Pengantar Kalkulus, Aritmatika, dan Aljabar

Dari masalah perubahan ke model matematika: fondasi untuk fungsi, limit, turunan, dan integral

Kalkulus · Pertemuan 01

Peta pertemuan

1. Kalkulus memodelkan perubahan dan akumulasi.
2. Aritmatika menjaga operasi dan satuan tetap benar.
3. Aljabar menyatakan hubungan melalui simbol dan persamaan.
4. Fungsi menghubungkan input, output, dan model masalah.

Bagian 1 — Mengapa kalkulus?

Kasus: memantau perjalanan kurir

Kasus: perjalanan kurir

KASUS

Bayangkan sebuah aplikasi logistik melacak lokasi kurirnya setiap menit. Dari data ini, tim operasional ingin mencari tahu: kapan kurir melaju paling kencang, seberapa lama waktu tambahannya jika rute diubah, dan berapa total energi yang dihabiskan dari awal sampai tujuan.

PERTANYAAN PEMICU

- Variabel apa saja yang nilainya **terus berubah** selama di perjalanan?
- Apa bedanya mengukur “kecepatan pas di menit ke-10” dengan “total jarak tempuh setelah 30 menit”?
- Bagaimana cara memodelkan masalah ini ke dalam rumus matematika agar bisa dihitung?

Mengapa Aljabar Biasa Tidak Cukup?

Dalam kondisi ideal (misalnya kecepatan konstan selalu 40 km/jam), kita cukup memakai rumus linier dasar: $\text{Jarak} = \text{Kecepatan} \times \text{Waktu}$. Namun, **realita data selalu bersifat dinamis dan non-linear**.

Tantangan 1: Kecepatan pas di menit ke-10

Kecepatan riil kurir selalu fluktuatif setiap detiknya karena macet, lampu merah, atau belokan. Mengukur kecepatan “pas di menit ke-10” berarti merekam laju perubahan **tepat pada satu timestamp spesifik**, bukan mengambil rata-rata dari menit 9 hingga menit 11.

Dalam **machine learning**, gagasan ini serupa dengan mencari nilai **gradient** tepat pada epoch tertentu.

Mengapa Aljabar Biasa Tidak Cukup? (lanjutan)

Tantangan 2: Total jarak selama 30 menit

Kecepatan kurir naik-turun secara kontinu, sehingga jarak tempuh tidak dapat dihitung dengan sekadar mengalikan satu kecepatan dengan 30 menit. Jarak perlu diakumulasikan dari setiap **frame** atau detik perjalanan yang nilainya terus berubah.

Gagasan ini setara dengan menghitung luas area di bawah kurva (**area under the curve**) pada data **time-series**.

Kesimpulan: Ketika kita berhadapan dengan sistem yang nilainya berubah-ubah secara dinamis dan tidak konstan, aritmatika dasar menjadi tidak akurat. Kita membutuhkan **framework** matematika baru untuk memodelkannya.

Dua ide utama kalkulus

Laju Perubahan Sesaat

Fokus pada satu titik: seberapa drastis posisi benda, suhu server, atau nilai **loss function** berubah **tepat pada detik atau iterasi ini**? Konsep ini melahirkan **Turunan (Diferensial)**.

Akumulasi Total

Fokus pada gambaran besar: berapa jumlah keseluruhan jarak, energi, atau probabilitas yang terakumulasi dari elemen-elemen perubahan yang sangat kecil (**infinitesimal**)? Konsep ini melahirkan **Integral**.

Catatan 1: Kalkulus tetaplah bentuk tingkat lanjut dari aljabar dasar. Semua operasi hitungnya harus valid dan logis.

Catatan 2: Loss function adalah metrik yang menghitung seberapa besar tingkat error/kesalahan prediksi pada model Machine Learning. Kita menggunakan turunan/diferensial untuk mencari arah yang tepat guna memperkecil error tersebut.

Evolusi Paradigma: Aritmatika, Aljabar, dan Kalkulus

Aritmatika (Statis)

Komputasi absolut dengan angka pasti: $7 + 3$, $\frac{12}{4}$, atau 2^3. Ini ibarat instruksi eksekusi berhitung paling dasar (ALU) pada prosesor.

Aljabar (Abstraksi)

Memetakan pola menggunakan variabel: $3x + 5 = 20$. Serupa dengan mendefinisikan sebuah fungsi di kode yang menerima parameter variabel x.

Kalkulus (Dinamis)

Menganalisis sistem yang nilainya berubah secara kontinu (*continuous state*), seperti fluktuasi *gradient* arah atau laju konvergensi data.

Contoh Sinergi: Jika tren pergerakan data dirumuskan sebagai fungsi $s(t)$, **Aljabar** membantu kita menuliskan sintaks rumusnya. Kemudian, **Kalkulus** bertindak sebagai mesin analitik untuk mengekstrak seberapa tajam laju perubahannya (*derivative*) dan berapa total akumulasinya (*integral*).

