — Scope: jangan membuat tiga proyek sekaligus

Kesalahan umum capstone adalah scope terlalu besar. Pembaca ingin membuat dashboard, chatbot, RAG, mobile app, model prediksi, RL agent, dan deployment cloud sekaligus. Hasilnya sering tidak selesai.

Gunakan tiga lingkaran scope:

Must-have = wajib ada agar proyek menjawab masalah
Should-have = bagus jika waktu cukup
Could-have = bonus, bukan janji utama

Cara membacanya: must-have adalah kontrak. Should-have adalah peningkatan. Could-have adalah ide tambahan. Jika waktu sempit, could-have dipotong tanpa rasa bersalah.

Contoh capstone rekomendasi promo:

Level	Isi
Must-have	data sintetis/riwayat kecil, baseline rule, model bandit, evaluasi laba
Should-have	visual EDA, risk register, simulasi beberapa segmen pelanggan
Could-have	dashboard web, integrasi LLM untuk laporan otomatis, API

Aturan capstone pemula:

Satu masalah utama.
Satu dataset utama.
Satu baseline wajib.
Satu model utama.
Satu evaluasi utama.
Satu demo sederhana.

Cara membacanya: angka “satu” bukan membatasi kreativitas, tetapi melindungi proyek dari melebar. Setelah versi inti selesai, barulah eksperimen tambahan boleh masuk.

Tes cepat subbab 4

1. Apa must-have untuk proyek klasifikasi keluhan pelanggan?
2. Mengapa dashboard sering sebaiknya masuk should-have/could-have?
3. Apa tanda scope proyek sudah terlalu besar?

Subbab 5 — Data card: sebelum model, pahami data

Data card adalah dokumen pendek yang menjawab: data berasal dari mana, apa isinya, boleh dipakai untuk apa, dan apa batasannya.

Minimal data card:

Nama dataset:
Sumber:

Jumlah baris:
 Fitur:
 Target/label:
 Periode:
 Izin/lisensi:
 Potensi bias:
 Kolom sensitif:
 Missing values:
 Catatan kualitas:

Cara membacanya: data card adalah KTP dataset. Ia membuat pembaca tidak memperlakukan data sebagai angka anonim. Untuk buku komersial dan proyek serius, data harus punya asal-usul dan batas penggunaan.

Contoh data promo UMKM:

Kolom	Arti	Risiko
day_type	weekday/weekend	pola lokal bisa berubah
weather	cerah/hujan	data cuaca bisa kasar
discount	diskon persen	eksperimen harga sensitif
units_sold	jumlah terjual	dipengaruhi stok
complaints	jumlah komplain	underreporting mungkin terjadi
profit	laba	rumus margin harus jelas

Hal yang harus dicek:

- Apakah ada leakage?
- Apakah label dibuat setelah keputusan terjadi?
- Apakah ada data pribadi?
- Apakah distribusi train dan test mirip?
- Apakah missing value punya makna bisnis?

Tes cepat subbab 5

1. Mengapa data card penting untuk reproducibility?
2. Apa contoh leakage dalam data promo?
3. Apa kolom sensitif pada proyek pendidikan?

Subbab 6 — Baseline: lawan pertama model bukan SOTA, tapi aturan sederhana

Baseline adalah pembanding minimal. Tanpa baseline, angka model tidak punya arti. Akurasi 85% terdengar bagus, tetapi jika baseline mayoritas sudah 84%, model tidak banyak membantu.

Jenis baseline:

Tipe	Contoh	Kapan dipakai
Majority baseline	selalu pilih kelas terbanyak	klasifikasi tidak seimbang
Mean/median baseline	prediksi rata-rata	regresi
Rule baseline	jika stok rendah, jangan diskon	keputusan bisnis
Random baseline	pilih acak	bandit/RL
Human baseline	keputusan staf	domain operasional

Rumus improvement sederhana:

$$\text{Improvement} = (\text{Skor_model} - \text{Skor_baseline}) / |\text{Skor_baseline}|$$

Cara membacanya: improvement mengukur seberapa jauh model lebih baik dari baseline relatif terhadap baseline. Jika baseline profit 100 dan model profit 120, improvement 20%. Jika baseline 100 dan model 102, model mungkin belum layak dipakai.

