**Tugas Besar I**

**IF 2123**

**Aljabar Linier dan Geometri**

**Sistem Persamaan Linier, Determinan, dan Aplikasinya**

**Semester I Tahun 2020/2021**



Nama (Foto Kiri) / NIM : Almeiza Arvin Muzaki / 13519066

Nama (Foto Tengah) / NIM : Jose Galbraith Hasintongan / 13519022

Nama (Foto Kanan) / NIM : Benidictus Galih Mahar Putra / 13519159

**Bab 1**

**Deskripsi Masalah**

# SPESIFIKASI TUGAS

Buatlah program dalam **Bahasa Java** untuk

* 1. Menghitung solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan *n* peubah dan *n* persamaan).
  2. Menyelesaikan persoalan interpolasi dan regresi linier.
  3. Menghitung matriks balikan
  4. Menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Spesifikasi program adalah sebagai berikut:

1. Program dapat menerima masukan (input) baik dari *keyboard* maupun membaca masukan dari file text. Untuk SPL, masukan dari *keyboard* adalah *m*, *n*, koefisien *aij* , dan *bi*. Masukan dari *file* berbentuk matriks *augmented* tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya,

3 4.5 2.8 10 12

-3 7 8.3 11 -4

0.5 -10 -9 12 0

1. Untuk persoalan menghitung determinan dan matriks balikan, masukan dari *keyboard* adalah *n* dan koefisien *aij*. Masukan dari *file* berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya,

3 4.5 2.8 10

-3 7 8.3 11

0.5 -10 -9 12

1. Untuk persoalan interpolasi, masukannya jika dari *keyboard* adalah *n*, (*x*0, *y*0), (*x*1, *y*1), ..., (*xn*, *yn*), dan nilai *x* yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

8.0 2.0794

9.0 2.1972

9.5 2.2513

1. Untuk persoalan regresi, masukannya jika dari *keyboard* adalah *n* (jumlah peubah *x*), semua nilai-nilai *x*1*i*, *x*2*i*, ..., *xni*, nilai *yi*, dan nilai-nilai *xk* yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.
2. Untup persoalan SPL, luaran (*output*) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya *x*4 = -2, *x*3 = 2*s* – *t*, *x*2 = *s*, dan *x*1 = *t*.)
3. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing
4. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada *x* yang diberikan.
5. Luaran program harus dapat ditampilkan **pada layar komputer dan dapat disimpan ke dalam file**.
6. Bahasa program yang digunakan adalah Java.
7. Program **tidak harus** berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas *Eclipse* misalnya).
8. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

MENU

* 1. Sistem Persamaaan Linier
  2. Determinan
  3. Matriks balikan
  4. Interpolasi Polinom
  5. Regresi linier berganda
  6. Keluar

Untuk pilihan menu nomor 1 ada sub-menu lagi yaitu pilihan metode:

1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Kaidah Cramer

Begitu juga untuk pilihan menu nomor 2 dan 3.

# PROSEDUR PENGERJAAN

1. Tugas dikerjakan secara berkelompok yang terdiri dari 3 orang. Kelompok dipilih secara mandiri. Daftarkan kelompok anda via tautan <http://bit.ly/PendataanTubesAlgeo1>.
2. Tugas ini dikumpulkan hari Jumat 2 Oktober 2020 paling lambat pukul 17.00 WIB sore. Asisten akan mengumumkan jadwal demo program secara daring.
3. Dilarang keras menyalin program dari sumber lain (buku, internet, program kakak tingkat, program kelompok lain)

# LAPORAN

Laporan terdiri dari:

* 1. *Cover*: *Cover* laporan ada foto anggota kelompok (foto bertiga kalau ada, atau foto masing-masing, bebas gaya). Foto ini menggantikan logo “gajah” ganesha.
  2. Bab 1: Deskripsi masalah (dapat meng-*copy paste* file tugas ini).
  3. Bab 2: Teori singkat mengenai metode eliminasi Gauss, metode eliminasi Gauss-Jordan, determinan, matriks balikan, matriks kofaktor, matriks adjoin, kaidah Cramer, interpolasi polinom, regresi linier berganda.
  4. Bab 3: Implementasi program dalam Java, meliputi struktur *class* yang didefinisikan (atribut dan method), garis besar program, dll.
  5. Bab 4: Eksperimen. Bab ini berisi hasil eksekusi program terhadap contoh- contoh kasus yang diberikan berikut analisis hasil eksekusi tersebut
  6. Bab 5: Kesimpulan, saran, dan refleksi (hasil yang dicapai, saran pengembangan, dan refleksi anda terhadap tugas ini).
  7. Tuliskan juga referensi (buku, web), yang dipakai/diacu di dalam Daftar Referensi.

# Keterangan laporan dan program:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar, tidak perlu panjang tetapi tepat sasaran dan jelas.
2. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).
3. *Listing* program tidak perlu disertakan pada laporan.
4. Program disimpan di dalam *repository github* dengan nama repository Algeo01-
   1. Lima digit terakhir adalah NIM anggota terkecil. Didalam *repository*

tersebut terdapat empat folder: bin, src, test dan doc yang masing-masing berisi:

* + - Folder *bin* berisi *java byte code* (.class)
    - Folder *src* berisi *source code* dari program java
    - Folder *test* berisi data uji.
    - Folder *doc* berisi laporan

Sertakan juga *readme* yang dibuat sejelas mungkin. Pastikan *repository* bersifat ***private*** dan telah mengundang asisten yang pembagiannya akan diumumkan kemudian.

