**SISTEM PERSAMAAN LINIER, DETERMINAN,**

**DAN APLIKASINYA**

**LAPORAN TUGAS BESAR**

Diajukan Untuk Memenuhi Tugas Aljabar Linier dan Geometri

oleh

**MICHAEL HANS 13518056**

**RAFAEL SEAN PUTRA 13518119**

**LIONNARTA SAVIRANDY 13518128**

****

**TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2019**

**BAB I**

**DESKRIPSI MASALAH**

Sistem persamaan linier (SPL) 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

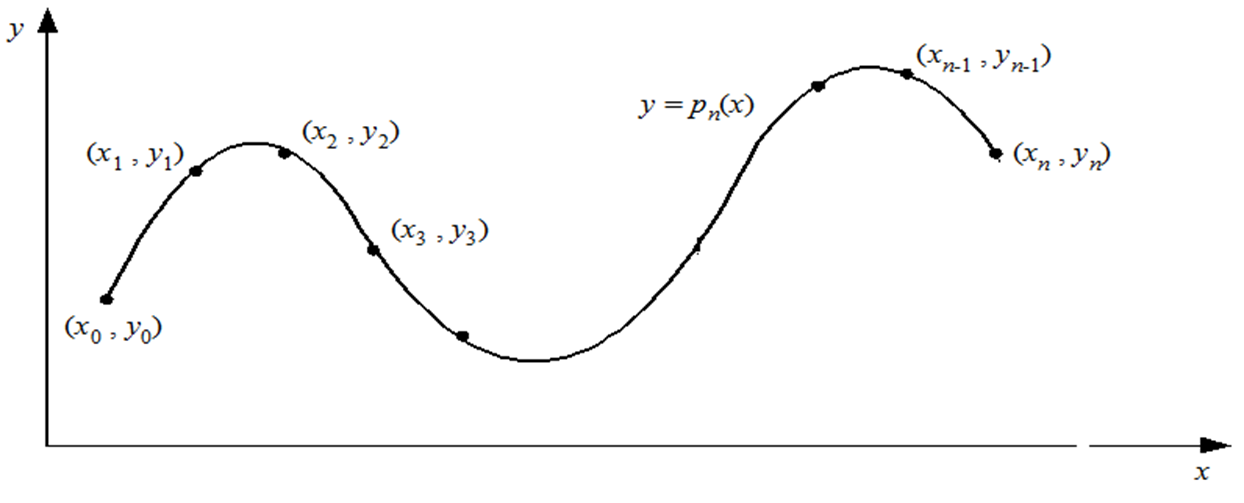
: :

: :

*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, dan metode Cramer. Solusi sebuah SPL mungkin tidak ada, banyak, atau hanya satu (unik/tunggal).

Sistem persamaan linier memiliki banyak aplikasi dalam bidang sains dan rekayasa, salah satunya adalah mengestimasi nilai fungsi dengan 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 menginterplolasi 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*,

*a*0 + *a*1*x*0 + *a*2*x*02 + ... + *an x*0*n* = *y*0

*a*0 + *a*1*x*1 + *a*2*x*12 + ... + *an x*1*n* = *y*1

... ...

*a*0 + *a*1*xn* + *a*2*xn*2 + ... + *an xnn* = *yn*

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 sisten 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 persamaandengan metode eliminasi Gauss menghasilkan *a*0 = 0.6762, *a*1 = 0.2266, dan *a*2 = -0.0064. Polinom interpolasi yang melaluiketiga 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.

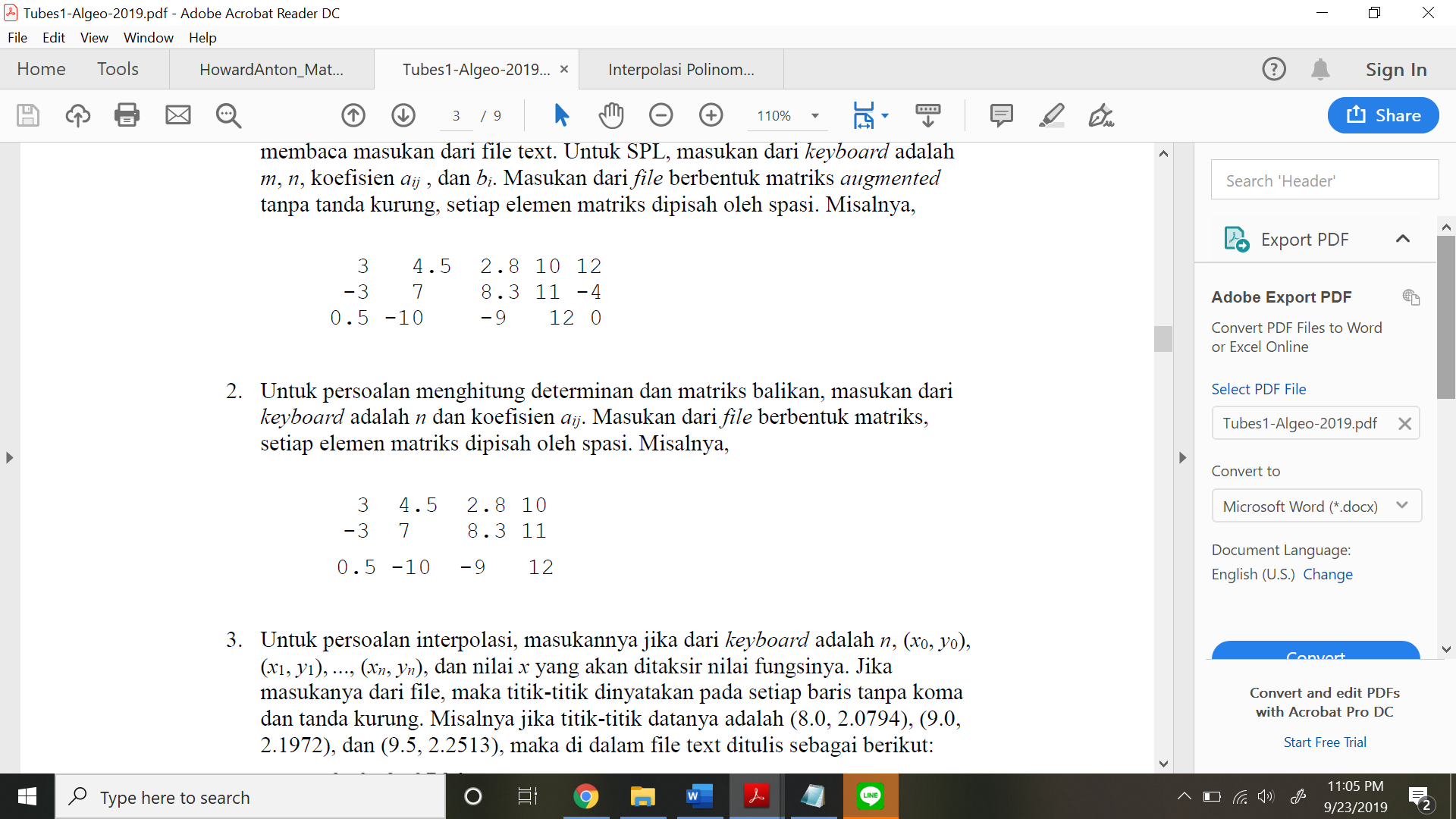
**SPESIFIKASI TUGAS**

Buatlah program dalam Bahasa Java untuk

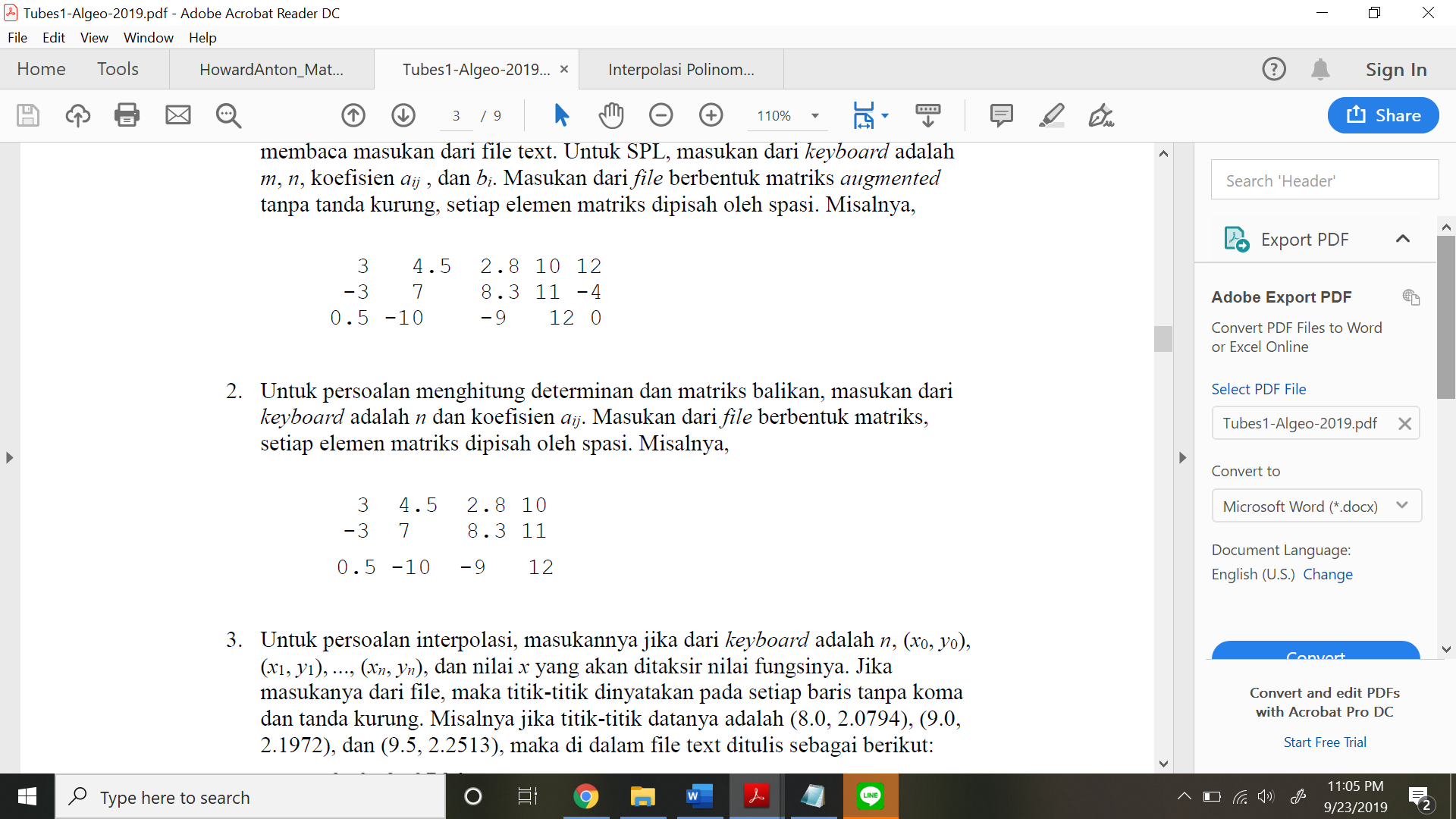
1. Menghitung solusi SPL dengan metode eliminasi metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan *n* pebuah dan *n* persamaan).
2. Menyelesaikan persoalan interpolasi.
3. Menghitung determinan matriks dengan berbagai cara yang disebutkan di atas, matriks kofaktor, dan matriks *adjoin* dari sebuah matriks n x n.

