

# Bab 05 — Probabilitas, Statistik, Kalkulus, dan Gradient

Status: Draft lengkap v0.4 — diperluas setelah kritik kedalaman Bagian buku: Level C — Matematika yang Ramah Target pembaca: Pemula serius yang ingin memahami ketidakpastian, statistik, loss, turunan, gradient, dan optimisasi secara teknis tetapi tetap manusiawi.

## Cara membaca bab ini

Bab ini sengaja dibuat lebih panjang daripada versi sebelumnya. Satu subtopik tidak lagi dipadatkan menjadi satu bagian ringkas. Beberapa ide dibahas melalui beberapa lapisan: pengertian, narasi dunia nyata, intuisi visual, contoh kode kecil, rumus pelan-pelan, kesalahan umum, dan motivasi.

Janji Bab 5: pembaca tidak hanya “pernah melihat” probabilitas, statistik, turunan, dan gradient, tetapi mulai bisa menjelaskan mengapa konsep-konsep itu menjadi mesin belajar model AI.

Prinsip belajar:

Kita tidak melompati rumus, tetapi kita juga tidak takut padanya. Rumus datang setelah cerita, gambar, dan eksperimen kecil.

## Pendalaman awal — sejarah probabilitas, statistik, kalkulus, dan optimisasi

Probabilitas modern berkembang dari masalah perjudian dan risiko, tetapi kemudian menjadi bahasa ilmiah untuk ketidakpastian. Statistik tumbuh dari kebutuhan negara, astronomi, biologi, ekonomi, dan industri untuk membaca data yang tidak sempurna. Kalkulus lahir dari kebutuhan memahami perubahan, gerak, dan kemiringan. Optimisasi berkembang karena manusia ingin memilih tindakan terbaik di antara banyak kemungkinan.

Machine learning mempertemukan semuanya. Probabilitas membantu model menyatakan keyakinan. Statistik membantu kita belajar dari sampel. Kalkulus memberi turunan dan gradient. Optimisasi menggunakan gradient untuk memperbaiki parameter. Karena itu, Bab 5 bukan empat topik terpisah, melainkan satu rantai: ketidakpastian → data → kesalahan → arah perbaikan → pembelajaran.

Secara formal, banyak training ML dapat dibaca sebagai masalah minimisasi:

$$\theta^* = \operatorname{argmin}_{\theta} L(\theta)$$

Artinya: cari parameter  $\theta$  terbaik yang membuat loss  $L$  sekecil mungkin. Gradient memberi arah perubahan loss:

$$\theta_{\text{baru}} = \theta_{\text{lama}} - \eta \nabla L(\theta)$$

$\eta$  adalah learning rate. Simbol  $\nabla$  dibaca nabla, yaitu gradient untuk banyak parameter.

## Subbab 1 — Mengapa AI harus bersahabat dengan ketidakpastian

## Mengapa AI harus bersahabat dengan ketidakpastian



Mengapa AI harus bersahabat dengan ketidakpastian

AI tidak hidup di ruang kelas yang semua soalnya punya satu jawaban bersih. AI dipakai untuk memprediksi permintaan es teh, memperkirakan risiko kredit UMKM, mengurutkan rekomendasi produk, mendeteksi spam, dan memilih rute perjalanan. Semua contoh itu punya satu kesamaan: hasilnya belum pasti ketika keputusan harus dibuat. Karena itu, probabilitas bukan hiasan matematika; probabilitas adalah bahasa yang membuat model berani berkata “kemungkinan besar”, “masih ragu”, atau “data belum cukup”.

Bayangkan pemilik warung hendak menyiapkan stok. Jika ia selalu menyiapkan stok berdasarkan hari paling ramai, banyak bahan terbuang. Jika berdasarkan hari paling sepi, pelanggan kecewa. Keputusan yang baik tidak hanya bertanya “berapa prediksi penjualan?”, tetapi juga “seberapa besar ketidakpastiannya?”. Di sinilah statistik masuk. Statistik membantu kita meringkas data lama, melihat pola, mengukur sebaran, dan menyadari kapan data bisa menipu.

Dalam machine learning, ketidakpastian muncul di banyak tempat: data latih bisa terbatas, label bisa salah, pengguna bisa berubah perilaku, sensor bisa bising, dan model bisa terlalu percaya diri. Bab ini mengajak pembaca melihat probabilitas, statistik, kalkulus, dan gradient sebagai satu cerita besar: model belajar dari dunia yang tidak sempurna dengan cara mengukur kesalahan lalu memperbaiki diri sedikit demi sedikit.

Rumus/ide kecil dan/atau latihan ketik manual

Keputusan AI sehat = prediksi + ukuran ketidakpastian + batas penggunaan

Tes cepat subbab 1

1. Jelaskan dengan contoh sendiri mengapa AI yang baik tidak boleh selalu merasa 100% yakin.
2. Apa risiko bisnis jika model hanya memberi jawaban pasti tanpa skor keyakinan?
3. Buat contoh keputusan stok warung dengan prediksi dan tingkat ketidakpastian.

## Subbab 2 — Tiga jenis pertanyaan: peluang, ringkasan, dan perbaikan

Bab 05 · Halaman 2

### Tiga jenis pertanyaan: peluang, ringkasan, dan perbaikan



Tiga jenis pertanyaan: peluang, ringkasan, dan perbaikan

Agar bab ini tidak terasa seperti kumpulan rumus terpisah, kita pakai tiga pertanyaan pemandu. Pertama: “seberapa mungkin sesuatu terjadi?” Ini wilayah probabilitas. Kedua: “apa yang bisa kita simpulkan dari data?” Ini wilayah statistik. Ketiga: “bagaimana model memperbaiki parameter agar kesalahan mengecil?” Ini wilayah kalkulus, gradient, dan optimisasi.

Contoh warung membuat hubungan ini jelas. Probabilitas bertanya: “berapa peluang penjualan es teh di atas 40 gelas besok?” Statistik bertanya: “berdasarkan 30 hari terakhir, rata-rata dan sebarannya seperti apa?” Optimisasi bertanya: “kalau model prediksi stok masih sering meleset, bagaimana bobot model diubah agar loss turun?”

Tiga pertanyaan ini akan muncul berulang-ulang dalam buku. Ketika nanti membaca supervised learning, neural network, deep learning, sampai reinforcement learning, pembaca akan bertemu kata seperti probability, likelihood, loss, gradient, learning rate, policy, value, dan reward. Jangan khawatir: semuanya lebih mudah jika fondasinya dibangun pelan-pelan sekarang.

Rumus/ide kecil dan/atau latihan ketik manual

Probabilitas: seberapa mungkin?  
Statistik: apa pola dari data?  
Optimisasi: bagaimana parameter diperbaiki?

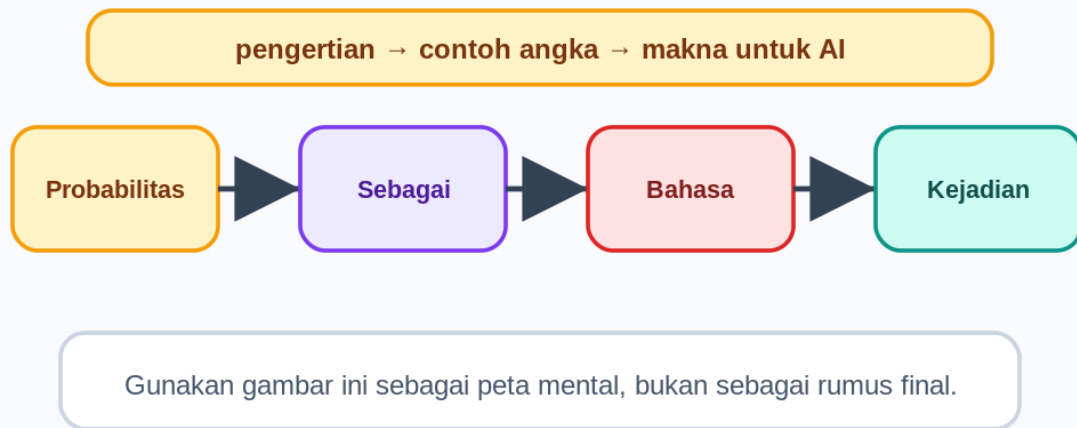
Tes cepat subbab 2

1. Sebutkan satu contoh pertanyaan probabilitas, satu contoh pertanyaan statistik, dan satu contoh pertanyaan optimisasi.
2. Bagaimana tiga pertanyaan pemandu itu saling tersambung dalam proses training?
3. Tuliskan satu mini-skenario yang punya peluang, ringkasan data, dan langkah perbaikan.