Bagian 2 — Aritmatika yang dapat diperiksa

Urutan operasi, pecahan, tanda, dan satuan

Hierarki Eksekusi Operasi (Precedence)

Seperti **compiler** pada bahasa pemrograman, matematika mengevaluasi sebuah ekspresi berdasarkan aturan hierarki yang ketat. Jika operasinya setingkat, eksekusi selalu berjalan dari **kiri ke kanan**.

Kurung → Pangkat / Akar → Kali / Bagi → Tambah / Kurang

> **Best Practice:** Selalu gunakan tanda kurung secara eksplisit `()` jika ekspresi berpotensi ambigu. Menuliskan ekspresi langkah demi langkah tidak hanya membuat logika lebih jelas, tapi juga sangat memudahkan proses *debugging* bila terjadi kesalahan perhitungan (**error**).

Contoh hitungan 1 — Pertanyaan

HITUNG DENGAN URUTAN OPERASI

Tentukan nilai ekspresi berikut tanpa memakai kalkulator terlebih dahulu.

$$3 + 2\frac{(5-1)^2}{4}$$

Tandai operasi mana yang harus dikerjakan paling awal, lalu tulis setiap langkah pada baris terpisah.

Contoh Eksekusi Operasi 1 — Jawaban

$$3 + 2\frac{(5-1)^2}{4}$$

$$= 3 + 2\frac{(4)^2}{4} \quad \rightarrow \quad \textit{Evaluasi dalam kurung: } (5-1)$$

$$= 3 + 2\frac{16}{4} \quad \rightarrow \quad \textit{Eksekusi eksponen/pangkat: } 4^2$$

$$= 3 + \frac{32}{4} \quad \rightarrow \quad \textit{Eksekusi perkalian kiri-ke-kanan: } 2(16)$$

$$= 11 \quad \rightarrow \quad \textit{Eksekusi pembagian } \frac{32}{4} = 8\textit{, lalu tambah } 3$$

Trace Log: Eksekusi prioritas selalu mendahulukan pangkat dibandingkan perkalian/pembagian. Setelah pangkat selesai, instruksi $2\frac{16}{4}$ diproses secara sekuensial (berurutan) dari kiri ke kanan.

Fundamental: Pecahan & Aturan Tanda (Polaritas)

Dua hal dasar ini adalah sumber *bug* logika matematis yang paling sering terjadi saat memformulasikan kode atau algoritma.

1. Manipulasi Pecahan

Penjumlahan variabel pecahan menuntut penyamaan penyebut terlebih dahulu (*common denominator*).

$$\frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$$

2. Aturan Tanda / Polaritas

$$(-a) \times (-b) = ab$$

$$(-a) \times b = -(ab)$$

Catatan Kritis: Di ranah Machine Learning, kesalahan sepele pada tanda minus ($-$) saat menghitung turunan (gradient) sangat fatal. Arah pencarian solusi yang seharusnya turun (minimasi) bisa terbalik menjadi naik dan membuat nilai error meledak.

Validasi Hasil (**Sanity Check** / Uji Kewajaran)

Jangan langsung menerima hasil *output* komputasi secara mentah-mentah tanpa validasi logika dunia nyata. Selalu cek kewajaran angka (orde besaran) dan kesesuaian satuannya.

Contoh Kasus Validasi

Sebuah kendaraan menempuh **jarak 120 km** dalam waktu **2,5 jam**.

$$v = \frac{s}{t} = \frac{120}{2.5} = 48 \textbf{ km/jam}$$

Hasil Evaluasi: Masuk akal secara logika fisika.

Jika sistem Anda mencetak 48.000 *km/jam karena ada kesalahan konversi satuan meter ke kilometer di dalam kode, pengecekan "Sanity Check" ini akan langsung menemukan* bug *tersebut sebelum kode dirilis ke produksi.*

Studi Kasus 2: Kecepatan Transfer Data (Pertanyaan)

Sebuah server berhasil memindahkan *file* sebesar **120 Gigabyte (GB)** dalam waktu **2,5 jam**. Tentukan *bandwidth* atau kecepatan transfer rata-ratanya beserta satuannya.

$$\text{Kecepatan Transfer} = \frac{\text{Total Ukuran Data}}{\text{Waktu Tempuh}}$$

> **Logika Estimasi Kasar:** Sebelum masuk ke kalkulator atau kodingan, mari gunakan intuisi analitik: Apakah *output* kecepatannya nanti akan berada di atas atau di bawah 60 GB/jam?

Studi Kasus 2: Kecepatan Transfer Data (Jawaban)

$$\text{Kecepatan} = \frac{120}{2.5} \ \text{GB/jam}$$

$$= 48 \ \text{GB/jam}$$

Mengapa Estimasi (Sanity Check) Itu Penting?

Secara nalar, jika waktunya pas 2 jam, maka kecepatannya adalah $\frac{120}{2} = 60$ GB/jam.

Karena waktu nyatanya (2,5 jam) lebih lama dari 2 jam, maka logikanya kecepatan **pasti lebih lambat (lebih kecil)** dari 60 GB/jam. Hasil hitungan 48 GB/jam memenuhi logika tersebut, membuktikan kalkulasi kita aman dari kesalahan besar.

