

# Spesifikasi Tugas Besar 1

IF2123 ALJABAR LINIER DAN GEOMETRI

SISTEM PERSAMAAN LINIER, DETERMINAN, DAN PENERAPANNYA

SEMESTER I TAHUN 2025/2026

v1.0

DEADLINE: RABU, 2025/10/8 21:23 WIB



LABORATORIUM ILMU DAN REKAYASA KOMPUTASI

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG



Dokumen spesifikasi tugas besar ini disusun oleh asisten Laboratorium Ilmu dan Rekayasa Komputasi dari mahasiswa Teknik Informatika angkatan 2023. Untuk berjaga-jaga dengan adanya revisi, salinan terbaru dari dokumen ini selalu dapat diakses di [sini](#).

## 0 Daftar Revisi

- v1.0 — [2025/09/17 13.00] — Versi rilis awal.

## Daftar Isi

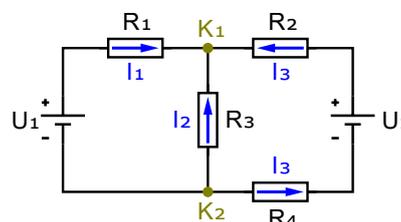
|          |   |           |
|----------|---|-----------|
| <b>0</b> | <b>Daftar Revisi</b>  | <b>1</b>  |
| <b>1</b> | <b>Pendahuluan</b>  | <b>2</b>  |
| <b>2</b> | <b>Dasar Teori</b>  | <b>3</b>  |
| 2.1      | Sistem Persamaan Linear   | 3         |
| 2.2      | Determinan Matriks  | 3         |
| 2.3      | Invers Matriks  | 3         |
| 2.4      | Interpolasi Polinomial  | 3         |
| 2.5      | Interpolasi Splina Bézier Kubik   | 4         |
| 2.6      | Regresi Polinomial Berganda ( <i>Multivariate Polynomial Regression</i> ) | 5         |
| <b>3</b> | <b>Spesifikasi</b>  | <b>7</b>  |
| 3.1      | Spesifikasi Wajib Program   | 7         |
| 3.2      | Spesifikasi Bonus   | 12        |
| 3.3      | Prosedur Pengerjaan   | 14        |
| 3.4      | Spesifikasi Laporan   | 14        |
| 3.5      | Kriteria Penilaian  | 15        |
| <b>4</b> | <b>Pengumpulan</b>  | <b>16</b> |
| 4.1      | <i>Deliverables</i>   | 16        |
| 4.2      | Teknis Pengumpulan  | 16        |
| <b>5</b> | <b>Studi dan Kasus Uji</b>  | <b>17</b> |
| 5.1      | Determinan Matriks  | 17        |
| 5.2      | Invers Matriks  | 17        |
| 5.3      | Sistem Persamaan Linear $Ax = b$  | 17        |
| 5.4      | Sistem Persamaan Linear Berbentuk Matriks <i>Augmented</i>                | 18        |
| 5.5      | Sistem Persamaan Linier Berbentuk Umum                                    | 19        |
| 5.6      | Aplikasi Sistem Persamaan Linier  | 19        |
| 5.7      | Studi Kasus Interpolasi Polinomial  | 20        |
| 5.8      | Studi Kasus Interpolasi Splina Bézier Kubik                               | 21        |
| 5.9      | Studi Kasus Regresi Polinomial Berganda                                   | 21        |
| <b>6</b> | <b>Referensi</b>  | <b>22</b> |
| 6.1      | SPL, Determinan, dan Matriks  | 22        |
| 6.2      | Interpolasi Polinomial  | 22        |
| 6.3      | Interpolasi Splina Bézier Kubik   | 22        |
| 6.4      | Regresi Polinomial Berganda   | 22        |
| <b>7</b> | <b>Surat Cinta dari Asisten</b>   | <b>23</b> |

## 1 Pendahuluan

Aplikasi seperti Photomath telah menjadi solusi populer dalam membantu menyelesaikan berbagai permasalahan matematika, termasuk soal matriks dan sistem persamaan linier yang kompleks. Kemampuan aplikasi tersebut dalam memberikan langkah-langkah penyelesaian secara otomatis tidak terlepas dari penerapan algoritma dan konsep matematika yang kuat, khususnya aljabar linier.

Aljabar linier merupakan salah satu cabang matematika yang sangat penting dalam pengembangan ilmu komputer, teknik, fisika, ekonomi, dan berbagai bidang lainnya. Dalam kehidupan sehari-hari, konsep matriks, dan sistem persamaan linier sering kali diaplikasikan untuk menyelesaikan berbagai persoalan nyata, mulai dari pemodelan data, analisis statistik, hingga simulasi sistem dinamis.

Pada tugas besar ini, Anda akan mengimplementasikan berbagai metode penyelesaian sistem persamaan linier, perhitungan determinan, pencarian invers matriks, interpolasi polinomial, interpolasi kurva Bézier kubik, dan regresi polinomial menggunakan bahasa pemrograman Java dalam bentuk pustaka (*library*) yang dapat digunakan secara modular dan terdokumentasi dengan baik.



Gambar 1: Aplikasi Photomath (kiri) dan diagram rangkaian listrik (kanan) . (sumber 1 2)

Penyelesaian sistem persamaan linier, misalnya, sangat krusial dalam pemodelan sirkuit listrik, analisis struktur bangunan, hingga pemrosesan citra digital. Sementara itu, interpolasi dan regresi polinomial digunakan untuk memperkirakan nilai di antara data yang tersedia atau memprediksi tren data di masa depan.

Melalui tugas besar ini, diharapkan Anda tidak hanya memahami teori di balik metode-metode tersebut, tetapi juga mampu mengimplementasikannya secara nyata yang dapat digunakan untuk menyelesaikan berbagai kasus uji (*test cases*) yang telah disediakan.

Secara formal, tujuan dari tugas besar ini adalah:

- Mengimplementasikan berbagai metode penyelesaian sistem persamaan linier, perhitungan determinan, invers matriks, interpolasi, dan regresi polinomial secara mandiri dalam bahasa Java.
- Membuat pustaka (*library*) yang dapat digunakan secara modular dan terdokumentasi dengan baik.
- Mengintegrasikan pustaka yang dibuat ke dalam sebuah program yang dapat menerima masukan dari pengguna dan menampilkan hasilnya dengan format yang jelas.
- Menguji pustaka yang dibuat pada berbagai kasus uji dan menganalisis hasilnya.

Dengan demikian, tugas besar ini diharapkan dapat menjadi sarana pembelajaran yang efektif dalam memahami dan mengaplikasikan konsep-konsep aljabar linier secara komputasional, serta membekali Anda dengan keterampilan praktis yang sangat dibutuhkan di dunia profesional.

## 2 Dasar Teori

### 2.1 Sistem Persamaan Linear

Sistem Persamaan Linear (SPL) adalah himpunan berhingga dari persamaan-persamaan linear dengan variabel-variabel yang sama. Bentuk umum dari SPL dengan  $m$  persamaan dan  $n$  variabel adalah

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n \end{cases}$$

atau, dalam bentuk perkalian matriks (kiri) dan matriks *augmented* (kanan),

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_A \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_x = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_b \quad \longrightarrow \quad \left[ \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array} \right]$$

Solusi dari SPL dapat berupa solusi tunggal, banyak solusi, atau tidak ada solusi. Metode penyelesaian yang umum digunakan antara lain eliminasi Gauss, eliminasi Gauss-Jordan, kaidah Cramer, dan metode matriks balikan.