# PENGUMPULAN TUGAS

1. Yang diserahkan saat pengumpulan tugas adalah:
   1. Laporan (*soft copy*)
   2. Kode program + Java *bytecode*

Kedua komponen tersebut di*-commit* sebelum Jumat 2 Oktober 2020 pukul 17.00 WIB. Commit setelah waktu tersebut akan mendapat pengurangan nilai.

1. *Java bytecode* dapat dijalankan. Asisten pemeriksa tidak akan melakukan *setting* atau kompilasi lagi agar program dapat berjalan. Program yang tidak dapat dijalankan tidak akan diberi nilai.

# PENILAIAN

Komposisi penilaian umum adalah sebagai berikut :

1. Program: 80 %
2. Laporan : 20 %

**Bab 2**

**Teori Singkat**

* 1. **Sistem Persamaan Linear**

Sistem persamaan linier (SPL) *Ax* = *b* dengan *n* peubah (*variable*) dan *m* persamaan adalah berbentuk

*a*11 *x*1 + *a*12 *x*2 + + *a*1*n xn* = *b*1

*a*21 *x*1 + *a*22 *x*2 + + *a*2*n xn* = *b*2

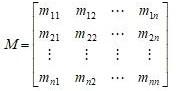
: :

: :

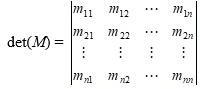
*am*1 *x*1 + *am*2 *x*2 + + *amn xn* = *bm*

yang dalam hal ini *xi* adalah peubah, *aij* dan *bi* adalah koefisien ∈ R. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan (*x* = *A*-1*b*), dan kaidah *Cramer* (khusus untuk SPL dengan *n* peubah dan *n* persamaan). Solusi sebuah SPL mungkin tidak ada, banyak, atau hanya satu (unik/tunggal).

Sebuah matriks *M* berukuran *n* × *n* berikut



dijabarkan determinannya dalam bentuk

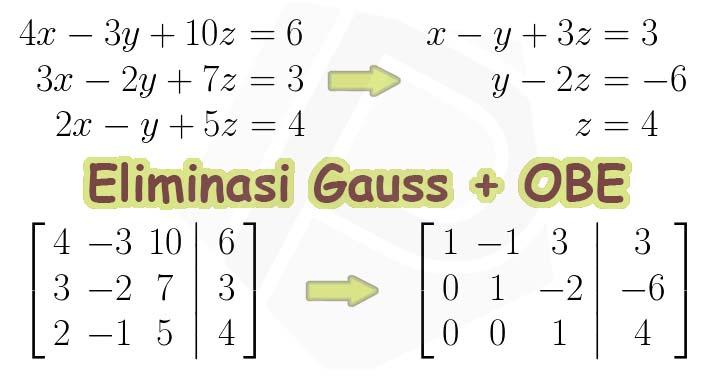


Nilai determinan matriks *M* berukuran *n* × *n* dapat dihitung dengan beberapa cara, diantaranya yaitu reduksi baris dan ekspansi kofaktor.

SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, dua diantaranya diterapkan pada tugas besar ini, yaitu interpolasi polinom dan regresi linier.

# 

# Metode Eliminasi Gauss

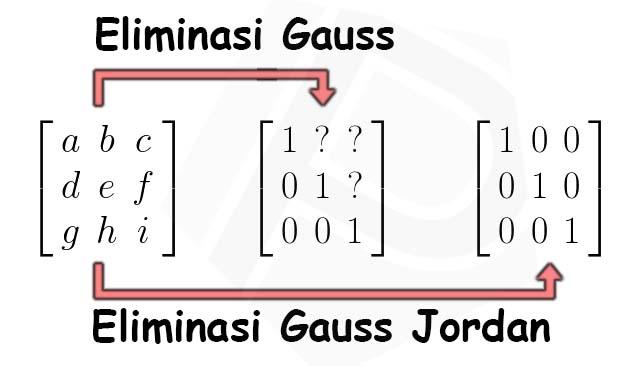
Eliminasi gauss dimanfaatkan untuk memecahkan sistem persamaan linear dengan merepresentasikan (mengubah) menjadi bentuk matriks, matriks tersebut lalu diubah ke bentuk Eselon Baris melalui [Operasi Baris Elementer.](https://www.profematika.com/pengenalan-operasi-baris-elementer/) Matriks eselon baris sendiri merupakan matriks yang memenuhi tiga syarat, yaitu memiliki satu utama sebagai bilangan tidak nol pertama, seluruh satu utama letaknya semakin ke bawah semakin bergeser ke kanan (tidak boleh pula dalam satu kolom yang sama) dan seluruh bilangan di bawah setiap satu utama wajib bernilai 0. 

**Gambar 1** Eliminasi Gauss

Matriks yang telah berbentuk eselon baris bisa dimanfaatkan untuk menyelesaikan beberapa persamaan.