Namun metrik harus sesuai arah. Untuk error seperti MAE/RMSE, lebih kecil lebih baik. Maka improvement error biasanya dibalik:

$$\text{Error_reduction} = (\text{Error_baseline} - \text{Error_model}) / \text{Error_baseline}$$

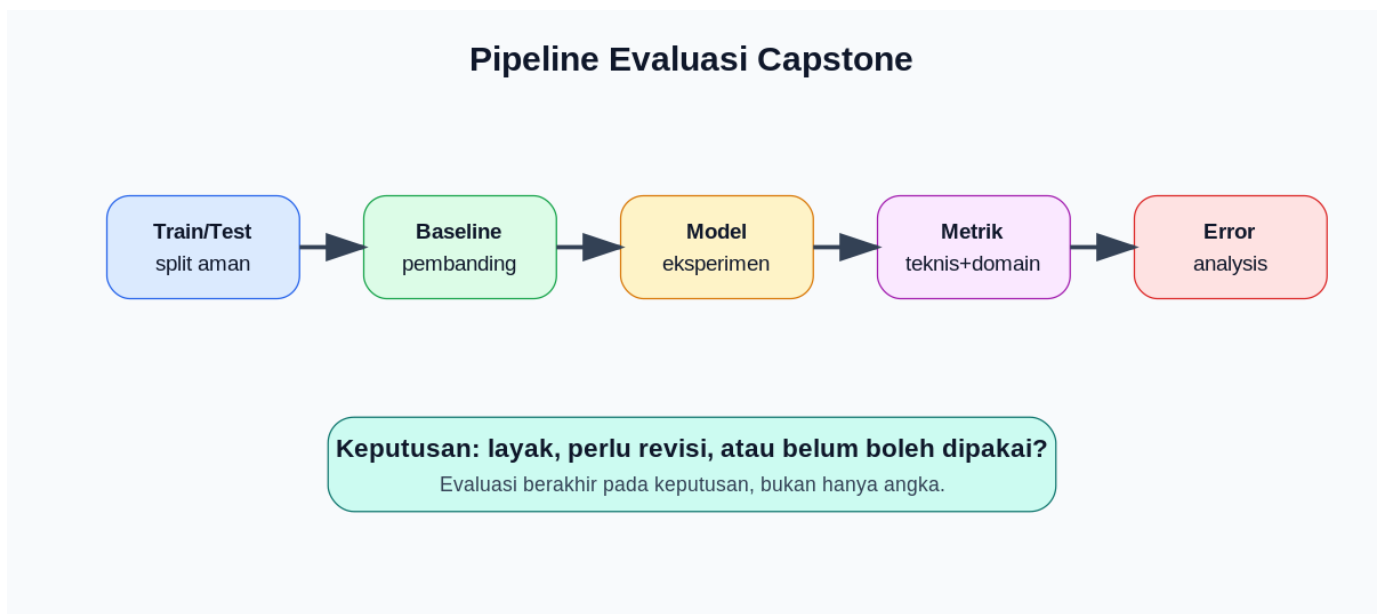
Cara membacanya: jika baseline error 10 dan model error 7, error reduction 30%. Model mengurangi kesalahan sebesar 30% dibanding baseline.

Tes cepat subbab 6

1. Mengapa baseline mayoritas bisa menipu pada data tidak seimbang?
2. Buat rule baseline untuk rekomendasi promo.
3. Hitung error reduction jika baseline MAE 20 dan model MAE 15.

Subbab 7 — Peta model: pilih model sesuai bukti, bukan gengsi

Capstone AI-first boleh memakai beberapa keluarga model, tetapi harus ada alasan.



Pipeline evaluasi capstone

Peta pemilihan:

Situasi	Kandidat
data tabular kecil	rule baseline, logistic regression, decision tree
data tabular sedang	random forest/boosting jika library tersedia
teks pendek	TF-IDF/logistic regression, embedding sederhana
dokumen panjang	semantic search/RAG mini
keputusan promo berulang	bandit/contextual bandit
reward tertunda dan simulator valid	tabular RL atau RL sederhana
output natural language	LLM/RAG dengan evaluasi groundedness

Jangan memakai deep learning hanya karena modern. Deep learning butuh data, komputasi, dan evaluasi lebih ketat. Untuk capstone edukatif, model sederhana sering lebih kuat karena mudah dijelaskan.

Ablation membantu membuktikan komponen mana yang berguna:

Model penuh - satu komponen = model ablation

Cara membacanya: hilangkan satu fitur/komponen, lalu lihat performa. Jika performa tidak turun, komponen itu mungkin tidak penting. Jika performa turun banyak, komponen itu berkontribusi.