### 2.2 Determinan Matriks

Determinan adalah sebuah nilai skalar khusus yang dapat dihitung dari elemen-elemen sebuah matriks persegi. Determinan dari matriks  $A$  dinotasikan sebagai  $\det(A)$  atau  $|A|$ . Nilai determinan sangat krusial karena dapat mengindikasikan apakah matriks tersebut memiliki invers. Beberapa metode untuk menghitung determinan matriks mencakup metode ekspansi kofaktor dan metode reduksi baris.

### 2.3 Invers Matriks

Invers dari sebuah matriks persegi  $A$  adalah matriks  $A^{-1}$  yang memenuhi properti  $AA^{-1} = A^{-1}A = I$ , di mana  $I$  adalah matriks identitas. Sebuah matriks hanya memiliki invers jika dan hanya jika determinannya tidak nol ( $\det(A) \neq 0$ ). Matriks yang memiliki invers disebut matriks *invertible* atau non-singular. Perhitungan invers dari suatu matriks dapat dilakukan dengan metode matriks adjoin maupun metode eliminasi Gauss-Jordan.

### 2.4 Interpolasi Polinomial

Interpolasi polinomial adalah metode untuk menentukan suatu polinomial  $P_n(x)$  berderajat  $n$  yang melalui  $n + 1$  titik data  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ ,  $x_i$  diurutkan dengan  $x_0$  terkecil dan  $x_n$  terbesar. Secara umum, polinomial tersebut dapat dituliskan sebagai:

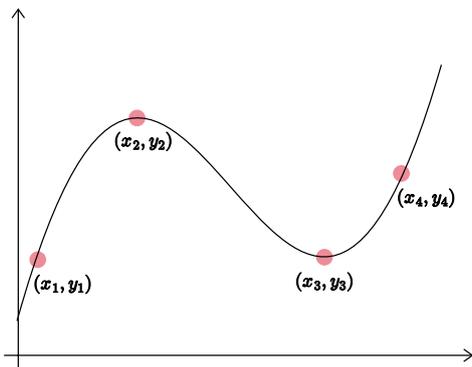
$$P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

dengan koefisien  $a_0, a_1, \dots, a_n$  yang ditentukan sedemikian rupa sehingga  $P_n(x_{\text{perkiraan}}) = y_{\text{perkiraan}}$  untuk semua nilai  $x_0 < x_{\text{perkiraan}} < x_n$ . Derajat persamaan polinom ditentukan oleh jumlah titik data yang diberikan dikurangi satu.

Koefisien polinomial dapat dicari dengan menyulihkan titik-titik data ke dalam persamaan polinomial dan menyelesaikan sistem persamaan linear yang dihasilkan dengan metode eliminasi Gauss yang dijelaskan sebelumnya. Kita dapat menyusun sistem persamaan linear sebagai berikut.

$$\begin{cases} a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0 \\ a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = y_1 \\ \vdots \\ a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = y_n \end{cases}$$

Persamaan-persamaan di atas dapat dipetakan menjadi sebuah permasalahan sistem persamaan linear, sehingga dapat diselesaikan dengan menggunakan metode eliminasi Gauss untuk menyelesaikan



Gambar 2: Contoh interpolasi polinomial melalui empat titik data.

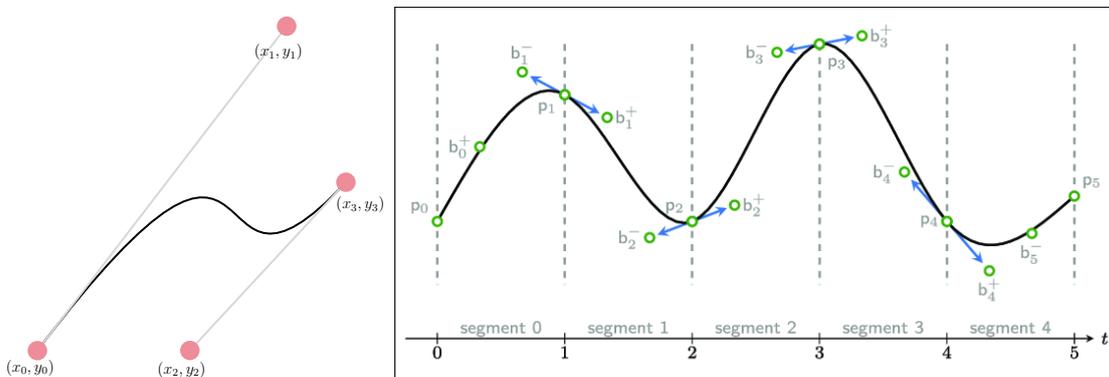
sistem tersebut dan menemukan koefisien polinomial  $a_0, a_1, \dots, a_n$ . Dari polinom  $P_n(x)$  yang dibentuk dengan koefisien tersebut, kita dapat memperkirakan nilai  $y$  untuk nilai  $x$  yang berada di antara titik-titik data yang diberikan (interpolasi) maupun di luar titik-titik data tersebut (ekstrapolasi).

### 2.5 Interpolasi Splina Bézier Kubik

Ketika menggunakan *pen tool* di Adobe Photoshop/Illustrator, kita akan berhadapan dengan kurva Bézier, atau lebih spesifiknya kurva Bézier kubik. Kurva ini sangat berguna untuk menggambar bentuk-bentuk halus dan kompleks dengan kontrol yang presisi.

Kurva Bézier kubik adalah kurva parametrik yang didefinisikan oleh empat titik kontrol yang mempengaruhi bentuk kurva tersebut. Titik kontrol pertama ( $P_0$ ) dan terakhir ( $P_3$ ) adalah titik awal dan akhir kurva, sedangkan titik-titik kontrol lainnya ( $P_1$  dan  $P_2$ ) menentukan arah dan kelengkungan kurva. Secara matematis, kurva Bézier kubik dapat ditulis sebagai persamaan parametrik

$$B(t) = (1 - t)^3 P_0 + 3t(1 - t)^2 P_1 + 3t^2(1 - t) P_2 + t^3 P_3, \quad 0 \leq t \leq 1.$$



Gambar 3: Kurva Bézier kubik dengan empat titik kontrol (kiri) dan kurva komposit yang tersusun atas banyak splina Bézier kubik (kanan, sumber).

Misalkan Kita diberikan titik-titik interpolasi  $S_0, S_1, \dots, S_n$  dan kita perlu menemukan titik-titik kontrol  $B_0, B_1, \dots, B_n$  yang sesuai untuk membangun kurva halus<sup>1</sup> yang melewati semua titik  $S_i$ . Kita dapat membangun kurvanya dari **splina Bézier kubik**, yakni himpunan kurva Bézier kubik yang didefinisikan pada segmen-segmen  $\overline{Q_i Q_{i+1}}$  untuk  $0 \leq i \leq n - 1$ .