# Metode Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan adalah prosedur pemecahan sistem persamaan linear dengan mengubahnya menjadi bentuk matriks eselon baris tereduksi dengan Operasi Baris Elementer.



**Gambar 2** Eliminasi Gauss-Jordan

# Metode Matriks Balikan

SPL Ax = b. Kalikan kedua ruas persamaan dengan A-1

(A-1 )Ax = (A-1 ) b

Ix = A-1 b (karena A-1A = I)

x =A-1 b (karena Ix = x)

•Jadi, solusi SPL Ax = b adalah x =A-1 b

# Kaidah Cramer

Jika Ax = b adalah SPL yang terdiri dari n persamaan linier dengan n peubah (variable) sedemikian sehingga det(A) ≠ 0, maka SPL tersebut memiliki solusi yang unik yaitu

x1= det(A1)/ det(A), x2= det(A2)/ det(A), x3= det(A3)/ det(A)

yang dalam hal ini, Aj adalah matriks yang diperoleh dengan mengganti entri pada kolom ke-j dari A dengan entri dari matriks

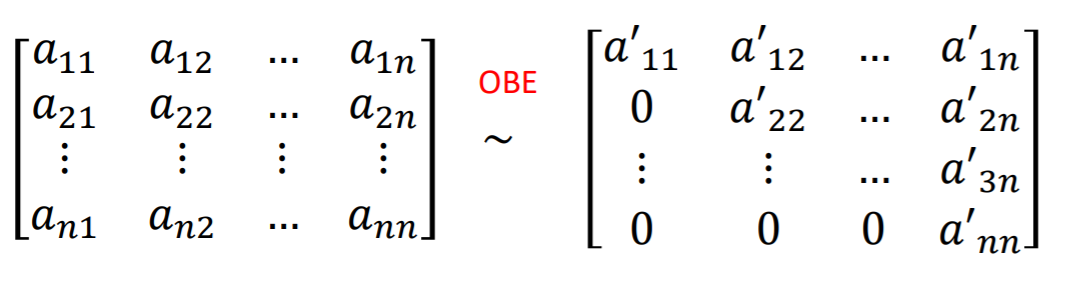
bT = [b1 b2…..bn ]

# Determinan

# Metode Reduksi Baris

Determinan matriks A dapat diperoleh dengan melakukan OBE pada matriks A sampai diperoleh matriks segitiga (segitiga bawah atau atas)

[𝐴] ~ [matriks segitiga bawah] dengan OBE

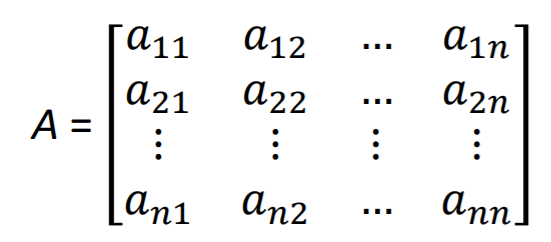


maka det(A) = (−1)p 𝑎′11𝑎′22 … 𝑎′nn

p menyatakan banyaknya operasi pertukaran baris di dalam OBE

# Metode Ekspansi Kofaktor

Misalkan A adalah matriks berukuran n x n



Didefinisikan:

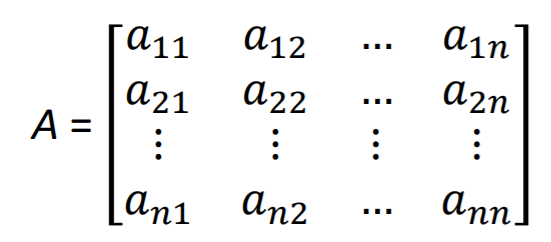
Mij = minor entri aij

= determinan upa-matriks (submatrix) yang elemen-elemennya tidak berada pada baris i dan kolom j

Cij = (–1)i+j Mij = kofaktor entri aij

Kofaktor Cij berkoresponden dengan minor entri Mij, hanya berbeda tanda (positif atau negatif, tergantung nilai i dan j)

Dengan menggunakan kofaktor, maka determinan matriks



dapat dihitung dengan salah satu dari persamaan berikut:

Secara baris:

det(A) = a11C11 + a12C12 + … + a1nC1n

det(A) = a21C21 + a22C22 + … + a2nC2n

⋮

det(A) = an1Cn1 + an2Cn2 + … + annCnn

Secara kolom:

det(A) = a11C11 + a21C21 + … + an1Cn1

det(A) = a12C12 + a22C22 + … + an2Cn2

⋮

det(A) = a1nC1n + a2nC2n + … + annCnn

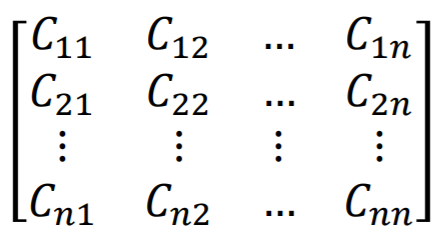
# Matriks Balikan

# Metode Matriks Identitas

Balikan (inverse) matriks A adalah A sedemikian sehingga AA-1 = A-1A = I. Metode eliminasi Gauss-Jordan (G-J) dapat digunakan untuk menghitung matriks balikan. Untuk matriks A yang berukuran n x n, matriks balikannya, yaitu A-1 dicari dengan cara berikut: [𝐴|𝐼] ~ [𝐼|A-1] dengan metode Gauss-Jordan.