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,



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,



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 masukanya 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 2.07944

9.0 2.1972

9.5 2.2513

Untuk 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*.)

1. Untuk persoalan determinan, matriks balikan, matriks kofator, dan adjoin, maka luarannya sesuai dengan persoalan masing-masing
2. Untuk persoalan polinom interpolasi, luarannya adalah persamaan polinom dan taksiran nilai fungsi pada *x* yang diberikan.
3. Luaran program harus dapat ditampilkan **pada layar komputer dan dapat disimpan ke dalam file**.
4. Bahasa program yang digunakan adalah Java.
5. Program **tidak harus** berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas *Eclipse* misalnya).
6. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan ditrancang masing-masing. Misalnya, menu:

MENU

1. Sistem Persamaaan Linier

2. Determinan

3. Matriks balikan

4. Matriks kofaktor

5. Adjoin

6. Interpolasi Polinom

7. 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.

1. Sebagai pembanding, bandingkan solusi program anda dengan hasil dari Wolfram Alpha.

**STUDI KASUS**

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

1. SPL berbentuk

*x*1 + 3*x*2 - 2*x*3 + 2*x*5 = 0

2*x*1 + 6*x*2– 5*x*3 - 2*x*4 + 4*x*4 - 3*x*6 = -1

5*x*3 + 10*x*4 + 15*x*6 = 5

2*x*1 + 6*x*2 + 8*x*4 + 4*x*4 + 18*x*6 = 6

1. SPL berbentuk matriks *augmented*



1. SPL berbentuk matriks *augmented*



1. SPL berbentuk



1. Untuk persoalan determinan, matrisk balikan, dan matriks kofaktor, silakan cari masing-masing dua buah matriks yang berukuran 5 x 5 dan 10 x 10.
2. Misalkan seorang insinyur Teknik Sipil merancang sebuah rangka statis yang berbentuk segitiga (Gambar 1). Ujung segitiga yang bersudut 30° bertumpu pada sebuah penyangga statis, sedangkan ujung segitiga yang lain bertumpu pada penyangga beroda.

Rangka mendapat gaya eksternal sebesar 1000 pon. Gaya ini disebar ke seluruh bagian rangka. Gaya *F* menyatakan tegangan atau kompresi pada anggota rangka. Reaksi eksternal (*H*2, *V*2, dan *V*3) adalah gaya yang mencirikan bagaimana rangka berinteraksi dengan permukaan pendukung. Engsel pada simpul 2 dapat menjangkitkan gaya mendatar dan tegak pada permukaan, sedangkan gelinding pada simpul 3 hanya menjangkitkan gaya tegak.



**Gambar 1** Gaya-gaya pada rangka statis tertentu

Struktur jenis ini dapat diuraikan sebagai sistem persamaan aljabar lanjar simultan. Diagram gaya-benda-bebas diperlihatkan untuk tiap simpul dalam Gambar 2.

*F*1, *v*

*F*1, *h*

•

*F*1

1

60°

30°

*F*3

*F*3

*F*2

*F*3, *v*

*F*3, *h*

3

•

60°

*V*3

•

*F*2

*F*1

*F*2, *v*

*H*2

2

30°

*V*2

**Gambar 2** Diagram gaya-benda-bebas untuk simpul-simpul rangka statis

Menurut hukum Newton, resultan gaya dalam arah mendatar maupun tegak harus nol pada tiap simpul, karena sistem dalam keadaan diam (statis). Oleh karena itu, untuk simpul 1,

∑ *FH* = 0 = -*F*1 cos 30° + *F*3 cos 60° + *F*1, *h*

∑ *FV* = 0 = -*F*1 sin 30° - *F*3 sin 60° + *F*1, *v*

untuk simpul 2,

∑ *FH* = 0 = *F*2 + *F*1 cos 30° + *F*2, *h* + *H*2

∑ *FV* = 0 = *F*1 sin 30° - *F*2, *v* + *V*2

dan untuk simpul 3,

∑ *FH* = 0 = -*F*2 - *F*3 cos 60° + *F*3, *h*

∑ *FV* = 0 = *F*3 sin 60° + *F*3, *v* + *V*3

Gaya 1000 pon ke bawah pada simpul 1 berpadanan dengan *F*1, *v*= -1000, sedangkan semua *Fi*, *v* dan *Fi*, *h* lainnya adalah nol. Persoalan rangka statis ini dapat dituliskan sebagai sistem yang disusun oleh enam persamaan lanjar dengan 6 peubah yang tidak diketahui:

∑ *FH* = 0 = -*F*1 cos 30° + *F*3 cos 60° + *F*1, *h* = -0.866*F*1 + 0.5 *F*3

∑ *FV* = 0 = -*F*1 sin 30° - *F*3 sin 60° + *F*1, *v* = -0.5*F*1 – 0.866 *F*3 + 1000

∑ *FH* = 0 = *F*2 + *F*1 cos 30° + *F*2, h + *H*2 = *F*2 + 0.866*F*1 + 0 + *H*2

∑ *FV* = 0 = *F*1 sin 30° - *F*2, *v* + *V*2 = 0.5 *F*1 + *V*2

∑ *FH* = 0 = -*F*2 - *F*3 cos 60° + *F*3, *h* = -*F*2 – 0.5 *F*3

∑ *FV* = 0 = *F*3 sin 60° + *F*3, *v* + *V*3 = 0.866 *F*3 + *V*3

Keenam persamaan di atas ditulis ulang kembali dalam susunan yang teratur berdasarkan urutan peubah *F*1, *F*2, *F*3, *H*2, *V*2, *V*3:

-0.866*F*1 + 0.5 *F*3 = 0

-0.5*F*1 – 0.866 *F*3 = -1000

-0.866*F*1 – *F*2 – *H*2 = 0

-0.5 *F*1 – *V*2 = 0

– *F*2 – 0.5 *F*3 = 0

–0.866 *F*3 – *V*3 = 0

Tentukan solusi sistem di atas!

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. **(Interpolasi)** Jumlah penduduk Jawa Barat dari tahun 1971 hingga 2010 (dibulatkan ke juta) adalah sebagai berikut:

|  |  |
| --- | --- |
| Tahun | Jumlah x 106) |
| 1971 | 21,6 |
| 1980 | 27,4 |
| 1990 | 35,4 |
| 1995 | 39,2 |
| 2000 | 35,7 |
| 2010 | 43,2 |
| 2015 | 46,7 |
| 2019 | 49.1 |

Berdasarkan data tersebut prediksilah jumlah penduduk Jawa Barat pada tahun 1975, 1983, 1992, 2005, 2012 (atau nilai lain sesuai masukan user) dengan menggunakan polinom interpolasi.

1. Sederhanakan fungsi y = 1/(√(x2 + 2) dengan polinom interpolasi derajat *n* di dalam selang [1, 5]. Sebagai contoh, jika *n* = 5, maka titik-titik *x* yang diambil di dalam selang [1, 5] berjarak *h* = (5 – 1)/5.