Masalah ini dapat direduksi menjadi penyelesaian sistem persamaan linear yang didapatkan dari hubungan antara titik-titik  $S_i$  dan titik-titik kontrol  $B_i$  (silakan baca referensi). Sistem persamaan

<sup>1</sup>Pada *paper* referensi kita, “halus” maksudnya adalah memenuhi kontinuitas  $C^2$ , yakni turunan keduanya selalu kontinu di mana-mana.

tersebut adalah

$$\begin{cases} 4\mathbf{b}_1 + \mathbf{b}_2 & = 6\mathbf{s}_1 - \mathbf{s}_0 \\ \mathbf{b}_1 + 4\mathbf{b}_2 + \mathbf{b}_3 & = 6\mathbf{s}_2 \\ \mathbf{b}_2 + 4\mathbf{b}_3 + \mathbf{b}_4 & = 6\mathbf{s}_3 \\ \dots & \\ \mathbf{b}_{n-2} + 4\mathbf{b}_{n-1} & = 6\mathbf{s}_{n-1} - \mathbf{s}_n \end{cases},$$

dengan  $\mathbf{b}_k$  dan  $\mathbf{s}_k$  adalah vektor posisi dari titik-titik kontrol  $B_k$  dan titik-titik interpolasi  $S_k$ , atau dapat dinyatakan dalam bentuk perkalian matriks,

$$\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & & 1 & 4 & \ddots \\ & & & \ddots & \ddots & 1 \\ & & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_{n-1} \end{bmatrix} = \begin{bmatrix} 6\mathbf{s}_1 - \mathbf{s}_0 \\ 6\mathbf{s}_2 \\ 6\mathbf{s}_3 \\ \vdots \\ 6\mathbf{s}_{n-1} - \mathbf{s}_n \end{bmatrix}.$$

Karena vektor posisi  $\mathbf{b}_k$  dan  $\mathbf{s}_k$  memiliki dua komponen (koordinat  $x$  dan  $y$ ), kita dapat memisahkan sistem persamaan di atas menjadi dua sistem persamaan linear yang terpisah, satu untuk komponen  $x$  dan satu untuk komponen  $y$ .

## 2.6 Regresi Polinomial Berganda (*Multivariate Polynomial Regression*)

Diberikan  $n$  titik data  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  dengan  $x_i$  adalah variabel dan  $y_i$  adalah hasil pengukuran ke- $i$  ( $1 \leq i \leq n$ ), regresi linear sederhana<sup>2</sup> bertujuan untuk menemukan garis lurus  $y = \beta_0 + \beta_1 x$  yang paling sesuai dengan data tersebut.<sup>3</sup> Namun, dalam banyak kasus, hubungan antarvariabel tidak selalu linier dan suatu besaran bisa dipengaruhi oleh lebih dari satu variabel. Untuk mengatasi hal ini, kita dapat menggunakan **regresi polinomial berganda** (*multivariate polynomial regression*).

Idenya masih sama: mencari fungsi polinomial yang kurvanya paling sesuai dengan data yang diberikan.<sup>4</sup> Misalnya, untuk dua variabel independen  $x_1, x_2$ , dan  $x_3$  dan derajat 2, fungsi polinomial yang kita hasilkan bisa berbentuk

$$\begin{aligned} y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1^2 + \beta_5 x_1 x_2 + \beta_6 x_1 x_3 + \beta_7 x_2^2 + \beta_8 x_2 x_3 + \beta_9 x_3^2 + \epsilon \\ &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_9 x_9 + \epsilon. \end{aligned}$$

dengan  $\beta_0, \beta_1, \dots, \beta_9$  adalah koefisien yang akan kita cari,  $x_4 = x_1^2, x_5 = x_1 x_2, x_6 = x_1 x_3, x_7 = x_2^2, x_8 = x_2 x_3$ , dan  $x_9 = x_3^2$ , dan  $\epsilon$  adalah galat. Secara umum, jika kita memiliki  $k$  variabel independen dan ingin membuat polinomial dengan derajat  $d$ , jumlah total suku dalam polinomial tersebut adalah  $p = \binom{d+k}{k}$ .

Misalkan kita mempunyai  $n$  buah titik data dan ingin membuat model regresi polinomial berganda dengan derajat  $d$ . Misalkan juga  $x_{ij}$  adalah nilai dari variabel  $x_j$  pada titik data ke- $i$  dan  $y_i$  adalah nilai hasil pengukuran pada titik data ke- $i$ . Dengan demikian, kita dapat menuliskan sistem persamaan linear berikut untuk  $n$  titik data,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix}_{n \times p} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}_{p \times 1} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}_{n \times 1},$$

atau dalam bentuk yang lebih ringkas,

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

Tugas kita pada dasarnya adalah mengestimasi nilai vektor koefisien<sup>5</sup>  $\hat{\boldsymbol{\beta}}$  yang meminimalkan jumlah kuadrat galat, yaitu  $\|\boldsymbol{\epsilon}\|^2 = \|\mathbf{y} - X\boldsymbol{\beta}\|^2$ . Dengan menggunakan metode kuadrat terkecil (*least squares*), kita dapat menemukan solusi untuk  $\hat{\boldsymbol{\beta}}$  dengan menyelesaikan persamaan normal berikut:

$$X^T X \hat{\boldsymbol{\beta}} = X^T \mathbf{y},$$

<sup>2</sup>Anda sudah pernah berurusan dengan regresi linear sederhana pada praktikum Fisika Dasar semasa TPB.

<sup>3</sup>*Best-fit line*.  $\beta_0$  adalah titik potong dengan sumbu- $y$  dan  $\beta_1$  adalah gradien garis.

<sup>4</sup>Namun, kali ini, fungsi tersebut dapat melibatkan lebih dari satu variabel dan memiliki derajat yang lebih tinggi.

<sup>5</sup>Tanda  $\hat{\phantom{x}}$  menunjukkan bahwa ini adalah estimasi dari nilai sebenarnya.



atau

$$\hat{\beta} = (X^T X)^{-1} X^T \mathbf{y}. \quad (1)$$

## 3 Spesifikasi

### 3.1 Spesifikasi Wajib Program

Secara ringkas, luaran minimum yang diharapkan pada tugas besar ini terdapat tiga:

1. Buatlah pustaka (*library* atau *package*) dalam bahasa Java untuk segala perhitungan matriks, regresi, dan interpolasi yang disebut di bagian 2. Bentuk dari pustaka ini adalah file `.jar` yang dapat digunakan pada program Java lainnya.
2. Gunakan pustaka yang dibuat pada poin 1 untuk membuat program berbasis **antarmuka baris perintah** (*command line interface*, CLI) atau **antarmuka pengguna grafis** (*graphical user interface*, GUI) yang memperlihatkan fungsionalitas dari pustaka tersebut disertai dengan dokumentasi README pada repository GitHub. Untuk setiap fitur wajib, program dapat menerima masukan secara manual (input keyboard) maupun dari berkas `.txt` dengan format yang ditentukan.
3. Laporan mengenai pustaka dan program yang dibuat.

#### 3.1.1 Batasan Umum Tugas Besar

1. Bahasa program yang digunakan adalah **Java**. Anda bebas untuk menggunakan versi Java apapun di atas Java versi 8.
2. Build tool yang digunakan untuk membuat program adalah **Maven** yang sudah disediakan pada repository template yang disediakan asisten.
3. Semua pustaka Java diperbolehkan **kecuali** segala pustaka yang membantu implementasi perhitungan matriks, regresi, dan interpolasi.
  - (a) Seluruh perhitungan tersebut, termasuk implementasi kelas dasar seperti matriks, harus dibuat dan diimplementasikan sendiri dalam pustaka yang dikembangkan.
  - (b) Untuk perhitungan dasar matematika, hanya diperbolehkan menggunakan `java.lang.Math`.
  - (c) Silakan konfirmasi di `sheets QnA` jika ada keraguan mengenai pustaka yang boleh digunakan.
4. Untuk program akhir, **error handling** untuk setiap tahapan program adalah **kewajiban**.<sup>6</sup> Harus dipastikan program tidak mungkin crash atau berhenti akibat suatu input, proses, atau output yang membuat program error. Satu cara umum yang direkomendasi untuk error handling adalah `Java Exception (try & catch)`.
5. Tanda desimal yang digunakan pada tugas besar ini dibebaskan untuk menggunakan koma (,) atau titik (.). Untuk membatasi jumlah angka di belakang desimal pada penampilan, setiap bilangan dibulatkan hingga maksimal 3 angka setelah desimal. Input dari file ataupun manual harus dapat mendukung kedua pembatas koma atau titik. *Diasumsikan bilangan input tidak menggunakan pemisah ribuan.*

**Contoh bilangan valid:**

- 124320,42353
  - -42354.2354
  - 0.343
  - -79.323434
6. Matriks yang diinput secara manual dapat dipastikan dimensinya **tidak lebih dari**  $11 \times 11$ . Sementara itu, matriks yang diinput melalui file `.txt` dapat dipastikan dimensinya **tidak lebih dari**  $1001 \times 1001$ . Tambahan +1 pada batasan input memperhitungkan kolom konstanta pada sistem persamaan linier.