# Metode Matriks Kofaktor

Misalkan A adalah matriks n x n dan Cij adalah kofaktor entri aij . Maka matriks kofaktor dari A adalah



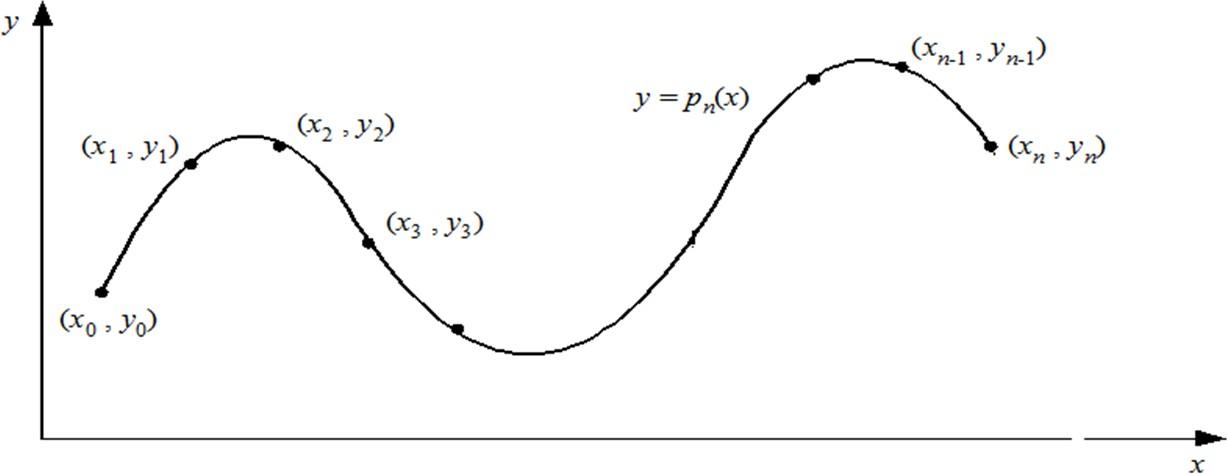
Adjoin dari A adalah transpose matriks kofaktor

Balikan matriks A dapat dihitung dengan menggunakan rumus

A-1 = (1/det(A)) x adj(A)

# Interpolasi Polinom

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan *n*+1 buah titik berbeda, (*x*0, *y*0), (*x*1, *y*1), ..., (*xn*, *yn*). Tentukan polinom *pn*(*x*) yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga *yi* = *pn*(*xi*) untuk *i* = 0, 1, 2, …, *n*.



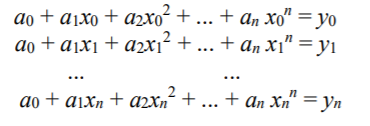
Setelah polinom interpolasi *pn*(*x*) ditemukan, *pn*(*x*) dapat digunakan untuk menghitung perkiraan nilai *y* di sembarang titik di dalam selang [*x*0, *xn*].

Polinom interpolasi derajat *n* yang menginterpolasi titik-titik (*x*0, *y*0), (*x*1, *y*1), ..., (*xn*, *yn*). adalah berbentuk *pn*(*x*) = *a*0 + *a*1*x* + *a*2*x*2 + … + *anxn*.

Jika hanya ada dua titik, (*x*0, *y*0) dan (*x*1, *y*1), maka polinom yang menginterpolasi kedua titik tersebut adalah *p*1(*x*) = *a*0 + *a*1*x* yaitu berupa persamaan garis lurus.

Jika tersedia tiga titik, (*x*0, *y*0), (*x*1, *y*1), dan (*x*2, *y*2), maka polinom yang menginterpolasi ketiga titik tersebut adalah *p*2(*x*) = *a*0 + *a*1*x* + *a*2*x*2 atau persaman kuadrat dan kurvanya berupa parabola.

Jika tersedia empat titik, (*x*0, *y*0), (*x*1, *y*1), (*x*2, *y*2), dan (*x*3, *y*3), polinom yang menginterpolasi keempat titik tersebut adalah *p*3(*x*) = *a*0 + *a*1*x* + *a*2*x*2 + *a*3*x*3, demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat *n* untuk *n* yang lebih tinggi asalkan tersedia (*n*+1) buah titik data. Dengan menyulihkan (*xi*, *yi*) ke dalam persamaan polinom *pn*(*x*) = *a*0 + *a*1*x* + *a*2*x*2 + … + *anxn* untuk *i* = 0, 1, 2, …, *n*, akan diperoleh *n* buah sistem persamaan lanjar dalam *a*0, *a*1, *a2*, …, *an*,



Solusi sistem persamaan lanjar ini, yaitu nilai *a*0, *a*1, …, *an*, diperoleh dengan menggunakan metode eliminasi Gauss yang sudah anda pelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada *x* = 9.2. Polinom kuadratik berbentuk *p*2(*x*) = *a*0 + *a*1*x* + *a*2*x*2. Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sistem persamaan lanjar yang terbentuk adalah