## Subbab 3 — Probabilitas sebagai bahasa kejadian

Bab 05 · Halaman 3

### Probabilitas sebagai bahasa kejadian



Probabilitas sebagai bahasa kejadian

Probabilitas mengukur peluang sebuah kejadian. Jika  $A$  adalah kejadian “pesan masuk adalah spam”, maka  $P(A)$  berarti peluang pesan itu spam. Nilainya berada antara 0 dan 1. Nilai 0 berarti mustahil dalam model kita, nilai 1 berarti pasti dalam model kita, dan nilai 0,5 berarti peluangnya setara dengan tebak koin adil.

Penting: probabilitas bukan sekadar angka formal. Ia cara berpikir. Saat model memberi skor 0,8 untuk “pelanggan mungkin membeli”, artinya model melihat bukti yang cukup kuat, tetapi tetap membuka kemungkinan salah. Dalam produk AI nyata, angka probabilitas sering dipakai untuk menentukan ambang keputusan: jika peluang fraud di atas 0,9, transaksi ditahan; jika hanya 0,55, mungkin perlu verifikasi ringan, bukan langsung ditolak.

Rumus paling dasar untuk probabilitas berbasis frekuensi adalah  $P(A) = \text{jumlah kejadian } A / \text{jumlah percobaan}$ . Rumus ini sederhana, tetapi mengajarkan ide besar: keyakinan sebaiknya didukung pengamatan. Jika dari 1000 pesan, 120 adalah spam, estimasi peluang spam di data itu adalah  $120/1000 = 0,12$ .

Rumus/ide kecil dan/atau latihan ketik manual

$$P(A) = \text{jumlah kejadian } A / \text{jumlah percobaan}$$
$$0 \leq P(A) \leq 1$$

Tes cepat subbab 3

1. Jika 45 dari 300 transaksi terlambat diproses, berapa estimasi peluang transaksi terlambat?
2. Mengapa nilai probabilitas harus berada di antara 0 dan 1?
3. Hitung peluang 45 transaksi terlambat dari 300 transaksi.

## Subbab 4 — Frekuensi relatif: mengapa percobaan banyak membuat estimasi lebih stabil

Bab 05 · Halaman 4

### Frekuensi relatif: mengapa percobaan banyak membuat estimasi lebih stabil



Frekuensi relatif: mengapa percobaan banyak membuat estimasi lebih stabil

Lempar dadu 6 kali dan angka 6 mungkin tidak muncul sama sekali. Lempar 6000 kali dan proporsi angka 6 biasanya mendekati  $1/6$ . Inilah intuisi frekuensi relatif: semakin banyak percobaan yang sebanding, estimasi peluang cenderung lebih stabil.

Namun kata “cenderung” penting. Probabilitas bukan mesin yang menjamin hasil pendek selalu rapi. Banyak pemula kecewa karena 10 lemparan koin tidak menghasilkan tepat 5 gambar dan 5 angka. Padahal probabilitas memang bicara pola jangka panjang, bukan janji mikro.

Dalam AI, ide ini membantu kita memahami mengapa dataset terlalu kecil sering berbahaya. Jika kita hanya melihat 10 pelanggan, kesimpulan bisa berubah drastis karena satu pelanggan ekstrem. Jika kita melihat 10.000 pelanggan yang representatif, pola umum lebih bisa dipercaya. Tetapi ukuran besar saja belum cukup; data juga harus dikumpulkan dengan cara yang tidak bias.

Rumus/ide kecil dan/atau latihan ketik manual

```
import random
random.seed(42)
n = 1000
muncul_enam = sum(1 for _ in range(n) if random.randint(1, 6) == 6)
print(muncul_enam / n)
```

Tes cepat subbab 4

1. Mengapa 10 percobaan sering belum cukup untuk menyimpulkan peluang dengan stabil?
2. Apa yang berubah ketika jumlah percobaan dadu dinaikkan dari 100 menjadi 10.000?
3. Simulasikan atau bayangkan 10 vs 1000 lemparan koin dan bandingkan kestabilannya.

## Subbab 5 — Aturan komplemen, AND, dan OR tanpa pusing

Bab 05 · Halaman 5

### Aturan komplemen, AND, dan OR tanpa pusing

pengertian → contoh angka → makna untuk AI



Gunakan gambar ini sebagai peta mental, bukan sebagai rumus final.

Aturan komplemen, AND, dan OR tanpa pusing

Tiga aturan dasar sering muncul dalam pemodelan. Komplemen: peluang “tidak A” adalah  $1 - P(A)$ . Jika peluang hujan 0,3, peluang tidak hujan 0,7. Aturan AND bertanya peluang dua kejadian terjadi bersama. Aturan OR bertanya peluang minimal salah satu terjadi.

Untuk kejadian independen,  $P(A \text{ dan } B) = P(A) \times P(B)$ . Misalnya peluang koin pertama gambar 0,5 dan koin kedua gambar 0,5, maka peluang keduanya gambar adalah 0,25. Tetapi banyak kejadian dunia nyata tidak independen. Peluang membeli payung dan hujan jelas berhubungan.

Untuk OR, jika A dan B tidak saling tumpang tindih,  $P(A \text{ atau } B) = P(A) + P(B)$ . Jika tumpang tindih, bagian irisan harus dikurangi agar tidak dihitung dua kali. Ide visualnya seperti dua lingkaran Venn: area gabungan adalah jumlah dua lingkaran dikurangi area tengah yang tumpang tindih.

Rumus/ide kecil dan/atau latihan ketik manual

$$P(\text{tidak } A) = 1 - P(A)$$

$$P(A \text{ dan } B) = P(A) \times P(B) \quad \# \text{ jika } A \text{ dan } B \text{ independen}$$

$$P(A \text{ atau } B) = P(A) + P(B) - P(A \text{ dan } B)$$

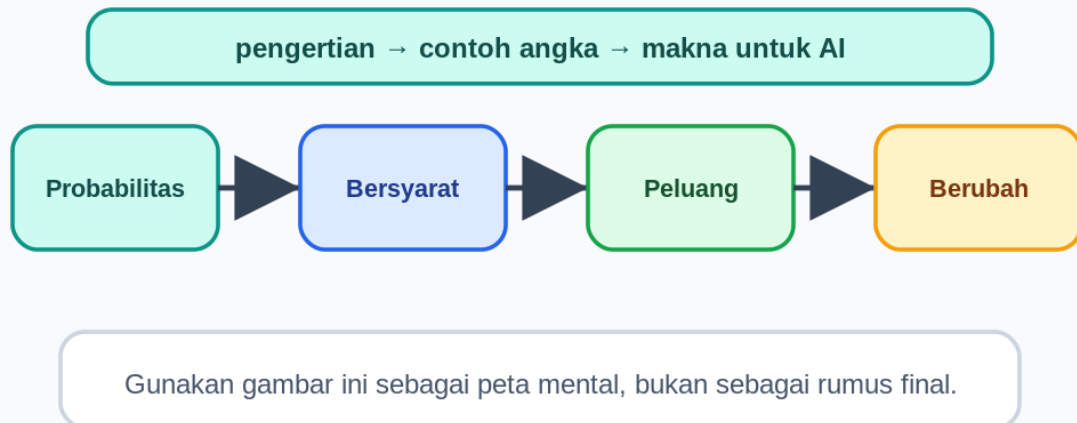
Tes cepat subbab 5

1. Jika peluang pesan bukan spam adalah 0,82, berapa peluang pesan spam?
2. Kapan aturan AND sederhana tidak boleh memakai perkalian langsung?
3. Hitung  $P(\text{tidak spam})$  jika  $P(\text{spam})=0,18$ , lalu hitung  $P(\text{dua koin gambar})$ .

## Subbab 6 — Probabilitas bersyarat: peluang berubah setelah melihat informasi baru

Bab 05 · Halaman 6

### Probabilitas bersyarat: peluang berubah setelah melihat informasi baru



Probabilitas bersyarat: peluang berubah setelah melihat informasi baru

Probabilitas bersyarat menjawab: “berapa peluang A jika kita sudah tahu B?” Notasinya  $P(A | B)$ . Garis vertikal dibaca “dengan syarat”. Misalnya  $P(\text{beli} | \text{klik})$  berarti peluang pengguna membeli setelah ia mengklik produk. Nilai ini bisa jauh lebih besar daripada  $P(\text{beli})$  secara umum.

Inilah mengapa fitur dalam model penting. Tanpa informasi, model hanya tahu peluang umum. Setelah melihat fitur—umur akun, jumlah klik, lokasi, jam akses, riwayat transaksi—model memperbarui keyakinannya. Probabilitas bersyarat adalah jembatan antara data mentah dan prediksi yang lebih tajam.