Bagian 3 — Bahasa Aljabar

Menerjemahkan Logika ke dalam Variabel, Ekspresi, dan Persamaan

Anatomi Aljabar 1: Ekspresi (*Expression*)

Ekspresi adalah susunan matematis yang membentuk suatu formula, namun belum menetapkan kondisi akhir apa pun.

$$3x + 5$$

Komponen Utama:

- x disebut **variabel** (wadah penyimpan nilai dinamis).
- 3 disebut **koefisien** (pengali penguat yang melekat pada variabel).
- 5 disebut **konstanta** (nilai statis yang absolut/tetap).

> **Kacamata Programming:** Ekspresi ibarat mendefinisikan sebuah algoritma tanpa menjalankannya (*un-evaluated function*). Ekspresi ini tidak bernilai "benar" atau "salah" sampai kita menyuapkan (*assign*) nilai spesifik ke dalam variabel x.

Anatomi Aljabar 2: Persamaan (*Equation*)

Kehadiran tanda sama dengan ($=$) mengubah sebuah ekspresi yang tadinya terbuka menjadi sebuah persamaan yang mengikat.

$$3x + 5 = 20$$

Persamaan secara tegas menyatakan bahwa nilai di ruas kiri dan ruas kanan adalah ekuivalen. Ini mengubah status matematisnya dari sekadar “rumus” menjadi “kasus yang bisa diselesaikan”.

> **Kacamata Programming:** Persamaan layaknya sebuah fungsi *solver* atau optimasi. Kita sedang menugaskan sistem untuk mencari: *“Nilai x berapakah yang akan membuat komputasi ini seimbang dan bernilai TRUE?”*

Menyelesaikan Persamaan (Mengekstrak Nilai **Unknown**)

$$3x + 5 = 20$$

Tujuan Utama: Mengisolasi variabel x (membuatnya berdiri sendiri). Kita harus menyingkirkan koefisien dan konstanta lain yang berada pada ruas yang sama dengan x.

> **Prinsip Konsistensi (Keseimbangan Ruas):** Persamaan ibarat timbangan memori yang seimbang. Operasi apa pun (tambah/kurang/kali/bagi) yang dieksekusi pada ruas kiri, **wajib** dieksekusi secara identik pada ruas kanan agar tidak merusak validitas logika.

Studi Kasus 3: Pencarian Nilai Variabel (Pertanyaan)

RESOLUSI PERSAMAAN LINEAR

Selesaikan persamaan $3x + 5 = 20$. Petakan setiap operasi matematika yang Anda eksekusi pada kedua ruas langkah demi langkah (buat *trace log*).

$$3x + 5 = 20$$

Praktik Validasi: Setelah mengekstrak nilai x, kita harus melakukan *back-substitution* (memasukkan kembali nilai tersebut ke dalam persamaan awal) untuk menguji kebenarannya.

Studi Kasus 3: Pencarian Nilai Variabel (Jawaban)

$$3x + 5 = 20 \quad \rightarrow \quad \textit{State awal persamaan}$$

$$3x + 5 - 5 = 20 - 5 \quad \rightarrow \quad \textit{Kurangi kedua ruas dengan 5 untuk mengeliminasi konstanta}$$

$$3x = 15 \quad \rightarrow \quad \textit{Hasil penyederhanaan konstanta}$$

$$\frac{3x}{3} = \frac{15}{3} \quad \rightarrow \quad \textit{Bagi kedua ruas dengan 3 untuk menetralisir koefisien } x$$

$$x = 5 \quad \rightarrow \quad \textit{Variabel } x \textit{ berhasil diisolasi (Solusi akhir ditemukan)}$$

Unit Testing (Verifikasi): Masukkan $x = 5$ ke rumus awal $\rightarrow 3(5) + 5 = 15 + 5 = 20$.
Karena ruas kiri bernilai 20 dan ruas kanan bernilai 20, maka eksekusi pencarian nilai kita dipastikan valid dan bebas *bug*.

Latihan Mandiri: *Unit Testing* Persamaan Linear

Uji pemahaman analitik Anda dengan mengekstrak nilai variabel (*unknowns*) dari 10 arsitektur persamaan berikut, mulai dari operasi dasar hingga manipulasi tipe data pecahan (*float*).

Level 1: Basic Isolation

1. $2x - 4 = 10$
2. $5y + 7 = -8$

Level 2: Multi-Node (Variabel di Kedua Ruas)

3. $4a - 3 = 2a + 9$
4. $7m + 12 = 3m - 4$

Level 3: Distribusi (Parameter Expansion)

5. $3(p - 2) = 15$

Level 3 (Lanjutan)

6. $2(4k + 1) - 3 = 11$

Level 4: Tipe Data Float/Pecahan

7. $\frac{x}{4} + 3 = 8$
8. $\frac{2n-5}{3} = 5$

Level 5: Kompleksitas Gabungan

9. $5(2w - 1) + 3 = 3(w + 4) - 1$
10. $0.5z + 2.5 = 0.2z - 0.5$

Langkah Penyelesaian (Soal 1 - 4)