*a*0 + 8.0*a*1 + 64.00*a*2 = 2.0794 *a*0 + 9.0*a*1 + 81.00*a*2 = 2.1972 *a*0 + 9.5*a*1 + 90.25*a*2 = 2.2513

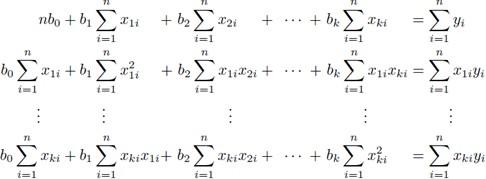
Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan *a*0 = 0.6762, *a*1 = 0.2266, dan *a*2 = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah *p*2(*x*) = 0.6762 + 0.2266*x* - 0.0064*x*2. Dengan menggunakan polinom ini, maka nilai fungsi pada *x* = 9.2 dapat ditaksir sebagai berikut: *p*2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)2 = 2.2192.

# Regresi Linier Berganda

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.



Untuk mendapatkan nilai dari setiap *βi* dapat digunakan *Normal Estimation Equation for Multiple Linear Regression* sebagai berikut:



Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

**Bab 3**

**Implementasi Program dalam Java**

Secara garis besar, program yang kami buat terdiri dari dua file berekstensi .java, yakni main.java dan matriks.java. Main.java merupakan file utama yang memilki class bernama main disertai algoritma program utama disertai beberapa fungsi dasar pengoperasian program. Sementara itu, matriks.java mengandung class bernama matriks berisi konstruktor tipe data matriks, dilanjutkan operasi-operasi yang memanfaatkan tipe data matriks tersebut.

Berikut ini daftar method yang kami gunakan:

1. Pada main.java

* static void SPLMenu()

Metode untuk mencetak menu untuk sistem persamaan linier.

* static void DETMenu()

Metode untuk mencetak menu metode determinan.

* static void BalikanMenu()

Metode untuk mencetak menu metode matriks balikan.

* static boolean SaveHasil()

Fungsi validasi apakah pengguna ingin melakukan save atau tidak.

* static void SaveFile(String hasil)

Metode untuk menyimpan hasil program ke dalam file .txt.

* public static String MatriksToString(matriks M1, int m, int n)

Fungsi untuk mengubah tipe data matriks menjadi string.

1. Pada matriks.java

* public matriks InputMatriksSQ(matriks M1)

Metode untuk melakukan input nilai pada setiap indeks efektif matriks M1.

* public matriks InputMatriks(matriks M1, int m, int n)

Metode untuk melakukan input nilai pada setiap indeks efektif matriks persegi M1.

* public matriks InputMatriksAug(matriks M1)

Metode untuk melakukan input ke dalam sebuah matriks augmented.

* public int GetLastIdxBrs(matriks M1)

Fungsi yang mengembalikan indeks baris terakhir sebuah matriks M1.

* public int GetLastIdxkol(matriks M1)

Fungsi yang mengembalikan indeks kolom terakhir sebuah matriks M1.

* float GetElmtDiagonal (matriks M1, int i){

Fungsi yang mengembalikan nilai elemen diagonal matriks M1 pada baris ke-i dan kolom ke-i.

* public void PrintMatriks(matriks M1)

Metode untuk mencetak matriks M1.

* public matriks CopyMatriks (matriks MIn){

Fungsi untuk membuat sebuah matriks MOut, yakni salinan dari matriks MIn.

* public matriks Transpose(matriks M1)

Metode untuk menghasilkan matriks transpose dari M1 (menukar seluruh baris menjadi kolom, dan sebaliknya).

* public matriks TukarBaris(matriks M3, int brs1, int brs2)

Fungsi yang akan mengembalikan matriks M3 setelah posisi brs1 dan brs2 di dalamnya ditukar.

* public boolean SearchOneKol(matriks M1, int idxbrs, int idxkol)

Fungsi yang mengembalikan nilai keberadaan satu utama dalam sebuah kolom.

* public int SearchOneIdxKol(matriks M1, int idxbrs, int idxkol){

Fungsi yang mengembalikan nilai posisi satu utama pada sebuah kolom.

* public boolean SearchNonZeroKol(matriks M1, int idxbrs1, int idxbrs2, int idxkol){

Fungsi yang bernilai true jika menemukan elemen tidak 0 dalam kolom idxkol.

* public int SearchNonZeroIdxKol(matriks M1, int idxbrs1, int idxbrs2, int idxkol){

Fungsi yang mengembalikan posisi indeks baris saat ditemukan elemen tidak 0 pada kolom idxkol.

* public matriks OperasiBaris(matriks M1, int idxbrs1, int idxbrs2, int idxkol)

Metode yang mengembalikan nilai operasi baris elementer matriks pada baris idxbrs2.

* public matriks MakeLeadOne(matriks M1, int idxbrs, int idxkol){

Metode untuk menghasilkan satu utama pada idxbrs.

* public matriks MatriksEselon(matriks M1){

Metode untuk menghasilkan matriks eselon.

* public float Determinan (matriks A){

Metode untuk menghasilkan nilai determinan dari matriks A.

* public matriks Kofaktor(matriks A, int bar, int kol)

Metode untuk menghasilkan matriks kofaktor pada baris bar dan kolom kol.