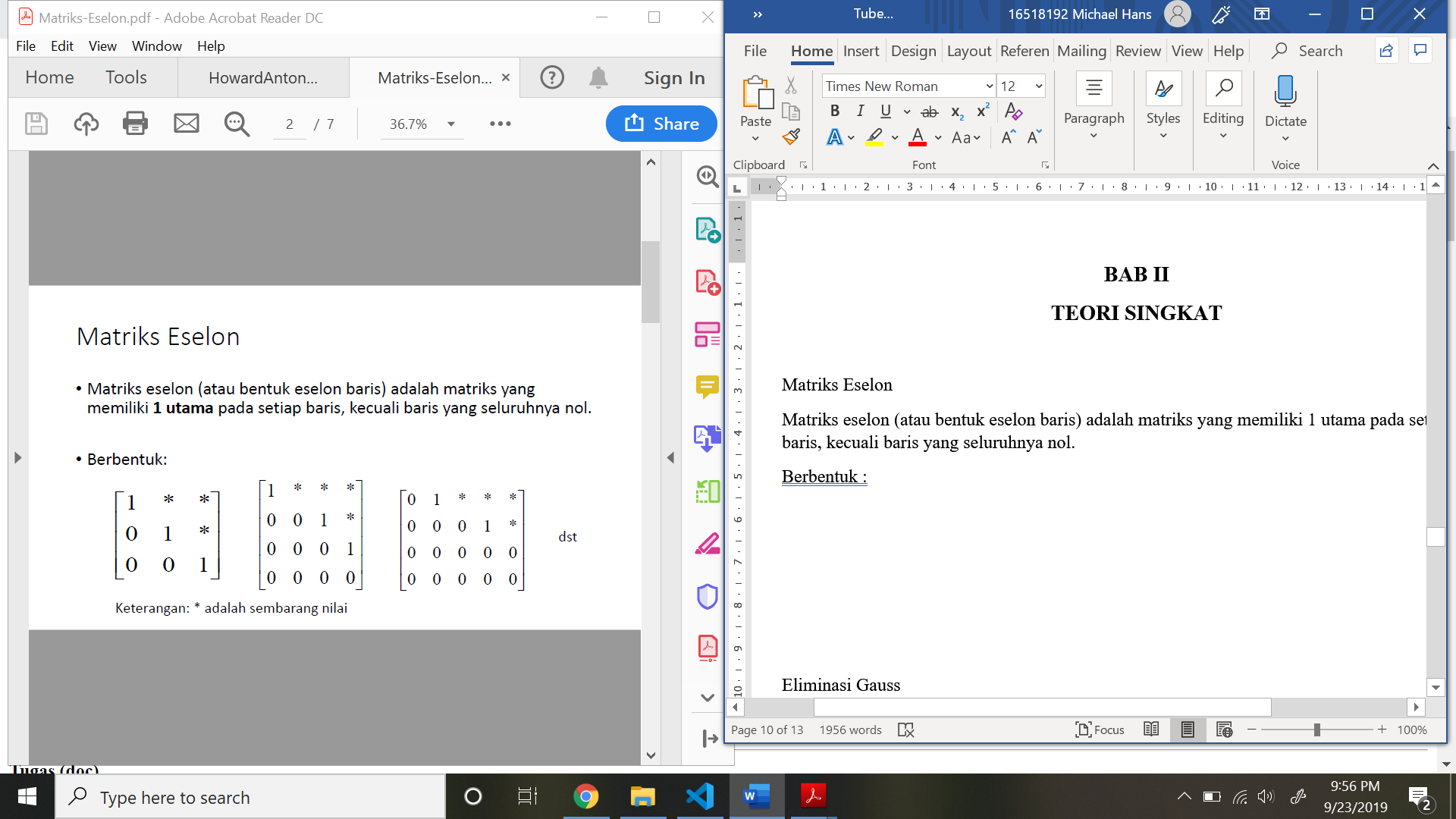
**BAB II**

**TEORI SINGKAT**

**2.1 Matriks Eselon**

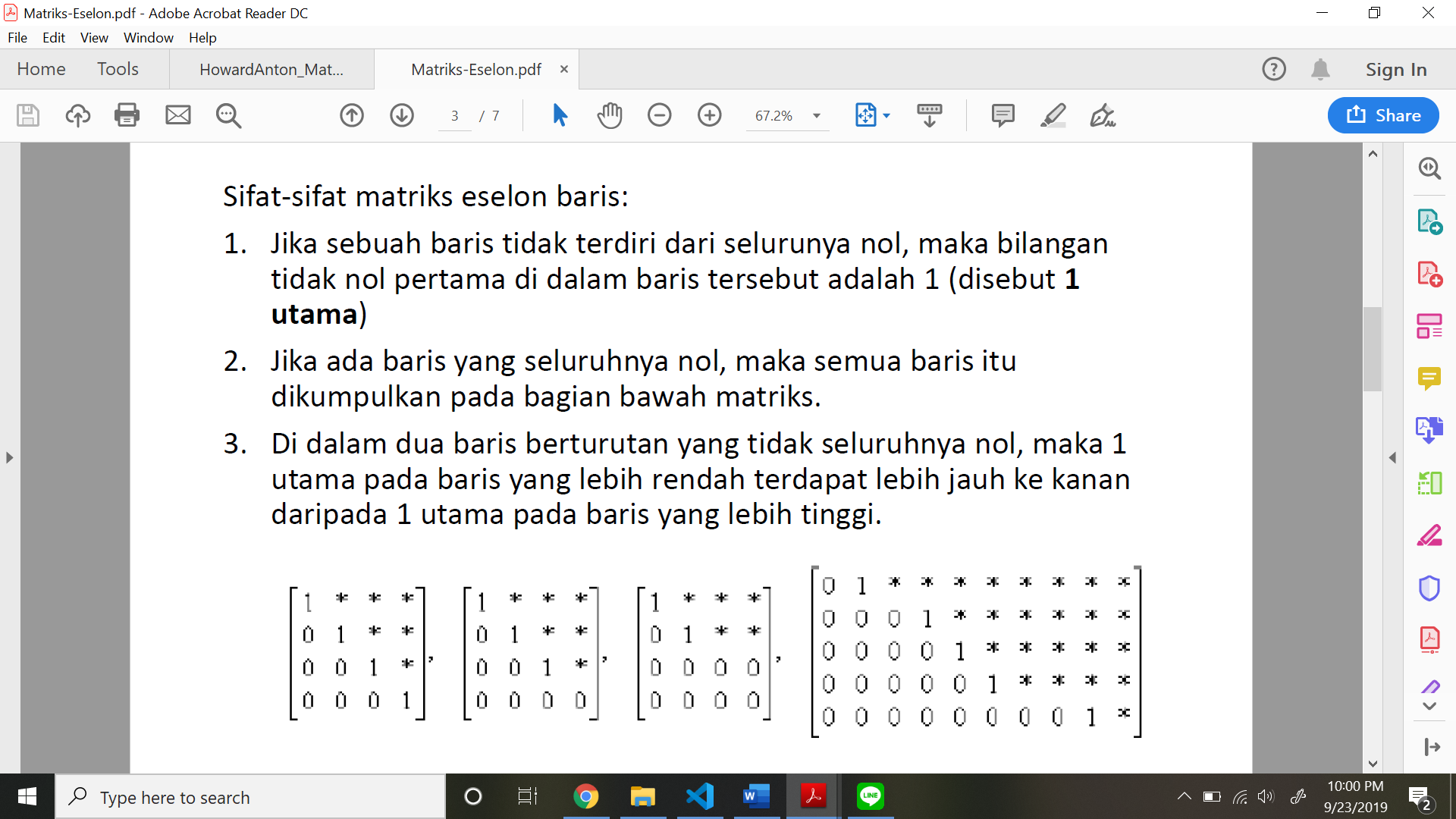
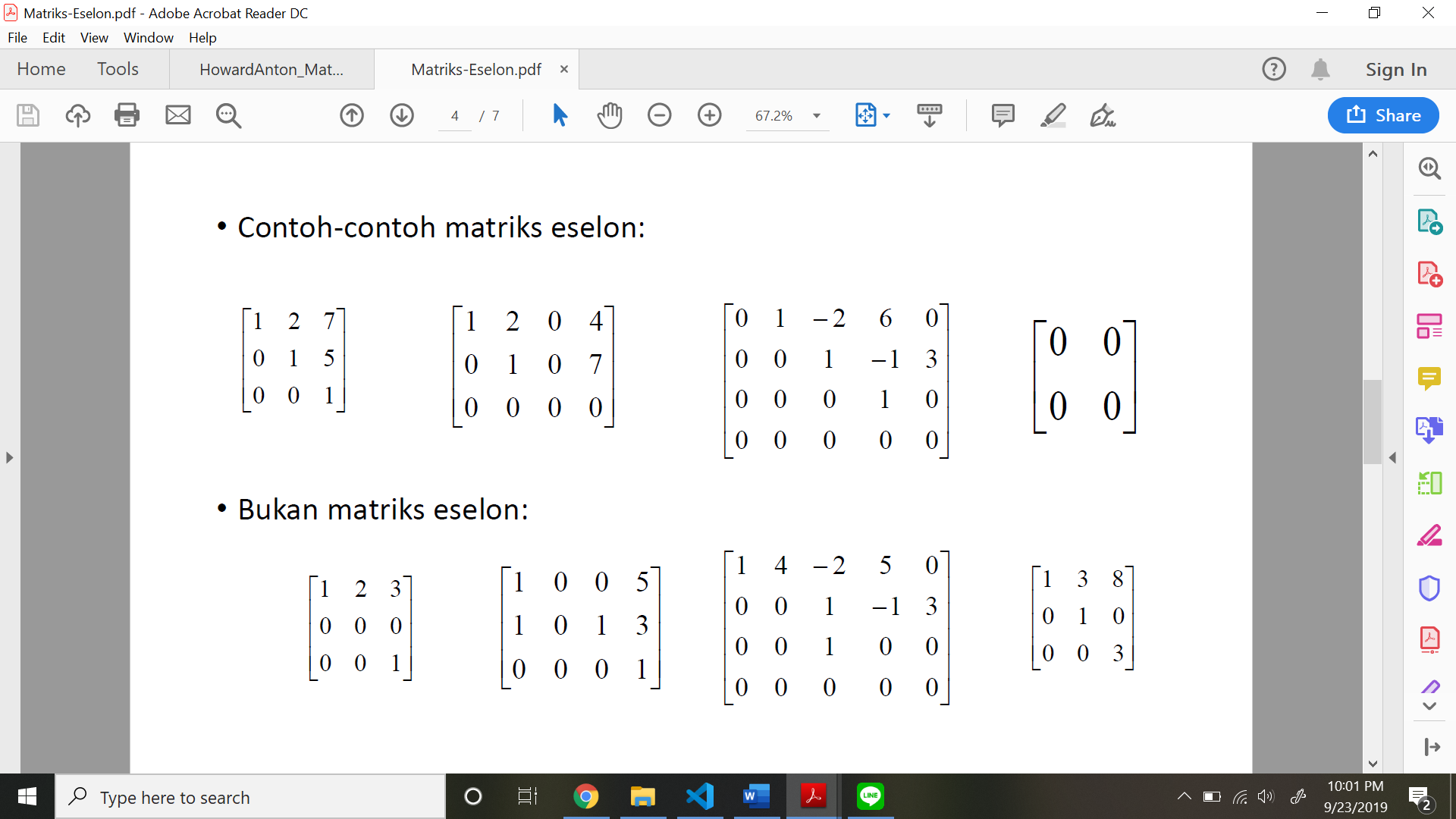
Matriks eselon (atau bentuk eselon baris) adalah matriks yang memiliki 1 utama pada setiap baris, kecuali baris yang seluruhnya nol.