*Catatan:* Batasan input ini tidak melingkupi augmentasi yang mungkin dilakukan pada tahap perhitungan. Sebagai contoh, pada pencarian invers dengan augmentasi bisa jadi input adalah matriks  $1000 \times 1000$ , sehingga pada proses perhitungan invers digunakan matriks  $1000 \times 2000$ .

<sup>6</sup>Siapkan program Anda untuk beragam kemungkinan kesalahan format input! :D

7. Untuk interpolasi, dipastikan **jumlah sampel tidak lebih dari 10** baik untuk input manual maupun input file. Sementara itu, untuk regresi polinomial, dipastikan **jumlah peubah tidak lebih dari 5** dan **derajat polinom tidak lebih dari 5** baik untuk input manual maupun input file.

### 3.1.2 Rincian Pustaka

Secara garis besar, pustaka minimal mempunyai modul sebagai berikut:

- a. Penyelesaian Sistem Persamaan Linier
  - a.1. Eliminasi Gauss
  - a.2. Eliminasi Gauss-Jordan
  - a.3. Kaidah Cramer
  - a.4. Metode Matriks Balikan
- b. Mencari Determinan Matriks
  - b.1. Metode Ekspansi Kofaktor
  - b.2. Metode Reduksi Baris (OBE)
- c. Mencari Invers Matriks
  - c.1. Metode Augment
  - c.2. Metode Adjoin
- d. Interpolasi
  - d.1. Interpolasi Polinomial
  - d.2. Interpolasi splina Bézier kubik
- e. Regresi Polinomial Berganda

### 3.1.3 Rincian Umum Program

Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Sebagai contoh (boleh berbeda), menu utama dapat terlihat seperti ini:

1. Sistem Persamaan Linier
2. Determinan
3. Matriks Balikan
4. Interpolasi
5. Regresi Polinomial

Ada submenu pada tiap fitur untuk menentukan metode sesuai dengan isi pustaka pada bagian 3.1.2. Sebagai contoh setelah memilih opsi 1 pada menu utama, muncul menu untuk memilih metode sebelum input matriks:

1. Eliminasi Gauss
2. Eliminasi Gauss-Jordan
3. Kaidah Cramer
4. Metode Invers

Rincian dari submenu untuk tiap menu dapat disesuaikan dengan isi pustaka pada bagian 3.1.2.

### 3.1.4 Rincian Input dan Output Program

Untuk semua modul, jika format file input tidak sesuai, wajib dilakukan *error handling*. Format penamaan file input hanya untuk memperjelas isi dari file tanpa perlu membuka file. Nama file tidak boleh digunakan untuk pengecekan input.

#### A. Sistem Persamaan Linier (SPL)

##### Input:

- Sistem persamaan linier yang dimasukkan secara manual (diketik) atau melalui satu file `.txt` dengan format tertera di bawah. (setiap bilangan dipisahkan oleh spasi).
- Metode penyelesaian SPL, sesuai daftar pada Bagian 3.1.2.

##### Format Input File SPL:

Nama file: `spl_xxx.txt`

Keterangan: `xxx` adalah deskripsi bebas dari isi file.

##### Format isi file (matriks augmented):

```
x11  x12  ...  x1n  y1
x21  x22  ...  x2n  y2
...  ...  ...  ...  ...
xm1  xm2  ...  xmn  ym
```

##### Contoh:

```
3    4.5  2.8  10  12
-3   7    8.3  11  -4
0.5 -10  -9    12  0
```

##### Output:

- Solusi SPL:
  - Jika solusi tunggal: tampilkan nilainya.
  - Jika tidak ada solusi: tampilkan *solusi tidak ada*.
  - Jika tak hingga solusi: tampilkan dalam bentuk parametrik. Contoh:

$$x_4 = -2, \quad x_3 = 2s - t, \quad x_2 = s, \quad x_1 = t.$$

- Langkah-langkah proses sesuai metode.
- Output dapat disimpan dalam file `.txt`, berisi:
  - Metode penyelesaian SPL
  - Input yang digunakan
  - Hasil SPL

#### B. Determinan

##### Input:

- Matriks yang dimasukkan secara manual (diketik) atau melalui satu file `.txt` dengan format tertera di bawah (setiap bilangan dipisahkan oleh spasi).
- Metode pencarian determinan, sesuai daftar pada Bagian 3.1.2.

##### Format Input File Determinan:

**Nama file:** `determinan_xxx.txt`

*Keterangan:* xxx adalah deskripsi bebas dari isi file.

**Format isi file (matriks):**

```
x11  x12  ...  x1n
x21  x22  ...  x2n
...  ...  ...  ...
xn1  xn2  ...  xnn
```

**Contoh:**

```
3    4.5  2.8
-3   7    8.3
0.5 -10  -9
```

**Output:**

- Nilai determinan matriks.
- Langkah-langkah perhitungan sesuai metode.
- Output dapat disimpan dalam file `.txt`, berisi:
  - Metode pencarian determinan
  - Input yang digunakan
  - Hasil determinan

### C. Matriks Balikan (Invers)

**Input:**

- Matriks yang dimasukkan secara manual (diketik) atau melalui satu file `.txt` dengan format tertera di bawah (setiap bilangan dipisahkan oleh spasi).
- Metode pencarian matriks balikan, sesuai daftar pada Bagian 3.1.2.

**Format Input File Matriks Balikan:**

**Nama file:** `inverse_xxx.txt`

*Keterangan:* xxx adalah deskripsi bebas dari isi file.

**Format isi file (matriks):**

```
x11  x12  ...  x1n
x21  x22  ...  x2n
...  ...  ...  ...
xn1  xn2  ...  xnn
```

**Contoh:**

```
3    4.5  2.8
-3   7    8.3
0.5 -10  -9
```

**Output:**

- Matriks balikan hasil.
- Langkah-langkah perhitungan sesuai metode.
- Output dapat disimpan dalam file `.txt`, berisi:
  - Metode pencarian matriks balikan
  - Input yang digunakan

- Matriks balikan hasil

#### D. Interpolasi

##### Input:

- Jumlah titik sampel  $n$ .
- Data titik sampel  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , dimasukkan secara manual atau melalui satu file `.txt` dengan format tertera di bawah.
- Metode interpolasi, sesuai daftar pada Bagian 3.1.2.

##### Format Input File Interpolasi:

Nama file: `interpolasi_xxx.txt`

*Keterangan:* xxx adalah deskripsi bebas dari isi file.