Contoh sederhana: dari 1000 pengunjung, 100 membeli.  $P(\text{beli})=0,1$ . Dari 200 pengunjung yang memasukkan barang ke keranjang, 80 membeli. Maka  $P(\text{beli} | \text{masuk keranjang})=80/200=0,4$ . Informasi “masuk keranjang” mengubah estimasi secara besar.

Rumus/ide kecil dan/atau latihan ketik manual

$$P(A | B) = P(A \cap B) / P(B), \text{ dengan } P(B) > 0$$

```
p_beli_dan_keranjang = 80
pengunjung_keranjang = 200
print(p_beli_dan_keranjang / pengunjung_keranjang) # 0.4
```

Tes cepat subbab 6

1. Apa arti  $P(\text{beli} | \text{klik})$  dalam bahasa sehari-hari?
2. Mengapa informasi “masuk keranjang” dapat mengubah peluang membeli?
3. Buat tabel kecil 200 pengunjung keranjang dan 80 pembeli, lalu hitung  $P(\text{beli} | \text{keranjang})$ .

## Subbab 7 — Bayes: cara elegan memperbarui keyakinan

Bab 05 · Halaman 7

### Bayes: cara elegan memperbarui keyakinan



Bayes: cara elegan memperbarui keyakinan

Teorema Bayes sering terlihat menakutkan, tetapi intinya sangat manusiawi: keyakinan awal + bukti baru = keyakinan yang diperbarui. Dokter melakukannya saat melihat gejala dan hasil tes. Sistem spam melakukannya saat melihat kata tertentu dalam email. Sistem rekomendasi melakukannya saat melihat riwayat klik.

Bentuk ringkasnya:  $P(A | B) = P(B | A) P(A) / P(B)$ . Jangan buru-buru menghafal. Baca pelan: peluang hipotesis A setelah bukti B bergantung pada seberapa mungkin bukti B muncul jika A benar, seberapa masuk akal A sebelum bukti, dan seberapa umum bukti B secara keseluruhan.

Kesalahan umum adalah mengabaikan base rate. Misalnya sebuah tes sangat akurat, tetapi penyakitnya sangat langka. Hasil positif belum tentu berarti peluang sakit menjadi hampir pasti. AI yang dipakai untuk risiko medis, fraud, atau keamanan harus hati-hati dengan base rate agar tidak membuat alarm berlebihan.

Rumus/ide kecil dan/atau latihan ketik manual

$$P(A | B) = P(B | A) \times P(A) / P(B)$$

posterior = likelihood × prior / evidence

Contoh base-rate: jika fraud hanya 1%, detektor bagus tetap bisa menghasilkan banyak alarm palsu bila alarm total sering terjadi.

Tes cepat subbab 7

1. Mengapa base rate penting ketika membaca hasil tes atau deteksi fraud?
2. Mengapa tes yang akurat tetap bisa memberi kesimpulan salah jika base rate diabaikan?
3. Buat contoh alarm fraud dengan prior kecil dan jelaskan base rate.

## Subbab 8 — Variabel acak: nama untuk angka yang belum terjadi

Bab 05 · Halaman 8

### Variabel acak: nama untuk angka yang belum terjadi



Variabel acak: nama untuk angka yang belum terjadi

Variabel acak adalah kuantitas yang nilainya belum pasti.  $X$  bisa berarti jumlah pesanan besar, waktu tunggu kurir, skor ujian, atau jumlah klik iklan. Kita belum tahu nilainya sekarang, tetapi kita bisa membicarakan kemungkinan nilai-nilainya.

Ada variabel acak diskret dan kontinu. Diskret berarti nilainya terpisah, seperti jumlah pembeli: 0, 1, 2, 3. Kontinu berarti nilainya berada pada rentang, seperti suhu, waktu, atau berat. Dalam praktik AI, keduanya muncul. Kelas gambar adalah diskret; nilai harga rumah adalah kontinu; skor probabilitas model berada pada rentang 0 sampai 1.

Memberi nama variabel acak membantu kita berpikir rapi. Daripada berkata “penjualan yang belum tahu itu”, kita menulis  $X$ . Daripada berkata “waktu pengantaran yang berubah-ubah”, kita menulis  $T$ . Simbol bukan untuk membuat rumit; simbol adalah label singkat agar penjelasan panjang bisa diringkas.

Rumus/ide kecil dan/atau latihan ketik manual

$X$  = variabel acak  
 $X$  diskret: jumlah pembeli  
 $X$  kontinu: waktu tunggu

Tes cepat subbab 8

1. Berikan contoh variabel acak diskret dan variabel acak kontinu dari kehidupan sehari-hari.
2. Mengapa simbol  $X$  membantu kita membahas nilai yang belum diketahui?
3. Tentukan apakah jumlah pembeli, suhu, dan skor probabilitas termasuk diskret atau kontinu.

## Subbab 9 — Distribusi diskret: tabel peluang yang bisa dibaca manusia

Bab 05 · Halaman 9

### Distribusi diskret: tabel peluang yang bisa dibaca manusia



Distribusi diskret: tabel peluang yang bisa dibaca manusia

Distribusi diskret bisa ditulis sebagai tabel: nilai yang mungkin dan peluangnya. Contoh dadu adil: setiap angka 1 sampai 6 punya peluang  $1/6$ . Contoh yang lebih realistis: jumlah pesanan dalam satu jam mungkin 0, 1, 2, 3, 4, dengan peluang berbeda-beda.

Tabel distribusi berguna karena membuat ketidakpastian terlihat. Kita tidak hanya punya “prediksi 2 pesanan”, tetapi juga tahu apakah 2 pesanan sangat dominan atau hanya salah satu kemungkinan. Dua distribusi bisa punya rata-rata sama tetapi risiko berbeda.

Dalam machine learning, output classifier sering berupa distribusi probabilitas atas kelas. Untuk gambar makanan, model mungkin menghasilkan: nasi goreng 0,72, mie ayam 0,18, sate 0,10. Angka tertinggi menjadi prediksi, tetapi distribusi lengkap memberi informasi tingkat keyakinan dan alternatif.

Rumus/ide kecil dan/atau latihan ketik manual

```
distribusi = {"nasi goreng": 0.72, "mie ayam": 0.18, "sate": 0.10}
print(max(distribusi, key=distribusi.get))
print(sum(distribusi.values()))
```

Tes cepat subbab 9

1. Mengapa distribusi probabilitas lebih informatif daripada hanya kelas dengan skor tertinggi?
2. Apa yang hilang jika classifier hanya menampilkan kelas pemenang tanpa skor kelas lain?
3. Buat distribusi tiga kelas makanan dengan total probabilitas 1.

## Subbab 10 — Histogram dan bentuk data: membaca gunung, ekor, dan outlier

Bab 05 · Halaman 10

### Histogram dan bentuk data: membaca gunung, ekor, dan outlier



Histogram dan bentuk data: membaca gunung, ekor, dan outlier

Histogram mengelompokkan data ke dalam interval lalu menghitung frekuensinya. Ini seperti melihat siluet pegunungan data. Apakah datanya menumpuk di tengah? Apakah miring ke kanan? Apakah ada dua puncak? Apakah ada nilai yang sangat jauh?

Bentuk data penting untuk keputusan. Jika penjualan harian biasanya 30 tetapi kadang melonjak ke 200 saat event, rata-rata saja bisa menipu. Jika waktu pengantaran punya ekor panjang, kebanyakan pesanan cepat tetapi sebagian kecil sangat terlambat; pengalaman pelanggan yang buruk mungkin tersembunyi jika kita hanya membaca rata-rata.

AI practitioner yang baik tidak langsung training model. Ia melihat distribusi dulu. Ia bertanya: ada missing value? outlier? kelas tidak seimbang? data berubah antar wilayah? Ini bukan pekerjaan membosankan; ini detektif data. Banyak bug model sebenarnya sudah terlihat dari histogram sederhana.

Rumus/ide kecil dan/atau latihan ketik manual

Histogram menjawab: data menumpuk di mana, miring ke mana, dan apakah ada outlier?

Tes cepat subbab 10

1. Apa yang bisa disembunyikan oleh rata-rata tetapi terlihat pada histogram?
2. Bagaimana histogram membantu menemukan outlier sebelum training model?
3. Sebutkan tiga bentuk histogram yang perlu dicurigai sebelum training.

## Subbab 11 — Mean: pusat gravitasi data, bukan ramalan sakti

Bab 05 · Halaman 11

### Mean: pusat gravitasi data, bukan ramalan sakti



Mean: pusat gravitasi data, bukan ramalan sakti

Mean atau rata-rata adalah jumlah semua nilai dibagi banyaknya nilai. Ia sering menjadi tebakan dasar yang masuk akal: jika tidak tahu apa-apa lagi, prediksi rata-rata historis bisa menjadi baseline. Tetapi mean bukan ramalan sakti. Mean adalah ringkasan pusat data.