Berbentuk :



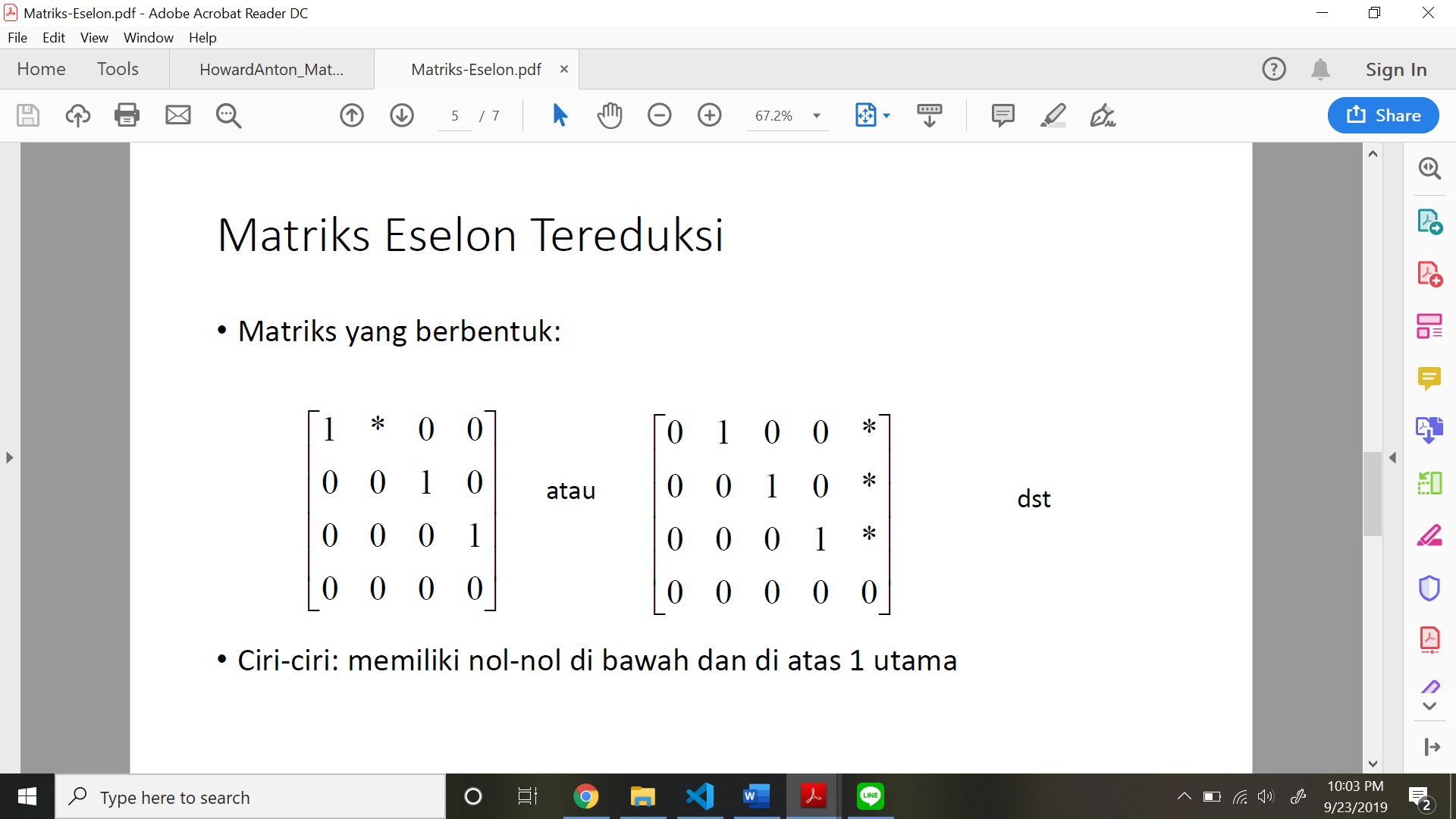
Sifat-sifat matriks eselon baris:

1. Jika sebuah baris tidak terdiri dari seluruhnya nol, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 (disebut 1 utama)
2. Jika ada baris yang seluruhnya nol, maka semua baris itu dikumpulkan pada bagian bawah matriks.
3. Di dalam dua baris berurutan yang tidak seluruhnya nol, maka 1 utama pada baris yang lebih rendah terdapat lebih jauh ke kanan daripada 1 utama pada baris yang lebih tinggi.

Contoh matriks eselon baris:

**2.2. Matriks Eselon Tereduksi**

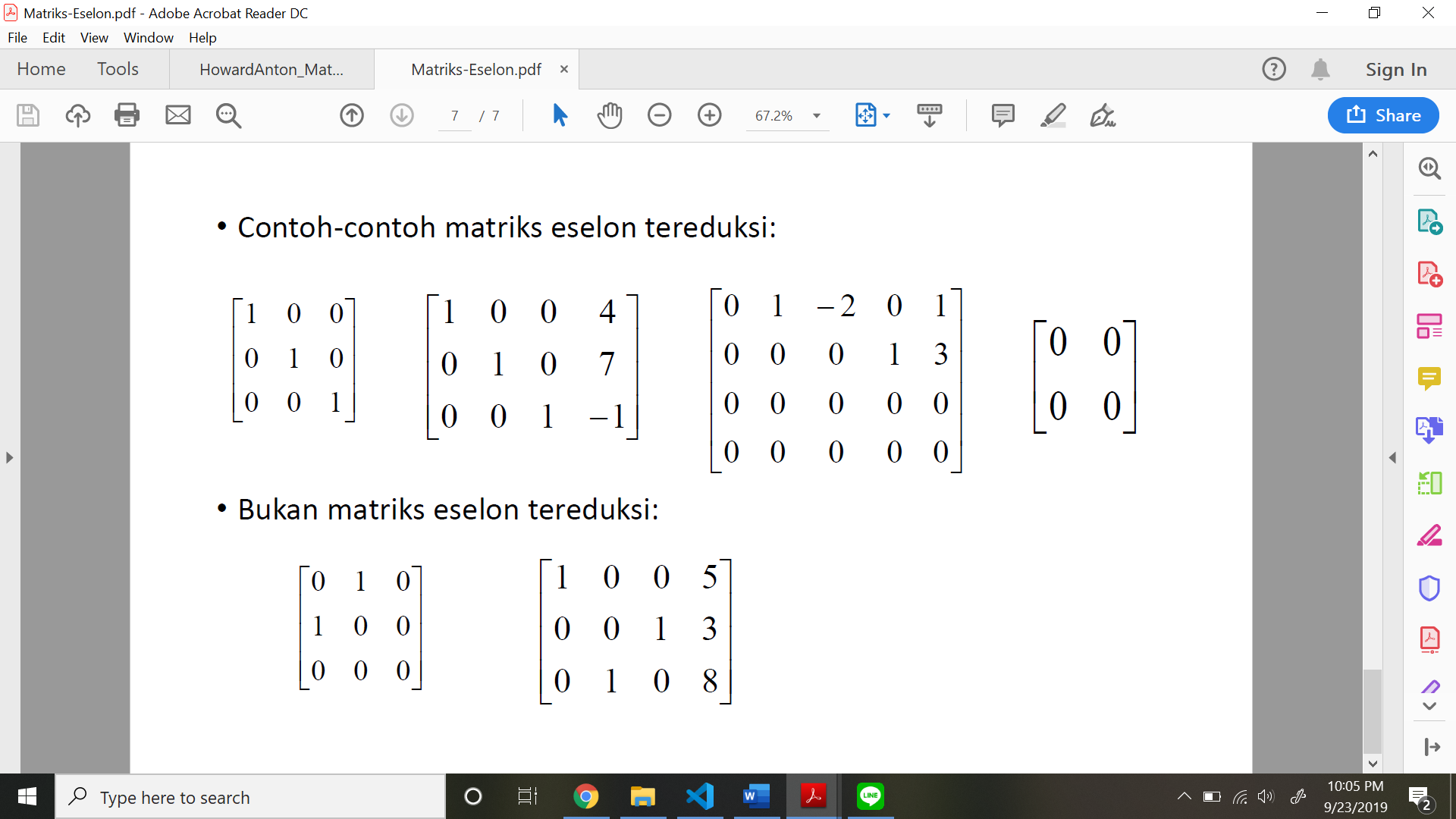
Matriks eselon tereduksi adalah matriks yang memiliki nol-nol di bawah dan di atas 1 utama.



Sifat-sifat matriks eselon baris tereduksi:

1. Jika sebuah baris tidak terdiri dari seluruhnya nol, maka bilangan tidak nol pertama di dalam baris tersebut adalah 1 (disebut 1 utama)
2. Jika ada baris yang seluruhnya nol, maka semua baris itu dikumpulkan pada bagian bawah matriks.
3. Di dalam dua baris berurutan yang tidak seluruhnya nol, maka 1 utama pada baris yang lebih rendah terdapat lebih jauh ke kanan daripada 1 utama pada baris yang lebih tinggi.
4. Setiap kolom yang memiliki 1 utama memiliki nol di tempat lain.