##### Format isi file (titik sampel):

```
x1  y1
x2  y2
...  ...
xn  yn
```

##### Contoh (3 titik sampel):

```
3    4.5
-3   7
0.5 -10
```

##### Output (interpolasi polinomial):

- Persamaan interpolasi  $y(x)$ .
- Dari persamaan yang dibentuk, program dapat menerima input nilai-nilai  $x_t$  tertentu dalam domain interpolasi untuk mengevaluasi nilai  $y_t$ .
- Output dapat disimpan dalam file `.txt`, berisi:
  - Metode interpolasi
  - Input yang digunakan
  - Domain interpolasi
  - Persamaan hasil

##### Output (interpolasi splina Bézier kubik):

- Titik kontrol untuk setiap segmen kurva Bézier kubik.

#### E. Regresi Polinomial Berganda Input:

- Jumlah titik sampel  $n$ .
- Derajat regresi  $m$ .
- Data titik sampel  $(x_{1,1}, x_{1,2}, \dots, x_{1,k}, y_1), (x_{2,1}, x_{2,2}, \dots, x_{2,k}, y_2), \dots, (x_{n,1}, x_{n,2}, \dots, x_{n,k}, y_n)$ , di mana  $k$  adalah jumlah peubah (variabel independen) pada regresi,  $k \leq 5$ , dimasukkan secara manual atau melalui satu file `.txt` dengan format tertera di bawah.

##### Format Input File Regresi:

Nama file: regresi\_xxx.txt

Keterangan: xxx adalah deskripsi bebas dari isi file.

Format isi file (titik sampel dan derajat pangkat):

```
x_11 x_12 ... y_1
x_21 x_22 ... y_2
...   ...   ...
x_n1 x_n2 ... y_n
m
```

Contoh (6 titik sampel, dengan jumlah peubah = 2, dan derajat pangkat = 5):

```
3    4.5
-3   7
0.5 -10
9    3.4
-8   7.9
10   20
5
```

**Output:**

- Persamaan regresi  $y(x)$ .<sup>7</sup>
- Dari persamaan yang dibentuk, program dapat menerima input nilai-nilai  $x_{t,1}, x_{t,2}, \dots, x_{t,k}$  tertentu dalam domain regresi untuk mengevaluasi nilai  $y_t$ .
- Output dapat disimpan dalam file .txt, berisi:
  - Input yang digunakan
  - Persamaan hasil

## 3.2 Spesifikasi Bonus

### 3.2.1 Antarmuka Pengguna Grafis (APG/GUI)

Silahkan gunakan GUI jika ingin membuat aplikasi yang lebih interaktif. Perlu diingat bahwa tugas besar ini menggunakan Java, sehingga kakas yang digunakan dalam membuat GUI haruslah berasal dari bahasa Java. Anda juga hanya diperbolehkan untuk membuat Desktop App dari tugas besar ini sehingga **tidak diperbolehkan untuk membuat aplikasi web**.

Kakas GUI yang boleh digunakan pada tugas besar ini adalah **JavaFX**.<sup>8</sup> IDE yang digunakan dibebaskan (Contoh: IntelliJ IDEA, NetBeans, dll). Untuk mempermudah, asisten telah memberikan build tool beserta file GUI yang bisa kalian modifikasi di repositori kalian. Semua fitur wajib harus dapat dijalankan dengan GUI jika anda membuat bonus ini.

### 3.2.2 Seamless Cloning dengan Persamaan Poisson

Ini adalah penerapan dari penyelesaian SPL yang bisa menanam suatu gambar pada gambar lain dengan seamless (terpadu). Berikut adalah contoh hasilnya,

<sup>7</sup>Variabel  $x_1^2$  dapat dituliskan sebagai  $x_1^2$ .

<sup>8</sup>Gunakan **Scene Builder** untuk membantu merancang desain GUI secara interaktif (*drag-and-drop*).

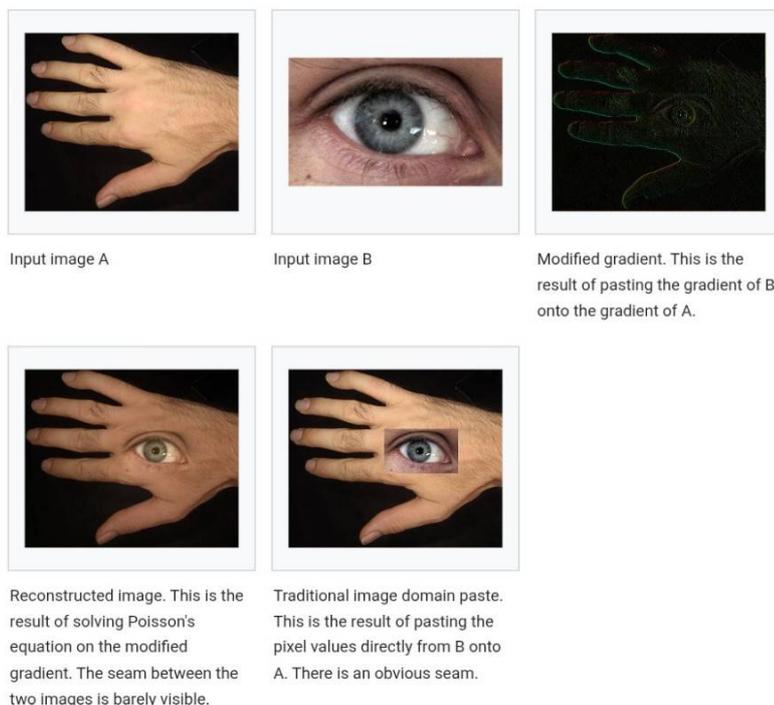


Image B ditanam pada Image A secara seamless.

Biarkan gambar A sebagai gambar dasar dan gambar B sebagai gambar yang akan ditanam pada A. Untuk bonus ini, berbeda dengan gambar di atas, gambar yang hanya perlu diubah adalah gambar B sehingga hanya gambar B yang akan menyesuaikan diri di gambar A. Selain itu, gambar B yang valid adalah gambar B yang lebih kecil di A dan *offset*-nya sedemikian rupa sehingga masih dalam range gambar A. Jika tidak valid, harus ditolak program menggunakan *error handling*.

Poisson Image Editing dapat dicapai dengan memperhitungkan nilai intensitas pada suatu pixel yang ditinjau menggunakan pixel-pixel tetangganya. Selengkapnya dapat dibaca dari artikel berikut: [Poisson Image Editing](#) Persamaannya sebagai berikut yang diterapkan pada gambar B:

$$4f(c) - f(r) - f(l) - f(u) - f(d) = g(r) + g(l) + g(u) + g(d)$$

- $c$  (current): pixel yang dicari pada gambar B.
- $r, l, u, d$  (right, left, up, down): tetangga dari pixel yang dicari.
- $f(x)$ : fungsi intensitas<sup>9</sup> dari setiap pixel.
- $g(x)$ : fungsi gradien<sup>10</sup> dari suatu tetangga pixel yang dicari.

Operasi matriks untuk bonus ini cukup intensif. Dengan gambar B berdimensi  $m \times n$ , jumlah variabel yang perlu dicari juga pada SPL  $m \times n$  juga. Ini akan terlalu besar untuk dicari dengan eliminasi Gauss-Jordan karena memerlukan matriks  $m^2 \times n^2$ .

Oleh karena itu, untuk menyelesaikan bonus ini, harus diselesaikan SPL-nya dengan metode lain, yakni **Metode Jacobi**. Metode Jacobi cocok untuk SPL yang *sparse*, yakni SPL yang memiliki banyak variabel, tetapi untuk satu persamaannya tidak menggunakan banyak variabel dibanding sistemnya secara keseluruhan. Jadi, untuk kasus Poisson Image Editing yang hanya menggunakan tidak lebih dari 5 variabel untuk setiap persamaan, metode ini sangat cocok.