1. $2x - 4 = 10$

$\rightarrow 2x = 10 + 4$

$\rightarrow 2x = 14 \quad \rightarrow \quad \boldsymbol{x = 7}$

2. $5y + 7 = -8$

$\rightarrow 5y = -8 - 7$

$\rightarrow 5y = -15 \quad \rightarrow \quad \boldsymbol{y = -3}$

3. $4a - 3 = 2a + 9$ *(Pindahkan variabel ke kiri, konstanta ke kanan)*

$\rightarrow 4a - 2a = 9 + 3$

$\rightarrow 2a = 12 \quad \rightarrow \quad \boldsymbol{a = 6}$

4. $7m + 12 = 3m - 4$

$\rightarrow 7m - 3m = -4 - 12$

$\rightarrow 4m = -16 \quad \rightarrow \quad \boldsymbol{m = -4}$

Langkah Penyelesaian (Soal 5 - 8)

5. $3(p-2)=15$ *(Bagi kedua ruas dengan 3 lebih dulu agar efisien)*

$\rightarrow p-2=5$

$\rightarrow p=5+2 \quad \rightarrow \quad \boldsymbol{p=7}$

6. $2(4k+1)-3=11$ *(Ekspansi kurung terlebih dahulu)*

$\rightarrow 8k+2-3=11$

$\rightarrow 8k-1=11$

$\rightarrow 8k=12 \quad \rightarrow \quad k=\frac{12}{8} \quad \rightarrow \quad \boldsymbol{k=1.5}$

7. $\frac{x}{4}+3=8$ *(Isolasi pecahan)*

$\rightarrow \frac{x}{4}=8-3$

$\rightarrow \frac{x}{4}=5 \quad \rightarrow \quad \boldsymbol{x=20}$

8. $\frac{2n-5}{3}=5$ *(Kalikan kedua ruas dengan 3 untuk menetralkan penyebut)*

$\rightarrow 2n-5=15$

$\rightarrow 2n=20 \quad \rightarrow \quad \boldsymbol{n=10}$

Langkah Penyelesaian (Soal 9 - 10)

9. $5(2w-1)+3=3(w+4)-1$ *(Ekspansi dan sederhanakan kedua sisi)*

$\rightarrow 10w-5+3=3w+12-1$

$\rightarrow 10w-2=3w+11$

$\rightarrow 10w-3w=11+2$

$\rightarrow 7w=13 \quad \rightarrow \quad \boldsymbol{w=\frac{13}{7}}$

10. $0.5z+2.5=0.2z-0.5$ *(Gabungkan koefisien desimal)*

$\rightarrow 0.5z-0.2z=-0.5-2.5$

$\rightarrow 0.3z=-3.0$

$\rightarrow z=-\frac{3.0}{0.3} \quad \rightarrow \quad \boldsymbol{z=-10}$

Aturan Aljabar 1: Anatomi Suku Sejenis (*Like Terms*)

Definisi Fundamental: Suku sejenis (*like terms*) adalah komponen-komponen aljabar yang memiliki **variabel yang persis sama** dan **pangkat (derajat) yang persis sama**.

Aturan Komputasi: Dalam pemrograman, kita akan terkena *Type Error* jika mencoba menjumlahkan tipe data *String* dengan *Integer*. Begitu pula di aljabar: operasi penjumlahan ($+$) dan pengurangan ($-$) **hanya valid dieksekusi antar suku yang sejenis**.

✅ **Valid (Suku Sejenis)**

- $4x$ dan $2x$ *(sama-sama x^1)*
- $5y^2$ dan $-3y^2$ *(sama-sama y^2)*
- $7ab$ dan $2ab$ *(kombinasi variabel identik)*

❌ **Invalid (Bukan Suku Sejenis)**

- $4x$ dan $4y$ *(variabel berbeda)*
- $3x$ dan $3x^2$ *(pangkat berbeda)*
- $5a$ dan $5ab$ *(kombinasi tidak identik)*

Peringatan Eksekusi: Beda Pangkat = Beda Entitas

Kesalahan pemula yang paling sering terjadi adalah memaksa menggabungkan variabel dengan pangkat yang berbeda.

Ilustrasi Kesalahan Fatal

$$x + x^2 = 2x^2 \quad \rightarrow \quad \textbf{(SALAH!)}$$

Konteks Machine Learning: Variabel x (linier) dan x^2 (kuadratik) merepresentasikan ruang dimensi yang sama sekali berbeda. Fitur x mungkin merepresentasikan “jarak tempuh”, sementara x^2 merepresentasikan “percepatan”. Menjumlahkan keduanya secara paksa akan merusak arsitektur algoritma Anda. Biarkan mereka tetap terpisah sebagai $x + x^2$.

Studi Kasus Lanjutan: *Pipeline* Gabungan (Pertanyaan)

Dalam pemrograman komputasi nyata, Anda harus mengeksekusi operasi ekspansi distributif (*broadcasting*) terlebih dahulu untuk membongkar kurung, sebelum Anda bisa memanggil fungsi agregasi (penggabungan suku sejenis).