Jika data simetris dan stabil, mean sangat berguna. Jika data timpang, mean bisa terseret outlier. Contoh: pendapatan 9 orang sekitar 3 juta, lalu 1 orang 100 juta. Mean kelompok naik besar, tetapi tidak menggambarkan pengalaman mayoritas. Karena itu, dalam data yang timpang, median sering lebih representatif.

Dalam machine learning, baseline mean penting. Sebelum membangun model canggih, coba prediksi nilai rata-rata. Jika model kompleks tidak mengalahkan baseline sederhana, ada masalah. Mungkin fitur kurang informatif, target terlalu bising, atau evaluasi salah.

Rumus/ide kecil dan/atau latihan ketik manual

```
data = [10, 20, 30]
print(sum(data) / len(data)) # 20
```

Tes cepat subbab 11

1. Mengapa mean bisa menyesatkan pada data yang punya outlier besar?
2. Kapan median lebih representatif daripada mean?
3. Bandingkan mean dan median untuk data [3, 3, 4, 100].

## Subbab 12 — Ekspektasi berbobot: ketika semua kemungkinan punya bobot peluang

Bab 05 · Halaman 12

### Ekspektasi berbobot: ketika semua kemungkinan punya bobot peluang



Ekspektasi berbobot: ketika semua kemungkinan punya bobot peluang

Ekspektasi dalam probabilitas adalah rata-rata berbobot peluang. Jika keuntungan 10 ribu terjadi dengan peluang 0,7 dan kerugian 20 ribu terjadi dengan peluang 0,3, ekspektasinya bukan rata-rata biasa, tetapi  $0,7 \times 10.000 + 0,3 \times (-20.000)$ .

Ide ini penting untuk decision making. Keputusan yang tampak menguntungkan di satu skenario bisa buruk jika risiko skenario lain besar. Dalam reinforcement learning nanti, konsep expected reward sangat penting: agen memilih aksi bukan hanya berdasarkan reward terbaik yang mungkin, tetapi nilai harapan jangka panjang.

Ekspektasi berbobot juga membantu memahami prediksi model probabilistik. Jika model memprediksi distribusi permintaan, kita bisa menghitung stok optimal berdasarkan biaya kehabisan stok dan biaya sisa stok. AI yang berguna tidak hanya memprediksi; ia membantu membuat keputusan dengan konsekuensi.

Rumus/ide kecil dan/atau latihan ketik manual

```
nilai = [10000, -20000]
peluang = [0.7, 0.3]
ekspektasi = sum(v*p for v, p in zip(nilai, peluang))
print(ekspektasi)
```

Tes cepat subbab 12

1. Apa bedanya mean biasa dan ekspektasi berbobot peluang?
2. Mengapa keputusan berbasis ekspektasi perlu memasukkan risiko rugi?
3. Hitung ekspektasi dari untung 10.000 peluang 0,7 dan rugi 20.000 peluang 0,3.

## Subbab 13 — Variansi dan standar deviasi: ukuran gelombang data

Bab 05 · Halaman 13

### Variansi dan standar deviasi: ukuran gelombang data



Variansi dan standar deviasi: ukuran gelombang data

Variansi mengukur rata-rata kuadrat jarak data dari mean. Standar deviasi adalah akar variansi. Mengapa selisih dikuadratkan? Karena selisih negatif dan positif tidak boleh saling menghapus, dan kesalahan besar perlu terlihat lebih jelas.

Standar deviasi sering lebih mudah dibaca karena satuannya kembali seperti data asli. Jika penjualan rata-rata 32 gelas dengan standar deviasi 10, kita mendapat rasa bahwa penjualan biasa naik turun sekitar belasan gelas. Ini membantu stok, jadwal karyawan, dan perencanaan bahan baku.

Dalam AI, variance juga punya makna lebih luas: model bisa tidak stabil jika sangat sensitif terhadap data latih. Nanti kita akan bertemu bias-variance tradeoff. Untuk sekarang, cukup pegang intuisi: mean memberi pusat, variance memberi seberapa liar data bergerak dari pusat itu.

Rumus/ide kecil dan/atau latihan ketik manual

```
variance_populasi =  $\Sigma(x - \text{mean})^2 / n$   
variance_sampel =  $\Sigma(x - \text{mean})^2 / (n - 1)$  # sering dipakai untuk estimasi populasi dari sampel  
standard_deviation = sqrt(variance)
```

Bab ini memakai versi populasi/ sederhana agar intuisi awal lebih mudah.

Tes cepat subbab 13

1. Apa informasi tambahan yang diberikan standar deviasi dibanding mean?
2. Apa bedanya variance populasi dan variance sampel secara tujuan?
3. Hitung mean dan variance sederhana untuk data [30, 31, 29].

## Subbab 14 — Percentile dan kuantil: melihat posisi dalam antrean data

Bab 05 · Halaman 14

### Percentile dan kuantil: melihat posisi dalam antrean data



Percentile dan kuantil: melihat posisi dalam antrean data

Percentile menjawab posisi relatif. Persentil ke-90 berarti 90% data berada di bawah atau sama dengan nilai itu. Untuk waktu pengantaran, p90 jauh lebih informatif daripada rata-rata jika kita peduli pengalaman buruk. Rata-rata 20 menit terdengar bagus, tetapi p90 60 menit berarti 10% pelanggan menunggu sangat lama.

Kuantil sering dipakai dalam dashboard dan SLA. Median adalah persentil ke-50. Q1 sekitar persentil ke-25, Q3 sekitar persentil ke-75. Dengan kuantil, kita bisa melihat rentang tengah data tanpa terlalu dipengaruhi outlier.

Dalam evaluasi AI, percentile berguna untuk latensi model, error prediksi, dan risiko. Produk komersial tidak cukup bertanya “rata-rata cepat?”. Ia harus bertanya: “bagaimana pengalaman 5% pengguna terburuk?”.

Rumus/ide kecil dan/atau latihan ketik manual

```
data = sorted([10, 20, 30, 40, 50])
median = data[len(data)//2]
print(median)
```

Tes cepat subbab 14

1. Mengapa p90 waktu pengantaran bisa lebih penting daripada rata-rata?
2. Mengapa p90 berguna untuk melihat pengalaman pengguna terburuk?
3. Urutkan [10,20,30,40,50] dan tentukan median serta perkiraan p90.

## Subbab 15 — Sampling: jendela kecil untuk melihat populasi besar

Bab 05 · Halaman 15

### Sampling: jendela kecil untuk melihat populasi besar



Sampling: jendela kecil untuk melihat populasi besar

Populasi adalah seluruh kelompok yang ingin kita pahami. Sampel adalah sebagian yang benar-benar kita amati. Kita jarang punya seluruh populasi. Kita melihat transaksi beberapa bulan, pengguna yang aktif, responden survei, atau data dari wilayah tertentu.

Sampel yang baik harus mewakili populasi tujuan. Jika aplikasi ingin dipakai di seluruh Indonesia tetapi data hanya dari kota besar dan ponsel mahal, model bisa gagal pada daerah dengan koneksi lambat atau perangkat sederhana. Ini bukan sekadar masalah teknis; ini masalah keadilan dan kegunaan.

Sampling mengajarkan kerendahan hati. Kesimpulan selalu punya konteks: dari data siapa, periode kapan, dengan cara kumpul seperti apa. AI komersial yang serius harus mencatat asal data, batas cakupan, dan kelompok yang mungkin kurang terwakili.

Rumus/ide kecil dan/atau latihan ketik manual

Populasi = semua yang ingin dipahami  
Sampel = sebagian yang diamati  
Sampel baik  $\approx$  mewakili populasi tujuan

Tes cepat subbab 15

1. Mengapa sampel dari kota besar saja bisa menjadi masalah untuk produk nasional?
2. Apa ciri sampel yang representatif untuk produk nasional?
3. Rancang sampel pengguna aplikasi edukasi yang tidak hanya berasal dari kota besar.

## Subbab 16 — Sampling error: mengapa hasil sampel bisa naik turun

Bab 05 · Halaman 16

### Sampling error: mengapa hasil sampel bisa naik turun



Sampling error: mengapa hasil sampel bisa naik turun

Sampling error adalah perbedaan alami antara statistik sampel dan statistik populasi. Jika kita ambil sampel 10 orang, mean bisa berbeda jauh dari populasi. Jika sampel 1000 orang, biasanya lebih stabil. Ini bukan karena peneliti selalu salah, tetapi karena sampel memang hanya sebagian.