Metode tersebut merupakan cara untuk mengaproksimasi nilai dari SPL secara iteratif. Metode ini dapat dibaca lebih lanjut berikut: [Iterative methods for sparse linear systems](#).

<sup>9</sup>Intensitas pada pixel gambar berupa nilai R, G, B, A. Ingat bahwa intensitas tetangga yang juga termasuk gambar B belum diketahui sehingga merupakan variabel yang harus dicari pada SPL

<sup>10</sup>Tujuan dari fungsi gradien ini adalah untuk memperoleh mask/bentuk dari gambar B. Dari itu, seandainya tetangga adalah pixel gambar A, nilainya akan nol. Gradien dari pixel tetangga adalah intensitas pixel dicari dikurangi intensitas pixel tetangga

**Input** untuk bonus ini adalah:

- **Gambar A:** gambar dasar
- **Gambar B:** gambar yang ingin ditempelkan
- *Offset* lokasi gambar B pada gambar A:  $(x, y)$  dengan origin pada atas kiri gambar A

**Output** dari bonus adalah gambar gabungan A dan B yang dapat disimpan.

### 3.2.3 Menggunakan $\LaTeX$ dalam Penulisan Laporan

$\LaTeX$  adalah sistem penyusunan dokumen yang sering digunakan di ranah ilmiah, khususnya yang membutuhkan penulisan rumus matematika secara intensif, tetapi tidak hanya terbatas pada itu. Bahkan dokumen spesifikasi yang super rapi ini pun ditulis menggunakan  $\LaTeX$ .

Salah satu keuntungan utama  $\LaTeX$  adalah dapat menghasilkan berkas dalam format teks biasa (.tex), sehingga sangat cocok digunakan secara kolaboratif lewat Git untuk *version control*. Untuk mendapatkan bonus ini, buatlah dokumen menggunakan  $\LaTeX$  lalu export ke format .pdf. Pastikan file mentah (.tex) juga disertakan dalam folder yang sama dengan laporan sebagai bukti penggunaan  $\LaTeX$ .

### 3.2.4 Video Pengenalan Program

Terdapat banyak cara untuk menjelaskan cara kerja sebuah program. Namun penjelasan secara visual dan kreatif adalah teknik penjelasan yang sangat menarik perhatian orang-orang. Oleh karena itu, anda diperbolehkan membuat sebuah video penjelasan tentang program yang dibuat dengan durasi maksimal 15 menit. Buatlah video ini sekreatif mungkin (Jangan hanya menjelaskan program saja, tunjukkan skill acting kalian :D). Untuk referensi, silakan cek [tautan ini](#).

## 3.3 Prosedur Pengerjaan

1. Tugas dikerjakan secara berkelompok dengan anggota sebanyak **3 orang** (boleh lintas kelas maupun lintas kampus) yang **dipilih sendiri**. Buka [sheets pendataan kelompok](#) untuk mendaftarkan kelompok Anda dan memilih asisten sebelum **19 September 2025 pukul 23:59 WIB** karena peserta mata kuliah yang tidak terdata akan dipasangkan secara acak.
2. Masuk ke GitHub Classroom [Algeo25](#) dan buat kelompok menggunakan nama yang **sama persis dengan yang didaftarkan di sheets**.<sup>11</sup> Kerjakan tugas besar di repositori kelompok Anda (<https://github.com/IRK-23/algeo1-<namakelompok>>).
3. Anda dapat mengajukan pertanyaan kepada asisten melalui [sheets QnA](#).
4. Ajukan asistensi kepada asisten jika dibutuhkan. Asisten kelompok Anda juga akan menjadi penguji saat demo.
5. Setelah [tenggat pengumpulan tugas besar ini](#), jadwal dan teknis demo akan diumumkan melalui milis.
6. **DILARANG** menyontek pekerjaan orang lain atau menggunakan bantuan *generative AI* seperti ChatGPT, DeepSeek, Claude, dan Gemini (tetapi tidak terbatas pada itu). Segala bentuk kecurangan akan ditindaklanjuti sesuai dengan sanksi akademik yang ada.

## 3.4 Spesifikasi Laporan

1. **Sampul:** mirip seperti sampul dokumen ini<sup>12</sup> tetapi ganti logo IRK dengan foto kalian bertiga bersama-sama dan tulisan "Spesifikasi" menjadi "Laporan".
2. **Bab 1 (Pendahuluan):** deskripsi persoalan; boleh menyalin bab pendahuluan dari dokumen ini.
3. **Bab 2 (Dasar Teori):** teori singkat mengenai sistem persamaan linier, eliminasi Gauss dan Gauss-Jordan, matriks kofaktor, matriks adjoin, determinan, matriks invers, kaidah Cramer, interpolasi polinomial, interpolasi splina Bézier kubik, dan regresi polinomial berganda. Jika mengerjakan bonus, tambahkan juga teori mengenai metode iteratif Jacobi dan Poisson *image editing*.<sup>13</sup>

<sup>11</sup> Atau bergabung dengan tim Anda jika sudah ada anggota kelompok yang membuatnya lebih dahulu.

<sup>12</sup> Tidak harus dalam *dark mode* dan tanpa keterangan tenggat (DL).

<sup>13</sup> Untuk determinan dan invers, sertakan pembahasan untuk kedua metode yang ada.

4. **Bab 3 (Implementasi)**: penjelasan implementasi pustaka dan program dalam Java, meliputi struktur program dan kelas-kelas penting serta cara kerja dan alur program.
5. **Bab 4 (Eksperimen)**: Hasil eksekusi program terhadap seluruh kasus uji pada Bab 5 beserta analisisnya.<sup>14</sup>
6. **Bab 5 (Penutup)**: kesimpulan dari seluruh pengerjaan tugas besar, saran kepada asisten, dan komentar tiap anggota kelompok terhadap tugas besar.
7. **Daftar Pustaka**: seluruh referensi yang Anda gunakan untuk bagian dasar teori maupun keseluruhan pengerjaan tugas besar ini. Gunakan format `sitasi IEEE`.<sup>15</sup>
8. **Lampiran**: tautan menuju repositori GitHub dan video (jika mengerjakan bonus tersebut) kelompok Anda.

### 3.5 Kriteria Penilaian

Penilaian tugas besar ini dirincikan pada tabel di bawah. Nilai terdiri atas tiga komponen utama, yaitu implementasi dan demo program (80 poin), laporan (20 poin), dan bonus. Total poin pada kolom bonus memang berjumlah 30, tetapi skor bonus yang dihitung maksimal 20 poin. Artinya, jika kalian mengerjakan semua bonus dan memperoleh nilai lebih dari 20, maka skor bonus akan tetap dicatat sebagai 20 (tidak bisa melebihi 20 poin).

Tabel 1: Rincian Skor Maksimum untuk Tiap Komponen Penilaian

| Komponen  | Skor       |
|---|------------|
| Implementasi dan Demo                                       | 80         |
| Dokumentasi   | 20         |
| (Bonus) Antarmuka Pengguna Grafis (GUI)                     | 15         |
| (Bonus) Seamless Cloning dengan Persamaan Poisson           | 10         |
| (Bonus) Menggunakan $\text{\LaTeX}$ dalam Penulisan Laporan | 5          |
| (Bonus) Video Pengenalan Program                            | 3          |
| <b>Total</b>  | <b>120</b> |

<sup>14</sup>Untuk interpolasi polinomial, cantumkan juga plot titik dan persamaan garis pada aplikasi visualisasi grafik seperti Geogebra, Desmos, dan lain sebagainya.