Sederhanakan 6 arsitektur persamaan berikut ini:

Level 1: Dua Kurung Dasar

1. $2(3x - 4) + 3(x + 2)$
2. $5(y + 1) - 2(y - 3)$

Level 2: Kurung dan Suku Lepas

3. $4(2m - 1) + 5m - 2$

Level 3: Distribusi Variabel

4. $a(2a + 3) - a^2$
5. $2p(p - 4) + 3p^2 + 5p$

Level 4: Jebakan Minus Berganda

6. $-3(2k - 5) - 4(-k + 2)$

Langkah Penyelesaian (Kasus 1 - 3)

Kasus 1: $\mathbf{2(3x-4)+3(x+2)}$

$\rightarrow (6x-8)+(3x+6)$ $\rightarrow$ *Ekspansi distributif*

$\rightarrow 6x+3x-8+6$

$\rightarrow \mathbf{9x-2}$ $\rightarrow$ *Agregasi suku sejenis*

Kasus 2: $\mathbf{5(y+1)-2(y-3)}$

$\rightarrow (5y+5)-(2y-6)$ $\rightarrow$ *Awas:* -2×-3 *menjadi* $+6$

$\rightarrow 5y+5-2y+6$

$\rightarrow (5y-2y)+(5+6)$

$\rightarrow \mathbf{3y+11}$

Kasus 3: $\mathbf{4(2m-1)+5m-2}$

$\rightarrow (8m-4)+5m-2$ $\rightarrow$ *Hanya operasikan distributif pada kurung*

$\rightarrow 8m+5m-4-2$

$\rightarrow \mathbf{13m-6}$

Langkah Penyelesaian (Kasus 4 - 6)

Kasus 4: $a(2a+3)-a^2$

$\rightarrow (2a^2+3a)-a^2 \quad \rightarrow$ *Ingat:* $a \times 2a = 2a^2$

$\rightarrow (2a^2-1a^2)+3a$

$\rightarrow \mathbf{a^2+3a}$

Kasus 5: $2p(p-4)+3p^2+5p$

$\rightarrow (2p^2-8p)+3p^2+5p$

$\rightarrow (2p^2+3p^2)+(-8p+5p) \quad \rightarrow$ *Kelompokkan pangkat 2 dengan pangkat 2*

$\rightarrow \mathbf{5p^2-3p}$

Kasus 6: $-3(2k-5)-4(-k+2)$

$\rightarrow (-6k+15)-(-4k+8) \quad \rightarrow$ *Bongkar kurung dengan skalar minus*

$\rightarrow -6k+15+4k-8 \quad \rightarrow$ *Minus bertemu minus menjadi plus* $(+4k)$

$\rightarrow (-6k+4k)+(15-8)$

$\rightarrow \mathbf{-2k+7}$

Aturan Aljabar 2: Ekspansi Distributif

Hukum distributif adalah mekanisme untuk mendistribusikan sebuah pengali (di luar kurung) ke setiap elemen di dalam kurung. Operasi ini wajib dilakukan untuk “membongkar” kurung agar persamaannya bisa disederhanakan.

$$a(b + c) = ab + ac$$

> **Kacamata Programming (Broadcasting):** Konsep ini sangat identik dengan *scalar multiplication* atau *broadcasting* pada *Array/Tensor* (seperti di NumPy / PyTorch). Jika kita mengalikan sebuah matriks/array $[b, c]$ dengan skalar a, maka sistem akan mengalikan bobot a tersebut ke setiap elemen individu di dalam matriks menjadi $[ab, ac]$.

Studi Kasus Ekspansi Distributif

Kasus 1: Pengali Positif (Skalar)

$3(2x + 5)$

$\rightarrow (3 \times 2x) + (3 \times 5)$ $\rightarrow$ *Kalikan 3 ke komponen pertama dan kedua*

$\rightarrow \mathbf{6x + 15}$

Kasus 2: Awas Jebakan Tanda Minus (−)!

$-2(4y - 3)$

$\rightarrow (-2 \times 4y) + (-2 \times -3)$ $\rightarrow$ *Minus dikali minus menjadi plus*

$\rightarrow \mathbf{-8y + 6}$

Kasus 3: Pengali Berupa Variabel

$x(3x - 2)$

$\rightarrow (x \times 3x) + (x \times -2)$ $\rightarrow$ *Ingat eksponen dasar:* $x \times x = x^2$

$\rightarrow \mathbf{3x^2 - 2x}$

Studi Kasus *Pipeline* Gabungan (Distributif + Suku Sejenis)

Dalam pemrograman dunia nyata, Anda harus melakukan komputasi ekspansi (*broadcasting*) terlebih dahulu, baru memanggil fungsi agregasi (penggabungan data).