Konsep ini membantu kita tidak terlalu cepat percaya pada angka tunggal. Jika survei mengatakan kepuasan 82%, kita perlu bertanya: berapa jumlah responden? bagaimana dipilih? margin of error berapa? Jika A/B test menunjukkan kenaikan 0,5%, apakah itu sinyal nyata atau noise?

Dalam machine learning, train-test split juga mengandung sampling. Skor model pada test set adalah estimasi performa masa depan, bukan kebenaran absolut. Karena itu, validasi silang, ukuran test set, dan monitoring setelah rilis penting.

Rumus/ide kecil dan/atau latihan ketik manual

```
import random
populasi = list(range(1, 101))
for n in [5, 30]:
    sampel = random.sample(populasi, n)
    print(n, sum(sampel)/len(sampel))
```

Tes cepat subbab 16

1. Mengapa skor model di test set tetap hanya estimasi, bukan jaminan performa masa depan?
2. Mengapa mean sampel kecil dapat berbeda jauh dari mean populasi?
3. Ambil contoh populasi 1-10, pilih sampel 3 angka, lalu bandingkan mean sampel dan populasi.

## Subbab 17 — Bias data: ketika sampel bukan hanya kecil, tetapi miring

Bab 05 · Halaman 17

### Bias data: ketika sampel bukan hanya kecil, tetapi miring



Bias data: ketika sampel bukan hanya kecil, tetapi miring

Bias data terjadi ketika data tidak mewakili kenyataan yang ingin dilayani. Sampel kecil bisa diperbaiki dengan menambah data; sampel bias lebih licin karena jumlah besar pun bisa tetap salah arah. Jika semua data rekrutmen masa lalu berasal dari proses yang diskriminatif, model yang belajar dari data itu bisa mengulang diskriminasi.

Bias juga bisa muncul dari cara label dibuat. Jika “pelanggan berkualitas” didefinisikan hanya berdasarkan pembelian mahal, model mungkin mengabaikan pelanggan setia dengan nilai transaksi kecil. Jika data keluhan hanya dari pengguna yang punya internet stabil, masalah pengguna koneksi buruk bisa tidak terlihat.

Karena itu, statistik harus selalu dipasangkan dengan pertanyaan etis dan domain. Siapa yang masuk data? Siapa yang tidak? Label dibuat oleh siapa? Apa konsekuensi jika model salah? Bab ini tidak menyelesaikan semua isu fairness, tetapi menanam kebiasaan penting: jangan menyembah angka tanpa memahami asalnya.

Rumus/ide kecil dan/atau latihan ketik manual

Bias data = data sistematis miring dari populasi/tujuan nyata  
Data besar + bias tetap bisa menghasilkan model salah arah

```
kota_besar = [8, 9, 8, 10, 9]
daerah_lambat = [5, 6, 4, 6, 5]
print(sum(kota_besar)/len(kota_besar))
print(sum(daerah_lambat)/len(daerah_lambat))
```

Tes cepat subbab 17

1. Apa bedanya sampel kecil dan sampel bias?
2. Siapa yang mungkin tidak terwakili dalam data contohmu?
3. Bandingkan mean kepuasan kota besar vs daerah lambat dari contoh lab bias.

## Subbab 18 — Korelasi: dua hal bergerak bersama

Bab 05 · Halaman 18

### Korelasi: dua hal bergerak bersama



Korelasi: dua hal bergerak bersama

Korelasi mengukur hubungan gerak bersama antara dua variabel. Jika suhu naik dan penjualan minuman dingin naik, keduanya berkorelasi positif. Jika harga naik dan jumlah pembelian turun, mungkin berkorelasi negatif. Korelasi membantu menemukan sinyal awal.

Tetapi korelasi bisa lemah, kuat, linear, non-linear, atau dipengaruhi outlier. Satu titik ekstrem bisa membuat korelasi tampak kuat padahal mayoritas data tidak menunjukkan pola. Scatter plot sering lebih jujur daripada hanya satu angka korelasi.

Dalam AI, korelasi adalah bahan bakar prediksi. Model sering tidak perlu tahu sebab-akibat untuk memprediksi. Namun untuk keputusan intervensi, sebab-akibat penting. Jika model menemukan pelanggan yang sering komplain juga sering berhenti langganan, memberi diskon mungkin membantu, tetapi bisa juga masalah utamanya kualitas layanan.

Rumus/ide kecil dan/atau latihan ketik manual

Korelasi positif: X naik, Y cenderung naik  
Korelasi negatif: X naik, Y cenderung turun  
Korelasi  $\neq$  sebab-akibat

Tes cepat subbab 18

1. Mengapa scatter plot penting saat membaca korelasi?
2. Mengapa angka korelasi sebaiknya dibaca bersama scatter plot?
3. Buat pasangan suhu dan penjualan es teh yang menunjukkan korelasi positif.

## Subbab 19 — Kausalitas: jangan salah menekan tombol dunia nyata

Bab 05 · Halaman 19

### Kausalitas: jangan salah menekan tombol dunia nyata



Kausalitas: jangan salah menekan tombol dunia nyata

Kausalitas bertanya apakah perubahan A menyebabkan perubahan B. Ini lebih kuat daripada korelasi. Untuk membuktikan kausalitas, kita sering membutuhkan eksperimen, randomisasi, desain kuasi-eksperimen, atau pengetahuan domain yang kuat.

Kesalahan kausal bisa mahal. Jika perusahaan melihat pengguna yang menerima notifikasi lebih sering membeli, belum tentu notifikasi menyebabkan pembelian. Bisa jadi sistem memang mengirim notifikasi ke pengguna yang sudah aktif. Jika langsung menambah notifikasi ke semua orang, hasilnya bisa spam dan churn.

AI yang dipakai untuk rekomendasi kebijakan, kesehatan, pendidikan, atau kredit harus berhati-hati. Prediksi menjawab “apa yang mungkin terjadi?”. Kausalitas menjawab “apa yang terjadi jika kita melakukan sesuatu?”. Dua pertanyaan ini mirip, tetapi tidak sama.

Rumus/ide kecil dan/atau latihan ketik manual

Prediksi: apa yang mungkin terjadi?

Kausalitas: apa yang terjadi jika kita melakukan intervensi?

Tes cepat subbab 19

1. Apa perbedaan pertanyaan prediktif dan pertanyaan kausal?
2. Mengapa rekomendasi intervensi membutuhkan bukti kausal, bukan hanya prediksi?
3. Ubah pertanyaan “siapa akan churn?” menjadi pertanyaan kausal “apa efek diskon?”.

## Subbab 20 — Loss: bahasa resmi untuk “model masih salah”

Bab 05 · Halaman 20

### Loss: bahasa resmi untuk “model masih salah”



Loss: bahasa resmi untuk “model masih salah”

Agar model belajar, kesalahan harus dijadikan angka. Angka itulah loss. Jika prediksi stok 32 dan aktual 40, errornya -8. Dari error ini kita bisa membuat absolute error 8 atau squared error 64. Loss mengubah rasa salah menjadi ukuran yang bisa dioptimalkan.

Loss harus cocok dengan tujuan. Untuk prediksi angka, MAE dan MSE umum dipakai. Untuk klasifikasi, cross-entropy sering dipakai. Untuk ranking, loss bisa dirancang agar urutan rekomendasi lebih baik. Salah memilih loss berarti model belajar mengejar tujuan yang salah.

Contoh bisnis: jika kehabisan stok jauh lebih mahal daripada sisa stok, loss sebaiknya tidak memperlakukan keduanya sama. Prediksi terlalu rendah mungkin harus dihukum lebih besar. Ini menunjukkan bahwa matematika AI selalu bertemu nilai dan keputusan manusia.

Rumus/ide kecil dan/atau latihan ketik manual

```
aktual = 40
prediksi = 32
error = prediksi - aktual
print(error, abs(error), error**2)
```

Tes cepat subbab 20

1. Mengapa loss harus disesuaikan dengan tujuan masalah?
2. Mengapa loss harus mengubah kesalahan menjadi angka?
3. Hitung error, absolute error, dan squared error untuk aktual 40 prediksi 32.

## Subbab 21 — MAE vs MSE: dua cara berbeda menghukum kesalahan

Bab 05 · Halaman 21

### MAE vs MSE: dua cara berbeda menghukum kesalahan



MAE vs MSE: dua cara berbeda menghukum kesalahan

MAE menghitung rata-rata nilai absolut error. Jika model meleset 5 unit, kontribusinya 5. MSE menghitung rata-rata kuadrat error. Jika model meleset 5 unit, kontribusinya 25. Karena dikuadratkan, MSE lebih keras terhadap error besar.