Contoh matriks eselon baris tereduksi :



**2.3. Eliminasi Gauss**

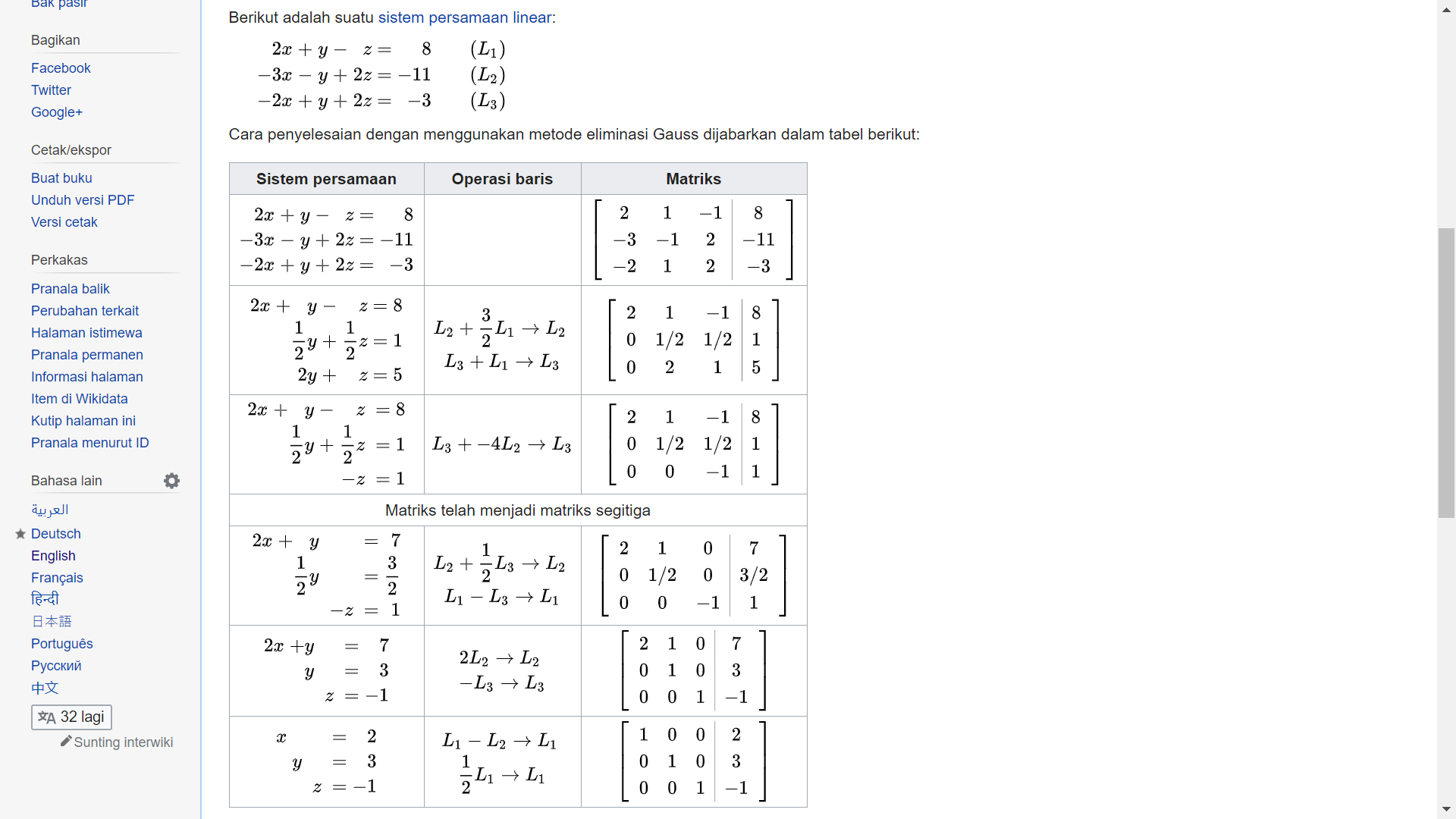
Eliminasi Gauss adalah algoritma yang digunakan untuk menyelesaikan sistem persamaan linear. Metode ini dinamai dari matematikawan Carl Friedrich Gauss (1777–1855), walaupun metode ini sudah dikenal oleh matematikawan Tionghoa semenjak tahun 179 M.

Terdapat tiga jenis operasi yang dapat dilakukan dalam metode ini:

* Mengganti urutan dua baris
* Mengalikan baris dengan angka yang bukan nol
* Menambah suatu baris dengan baris yang lainnya

Dengan cara ini, matriks dapat diubah menjadi matriks segitiga atas.

Contoh Penggunaan Eliminasi Gauss dalam Menyelesaikan Persamaan Linear :



**2.4 Eliminasi Gauss-Jordan**

Salah satu metode yang dapat digunakan untuk menyelesaikan sistem persamaan linier adalah metode eliminasi Gauss-Jordan. Metode ini diberi nama Gauss-Jordan untuk menghormati Carl Friedrich Gauss dan Wilhelm Jordan. Metode ini sebenarnya adalah modifikasi dari metode eliminasi Gauss, yang dijelaskan oleh Jordan di tahun 1887.

Metode Gauss-Jordan ini menghasilkan matriks dengan bentuk baris eselon yang tereduksi (reduced row echelon form), sementara eliminasi Gauss hanya menghasilkan matriks sampai pada bentuk baris eselon (row echelon form). Metode ini digunakan untuk mencari invers dari sebuah matriks.

Prosedur umum untuk metode eliminasi Gauss-Jordan ini adalah

1. Ubah sistem persamaan linier yang ingin dihitung menjadi matriks augmentasi.

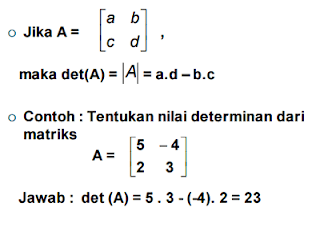
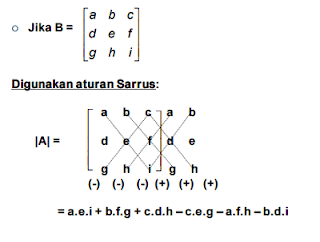
2. Lakukan operasi baris elementer pada matriks augmentasi (A|b) untuk mengubah matriks

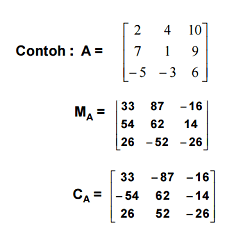
A menjadi dalam bentuk baris eselon yang tereduksi

**2.5 Determinan**

Determinan adalah suatu bilangan real yang diperoleh dari suatu proses dengan aturan tertentu terhadap matriks bujur sangkar. Determinan dinyatakan sebagai jumlah semua hasil kali dasar bertanda dari matriks bujur sangkar A. Determinan dari sebuah matriks bujur sangkar A, dinotasikan dengan det(A), atau |A|.

Berikut ini adalah cara-cara untuk menentukan determinan dari suatu matriks berukuran N x N.

1. [](https://2.bp.blogspot.com/-0xiHds8ygRU/WF-WDFvA2NI/AAAAAAAABGY/6GBbbndn2gov6RIEZ7a5gR_lZZviXnhrQCLcB/s1600/Determinan%2BMatrix%2BOrdo%2B2x2.png)Menentukan nilai determinan matriks berordo 2 x 2
2. [](https://1.bp.blogspot.com/-uMh7MdnXuTU/WF-WD2iEb7I/AAAAAAAABGc/RCb2mflY5mY1rAgOIKizx14ak9Zgv8ALQCLcB/s1600/Determinan%2BMatrix%2BOrdo%2B3x3.png)Menentukan nilai determinan matriks berordo 3 x 3 dengan Aturan Sarrus
3. Menentukan determinan matriks n x n dengan matriks Kofaktor
4. Minor dari suatu matr[i](http://sosmedpc.blogspot.com/2016/12/determinan-pengertian-dan-contoh.html)ks bujur sangkar A adalah harga determinan sub matriks yang tetap, setelah menghilangkan baris ke i dan kolom ke j. Minor dari baris ke i dan kolom ke j, dinotasikan dengan Mij.
5. Kofaktor dari suatu matriks bujur sangkar dilambangkan dengan cij, yaitu cij = (-1)i+jMij

[](https://2.bp.blogspot.com/-lUxMnLxlLis/WF-WAyiUjRI/AAAAAAAABF8/BGZWD6ZA6FwtUCqFpsohY0WHBVazK__MwCLcB/s1600/Contoh%2BDeterminan%2BMatrix%2Bn%2Bx%2Bn.png)

Ada 2 cara, yaitu :

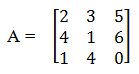
1. Ekspansi Kofaktor sepanjang baris ke i : det(A) = ai1ci1 + ai2ci2 + … + aincin
2. Ekspansi Kofaktor sepanjang kolom ke j : det(A) = a1jc1j + a2jc2j + … + anjcnj
3. Menentukan determinan matriks n x n dengan Transformasi Baris Elementer (TBE)
4. Menukarkan dua baris Notasi = bij Arti = menukarkan baris ke-[i](http://sosmedpc.blogspot.com/2016/12/determinan-pengertian-dan-contoh.html) dgn baris ke-j
5. Mengalikan suatu bari dengan skalar k, k ≠ 0 Notasi = k.bi Arti = mengalikan setiap elemen dari baris ke- i, dengan skalar k, k ≠ 0
6. Menambahkan baris ke- i dengan k kali baris ke- j (k ≠ 0) Notasi= bij(k) Arti = bi + k bj(Perubahan terjadi pada bi).
7. Menentukan Determinan Matriks dengan TBE Langkah :
8. Dengan menggunakan TBE, ubahlah matriks yang ada, menjadi Matriks Segitiga Atas / Bawah.
9. Harga determinannya adalah perkalian anta[r](http://sosmedpc.blogspot.com/2016/12/determinan-pengertian-dan-contoh.html) elemen–elemen pada diagonal utamanya.

**2.6 Minor**

Untuk mencari nilai kofaktor terlebih dahulu kita harus mencari nilai minor dari setiap elemen matrik. Untuk memudahkan, selanjutnya minor kita beri simbol dengan huruf M dan minor untuk setiap elemen matrik akan kita beri simbol dengan Mij dimana i adalah letak baris dan j adalah letak kolom dari setiap elemen matrik.