* public matriks MinorKofaktor(matriks A)

Metode untuk menghasilkan matriks kofaktor berdasarkan nilai minor setiap elemen matriks A.

* public matriks MatriksKolomKeI(int kolom){

Metode untuk mengganti nilai matriks pada kolom ke-”kolom”.

* public void Cramer() {

Metode kaidah cramer untuk menghasilkan nilai determinan.

* public matriks matriksEselonReduksi(matriks M1){

Metode untuk menghasilkan matriks eselon baris tereduksi.

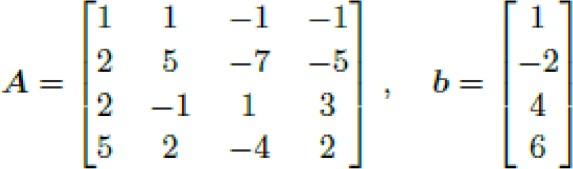
**Bab 4**

**Eksperimen**

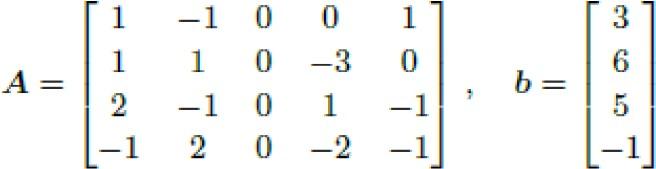
# STUDI KASUS

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

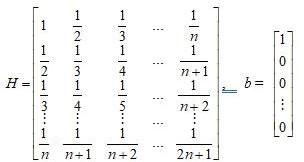
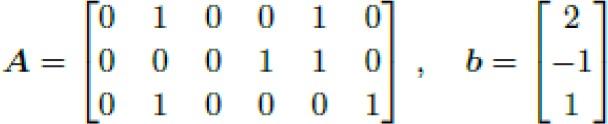
1. Temukan solusi SPL *Ax* = *b*,berikut: a.



b.



c.

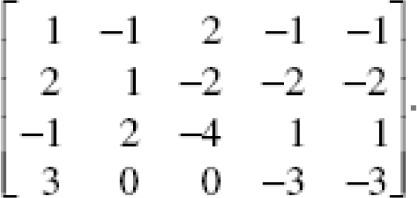
d.

*H adalah matriks Hilbert. Cobakan untuk n = 6 d*

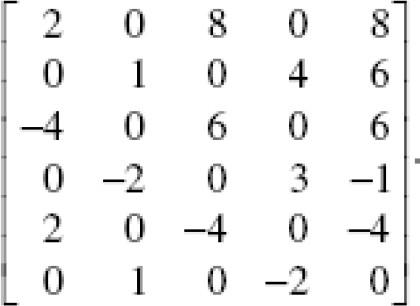
*an n = 10*.

H adalah matriks Hilbert. Cobakan untuk n = 6 dan n = 10

1. SPL berbentuk matriks *augmented*

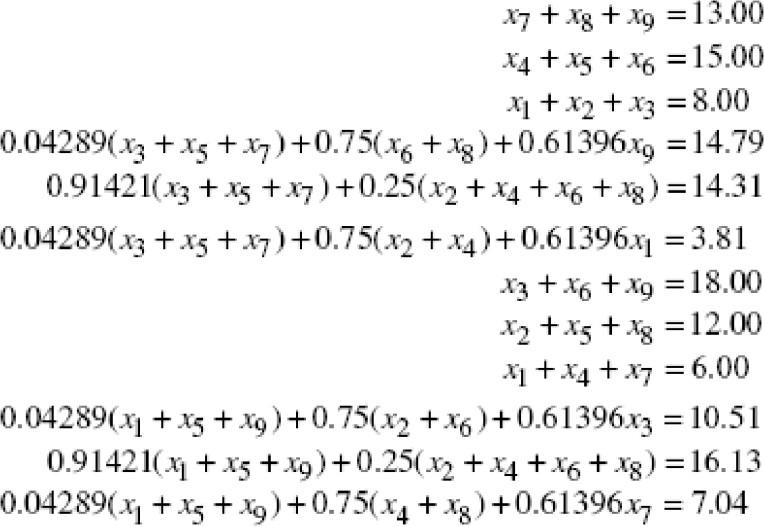
a.

b.



1. SPL berbentuk

a. 8*x*1 + *x*2 + 3*x*3 + 2*x*4 = 0 2*x*1 + 9*x*2 - *x*3 - 2*x*4 = 1 *x*1 + 3*x*2 + 2*x*3 - *x*4 = 2 *x*1 + 6*x*3 + 4*x*4 = 3

b.