Tes cepat subbab 7

1. Kapan bandit lebih cocok daripada supervised learning?
2. Mengapa deep learning tidak selalu lebih baik?
3. Apa manfaat ablation?

Subbab 8 — Evaluasi: satu angka tidak cukup

Evaluasi capstone harus menjawab tiga pertanyaan:

1. Apakah model bekerja secara kuantitatif?
2. Di mana model gagal?
3. Apakah kegagalan itu berbahaya?

Untuk klasifikasi:

Accuracy = prediksi benar / semua prediksi
Precision = true positive / semua prediksi positif
Recall = true positive / semua positif sebenarnya
 $F1 = 2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$

Cara membacanya: accuracy menjawab “berapa banyak benar secara total”. Precision menjawab “ketika model bilang positif, seberapa sering benar?”. Recall menjawab “dari semua kasus positif, berapa yang berhasil ditemukan?”. F1 menyeimbangkan precision dan recall.

Untuk keputusan bisnis, metrik teknis harus diterjemahkan ke metrik domain:

Metrik teknis	Metrik domain
akurasi	keputusan benar
MAE demand	selisih stok
reward simulasi	laba/stockout/komplain
groundedness RAG	jawaban punya sumber
regret bandit	biaya eksperimen

Error analysis minimal:

Ambil 20 kesalahan paling penting.
Kelompokkan pola salah.
Cari penyebab data/model/proses.
Tulis perbaikan yang realistis.

Cara membacanya: error analysis adalah proses membaca kegagalan, bukan hanya menampilkan confusion matrix. Model yang baik harus diketahui batasnya.

Tes cepat subbab 8

1. Kapan recall lebih penting daripada precision?
2. Mengapa metrik domain penting?

3. Apa manfaat membaca 20 error secara manual?

Subbab 9 — Komponen generative AI/RAG dalam capstone

Generative AI dapat masuk capstone, tetapi harus punya fungsi jelas. Jangan menambahkan LLM hanya agar proyek terlihat modern.

Contoh penggunaan yang masuk akal:

- membuat ringkasan laporan model;
- menjawab pertanyaan berdasarkan dokumen internal dengan RAG;
- menghasilkan penjelasan rekomendasi promo;
- membantu membuat checklist risiko;
- membuat tutor interaktif untuk hasil model.

RAG mini punya alur:

dokumen → chunk → embedding/retrieval → konteks → jawaban + sitasi

Cara membacanya: jawaban LLM tidak dibiarkan mengarang dari memori. Sistem mengambil potongan dokumen relevan, memasukkannya sebagai konteks, lalu meminta jawaban dengan sitasi.

Evaluasi RAG minimal:

Aspek	Pertanyaan
Retrieval	apakah konteks yang diambil relevan?
Groundedness	apakah klaim ada di sumber?
Helpfulness	apakah jawaban membantu pengguna?
Safety	apakah jawaban menolak saat data tidak cukup?

Tes cepat subbab 9

1. Kapan LLM tidak perlu dipakai dalam capstone?
2. Apa beda RAG dan chatbot biasa?
3. Mengapa sitasi penting?

Subbab 10 — Komponen RL dalam capstone: gunakan dengan jujur

RL cocok jika ada keputusan berurutan, reward, dan kemungkinan simulasi/evaluasi. Jika tidak ada simulator atau data historis cukup, jangan mengklaim RL produksi.

Level RL untuk capstone:

Level	Bentuk	Cocok untuk
Bandit	pilih aksi tanpa state panjang	promo, rekomendasi ringan
Contextual bandit	aksi bergantung konteks	segmentasi pelanggan
Tabular RL	state kecil, simulator sederhana	gridworld, stok mini
Offline RL konseptual	belajar dari log historis	diskusi risiko, bukan klaim besar
Deep RL	DQN/SAC/PPO	hanya jika lingkungan valid dan waktu cukup

Reward capstone harus ditulis eksplisit:

$$\text{reward} = \text{manfaat utama} - \text{penalti risiko} - \text{penalti biaya}$$

Cara membacanya: reward bukan hanya hasil yang ingin dinaikkan. Ia harus mengurangi biaya dan risiko. Dalam rekomendasi promo, manfaat utama bisa laba, penalti risiko bisa stockout/komplain, dan penalti biaya bisa diskon terlalu besar.