Contoh:

Diketahui matriks A sebagai berikut:

[](http://1.bp.blogspot.com/-3E0EeC72PPk/VRY2JWF-dNI/AAAAAAAACkY/ZGiOa7LeZ6I/s1600/Screenshot_17.png)

Maka minor elemen 2 yang terletak pada baris ke 1 kolom ke 1 diberi simbol dengan M11. Untuk mencari harga minornya dapat kita lakukan dengan mencoret atau menghilangkan baris ke 1 dan kolom ke 1 sehingga didapatkan matrik baru seperti berikut:

[](http://4.bp.blogspot.com/-JwtBsS3ft3w/VRY28YRAdgI/AAAAAAAACkg/QvZ1I3o7gIM/s1600/Screenshot_18.png)

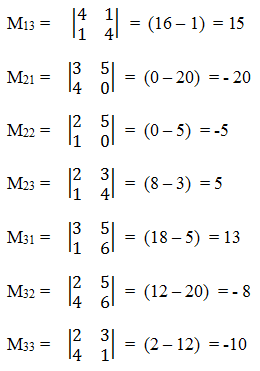
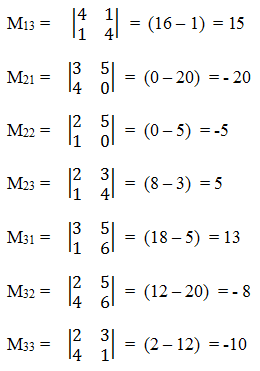
Jadi minor elemen 2 (M11) adalah :

[](http://2.bp.blogspot.com/-i3XYukz8iC0/VRY3NaxkbUI/AAAAAAAACko/rR03ujEn40Y/s1600/Screenshot_19.png)

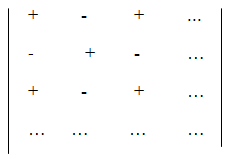
Serupa dengan cara di atas, minor elemen 3 (M12) adalah :

[](http://2.bp.blogspot.com/-FW7dzvoYZ6Q/VRY3lfbLB8I/AAAAAAAACkw/E6QPwKolipQ/s1600/Screenshot_20.png)

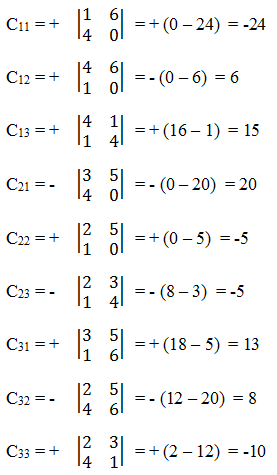
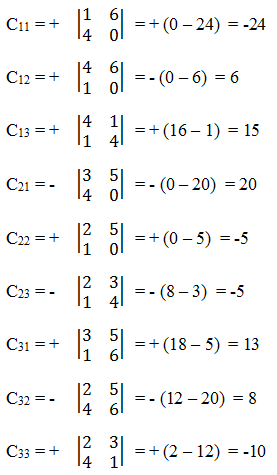
Untuk nilai M13, M21, M22, M23, M31, M32 dan M33 didapatkan hasil sebagai berikut:

[](http://2.bp.blogspot.com/--4zhzkCNBbM/VRY5mj69cBI/AAAAAAAACk8/KZAcVmp3t0M/s1600/Screenshot_22.png)

**2.7 Kofaktor**

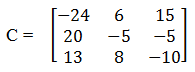
[](http://1.bp.blogspot.com/-9Wp4oS5ffAk/VRY6a-H-59I/AAAAAAAAClI/yjpyaoFZKps/s1600/Screenshot_23.png)Setelah mendapatkan harga minor dari masing-masing elemen matriks, dapat ditentukan nilai atau harga dari kofaktor. Cara mencarinya adalah dengan mengalikan masing-masing nilai minor di atas dengan tanda tempat masing-masing elemen. Adapun tanda tempatnya dapat dilihat pada gambar berikut:

Jadi berdasarkan tanda tempat di atas kita dapat mencari nilai kofaktor dari masing-masing elemen matriks. Untuk selanjutnya kita akan berikan simbol untuk nilai kofaktor masing-masing elemen dengan Cij, dimana i menandakan baris dan j menandakan kolom. jadi untuk setiap elemen di atas kita dapatkan harga kofaktornya sebagai berikut.

[](http://2.bp.blogspot.com/-1L2VrCqbIMg/VRY7TDryVfI/AAAAAAAAClU/YwxmhELUEyY/s1600/Screenshot_24.png)

**2.8 Matriks Kofaktor**

Setelah mendapatkan harga atau nilai kofaktor dari masing-masing elemen matrik di atas, maka sekarang akan disusun setiap nilai kofator tersebut sesuai dengan alamat tempatnya masing-masing. Susunan masing-masing elemen dari nilai kofaktor ini akan menghasilkan sebuah matrik baru yang kita namakan dengan matrik kofaktor. Untuk selanjutnya matrik kofaktor akan kita beri simbol dengan huruf C. Jadi matrik kofaktor (C) dari matrik di atas adalah:

[](http://2.bp.blogspot.com/-6FFWrBMvMPk/VRY8FNoAYBI/AAAAAAAAClc/2JQZJfyk4zo/s1600/Screenshot_25.png)

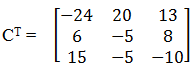
**2.9 Adjoin Matriks**

Matriks transpose dari matrik kofaktor dinamakan dengan matriks adjoin dari suatu matriks.

Untuk memperoleh adjoin dari suatu matrik bujur sangkar A, maka langkah-langkah yang harus dilakukan adalah:

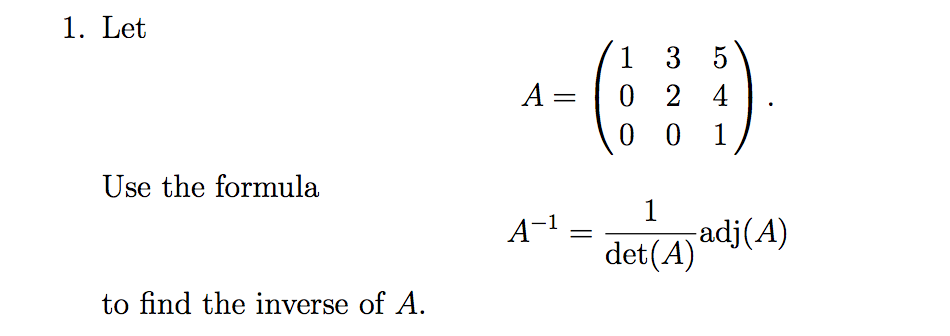
* Membentuk matrik kofaktor C
* Menuliskan transpose dari matrik C yaitu CT

Adjoin matriks C atau Transpos dari matriks C adalah :

[](http://4.bp.blogspot.com/-ul2_B6kiqlc/VRZCPqcPg5I/AAAAAAAACls/6Q2dTRNlrXI/s1600/Screenshot_26.png)

**2.10 Matriks Balikan (Invers)**

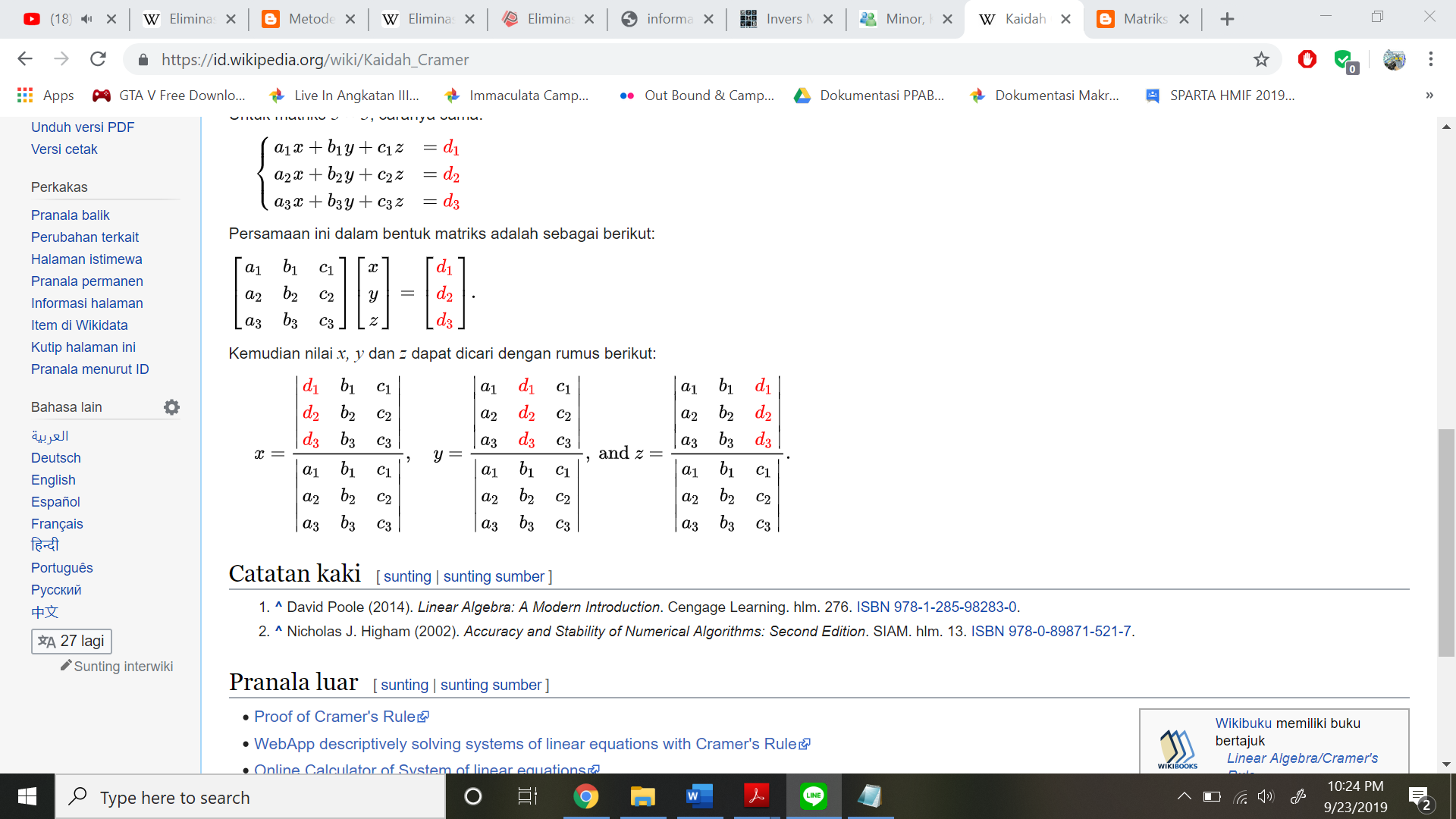
Invers atau Balikan dari suatu matriks dapat dihitung sebagai berikut.