MAE mudah dijelaskan ke orang bisnis: “rata-rata model meleset 3,75 gelas.” MSE lebih sensitif terhadap kesalahan ekstrem, sehingga sering berguna ketika error besar sangat berbahaya. Tetapi MSE juga bisa membuat model terlalu fokus pada outlier jika outlier berasal dari noise atau data salah.

Tidak ada loss universal terbaik. Pilihan loss adalah pilihan desain. Pembaca perlu membiasakan diri bertanya: kesalahan jenis apa yang paling mahal? apakah outlier valid? apakah interpretasi mudah dibutuhkan? apakah target punya satuan yang jelas?

Rumus/ide kecil dan/atau latihan ketik manual

$MAE = \text{mean}(|\text{prediksi} - \text{aktual}|)$   
 $MSE = \text{mean}((\text{prediksi} - \text{aktual})^2)$   
MSE menghukum error besar lebih keras.

Tes cepat subbab 21

1. Kapan MSE bisa lebih cocok daripada MAE? Kapan MAE lebih mudah dipakai?
2. Mengapa satu outlier membuat MSE melonjak?
3. Bandingkan MAE/MSE untuk error [1,1,10].

## Subbab 22 — Turunan dari perubahan kecil: melihat sensitivitas

Bab 05 · Halaman 22

### Turunan dari perubahan kecil: melihat sensitivitas



Turunan dari perubahan kecil: melihat sensitivitas

Turunan sering diajarkan dengan simbol yang membuat pemula mundur. Padahal intinya dekat: jika  $x$  berubah sedikit,  $y$  berubah seberapa cepat? Jika harga naik Rp1.000, permintaan turun berapa? Jika bobot model naik 0,01, loss naik atau turun?

Kita bisa memulai dari finite difference:  $(f(x+h)-f(x))/h$ . Ini bukan definisi formal lengkap, tetapi sangat membantu intuisi. Kita geser  $x$  sedikit, lihat perubahan fungsi, lalu bagi dengan ukuran geser. Jika  $h$  sangat kecil, kita mendekati kemiringan lokal.

Untuk  $\text{loss}(x)=(x-3)^2$ , turunan analitiknya  $2(x-3)$ . Ketika  $x$  lebih kecil dari 3, turunan negatif. Ketika  $x$  lebih besar dari 3, turunan positif. Di  $x=3$ , turunan nol. Titik turunan nol adalah kandidat lembah, puncak, atau dataran.

Rumus/ide kecil dan/atau latihan ketik manual

```
def f(x): return (x - 3)**2
h = 0.00001
print((f(5+h) - f(5)) / h) # mendekati 4
```

Tes cepat subbab 22

1. Apa arti turunan dalam kalimat “jika input berubah sedikit, output berubah seberapa cepat?”
2. Mengapa finite difference membantu memahami turunan sebelum rumus formal?
3. Hitung finite difference kasar untuk  $f(x)=(x-3)^2$  di  $x=5$  dengan  $h=0,1$ .

## Subbab 23 — Dari turunan ke gradient: banyak tombol, banyak arah

Bab 05 · Halaman 23

### Dari turunan ke gradient: banyak tombol, banyak arah



Dari turunan ke gradient: banyak tombol, banyak arah

Turunan satu variabel seperti satu tombol volume. Gradient adalah versi banyak tombol. Model AI punya banyak parameter: bobot fitur harga, bobot fitur klik, bobot fitur jarak, bias, bahkan jutaan bobot dalam neural network. Gradient memberi tahu efek perubahan kecil pada setiap parameter terhadap loss.

Jika gradient untuk bobot harga positif, menaikkan bobot itu membuat loss naik; maka kita cenderung menurunkannya. Jika gradient negatif, menaikkan bobot bisa menurunkan loss. Dengan cara ini, model punya petunjuk arah perbaikan untuk setiap parameter.

Gradient adalah vektor. Bab 4 membantu kita melihat vektor sebagai panah. Di Bab 5, panah gradient hidup di ruang parameter. Ia menunjuk arah loss naik tercepat. Untuk menurunkan loss, gradient descent berjalan ke arah sebaliknya.

Rumus/ide kecil dan/atau latihan ketik manual

$$\text{gradient} = [\partial\text{Loss}/\partial w_1, \partial\text{Loss}/\partial w_2, \dots, \partial\text{Loss}/\partial b]$$
$$\text{Arah turun} = -\text{gradient}$$

Tes cepat subbab 23

1. Mengapa gradient bisa dianggap vektor arah di ruang parameter?
2. Mengapa gradient untuk banyak parameter bisa dipandang sebagai vektor?
3. Tulis gradient dua parameter sebagai  $[d\text{Loss}/dw, d\text{Loss}/db]$ .

## Subbab 24 — Permukaan loss: bukit, lembah, dan dataran

Bab 05 · Halaman 24

### Permukaan loss: bukit, lembah, dan dataran



Permukaan loss: bukit, lembah, dan dataran

Bayangkan loss sebagai permukaan tanah. Parameter model adalah posisi kita di peta. Jika loss tinggi, posisi kita berada di bukit. Jika loss rendah, kita berada di lembah. Training adalah perjalanan mencari posisi parameter dengan loss rendah.

Pada fungsi sederhana, permukaan loss seperti mangkuk halus. Tetapi model nyata bisa punya permukaan rumit: banyak lembah, pelana, dataran, dan jalur berkeluk. Neural network modern bahkan punya jutaan dimensi, sehingga kita tidak bisa menggambarinya secara langsung. Visual bukit hanya analogi, tetapi berguna untuk memahami arah.

Masalah praktis muncul ketika gradient sangat kecil di dataran: model belajar lambat. Jika gradient sangat besar: update bisa tidak stabil. Optimisasi modern memakai banyak trik seperti momentum, adaptive learning rate, normalisasi, dan scheduler. Bab ini hanya fondasi agar trik itu tidak terasa seperti sihir.

Rumus/ide kecil dan/atau latihan ketik manual

Ruang parameter: posisi model  
Permukaan loss: tinggi-rendah kesalahan  
Training: mencari posisi dengan loss lebih rendah

Tes cepat subbab 24

1. Mengapa analogi permukaan loss berguna walaupun model nyata berdimensi sangat tinggi?
2. Apa keterbatasan analogi bukit-lembah untuk neural network besar?
3. Gambar secara verbal posisi parameter di bukit loss tinggi dan lembah loss rendah.

## Subbab 25 — Gradient descent langkah demi langkah

Bab 05 · Halaman 25

### Gradient descent langkah demi langkah



Gradient descent langkah demi langkah

Gradient descent punya empat langkah berulang. Pertama, mulai dari parameter awal. Kedua, hitung loss. Ketiga, hitung gradient. Keempat, ubah parameter ke arah negatif gradient dengan ukuran learning rate. Ulangi sampai loss cukup kecil atau tidak membaik.

Rumus intinya:  $\text{parameter\_baru} = \text{parameter\_lama} - \text{learning\_rate} \times \text{gradient}$ . Tanda minus adalah jantungnya. Gradient menunjuk arah naik; kita ingin turun. Learning rate menentukan panjang langkah.

Untuk  $\text{loss}(x) = (x-3)^2$ , jika  $x = -4$ ,  $\text{gradient} = 2(-4-3) = -14$ . Update dengan learning rate 0,1 menjadi  $x_{\text{baru}} = -4 - 0,1 \times (-14) = -2,6$ .  $x$  bergerak mendekati 3. Setelah banyak langkah,  $x$  mendekati lembah.

Rumus/ide kecil dan/atau latihan ketik manual

$$x_{\text{baru}} = x_{\text{lama}} - \text{learning\_rate} \times \text{gradient}$$

$$x = -4, \text{gradient} = -14, \text{lr} = 0.1$$

$$x_{\text{baru}} = -4 - 0.1 \times (-14) = -2.6$$

Tes cepat subbab 25

1. Hitung satu langkah update dari  $x = -4$ ,  $\text{gradient} = -14$ , dan learning rate 0,1.
2. Mengapa tanda minus muncul pada rumus update?
3. Hitung satu langkah update untuk  $x = 5$ ,  $\text{gradient} = 4$ , learning rate = 0,1.

## Subbab 26 — Learning rate: seni mengatur ukuran langkah

Bab 05 · Halaman 26

### Learning rate: seni mengatur ukuran langkah



Learning rate: seni mengatur ukuran langkah

Learning rate kecil membuat training aman tetapi lambat. Learning rate sedang bisa cepat dan stabil. Learning rate terlalu besar bisa melompat melewati lembah, berosilasi, atau divergen. Karena itu, learning rate sering menjadi hyperparameter paling penting.