Jika menggunakan RL, selalu tulis:

- state;
- action;
- reward;
- episode;
- baseline;
- batas eksplorasi;
- risiko reward hacking.

Tes cepat subbab 10

1. Kapan bandit lebih aman daripada RL penuh?
2. Buat reward untuk rekomendasi stok.
3. Mengapa simulator yang salah menghasilkan policy yang salah?

Subbab 11 — Risk register: bagian wajib, bukan formalitas

Risk register adalah tabel risiko yang membuat capstone lebih dewasa.

Risk Register Capstone

Risiko	Dampak	Peluang	Mitigasi
data bias	5	3	audit distribusi
stockout	4	2	guardrail stok
hallucination	4	3	RAG + sitasi
overclaim	3	4	model card
privasi	5	2	anonimisasi

Risk score = impact × probability; mitigasi harus spesifik dan bisa diuji.

Risk register capstone

Minimal kolom:

Risiko	Dampak	Peluang	Mitigasi	Owner
data bias	tinggi	sedang	audit distribusi	data lead

Risiko	Dampak	Peluang	Mitigasi	Owner
rekomendasi merugikan	tinggi	rendah	guardrail manual	product lead
privasi data	tinggi	sedang	anonimisasi	data lead
hallucination	sedang	sedang	RAG + sitasi	AI lead
overclaim hasil	sedang	tinggi	model card	semua

Skor risiko:

$$\text{Risk_score} = \text{impact} \times \text{probability}$$

Cara membacanya: risiko tinggi jika dampaknya besar dan peluangnya besar. Dampak tinggi tetapi peluang kecil tetap perlu mitigasi jika konsekuensinya serius.

Risk register bukan dokumen untuk menakuti tim. Ia membantu tim berkata jujur: “ini yang bisa salah, ini cara kami membatasinya.”

Tes cepat subbab 11

1. Apa risiko utama proyek chatbot sekolah?
2. Mengapa overclaim termasuk risiko?
3. Hitung risk score jika impact 5 dan probability 4.

Subbab 12 — Demo: yang penting bisa diuji, bukan harus mewah

Demo capstone harus membuat evaluator bisa mencoba ide utama. Demo boleh sederhana:

- script CLI;
- notebook yang rapi;
- halaman HTML statis;
- dashboard lokal;
- API mini;
- video singkat.

Urutan demo yang baik:

input contoh → proses model → output → interpretasi → batasan

Cara membacanya: demo bukan hanya menampilkan output. Demo harus menunjukkan input, proses, hasil, cara membaca hasil, dan kapan hasil tidak boleh dipercaya.

Checklist demo:

- Bisa dijalankan dari README
- Input contoh tersedia
- Output tersimpan di folder outputs
- Ada seed atau instruksi reproducible
- Ada contoh sukses dan contoh gagal
- Ada catatan batasan

Tes cepat subbab 12

1. Mengapa demo CLI bisa cukup untuk capstone?
2. Apa contoh input/output untuk proyek rekomendasi promo?
3. Mengapa contoh gagal perlu ditampilkan?

Subbab 13 — Laporan capstone: struktur yang menjual tanpa overclaim

Laporan capstone harus rapi dan jujur. Struktur yang disarankan:

1. Ringkasan eksekutif
2. Masalah dan pengguna
3. Data card
4. Metode dan baseline
5. Hasil evaluasi
6. Error analysis
7. Demo
8. Risk register
9. Batasan
10. Rencana lanjut

Cara membacanya: laporan dimulai dari keputusan dan dampak, bukan dari detail kode. Detail teknis tetap ada, tetapi pembaca bisnis harus bisa memahami ringkasan tanpa membaca semua notebook.

Kalimat yang baik:

Model logistic regression mengurangi error stok 18% dibanding baseline median pada data simulasi.

Kalimat yang buruk:

Model AI kami menyelesaikan masalah stok UMKM.

Yang pertama spesifik, terukur, dan jujur. Yang kedua terlalu besar dan tidak menyebut batasan.

Tes cepat subbab 13

1. Mengapa laporan harus menyebut baseline?
2. Apa contoh kalimat overclaim?
3. Mengapa ringkasan eksekutif penting?

Subbab 14 — Praktikum Bab 13: capstone promo UMKM AI-first

Praktikum utama bab ini memakai proyek kecil: AI Promo Copilot untuk UMKM. Tujuannya bukan membuat sistem produksi, tetapi melatih pola pikir end-to-end.