**2.11 Kaidah Cramer**

Kaidah Cramer adalah rumus yang dapat digunakan untuk menyelesaikan sistem persamaan linear. Metode ini menggunakan determinan suatu matriks dan matriks lain yang diperoleh dengan mengganti salah satu kolom dengan vektor yang terdiri dari angka di sebelah kanan persamaannya. Metode ini dinamai dari matematikawan Swiss Gabriel Cramer (1704–1752). Kaidah Cramer tidak efisien untuk sistem dengan lebih dari dua atau tiga persamaan.

Contoh penyelesaian Sistem Persamaan Linear menggunakan Kaidah Cramer adalah :

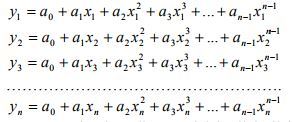


**2.12 Interpolasi Polinomial**

Interpolasi polinomial digunakan untuk mencari titik-titik antara dari n buah titik, P1(x1,y1), P2(x2,y2), P3(x3,y3), …, PN(xN,yN) dengan menggunakan pendekatan fungsi polinomial pangkat n-1:

e

Masukkan nilai dari setiap titik ke dalam persamaan polynomial di atas dan diperoleh persamaan simultan dengan n persamaan dan n variable bebas:



Penyelesaian persamaan simultan di atas adalah nilai-nilai a0, a1, a2, a3, …, an yang

merupakan nilai-nilai koefisien dari fungsi pendekatan polynomial yang akan

digunakan. Dengan memasukkan nilai x dari titik yang dicari pada fungsi polinomialnya, akan diperoleh nilai y dari titik tersebut.

**BAB III**

**IMPLEMENTASI PROGRAM DALAM JAVA**

**3.1 Struktur dan Definisi Class**

Tabel 3.1.1 Atribut dan Definisi Setiap Atribut

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Atribut** | **Definisi dan Spesifikasi** | **NIM Pembuat** |
| 1 | MatriksAdt | Berisi segala class dan method dari matriks | 13518128 |
| 2 | MATRIKS | MATRIKS class terdiri atas 3 komponen, yaitu Mem[100][100], NbrsEff, dan NkolEff | 13518056 |
| 3 | MainMatriks | Driver atau program utama untuk menjalankan program-program MatriksAdt | 13518056 |

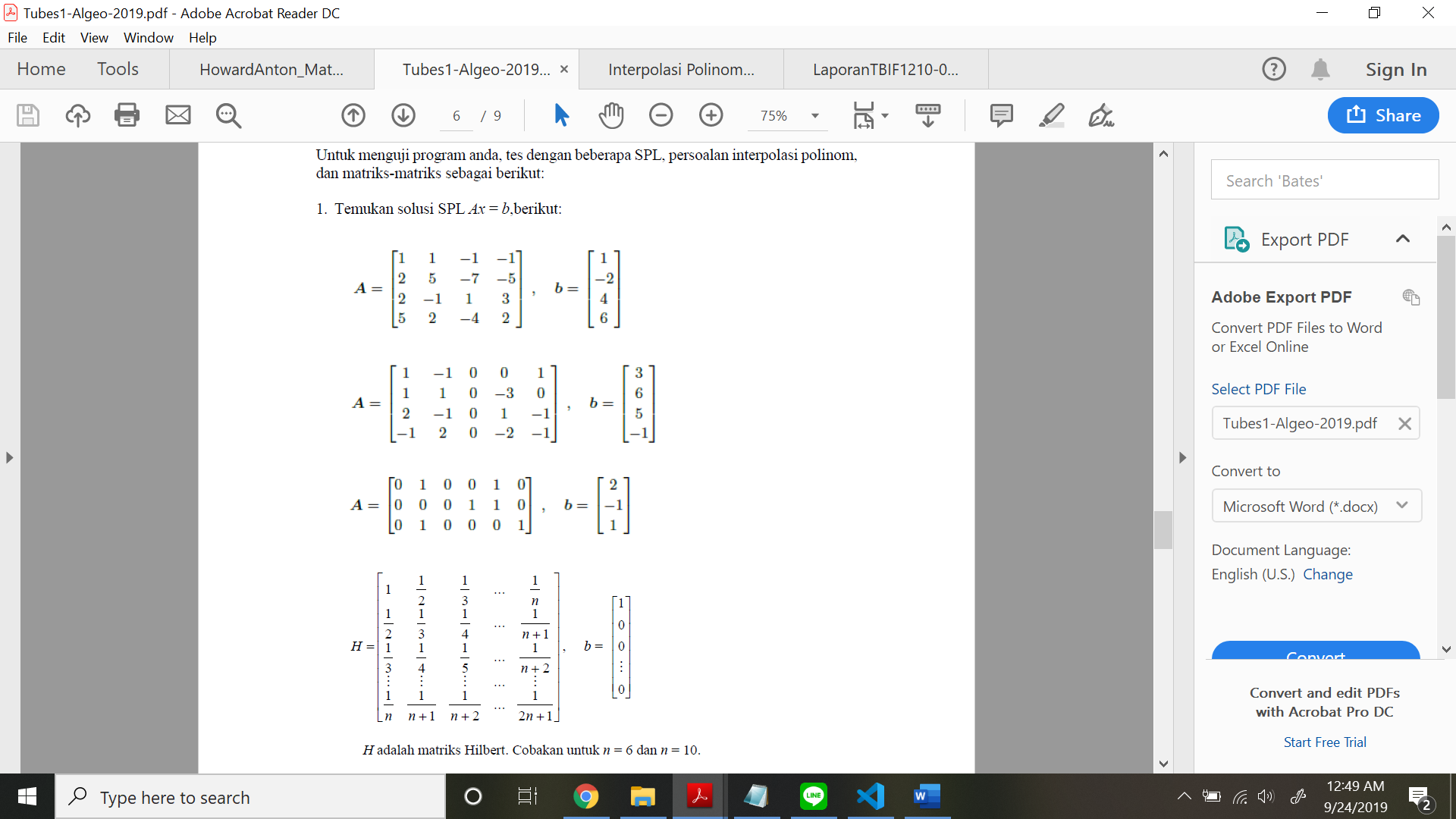
Tabel 3.1.2 Method dan Definisi Setiap Method