<sup>15</sup>Kalian seharusnya sudah pernah menjumpai ini ketika mengerjakan makalah IF1220 Matematika Diskrit. :D

## 4 Pengumpulan

### 4.1 Deliverables

Berkas yang perlu dikumpulkan tiap kelompok untuk tugas besar ini adalah **kode sumber** (*source code*) dan **laporan** sesuai spesifikasi dengan rincian sebagai berikut.

1. Program dan laporan disimpan di dalam repositori GitHub yang otomatis dibuat ketika bergabung ke dalam GitHub Classroom A1geo25.
2. Minimalnya terdapat empat folder berikut di dalam repositori:
  - (a) Folder `docs` yang berisi *softcopy* laporan dalam bentuk PDF dengan nama file `<nama_kelompok>.pdf`.
  - (b) Folder `bin` yang berisi file `JAR` dari program yang bisa dijalankan.
  - (c) Folder `src` yang berisi kode sumber dari program.
  - (d) Folder `test` yang berisi data kasus uji dalam file `txt`.
3. Pastikan menggunakan `semantic commit message` yang sesuai pada repositori Anda.
4. Sertakan juga README yang dibuat se jelas mungkin. README minimal memuat penjelasan singkat program, alur program, dan tata cara menjalankan program. Gunakan `README ini` sebagai contoh.
5. Pastikan repositori bersifat *private*<sup>16</sup> selama pengerjaan tugas besar.
6. Pastikan *bytecode* Java di dalam folder `bin` dapat dijalankan.
7. Selain dijalankan saat demo, program akan dicoba dijalankan asisten di luar waktu demo untuk salah satu aspek penilaian. Asisten pemeriksa tidak akan melakukan pengaturan atau kompilasi lagi agar program dapat berjalan. Program yang tidak dapat dijalankan tidak akan diberi nilai.

### 4.2 Teknis Pengumpulan

1. Program dan laporan dikumpulkan dengan membuat `rilis` pada repositori GitHub untuk tiap kelompok. Selagi belum mencapai tenggat pengumpulan, Anda boleh memperbaiki pengerjaan Anda dan mengumpulkannya kembali sebagai revisi.
2. Untuk rilis, gunakan tag berformat `v1.x` dengan  $x$  adalah nomor revisi ( $x = 0$  untuk pengumpulan pertama).
3. Versi yang digunakan untuk penilaian adalah revisi terakhir. Catat bahwa ini berarti jika Anda sudah selesai mengerjakan sebelum tenggat lalu membuat revisi dan mengumpulkannya setelah tenggat, pekerjaan Anda dianggap terlambat.<sup>17</sup>
4. Tenggat pengumpulan tugas besar<sup>18</sup> adalah hari **Rabu, 8 Oktober 2025 pukul 21:23 WIB**.
5. Terdapat penalti 1 poin untuk tiap 6 menit keterlambatan. Keterlambatan dihitung dari selisih (dibulatkan ke kelipatan 6 di atasnya<sup>19</sup>) antara tenggat pengerjaan tugas besar dengan waktu ketika rilis terbaru dibuat di repositori GitHub.

<sup>16</sup>Boleh diubah ke *public* setelah tenggat pengumpulan tugas besar berlalu.

<sup>17</sup>Anda boleh saja meminta kepada asisten untuk menggunakan revisi sebelumnya yang tidak melewati tenggat, tetapi ini berarti perbaikan pada revisi terakhir Anda tidak dianggap. :D

<sup>18</sup>Waktu paling lambat untuk membuat rilis di repositori GitHub.

<sup>19</sup>Misalnya, keterlambatan 3 menit maupun 6 menit sama-sama menghasilkan penalti 1 poin.

## 5 Studi dan Kasus Uji

Untuk menguji program yang telah dibuat, tes dengan beberapa SPL, persoalan interpolasi polinom, dan matriks-matriks sebagai berikut:

### 5.1 Determinan Matriks

Tentukan nilai determinan dari matriks-matriks berikut. Uji masing-masing kasus dengan dua metode yang terdapat di rincian pustaka 3.1.2.

#### 5.1.1 Kasus Uji 1

$$\begin{bmatrix} 5 & 7 & 9 & 11 \\ 10 & 14 & 18 & 22 \\ 15 & 21 & 27 & 33 \\ 20 & 28 & 36 & 44 \end{bmatrix}$$

#### 5.1.2 Kasus Uji 2

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$$

### 5.2 Invers Matriks

Tentukan invers dari matriks-matriks berikut. Uji masing-masing kasus dengan dua metode yang terdapat di rincian pustaka 3.1.2.

#### 5.2.1 Kasus Uji 1

$$\begin{bmatrix} 5 & 2 & -3 & 4 \\ 7 & 7 & 8 & -9 \\ 12 & 13 & 2 & 14 \\ 17 & 18 & 19 & 4 \end{bmatrix}$$

#### 5.2.2 Kasus Uji 2

$$\begin{bmatrix} 5 & 7 & 9 & 11 & 13 & 15 \\ 10 & 14 & 18 & 22 & 26 & 30 \\ 15 & 21 & 27 & 33 & 39 & 45 \\ 20 & 28 & 36 & 44 & 52 & 60 \\ 25 & 35 & 45 & 55 & 65 & 75 \\ 30 & 42 & 54 & 66 & 78 & 90 \end{bmatrix}$$

### 5.3 Sistem Persamaan Linear $Ax = b$

Tentukan solusi SPL  $Ax = b$  berikut. Uji masing-masing kasus dengan empat metode yang terdapat di rincian pustaka 3.1.2.

### 5.3.1 Kasus Uji 1

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$

### 5.3.2 Kasus Uji 2

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

### 5.3.3 Kasus Uji 3

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

### 5.3.4 Kasus Uji 4

$$A = H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$H$  adalah matriks Hilbert. Coba uji untuk nilai  $n = 6$  dan  $n = 10$ .

## 5.4 Sistem Persamaan Linear Berbentuk Matriks *Augmented*

Tentukan solusi SPL berbentuk matriks *augmented* berikut. Uji masing-masing kasus dengan empat metode yang terdapat di rincian pustaka 3.1.2.

### 5.4.1 Kasus Uji 1

$$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}$$

### 5.4.2 Kasus Uji 2

$$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 4 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}$$

## 5.5 Sistem Persamaan Linier Berbentuk Umum

Tentukan solusi SPL berbentuk umum berikut. Uji masing-masing kasus dengan empat metode yang terdapat di rincian pustaka 3.1.2.