Sederhanakan arsitektur persamaan berikut:

$$2(3x - 4) - 3(x + 2)$$

Langkah 1: Eksekusi Distributif (Bongkar kurung)

$\rightarrow (2 \times 3x) + (2 \times -4) - (3 \times x) - (3 \times 2)$

$\rightarrow 6x - 8 - 3x - 6$

Langkah 2: Eksekusi Agregasi (Gabungkan suku sejenis)

$\rightarrow (6x - 3x) + (-8 - 6)$

$\rightarrow \mathbf{3x - 14}$

Konsep Faktorisasi (*Reverse Engineering*)

Jika ekspansi distributif adalah proses “membongkar” kurung, maka **Faktorisasi** adalah proses *reverse engineering*. Kita merangkum ulang sebuah polinomial panjang kembali menjadi bentuk perkalian elemen-elemen dasarnya (*factors*).

Aturan 1: Ekstraksi Faktor Umum (*Greatest Common Factor*)

Langkah pertama (dan paling utama) yang wajib dicek pada setiap persamaan: apakah setiap suku memiliki “elemen data” yang sama (angka atau variabel) yang bisa ditarik keluar?

$$ab + ac = a(b + c)$$

Contoh Penerapan:

$3x^2 + 6x$

$\rightarrow$ *Analisis: Kedua suku memiliki variabel x dan bisa dibagi 3*

$\rightarrow$ $\mathbf{3x(x + 2)}$ *Ekstrak $3x$ keluar sistem*

Pola Faktorisasi 2: Selisih Dua Kuadrat

Pola ini terjadi secara eksklusif ketika sebuah nilai kuadrat dikurangi dengan nilai kuadrat lainnya ($a^2 - b^2$). Hasil faktorisasinya **selalu** menghasilkan pasangan kurung yang identik secara angka, namun berlawanan tanda (polaritas).

$$a^2 - b^2 = (a - b)(a + b)$$

Contoh Uji Pola:

Kasus 1: Konstanta Sederhana

$x^2 - 16 \quad \rightarrow \quad$ *Karena* 16 *adalah* 4^2

$\rightarrow (x - 4)(x + 4)$

Kasus 2: Kuadrat Kompleks

$9y^2 - 25 \quad \rightarrow \quad$ $9y^2$ *adalah* $(3y)^2$*, dan* 25 *adalah* 5^2

$\rightarrow (3y - 5)(3y + 5)$

Pola Faktorisasi 3: Kuadrat Sempurna

Pola ini terbentuk dari duplikasi kurung yang identik. Ciri khasnya mudah dideteksi: **suku pertama dan terakhir adalah bilangan kuadrat**, dan **suku tengahnya** adalah hasil kali $2 \times$ akar suku pertama $\times$ akar suku terakhir.

$$a^2 + 2ab + b^2 = (a + b)^2$$

$$a^2 - 2ab + b^2 = (a - b)^2$$

Contoh Uji Pola:

Ubah $x^2 + 10x + 25$ menjadi faktor dasarnya:

1. Akar suku awal $(x^2) \to x$
2. Akar suku akhir $(25) \to 5$
3. Cek suku tengah: $2(x)(5) = 10x$ *(Match! Terkonfirmasi kuadrat sempurna)*

$\to \boldsymbol{(x + 5)^2}$

Pola Faktorisasi 4: Trinomial Umum (Pencarian Logika)

Jika persamaan tidak memiliki faktor umum dan bukan kuadrat sempurna, kita akan menggunakan **Algoritma Pencarian** dengan dua syarat (*constraints*) untuk memecahkan bentuk standar $x^2 + bx + c$.

> **Constraint Satisfaction (Pencarian 2 Angka):**
> Cari dua angka (misal p dan q) yang secara bersamaan memenuhi:
> 1. Syarat Perkalian: $p \times q = c$ (Konstanta akhir)
> 2. Syarat Penjumlahan: $p + q = b$ (Koefisien tengah)

Studi Kasus: Faktorkan $x^2 + 7x + 10$

- *Filter 1 (Perkalian):* Faktor pembentuk 10 adalah $(1, 10)$ atau $(2, 5)$.
- *Filter 2 (Penjumlahan):* Cek mana yang jika dijumlah menghasilkan 7. $1 + 10 = 11$ *(Gagal)*, $2 + 5 = 7$ *(Lolos!)*.
- *Hasil Akhir:* $\boldsymbol{(x + 2)(x + 5)}$

Studi Kasus Terpadu: *Pattern Matching* Faktorisasi (Pertanyaan)

Uji kemampuan analitik Anda. Gunakan algoritma pengenalan pola (*pattern matching*) yang tepat untuk memfaktorkan 5 arsitektur persamaan berikut kembali ke bentuk perkalian dasarnya.

Level 1: Deteksi Pola Dasar

1. $5x^2 - 20x$
2. $x^2 - 64$
3. $x^2 - 14x + 49$

Level 2: Pencarian Constraint (Trinomial)

4. $x^2 - 4x - 12$

Level 3: Multi-Pass Pipeline

5. $3x^2 + 15x + 18$

> **Petunjuk:** Tidak semua persamaan diselesaikan dengan cara yang sama. Identifikasi dulu “tipe data/pola” dari persamaannya (apakah ada faktor umum? apakah ini selisih kuadrat?) sebelum mengeksekusi perhitungan.