Komponen praktikum:

1. membuat data sintesis transaksi promo;
2. membuat baseline rule;
3. melatih model sederhana untuk prediksi profit;
4. membandingkan strategi promo;
5. membuat risk register;
6. menghasilkan output JSON untuk laporan.

Jalankan:

```
cd zero-to-hero-menaklukkan-ai/chapters/13-capstone-ai-first/code
python3 capstone_ai_first_playground.py --self-test
python3 capstone_ai_first_playground.py
```

Output:

```
code/outputs/bab13_capstone_results.json
```

Cara membacanya: praktikum ini sengaja kecil agar pembaca memahami alur. Jika alur kecil sudah benar, barulah proyek nyata dapat diperbesar.

Tes cepat subbab 14

1. Apa baseline praktikum ini?
2. Mengapa data sintetis boleh dipakai untuk belajar tetapi tidak boleh dioverclaim?
3. Apa isi output JSON yang penting untuk laporan?

Subbab 15 — Rubrik penilaian capstone

Rubrik membuat penilaian lebih objektif.

Dimensi	Bobot	Kriteria unggul
Problem framing	15%	pengguna, keputusan, metrik, batasan jelas
Data card & EDA	15%	sumber, kualitas, bias, leakage dibahas
Baseline	10%	baseline relevan dan dijelaskan
Model & eksperimen	20%	model sesuai masalah, ablation/variasi ada
Evaluasi & error analysis	15%	metrik tepat, error dibaca manual
Demo & reproducibility	10%	bisa dijalankan ulang dari README
Risk register	10%	risiko nyata + mitigasi masuk akal
Komunikasi	5%	laporan jelas, tidak overclaim

Skor akhir:

$$\text{Skor_akhir} = \sum \text{bobot}_i \times \text{skor}_i$$

Cara membacanya: setiap dimensi punya bobot. Skor tiap dimensi dikalikan bobotnya, lalu dijumlahkan. Rubrik mencegah proyek yang demo-nya cantik tetapi tidak punya evaluasi mendapat nilai terlalu tinggi.

Tes cepat subbab 15

1. Mengapa problem framing diberi bobot khusus?
2. Apa yang terjadi jika demo bagus tetapi risk register kosong?
3. Bagaimana rubrik membantu pembaca belajar mandiri?

Subbab 16 — Checklist akhir sebelum publikasi

Sebelum capstone dikumpulkan atau dipublikasikan, jalankan checklist ini:

- README bisa diikuti orang lain
- Data card selesai
- Model card selesai
- Risk register selesai
- Baseline ada
- Metrik sesuai masalah
- Error analysis ada
- Output demo tersimpan
- Batasan ditulis jelas
- Tidak ada data pribadi/secret
- Tidak ada klaim berlebihan

Cara membacanya: checklist ini adalah rem terakhir. Ia memastikan proyek bukan hanya berjalan di laptop pembuatnya, tetapi bisa dibaca, diuji, dan dikritik orang lain.

Ringkasan bab

1. Capstone dimulai dari masalah, bukan algoritma.
2. Scope kecil yang selesai lebih baik daripada scope besar yang kabur.
3. Data card dan model card membuat proyek dapat dipercaya.
4. Baseline wajib ada agar model punya pembandingan.
5. Evaluasi harus menggabungkan metrik teknis, metrik domain, dan error analysis.
6. Generative AI/RAG hanya dipakai jika punya fungsi jelas.
7. RL hanya dipakai jika state/action/reward dan evaluasi masuk akal.
8. Risk register adalah bagian inti proyek AI.
9. Demo harus reproducible dan menunjukkan batasan.
10. Laporan yang baik menjual nilai proyek tanpa overclaim.

Tes akhir bab

1. Buat problem statement capstone dengan template pengguna-situasi-keputusan-metrik.
2. Buat data card mini untuk dataset pilihan Anda.
3. Buat baseline untuk proyek tersebut.
4. Pilih satu metrik teknis dan satu metrik domain.
5. Tulis tiga risiko dan mitigasinya.
6. Jelaskan apakah proyek Anda perlu generative AI/RAG atau tidak.
7. Jelaskan apakah proyek Anda perlu RL atau cukup supervised/bandit.
8. Buat struktur README capstone.
9. Tulis satu kalimat hasil yang jujur dan terukur.
10. Tulis satu rencana lanjut setelah capstone selesai.