Dalam contoh  $\text{loss}(x) = (x-3)^2$ , learning rate 0,01 bergerak lambat. Learning rate 0,1 mendekati lembah dengan baik. Learning rate 1,1 bisa meledak menjauh karena setiap update terlalu agresif. Melihat perbandingan ini membuat konsep “training unstable” menjadi nyata.

Dalam praktik deep learning, learning rate sering tidak tetap. Ada warmup, decay, scheduler, dan optimizer adaptif seperti Adam. Tetapi semua variasi itu tetap berakar pada pertanyaan sederhana: seberapa besar langkah yang aman untuk menurunkan loss?

Rumus/ide kecil dan/atau latihan ketik manual

lr terlalu kecil → lambat  
lr cukup → turun stabil  
lr terlalu besar → melompat/divergen

Tes cepat subbab 26

1. Mengapa learning rate terlalu besar bisa membuat loss naik, bukan turun?
2. Apa gejala training ketika learning rate terlalu besar?
3. Bandingkan hasil  $lr=0,01$ ,  $0,1$ , dan  $1,1$  dari praktikum.

## Subbab 27 — Stochastic gradient descent: belajar dari batch kecil

Bab 05 · Halaman 27

### Stochastic gradient descent: belajar dari batch kecil



Stochastic gradient descent: belajar dari batch kecil

Jika data sangat besar, menghitung gradient dari seluruh dataset setiap langkah bisa mahal. Stochastic gradient descent (SGD) memakai satu contoh atau batch kecil untuk memperkirakan gradient. Arah geraknya lebih berisik, tetapi jauh lebih cepat.

Analogi: jika ingin tahu arah angin di kota, kita tidak selalu mengukur setiap jalan. Kita ambil beberapa titik. Estimasinya mungkin berisik, tetapi cukup untuk bergerak. Mini-batch training memakai beberapa data sekaligus agar noise tidak terlalu liar.

Noise SGD kadang membantu model keluar dari lembah buruk atau dataran. Tetapi noise terlalu besar juga bisa membuat training tidak stabil. Inilah mengapa ukuran batch, learning rate, dan shuffle data saling berkaitan. Bab ini belum masuk detail optimizer modern, tetapi pembaca sudah punya intuisi utamanya.

Rumus/ide kecil dan/atau latihan ketik manual

Full batch: gradient dari semua data  
Mini-batch: gradient dari sebagian data  
SGD: gradient dari satu/mini data, lebih berisik tetapi praktis

Tes cepat subbab 27

1. Apa keuntungan dan kelemahan memakai mini-batch dibanding seluruh dataset?
2. Mengapa mini-batch gradient lebih berisik tetapi praktis?
3. Jelaskan dengan angka kecil perbedaan memakai 1 data vs 10 data untuk estimasi gradient.

## Subbab 28 — Gradient untuk model linear kecil

Bab 05 · Halaman 28

### Gradient untuk model linear kecil



Gradient untuk model linear kecil

Sekarang kita hubungkan Bab 4 dan Bab 5. Model linear sederhana:  $\text{prediksi} = w \times x + b$ . Parameter yang bisa dipelajari adalah  $w$  dan  $b$ . Loss bisa berupa squared error:  $(\text{prediksi} - \text{aktual})^2$ . Gradient memberi tahu bagaimana  $w$  dan  $b$  harus diubah.

Jika prediksi terlalu tinggi, error positif. Untuk menurunkan prediksi, model mungkin menurunkan  $w$  atau  $b$  tergantung nilai  $x$ . Jika  $x$  besar, perubahan  $w$  punya efek besar. Inilah alasan fitur perlu diskalakan: fitur dengan skala besar bisa membuat gradient besar dan training tidak seimbang.

Rumus gradient untuk satu data:  $d\text{Loss}/dw = 2 \times \text{error} \times x$  dan  $d\text{Loss}/db = 2 \times \text{error}$ . Rumus ini tampak teknis, tetapi artinya masuk akal: makin besar error, makin besar koreksi; makin besar fitur  $x$ , makin besar pengaruh  $w$ .

Rumus/ide kecil dan/atau latihan ketik manual

```
prediksi = w * x + b
error = prediksi - aktual
dLoss/dw = 2 * error * x
dLoss/db = 2 * error
```

Tes cepat subbab 28

1. Mengapa fitur berskala besar bisa membuat update bobot menjadi terlalu besar?
2. Mengapa fitur berskala besar memperbesar  $d\text{Loss}/dw$ ?
3. Hitung  $d\text{Loss}/dw$  jika  $\text{error}=3$  dan  $x=2$ .

## Subbab 29 — Training model linear mini dari nol

Bab 05 · Halaman 29

### Training model linear mini dari nol



Training model linear mini dari nol

Dengan beberapa baris Python, kita bisa melatih model linear kecil tanpa library ML. Kita mulai dari  $w=0$  dan  $b=0$ , hitung prediksi, hitung error, hitung gradient rata-rata, lalu update  $w$  dan  $b$ . Setelah banyak langkah, garis prediksi mendekati pola data.

Latihan ini penting bukan karena kita akan selalu menulis training loop manual. Dalam proyek nyata, kita memakai scikit-learn, PyTorch, TensorFlow, atau JAX. Tetapi menulis versi kecil membuat kita tahu apa yang terjadi di balik layar. Ketika nanti melihat `optimizer.step()`, kita tidak menganggapnya mantra.

Praktikum Bab 5 menampilkan dua training: fungsi satu dimensi ( $x$  menuju 3) dan model linear kecil. Pembaca diminta mengubah learning rate, jumlah langkah, dan data agar melihat sendiri perilaku training. Pemahaman terbaik lahir dari eksperimen kecil yang bisa diulang.

Rumus/ide kecil dan/atau latihan ketik manual

```
w, b = 0.0, 0.0
# hitung prediksi → error → gradient → update
# inilah versi kecil dari training loop ML
```

Tes cepat subbab 29

1. Mengapa menulis training loop manual berguna meskipun nanti memakai library?
2. Mengapa training loop manual membuat optimizer library lebih mudah dipahami?
3. Ubah target menjadi  $y=3x+2$  dan amati apakah  $w, b$  mendekat.

## Subbab 30 — Evaluasi training: jangan hanya melihat angka akhir

Bab 05 · Halaman 30

### Evaluasi training: jangan hanya melihat angka akhir



Evaluasi training: jangan hanya melihat angka akhir

Saat training, jangan hanya melihat loss akhir. Lihat kurva loss dari waktu ke waktu. Jika loss turun stabil, training sehat. Jika loss naik, learning rate mungkin terlalu besar atau data bermasalah. Jika loss datar, learning rate mungkin terlalu kecil, fitur kurang informatif, atau model tidak cocok.

Bandingkan juga train loss dan validation loss. Jika train loss turun tetapi validation loss naik, model mungkin overfitting: hafal data latih tetapi gagal pada data baru. Jika keduanya tinggi, model underfitting: belum cukup mampu menangkap pola.

Bab ini baru memberi dasar. Nanti di bab supervised learning dan neural network, kita akan memakai grafik learning curve, split data, regularisasi, dan metrik evaluasi. Tetapi kebiasaan kritis dimulai sekarang: angka bagus harus diperiksa, bukan langsung dipercaya.

Rumus/ide kecil dan/atau latihan ketik manual

```
Loss turun train & validasi: sehat  
Train turun, validasi naik: kemungkinan overfitting  
Keduanya tinggi: kemungkinan underfitting  
  
train_loss = [1.0, 0.6, 0.3]  
valid_loss = [1.1, 0.8, 0.9]  
print("validasi mulai naik?", valid_loss[-1] > valid_loss[-2])
```

Tes cepat subbab 30

1. Apa tanda sederhana bahwa model mungkin overfitting?
2. Apa yang harus diperiksa selain loss akhir?
3. Buat tiga angka train loss dan validation loss yang menunjukkan overfitting.

## Subbab 31 — Praktikum besar Bab 5: pasar kecil, loss, dan gradient

Bab 05 · Halaman 31

### Praktikum besar Bab 5: pasar kecil, loss, dan gradient



Praktikum besar Bab 5: pasar kecil, loss, dan gradient

Praktikum Bab 5 menggabungkan semua konsep. Kita mensimulasikan dadu untuk probabilitas, meringkas data penjualan, menghitung MAE/MSE, mencoba finite difference, menjalankan gradient descent satu variabel, lalu melatih model linear kecil. Semua memakai Python standard library agar bisa dijalankan di terminal, VS Code, Jupyter, Colab, atau Kaggle.

Jangan hanya menekan Run. Ketik ulang beberapa bagian. Ubah data penjualan. Ubah seed. Ubah learning rate. Tambahkan outlier. Lihat bagaimana mean, standard deviation, dan loss berubah. Cara ini membuat rumus menjadi pengalaman, bukan hafalan.