Langkah Penyelesaian Faktorisasi (Kasus 1 - 3)

Kasus 1: $\mathbf{5x^2 - 20x}$

$\rightarrow$ *Deteksi Pola:* Ekstraksi Faktor Umum. Kedua suku bisa dibagi $5x$.

$\rightarrow$ $\mathbf{5x(x - 4)}$

Kasus 2: $\mathbf{x^2 - 64}$

$\rightarrow$ *Deteksi Pola:* Selisih Dua Kuadrat. (Bentuk $a^2 - b^2$, dengan $a = x$ dan $b = 8$)

$\rightarrow$ $(x - 8)(x + 8)$ $\rightarrow$ *Pola plus/minus selalu identik*

$\rightarrow$ $\mathbf{(x - 8)(x + 8)}$

Kasus 3: $\mathbf{x^2 - 14x + 49}$

$\rightarrow$ *Deteksi Pola:* Kuadrat Sempurna. (Ujungnya kuadrat: x^2 dan 7^2)

$\rightarrow$ Cek suku tengah: $2 \times (x) \times (-7) = -14x$ $\rightarrow$ *(Match!)*

$\rightarrow$ $\mathbf{(x - 7)^2}$

Langkah Penyelesaian Faktorisasi (Kasus 4 - 5)

Kasus 4: $x^2 - 4x - 12$

→ *Deteksi Pola:* Trinomial Umum (Pencarian 2 angka).

→ *Constraint:* Dikalikan hasil $= -12$, Dijumlah hasil $= -4$.

→ *Uji Kandidat:*

a. (-3) dan $4 \rightarrow$ Jumlah: 1 *(Salah)*

b. $\mathbf{2}$ **dan** $(-6) \rightarrow$ Jumlah: -4 *(Benar! Match dengan koefisien tengah)*

→ $\mathbf{(x - 6)(x + 2)}$

Kasus 5: $3x^2 + 15x + 18$

→ *Deteksi Pola:* Multi-Pass (Ada faktor umum, lalu ada trinomial).

→ *Pass 1 (Ekstrak 3):* $3(x^2 + 5x + 6)$

→ *Pass 2 (Faktorkan Trinomial):* Cari angka dikali 6, dijumlah $5 \rightarrow$ Kandidat: 2 dan 3.

→ $\mathbf{3(x + 2)(x + 3)}$

Bagian 4 — Pangkat, Domain, dan Fungsi

Membangun Model Matematis dan Menjaga Integritas Batasan Input

Aturan Eksponen (Pangkat): Manipulasi Basis yang Sama

Eksponen atau pangkat adalah bentuk komputasi berulang. Saat memanipulasi variabel dengan **basis yang sama** (misalnya sama-sama x), kita bisa menyederhanakannya menggunakan aturan berikut:

1. Perkalian

$$x^m \times x^n = x^{m+n}$$

Pangkat ditambahkan

2. Pembagian

$$\frac{x^m}{x^n} = x^{m-n}$$

Pangkat dikurangkan

3. Pangkat Bertingkat

$$\left(x^m\right)^n = x^{m \times n}$$

Pangkat dikalikan

Contoh:

- Perkalian: $x^3 \times x^4 = x^{3+4} = x^7$
- Pembagian: $\frac{y^6}{y^2} = y^{6-2} = y^4$
- Bertingkat: $\left(a^2\right)^3 = a^{2 \times 3} = a^6$

Kasus Khusus Eksponen: Pangkat Negatif & Akar Kuadrat

Pangkat Negatif (Invers)

$$x^{-2} = \frac{1}{x^2}$$

Aturan: Pangkat negatif **bukan** berarti nilainya menjadi minus, melainkan berbalik menjadi pecahan (invers).
Syarat mutlak: $x \neq 0$ (karena pembagian dengan nol tidak diizinkan).

Akar Kuadrat (Nilai Mutlak)

$$\sqrt{(-3)^2} = \sqrt{9} = 3$$

Aturan: Hasil dari akar kuadrat utama (simbol $\sqrt{x}$) **selalu non-negatif**. Oleh karena itu, di dalam aljabar berlaku aturan nilai mutlak (`abs()`):
$\sqrt{x^2} = |x|$.

Domain Fungsi: Mencegah *ZeroDivisionError*

Dalam pemrograman, jika Anda membagi sebuah angka dengan nol, program akan **crash** dan mengeluarkan pesan `ZeroDivisionError`. Di matematika, kita mencegah hal ini dengan mendefinisikan **Domain** (batasan input yang valid).

Studi Kasus Pembatasan Domain

$$f(x) = \frac{1}{x-2}$$

Analisis Kerentanan:

Jika kita memasukkan input $x = 2$, maka penyebut akan menjadi $2 - 2 = 0$. Hal ini menyebabkan komputasi menjadi **tidak terdefinisi (undefined)**.

Kesimpulan Domain:
Domain dari fungsi di atas adalah **seluruh bilangan riil, KECUALI** $x = 2$. Mendeklarasikan domain sedini mungkin ibarat melakukan *input validation* sebelum data diproses oleh fungsi.