### 5.5.1 Kasus Uji 1

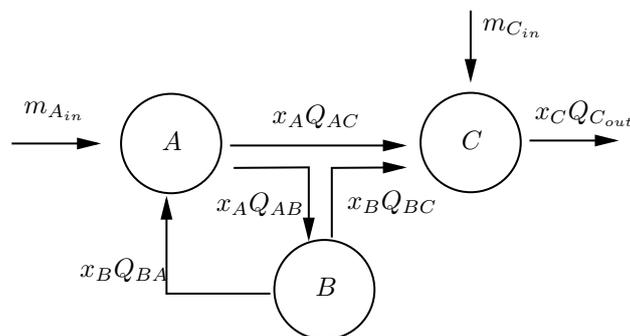
$$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$$

### 5.5.2 Kasus Uji 2

$$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\ x_3 + x_6 + x_9 &= 18.00 \\ x_2 + x_5 + x_8 &= 12.00 \\ x_1 + x_4 + x_7 &= 6.00 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\ 0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04 \end{aligned}$$

## 5.6 Aplikasi Sistem Persamaan Linier

Lihatlah sistem reaktor pada gambar berikut:



Gambar 4: Model rangkaian reaktor

Dengan laju volume  $Q$  dalam  $m^3/s$  dan input massa min dalam  $mg/s$ . Konservasi massa pada tiap inti reaktor adalah sebagai berikut:

$$\begin{aligned} \text{A: } m_{A_{in}} + Q_{BA}x_B - Q_{AB}x_A - Q_{AC}x_A &= 0 \\ \text{B: } Q_{AB}x_A - Q_{BA}x_B - Q_{BC}x_B &= 0 \\ \text{C: } m_{C_{in}} + Q_{AC}x_A + Q_{BC}x_B - Q_{C_{out}}x_C &= 0 \end{aligned}$$

Tentukan solusi  $x_A, x_B, x_C$  dengan menggunakan parameter berikut:

$$Q_{AB} = 40, Q_{AC} = 80, Q_{BA} = 60, Q_{BC} = 20, Q_{C_{out}} = 150 \text{ m}^3/s, m_{A_{in}} = 1300 \text{ dan } m_{C_{in}} = 200 \text{ mg/s.}$$

## 5.7 Studi Kasus Interpolasi Polinomial

### a. Studi Kasus 1

Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel. Program menerima masukan nilai  $x$  yang akan dicari nilai fungsi  $f(x)$ .

| $x$    | 0.1   | 0.3   | 0.5   | 0.7   | 0.9   | 1.1   | 1.3   |
|--------|-------|-------|-------|-------|-------|-------|-------|
| $f(x)$ | 0.003 | 0.067 | 0.148 | 0.248 | 0.370 | 0.518 | 0.697 |

Lakukan pengujian pada nilai-nilai berikut:

$$x = 0.2 \quad f(x) = ?$$

$$x = 0.55 \quad f(x) = ?$$

$$x = 0.85 \quad f(x) = ?$$

$$x = 1.28 \quad f(x) = ?$$

### b. Studi Kasus 2

Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2022 hingga 31 Agustus 2022:

| Tanggal    | Tanggal (desimal) | Jumlah Kasus Baru |
|------------|-------------------|-------------------|
| 17/06/2022 | 6.567             | 12.624            |
| 30/06/2022 | 6.967             | 21.807            |
| 08/07/2022 | 7.258             | 38.391            |
| 14/07/2022 | 7.451             | 19.541            |
| 21/07/2022 | 7.548             | 19.582            |
| 01/08/2022 | 8.032             | 28.935            |
| 08/08/2022 | 8.258             | 25.854            |
| 15/08/2022 | 8.484             | 20.935            |
| 22/08/2022 | 8.709             | 12.408            |
| 31/08/2022 | 8.997             | 10.534            |

Tanggal (desimal) adalah tanggal yang sudah diolah ke dalam bentuk desimal 3 angka di belakang koma dengan memanfaatkan perhitungan sebagai berikut:

$$\text{Tanggal (desimal)} = \text{bulan} + (\text{tanggal} / \text{jumlah hari pada bulan tersebut})$$

Sebagai contoh, untuk tanggal 17/06/2022 (dibaca: 17 Juni 2022) diperoleh tanggal (desimal) sebagai berikut:

$$\text{Tanggal (desimal)} = 6 + \frac{17}{30} = 6.567$$

Gunakan data di atas dengan memanfaatkan **interpolasi polinomial** untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

a. 16/07/2022

b. 10/08/2022

c. 05/09/2022

d. Masukan user lainnya berupa *tanggal (desimal)* yang sudah diolah dengan asumsi prediksi selalu dilakukan untuk tahun 2022.

## 5.8 Studi Kasus Interpolasi Splina Bézier Kubik

Tentukan titik kontrol dari tiap segmen kurva splina Bézier kubik yang melalui titik-titik berikut ini:

$$(0, 0), (3, 11), (6, -4), (8, 0), (11, -10), (17, 0).$$

## 5.9 Studi Kasus Regresi Polinomial Berganda

Wak Rusdi sedang meneliti pertumbuhan alga berdasarkan faktor cahaya, suhu, dan pH. Ia meneliti 50 sampel yang hasilnya dapat dilihat [di sini](#).

Silakan proses data tersebut menjadi file `txt` sesuai dengan format input yang telah dijelaskan sebelumnya lalu tentukan persamaan regresi polinomial berganda yang sesuai dengan data tersebut menggunakan polinomial kubik (derajat 3).

## 6 Referensi

### 6.1 SPL, Determinan, dan Matriks

- H. Anton, *Elementary Linear Algebra*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, 2010. [Online]. Available: <https://archive.org/details/ElementaryLinearAlgebraByHowardAnton10thEdition>

### 6.2 Interpolasi Polinomial

- G. Muntingh, *Topics in Polynomial Interpolation Theory*, Ph.D. dissertation, Centre of Mathematics for Applications, University of Oslo, Oslo, Norway, Dec. 2010. [Online]. Available: [https://www.researchgate.net/publication/265819309\\_Topics\\_in\\_Polynomial\\_Interpolation\\_Theory](https://www.researchgate.net/publication/265819309_Topics_in_Polynomial_Interpolation_Theory).

### 6.3 Interpolasi Splina Bézier Kubik

- L. P. Quan and T. A. Nhan, “A closed-form solution to the inverse problem in interpolation by a Bézier-spline curve,” *Arabian Journal of Mathematics*, vol. 9, pp. 155–165, 2020. [Online]. Available: <https://doi.org/10.1007/s40065-019-0241-0>

### 6.4 Regresi Polinomial Berganda

- M. Rosenfeld, *Matrix Formulation of OLS: Notes for SOC Methods Project 3*. Stanford, CA, USA: Stanford University. [Online]. Available: [https://web.stanford.edu/~mrosenfe/soc\\_meth\\_proj3/matrix\\_OLS\\_NYU\\_notes.pdf](https://web.stanford.edu/~mrosenfe/soc_meth_proj3/matrix_OLS_NYU_notes.pdf).
- A. B. Kashlak, 2022, *Applied Regression Analysis: Course Notes for STAT 378/502*. Edmonton, AB, Canada: University of Alberta, Dept. of Mathematical & Statistical Sciences. [Online]. Available: <https://sites.ualberta.ca/~kashlak/data/stat378.pdf>.
- S. Weisberg, *Applied Linear Regression*, 4th ed. Hoboken, NJ, USA: Wiley-Blackwell, 2013. [Online]. Available: <https://www.stat.purdue.edu/~qfsong/teaching/525/book/Weisberg-Applied-Linear-Regression-Wiley.pdf>

## 7 Surat Cinta dari Asisten

### Komentar 7.1: Nayaka

ကလေးတို့ကလေးတွေလိုပဲ  
စိတ်ကလေးပျက်ပျက်  
လေးလေးကလေးကလေး

### Komentar 7.2: Fayadh

Ada orang pernah bilang, “kenapa tubes itu menyenangkan?” Kadang ia penuh error, bikin pusing kepala. Kadang ia lancar, bikin hati lega dan bahagia. Tapi asisten selalu menerima hasil tubes apa adanya.

### Komentar 7.3: Weka

Pemanasan

### Komentar 7.4: Hakim

Semangat gess ngerjainnya, jangan lupa touch grass yak :3

### Komentar 7.5: Orvin



Kerjakanlah dengan sungguh-sungguh, karena Bayu Sang Pahlawan akan melakukan hal yang sama

### Komentar 7.6: Aryo



Spek tubes 1 algeo dirombak ulang setelah sekian lama ges! Ditunggu kerjaan tubesnya juga yang gk kalah gacor (-> U <-)