# Studi Kasus: Interpolasi

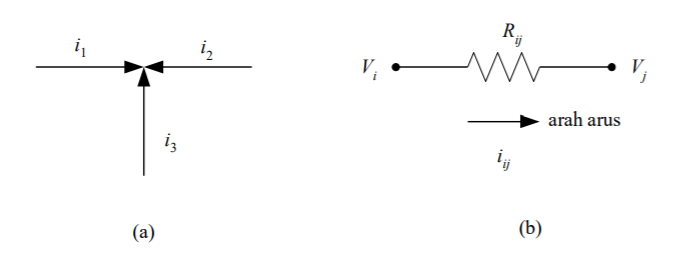
Seperti yang kalian telah pelajari di Mata kuliah Pengantar Analisis Rangkaian, dalam sebuah rangkaian listrik berlaku hukum-hukum arus Kirchoff menyatakan bahwa jumlah aljabar dari semua arus yang memasuki suatu simpul (Gambar 4.4a) haruslah nol:

Σ *i* = 0

Dalam hal ini, semua arus *i* yang memasuki simpul dianggap bertanda positif. Sedangkan hukum Ohm (Gambar 1) menyatakan bahwa arus *i* yang melalui suatu tahanan adalah :

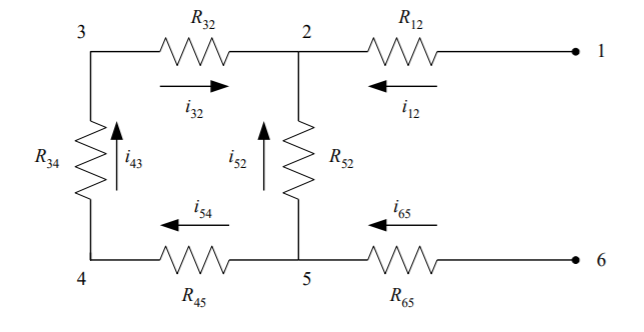
iij = **(𝑽i−𝑽j ) / Rij**

yang dalam hal ini *V* adalah tegangan dan *R* adalah tahanan.



**Gambar 1** (a) Hukum Kirchoff, (b) hukum Ohm

Diberikan sebuah rangkaian listrik dengan 6 buah tahanan seperti pada Gambar 2 Anda diminta menghitung arus pada masing-masing rangkaian.

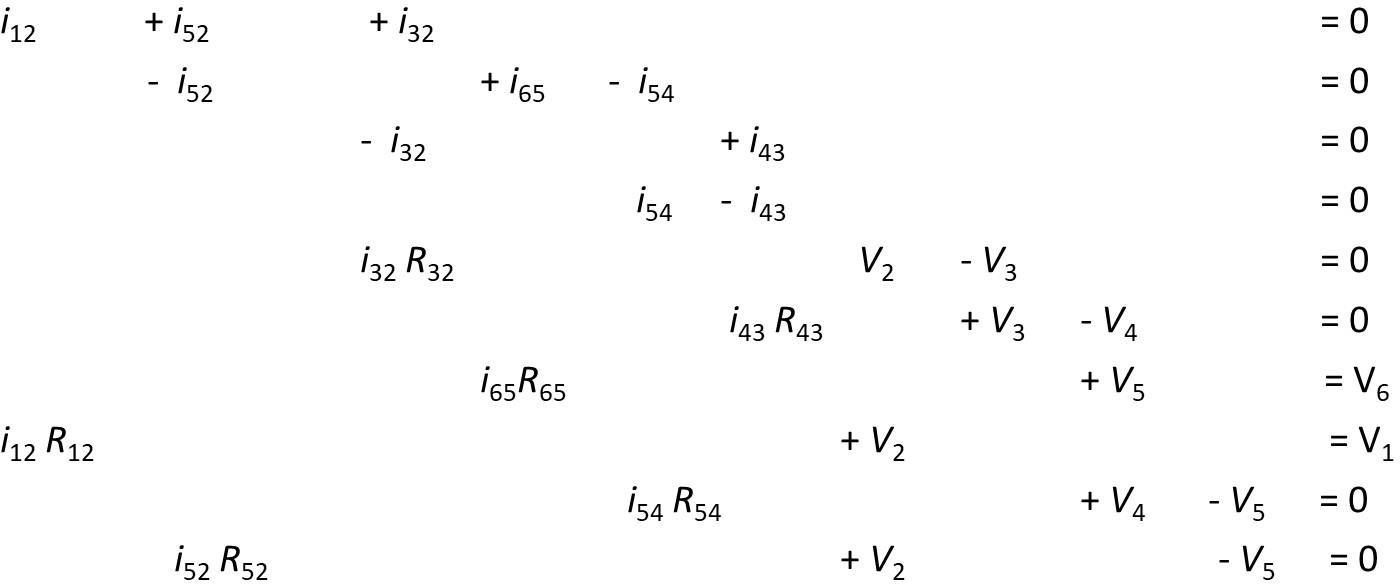


**Gambar 2** Rangkaian listrik dengan 6 buah tahanan

Arah arus dimisalkan seperti diatas. Dengan hukum Kirchoff diperoleh persamaan- persamaan berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  |  |
|  | |  |  |  |
| i12+ i52 + i32 = 0  i65 - i52 - i54 = 0  i43 - i32 = 0  i54 - i43 = 0 | |  |  |  |
| Dari hukum Ohm didapat : |  |  |  |  |
|  | *i*32 *R*32 | - *V*3 | + *V*2 | = 0 |
|  | *i*43 *R*43 *i*65 *R*65 | - *V*4 | + *V*3  + *V*5 | = 0  = *V*6 |
|  | *i*12 *R*12 |  | + *V*2 | = *V*1 |
|  | *i*54 *R*54 | - *V*5 | + *V*4 | = 0 |
|  | *i*52 *R*52 | - *V*5 | + *V*2 | = 0 |

Dengan menyusun kesepuluh persamaan diatas didapatkan sistem persamaan linier sebagai berikut :



Tentukan nilai dari:

*i*12 , *i*52 , *i*32 , *i*65 , *i*54 , *i*13 , *V*2 , *V*3 , *V*4 , *V*5

bila diketahui

*R*12 = 5 ohm , *R*52 = 10 ohm , *R*32 = 10 ohm

*R*65 = 20 ohm , *R*54 = 15 ohm , *R*14 = 5 ohm.