Fungsi Matematis: Arsitektur *Input-Output*

Fungsi matematika pada dasarnya sama persis dengan fungsi (`function` / `method`) di dalam kode pemrograman. Ia menerima parameter input, memprosesnya dengan aturan tertentu, dan mengembalikan (*return*) sebuah nilai output.

MODEL PREDIKSI BIAYA LOGISTIK

Biaya operasional sebuah perjalanan terdiri dari biaya awal tetap Rp7.000, ditambah tarif jarak Rp2.500 per kilometer. Jika d adalah jarak dalam kilometer, maka **model fungsi biayanya** adalah:

$$C(d) = 7000 + 2500d$$

Parameter *Input*

$$d = 4$$

(Memanggil fungsi untuk jarak 4 km)

Nilai *Output* (*Return*)

$$C(4) = 17.000$$

(Karena $7000 + 2500(4) = 17.000$*)*

Latihan Mandiri: Uji Pemahaman Komprehensif (Pertanyaan)

Uji penguasaan Anda terhadap konsep hierarki operasi, aljabar dasar, faktorisasi, dan arsitektur fungsi.

1. Hierarki Eksekusi: Hitung nilai dari ekspresi $2 + 3(4 - 2)^2$.

2. Ekstraksi Variabel: Selesaikan persamaan $5x - 4 = 21$ dan lakukan verifikasi.

3. Faktorisasi: Faktorkan bentuk kuadratik $x^2 - 4x - 12$.

4. Validasi Domain: Tentukan nilai input yang **terlarang** (menyebabkan error) pada arsitektur fungsi $g(x) = \frac{x+1}{x-5}$.

5. Ekstraksi Konteks: Pada pemodelan suhu mesin $T(t) = 24 + 3t$, apa representasi fisik dari komponen 24, 3, dan variabel t?

Langkah Penyelesaian (Kasus 1 - 2)

Kasus 1: $\mathbf{2 + 3(4 - 2)^2}$

Langkah:

$\rightarrow 2 + 3(2)^2 \quad \rightarrow$ *Eksekusi dalam kurung*

$\rightarrow 2 + 3(4) \quad \rightarrow$ *Eksekusi pangkat*

$\rightarrow 2 + 12 \quad \rightarrow$ *Eksekusi perkalian*

$\rightarrow \mathbf{14}$

Kasus 2: Selesaikan $\mathbf{5x - 4 = 21}$

Langkah:

$\rightarrow 5x = 21 + 4$

$\rightarrow 5x = 25$

$\rightarrow \boldsymbol{x = 5}$

Verifikasi: $5(5) - 4 = 25 - 4 = 21$ *(Valid!)*

Langkah Penyelesaian (Kasus 3 - 4)

Kasus 3: Faktorkan $x^2 - 4x - 12$

Langkah:

$\rightarrow$ Cari 2 angka: dikali hasil -12, dijumlah hasil -4.

$\rightarrow$ Angka yang tepat adalah -6 dan 2.

$\rightarrow$ $\mathbf{(x - 6)(x + 2)}$

Kasus 4: Domain $g(x) = \frac{x+1}{x-5}$

Langkah:

$\rightarrow$ Fungsi pecahan akan **crash** (tidak terdefinisi) jika penyebutnya bernilai nol.

$\rightarrow$ Atur penyebut sama dengan nol: $x - 5 = 0$

$\rightarrow$ $x = 5$

$\rightarrow$ **Kesimpulan: Nilai $x = 5$ dilarang (berada di luar domain).**

Langkah Penyelesaian (Kasus 5)

Kasus 5: Analisis Model $T(t) = 24 + 3t$

Langkah:

Berdasarkan struktur persamaan linear ($y = mx + c$):

$\rightarrow$ $\mathbf{24}$ adalah konstanta awal (*base state*), merepresentasikan suhu mula-mula mesin sebelum waktu berjalan.

$\rightarrow$ $\mathbf{3}$ adalah koefisien laju perubahan (*gradient*), merepresentasikan kenaikan suhu mesin per 1 satuan waktu (misal: naik 3 derajat per menit).

$\rightarrow$ t adalah variabel *input* independen, merepresentasikan waktu yang terus berjalan.

Peta Jalan (*Roadmap*) Perkuliahan Berikutnya

Bab 2 **Arsitektur Bilangan (Riil & Kompleks):**
Tipe data matematis untuk berbagai operasi komputasi dan pemodelan.

Bab 3 **Fungsi & Relasi:**
Mendalami aturan *input-output* serta representasi visual (grafik/kurva).

Bab 4+ **Core Kalkulus (Limit, Turunan, & Integral):**
Analisis pergerakan kontinu, laju perubahan sesaat, dan akumulasi data.

> **Empat Pilar Praktik Baik (Best Practices):**
> Biasakan diri Anda untuk: (1) Menulis model dengan jelas, (2) Mematuhi hierarki urutan operasi, (3) Mendeklarasikan domain & satuan (*sanity check*), dan (4) Selalu memverifikasi hasil eksekusi (uji substitusi).