Tujuan praktikum bukan membuat model paling akurat. Tujuannya membangun rasa: bagaimana data bergerak, bagaimana loss bereaksi, dan bagaimana gradient memperbaiki parameter. Rasa inilah yang akan menyelamatkan pembaca saat nanti menghadapi model lebih besar.

Rumus/ide kecil dan/atau latihan ketik manual

```
python3 gradient_descent_playground.py
```

Ubah seed, outlier, learning rate, dan data linear; tulis interpretasi setelah setiap run.

Tes cepat subbab 31

1. Sebutkan tiga eksperimen kecil yang bisa kamu lakukan di praktikum Bab 5.
2. Bagaimana kamu akan mencatat hasil eksperimen agar bisa diulang?
3. Tulis tabel eksperimen: seed, data, learning rate, loss akhir, catatan.

## Subbab 32 — Motivasi penutup: belajar AI juga seperti gradient descent

Bab 05 · Halaman 32

### Motivasi penutup: belajar AI juga seperti gradient descent



Motivasi penutup: belajar AI juga seperti gradient descent

Gradient descent adalah metafora belajar yang indah. Kita mulai dari posisi jauh dari lembah. Kita salah. Kita ukur kesalahan. Kita ambil langkah kecil. Kadang terlalu lambat, kadang terlalu besar, kadang harus menyesuaikan arah. Tetapi jika terus mengukur dan memperbaiki, kita makin dekat.

Belajar AI juga begitu. Tidak perlu langsung memahami semua rumus. Pahami satu cerita, satu visual, satu kode kecil, satu rumus. Jika bingung, itu bukan tanda gagal; itu tanda otak sedang membangun peta baru. Pembaca yang bertahan sampai subbab ini sudah melewati salah satu gerbang terpenting AI: probabilitas, statistik, loss, gradient, dan optimisasi.

Di bab berikutnya, konsep ini berubah menjadi machine learning fundamentals. Kita akan melihat bagaimana data, fitur, target, model, loss, training, evaluasi, dan generalisasi tersusun menjadi workflow utuh. Bab 5 adalah mesin kecil di dalamnya.

Rumus/ide kecil dan/atau latihan ketik manual

Belajar = ukur kesalahan → ambil langkah kecil → evaluasi → perbaiki arah

Tes cepat subbab 32

1. Tuliskan dengan bahasamu sendiri: mengapa belajar AI mirip gradient descent?
2. Jika belajar terasa lambat, apa “learning rate” pribadi yang bisa kamu atur?
3. Tulis satu rencana belajar 3 langkah kecil setelah Bab 5.

## Latihan hitung terstruktur Bab 5

Kerjakan pelan-pelan sebelum melihat kode.

### Soal 1 — Probabilitas frekuensi

Dalam 500 transaksi, 35 terlambat diproses.

$$\begin{aligned}P(\text{terlambat}) &= 35/500 = 0,07 \\P(\text{tidak terlambat}) &= 1 - 0,07 = 0,93\end{aligned}$$

Makna: jika proses serupa berulang, sekitar 7 dari 100 transaksi berisiko terlambat menurut data ini.

### Soal 2 — Conditional probability

Dari 200 pengunjung yang memasukkan barang ke keranjang, 80 membeli.

$$P(\text{beli} \mid \text{keranjang}) = 80/200 = 0,4$$

Bandungkan dengan  $P(\text{beli})$  umum, misalnya  $100/1000 = 0,1$ . Informasi keranjang menaikkan estimasi dari 0,1 ke 0,4.

### Soal 3 — Bayes dan base rate

Prior fraud 1%, detektor menangkap 90% fraud, dan total alarm 5%.

$$\begin{aligned}P(\text{fraud} \mid \text{alarm}) &= P(\text{alarm} \mid \text{fraud})P(\text{fraud})/P(\text{alarm}) \\&= 0,90 \times 0,01 / 0,05 \\&= 0,18\end{aligned}$$

Walau detektor terlihat kuat, peluang fraud setelah alarm hanya 18% karena base rate fraud kecil dan alarm total cukup sering.

### Soal 4 — Variance populasi dan sampel

Data [30, 31, 29], mean = 30.

$$\begin{aligned}\text{selisih} &= [0, 1, -1] \\ \text{kuadrat} &= [0, 1, 1] \\ \text{variance\_populasi} &= (0+1+1)/3 = 0,667 \\ \text{variance\_sampel} &= (0+1+1)/(3-1) = 1\end{aligned}$$

Sample variance lebih besar karena dipakai untuk mengestimasi populasi dari sampel kecil.

### Soal 5 — Turunan dan gradient descent

Loss  $(x-3)^2$ , turunan  $2(x-3)$ . Mulai  $x=-4$ , learning rate 0,1.

$$\begin{aligned}\text{grad} &= 2(-4-3) = -14 \\ x_{\text{baru}} &= -4 - 0,1(-14) = -2,6 \\ \text{loss lama} &= 49 \\ \text{loss baru} &= (-2,6-3)^2 = 31,36\end{aligned}$$

Loss turun. Ini contoh nyata bagaimana gradient memberi arah perbaikan.

### Soal 6 — Gradient model linear

Prediksi  $\hat{y} = wx+b$ , loss satu data  $(\hat{y}-y)^2$ . Jika  $x=2$ ,  $y=7$ ,  $w=1$ ,  $b=1$ :

```
 $\hat{y} = 1 \times 2 + 1 = 3$   
error =  $\hat{y} - y = -4$   
dLoss/dw =  $2 \times \text{error} \times x = 2 \times (-4) \times 2 = -16$   
dLoss/db =  $2 \times \text{error} = -8$ 
```

Gradient negatif berarti menaikkan w dan b akan menurunkan loss pada contoh ini.

## Praktikum terpandu Bab 5

File utama:

- `code/gradient_descent_playground.py`
- `code/gradient_descent_playground.ipynb`

Jalankan dari terminal:

```
cd zero-to-hero-menaklukkan-ai/chapters/05-probabilitas-statistik-gradient/code  
python3 gradient_descent_playground.py
```

Eksperimen wajib:

1. Ubah jumlah simulasi dadu: 100, 1000, 10000.
2. Tambahkan outlier pada data penjualan dan lihat mean/std berubah.
3. Bandingkan MAE dan MSE saat ada satu error besar.
4. Ubah learning rate: 0.01, 0.1, 1.1.
5. Ubah data model linear kecil dan lihat bobot akhir.

## Ringkasan Bab 5 yang diperluas

- Probabilitas adalah bahasa ketidakpastian.
- Statistik adalah alat untuk belajar dari data terbatas.
- Distribusi lebih kaya daripada satu angka rata-rata.
- Mean, variance, standard deviation, percentile, dan histogram menjawab pertanyaan berbeda.
- Sampling harus representatif; data besar tetap bisa bias.
- Korelasi membantu prediksi, tetapi kausalitas diperlukan untuk intervensi.
- Loss mengubah kesalahan model menjadi angka yang bisa dioptimalkan.
- Turunan mengukur sensitivitas perubahan kecil.
- Gradient adalah vektor arah perubahan loss terhadap parameter.
- Gradient descent memperbaiki parameter dengan langkah kecil berulang.
- Learning rate terlalu kecil membuat lambat; terlalu besar bisa divergen.
- SGD dan mini-batch membuat training besar menjadi praktis.
- Training perlu didiagnosis lewat kurva loss dan validasi, bukan angka akhir saja.

## Referensi utama bab

[R1] Blitzstein & Hwang. \*Introduction to Probability\*. Probabilitas dasar dan intuisi frekuensi. [R2] Bruce, Bruce, & Gedeck. \*Practical Statistics for Data Scientists\*. Statistik praktis, distribusi, sampling, korelasi. [R3] Deisenroth, Faisal, & Ong. \*Mathematics for Machine Learning\*. Probabilitas, kalkulus, optimisasi. [R4] Goodfellow, Bengio, & Courville. \*Deep Learning\*. Loss, gradient, optimisasi, SGD. [R5] Géron. \*Hands-On Machine Learning\*. Workflow ML praktis, evaluasi, training.

## Catatan validasi internal v0.4

Aspek	Status	Catatan
Kedalaman	Diperluas	Dari ringkasan awal menjadi 32 subbab bernarasi.
Pedagogi	Sesuai playbook	Pengertian → detail → visual → kode/rumus → tes.
Teknis	Lebih lengkap	Menambah aturan probabilitas, conditional probability, Bayes, percentile, sampling error, bias, SGD, linear gradient.
Praktikum	Diperluas	Script dan notebook mencakup probabilitas sampai model linear kecil.
Risiko	Terbuka	Perlu review ahli statistik/matematika sebelum edisi komersial final.