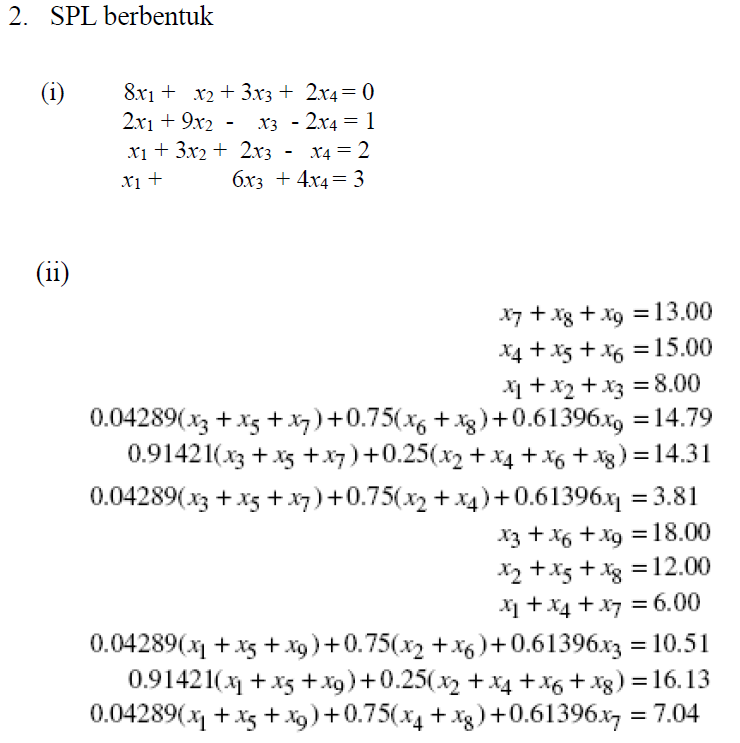
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Method** | **Definisi dan Spesifikasi** | **NIM Pembuat** |
| 1 | SPL | Menjalankan menu untuk operasi pada SPL | 13518056 |
|  |  |  |  |
| 2 | Gauss | Melakukan prosedur mengubah matriks augmented MAug dengan Eliminasi Gauss | 13518128 |
|  | Idx | Mengirimkan sebuah indeks dengan elemen ≠ 0 | 13518128 |
|  | SwapBaris | Melakukan operasi pertukaran baris | 13518128 |
|  | LeadOne | Mengecek apakah ada 1 utama atau tidak ada dalam suatu baris NK dari matriks M | 13518128 |
|  | BackwardSub | Melakukan substitusi mundur sehingga diperoleh solusi-solusi SPL berdasarkan augmented matrix | 13518056 |
|  |  |  |  |
| 2 | GaussJordan | Melakukan prosedur mengubah matriks augmented dengan Eliminasi Gauss Jordan | 13518119 |
|  |  |  |  |
| 3 | InverseMethod | Melakukan prosedur untuk mencari solusi SPL dengan metode matriks balikan dari matriks augmented MAug | 13518056 |
|  | GetMATRIKS | Membentuk matriks M berukuran NxN yang berisi matriks koefisien dari MAug | 13518056 |
|  | GetKONSTANTA | Membentuk matriks MK berukuran Nx1 yang berisi elemen-elemen konstanta dari MAug | 13518056 |
|  | Invers | Menghasilkan invers atau balikan dari matriks M | 13518056 |
|  | KaliMATRIKS | Mengirim hasil perkalian matriks M1 \* M2 | 13518056 |
|  |  |  |  |
| 4 | MetodeCramer | Melakukan operasi penyelesaian SPL dengan Metode Cramer dengan masukan berupa Matriks Augmented | 13518056 |
|  | MCramer | Membentuk sebuah matriks cramer dari matriks M dengan kolom ke-kol digantikan oleh matriks konstanta MK | 13518056 |
|  |  |  |  |
| 5 | MainDeterminan | Prosedur untuk menerima matriks M berukuran NxN kemudian menampilkan determinan dari M | 13518056 |
|  | HighTriangle | Mengubah matriks M menjadi matriks segitiga atas sehingga diperoleh determinan dari perkalian diagonal-diagonal utama | 13518119 |
|  | LowTriangle | Mengubah matriks M menjadi matriks segitiga bawah sehingga diperoleh determinan dari perkalian diagonal-diagonal utama | 13518128 |
|  | MetodeKofaktor | Prosedur untuk mendapatkan nilai determinan dengan metode kofaktor | 13518056 |
|  | Determinan | Menghasilkan nilai determinan dari matriks M | 13518056 |
|  |  |  |  |
| 6 | MainInvers | Prosedur untuk membaca dan menghasilkan invers dari matriks M | 13518056 |
|  | InversGJ | Prosedur untuk mendapatkan invers dari suatu matriks dengan metode Gauss Jordan | 13518056 |
|  |  |  |  |
| 7 | MainKofaktor | Prosedur untuk membaca dan menghasilkan matriks kofaktor dari matriks M | 13518056 |
|  | Minor | Menghasilkan minor ij dari suatu matriks M | 13518056 |
|  | MatriksKofaktor | Menghasilkan matriks kofaktor dari matriks M | 13518056 |
|  |  |  |  |
| 8 | MainAdjoin | Prosedur untuk membaca dan menghasilkan adjoin dari matriks M | 13518056 |
|  | Adjoin | Menghasilkan adjoin dari matriks M | 13518056 |
|  |  |  |  |
| 9 | Interpolasi | Melakukan prosedur interpolasi titik-titik | 13518056 |
|  | BackwardSubInterpolasi | Melakukan substitusi mundur sehingga diperoleh sebuah array Hasil yang berisi solusi augmented | 13518056 |
|  | InverseMethodInterpolasi | Melakukan prosedur invers untuk mendapatkan solusi untuk interpolasi dan menyimpannya dalam penampung nilai Hasil | 13518056 |
|  | MetodeCramerInterpolasi | Melakukan prosedur cramer untuk mendapatkan solusi untuk interpolasi dan menyimpannya dalam penampung nilai Hasil | 13518056 |
|  | TulisPolinom | Menuliskan hasil solusi matriks dalam bentuk fungsi interpolasi polinomial | 13518056 |
|  | HitungPolinom | Menghitung fungsi polinomial yang diperoleh dengan parameter nilai berupa x |  |
|  |  |  |  |
| 10 | MakeMATRIKS | Membuat Matriks dengan mengatur jumlah baris efektif dan jumlah kolom efektif | 13518056 |
|  | BacaMATRIKS | Membaca matriks baik input secara langsung maupun membaca file text | 13518128 |
|  | BacaAUG | Membaca matriks augmented baik input secara langsung maupun membaca file text | 13518128 |
|  | TulisMATRIKS | Menuliskan matriks ke layar tanpa enter di akhir | 13518056 |
|  | CopyMATRIKS | Menyalin matriks MIn ke MHsl | 13518056 |
|  | Identity | Menghasilkan matriks identitas berukuran NxN | 13518056 |
|  | Transpose | Mengubah setiap elemen baris menjadi kolom dan elemen kolom menjadi baris | 13518056 |
|  | GetINVERSE | Mengambil bagian dari Matriks Augmented yang merupakan hasil akhir invers matriks | 13518056 |
|  | GabungMATRIKS | Menghasilkan matriks yang merupakan gabungan antara matriks M dan matriks I yang berukuran sama | 13518056 |
|  |  |  |  |
| 11 | SaveFile | Menyimpan hasil input/output ke dalam file .txt | 13518128 |

**BAB IV**

**EKSPERIMEN**

**4.1 STUDI KASUS**





3. Untuk persoalan determinan, matrisk balikan, dan matriks kofaktor, silakan cari masing-masing dua buah matriks yang berukuran 5 x 5 dan 10 x 10.

4. Interpolasi

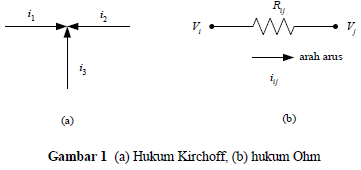
Dalam sebuah rangkaian listrik berlaku hukum-hukum arus Kirchoff menyatakan bahwa jumlah aljabar dari semua arus yang memasuki suatu simpul (Gambar 4.4a) haruslah nol: 8

Σ *i* = 0

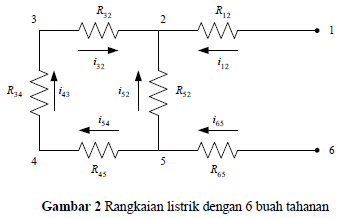
Dalam hal ini, semua arus *i* yang memasuki simpul dianggap bertanda positif. Sedangkan hukum Ohm (Gambar 1) menyatakan bahwa arus *i* yang melalui suatu tahanan adalah :

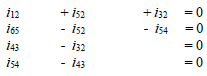


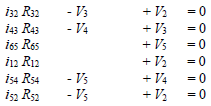
yang dalam hal ini *V* adalah tegangan dan *R* adalah tahanan.



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



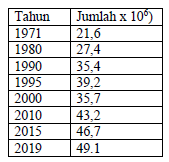
Arah arus dimisalkan seperti diatas. Dengan hukum Kirchoff diperoleh persamaan-persamaan berikut :

Dari hukum Ohm didapat :

Tentukan

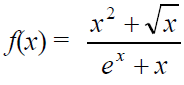
bila diketahui

5. **(Interpolasi)** Jumlah penduduk Jawa Barat dari tahun 1971 hingga 2019 (dibulatkan ke juta) adalah sebagai berikut:



Berdasarkan data tersebut prediksilah jumlah penduduk Jawa Barat pada tahun 1975, 1983, 1992, 2005, 2012 (atau nilai lain sesuai masukan user) dengan menggunakan polinom interpolasi.

**6.** 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.

**BAB V**

**PENUTUP**

**5.1 Kesimpulan**

Dari tugas besar berjudul “Sistem Persamaan Linear, Determinan, dan Aplikasinya”, maka program yang kami buat bernama MatriksAdt.java dan MainMatriks.java bekerja dengan baik dan sebagian besar telah memenuhi test case yang ada.

**5.2 Saran**

Berikut ini adalah saran-saran yang bisa kami berikan untuk tugas besar ini.

1. Persoalan sebaiknya dipecah menjadi persoalan-persoalan lebih kecil agar memudahkan proses debugging dan pembuatan kode itu sendiri.
2. Atribut dan method sebaiknya tidak dipadatkan dalam satu file .java saja, melainkan terpisah-pisah agar modul bisa lebih visible untuk dibaca karena terkesan lebih sedikit.
3. Perbanyak test case yang dibuat agar program bisa terjamin lebih benar kinerjanya.

**5.3 Refleksi**

Berikut ini adalah refleksi yang kami dapatkan setelah mengerjakan tugas besar ini.

1. Biasakan untuk mengerjakan tugas tepat waktu dan mulai dicicil.
2. Pembagian tugas sangat penting agar proses pembuatan tugas program bekerja dengan lancar.
3. Sikap pantang menyerah dan sabar harus tertanam dalam diri agar proses pembuatan program dapat berjalan dengan lancar.
4. Jangan menunda-nunda waktu ketika ada waktu luang untuk mengerjakan tugas ini.
5. Istirahat yang cukup agar pikiran lebih fresh untuk mengerjakan tugas nantinya.

**REFERENSI**

https://iragitawulandari.wordpress.com/2012/12/15/metode-gauss-jordan/

https://id.wikipedia.org/wiki/Eliminasi\_Gauss

http://team-aljabar.blogspot.com/2013/03/matriks-balikan-invers.html

https://sosmedpc.blogspot.com/2016/12/determinan-pengertian-dan-contoh.html

http://www.uniksharianja.com/2015/03/minor-kofaktor-matrik-kofaktor-dan.html

https://id.wikipedia.org/wiki/Kaidah\_Cramer

https://aimprof08.wordpress.com/2016/05/02/invers-matriks-menggunakan-adjoint/

https://d4anm2017a.wordpress.com/2017/11/23/interpolasi-linier-kuadratik-dan-polinomial/

http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2019-2020/algeo19-20.htm

https://www.tutorialspoint.com/cplusplus-program-to-compute-determinant-of-a-matrix

https://www.geeksforgeeks.org/java-pow-method-example/

https://www.w3schools.com/java/java\_constructors.asp

https://stackoverflow.com/questions/5168144/does-java-support-structs/5168216

https://www.mkyong.com/java/java-display-double-in-2-decimal-points/

https://www.codewithc.com/c-program-for-gauss-elimination-method/

https://www.geeksforgeeks.org/finding-inverse-of-a-matrix-using-gauss-jordan-method/