*V*1 = 200 volt , *V*6 = 0 volt.

1. **(Interpolasi)** 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 default berikut:

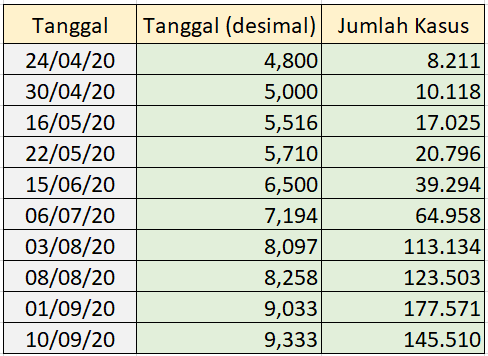
*x* = 0.2 f(x) = ?

*x* = 0.55 f(x) = ?

*x* = 0.85 f(x) = ?

*x* = 1.28 f(x) = ?

1. Jumlah kasus positif Covid-19 di Indonesia semakin bertambah dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus Covid-19 di Indonesia mulai dari tanggal 24 April 2020 hingga 10 September 2020:



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



Sebagai **contoh**, untuk tanggal 24/04/20 (dibaca: 24 April 2020) diperoleh tanggal(desimal) sebagai berikut:

Tanggal(desimal) = 4 + (24/30) = 4,800

Gunakanlah data di atas dengan memanfaatkan **polinom interpolasi** untuk melakukan prediksi jumlah kasus Covid-19 pada tanggal-tanggal berikut:

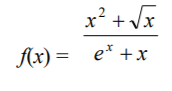
a. 25/05/20

b. 30/08/20

c. 15/09/20

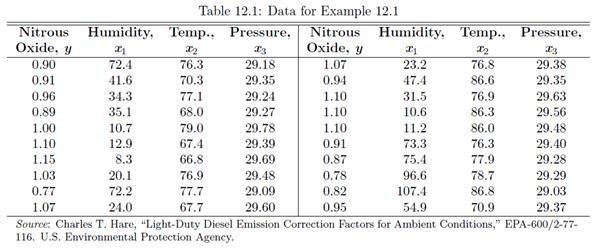
d. beserta masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2020.

1. Sederhanakan fungsi



dengan polinom interpolasi derajat *n* di dalam selang [0, 2]. Sebagai contoh, jika *n* = 5, maka titik-titik *x* yang diambil di dalam selang [0, 2] berjarak *h* = (2 – 0)/5 = 0.4.

1. Diberikan sekumpulan data sesuai pada tabel berikut ini.



Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.

20b0 + 863.1b1 + 1530.4b2 + 587.84b3 = 19.42

863.1b0 + 54876.89b1 + 67000.09b2 + 25283.395b3 = 779.477

1530.4b0 + 67000.09b1 + 117912.32b2 + 44976.867b3 = 1483.437

587.84b0 + 25283.395b1 + 44976.867b2 + 17278.5086b3 = 571.1219

**Bab 5**

**Kesimpulan, Saran, dan Refleksi**

Dari tugas besar Aljabar Linier dan Geometri, kami menjadi mengerti materi Sistem Persamaan Linier, Determinan, dan Aplikasinya, serta implementasi materi dalam program Java. Materi yang dicakup dalam Sistem Persamaaan Linier yaitu Metode eliminasi Gauss, Metode eliminasi Gauss-Jordan, Metode matriks balikan, dan Kaidah Cramer. Lalu untuk Determinan telah dipahami Metode Reduksi Baris dan Metode Ekspansi Kofaktor. Lalu untuk mencari Matriks Balikan kami memahami materi Metode Matriks Identitas dan Metode Matriks Kofaktor. Selain tiga materi tadi kami juga lebih paham mencari solusi dari permasalahan Interpolasi Polinom dan Regresi Linier Berganda. Sedikit saran untuk metode yang memakai eselon baris, mungkin dalam membentuk matriks eselon dapat ditambahkan fitur search elemen 1 lalu lakukan swap baris supaya hal ini mempermudah fungsi implementasinya. Hal yang dapat dipelajari dari mekanisme tugas besar ini adalah pelajari bagaimana cara Git Hub bekerja mulai dari fungsi dasar seperti git clone, pull, push, add, commit dll. Selain itu, pelajaran yang dapat diambil dari tugas besar ini adalah pelajari kembali bagaimana menginstall, meng-*compile,* dan menggunakan *syntax – syntax* dalam Java.

**Daftar Referensi**

* <https://stackoverflow.com/>
* <https://www.geeksforgeeks.org/java/?ref=leftbar>
* <https://beginnersbook.com/java-tutorial-for-beginners-with-examples/>
* <https://www.profematika.com/>
* http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/