Membentuk Algoritma untuk Pemecahan Sistem Persamaan Lanjar secara Numerik

Bervianto Leo P - 13514047
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514047@std.stei.itb.ac.id

***Abstrak*—Sistem persamaan lanjar merupakan suatu persoalan yang biasa ditemukan. Dalam sistem persamaan lanjar terdapat tiga pemecahan, yaitu terdapat pemecahan tak-hingga, pemecahan tunggal, dan tidak ada pemecahan. Lalu penulisan sistem persamaan lanjar dapat digunakan suatu matriks yang dinamakan matriks *augmented*. Dari matriks tersebut diberlakukan operasi baris elementer dengan menggunakan metode eliminasi Gauss atau eliminasi Gauss-Jordan sehingga mendapatkan suatu bentuk eselon baris atau bentuk eselon baris tereduksi yang memberikan suatu pemecahan dari sistem persamaan lanjar tersebut. Dengan metode tersebut diimplementasikan untuk pengerjaan dalam komputer atau sebagai algoritma. Pembentukan algoritma tersebut ada kemiripan dengan algoritma secara umum.**

***Kata Kunci*—algoritma, numerik, pemecahan, sistem persamaan lanjar**

I. Pendahuluan

Sistem persamaan lanjar atau sistem persamaan linear sudah banyak dikenal oleh pelajar-pelajar. Namun sering kali akan menghadapi suatu kesulitan jika sudah cukup banyak persamaan yang diberikan untuk mencari pemecahannya. Penyebab sulitnya untuk memecahkannya seperti penulisan persamaannya dalam pengerjaannya masih dituliskan secara lengkap.

Dari hal tersebut dibutuhkan suatu metode mempermudah mencari suatu pemecahan dari sistem persamaan lanjar. Salah satunya dengan merubah representasinya menjadi suatu matriks. Lalu melakukan berbagai operasi pada matriks tersebut.

Selain itu diperlukan kemampuan untuk menganalisis dengan baik metode yang terbaik untuk mengoperasikan matriks tersebut serta mengimplementasikannya dalam pengerjaan komputer. Sehingga diperlukan pemahaman untuk algoritma-algoritma yang dibutuhkan dalam pengerjaan atau pengoperasian suatu matriks tersebut.

II. Dasar Teori

A. Sistem Persamaan Lanjar

i. Bentuk umum persamaan lanjar

Persamaan lanjar (linear) secara umum memiliki bentuk sebagai berikut

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = b$$

dengan n peubah, a_1 , $a_2, \ldots$, a_n dan b adalah konstanta-konstanta riil. Berikut ini sebagai contoh persamaan-persamaan lanjar:

$$x + 2y = 1 \qquad 2x_1 + 3x_2 + 4x_5 = 10$$

Pada persamaan lanjar tidak melibatkan sesuatu hasil kali atau akar peubah. Semua peubah hanya terdapat sampai dengan angka pertama dan tidak muncul sebagai argumen untuk fungsi trigonometrik, fungsi logaritmik, atau untuk fungsi eksponensial. [1] Berikut ini sebagai contoh yang bukan persamaan lanjar:

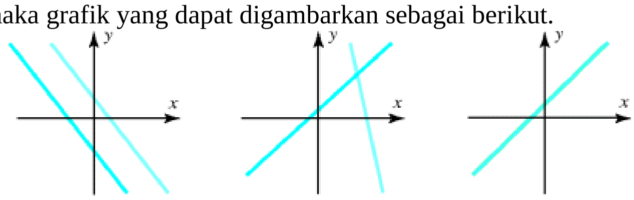
$$3x^2 + 4y = 1 \qquad 3xy + yz = 2$$

ii. Pemecahan Sistem Persamaan Lanjar

Pemecahan persamaan lanjar $a_1x_1 + a_2x_2 + \ldots + a_nx_n = b$ adalah urutan dari n bilangan s_1 , $s_2, \ldots$, s_n sehingga persamaan tersebut dipenuhi bila mensubsitusikannya terhadap $x_1 = s_1$, $x_2 = s_2, \ldots$, $x_n = s_n$. Himpunan semua pemecahan tersebut merupakan *himpunan pemecahannya*. [1] Dalam sistem persamaan lanjar memiliki tiga kemungkinan pemecahan. Hal ini akan ditunjukan dengan dua persamaan lanjar yang ditunjukan sebagai garis dalam bidang xy. Misalkan,

$$a_1x + b_1y = c_1 \; \{a_1, b_1 \text{ keduanya tidak nol}\}$$
$$a_2x + b_2y = c_2 \; \{a_2, b_2 \text{ keduanya tidak nol}\}$$

maka grafik yang dapat digambarkan sebagai berikut.



(a) (b) (c)

Gambar 2.1 Ilustrasi Pemecahan Sistem Persamaan Lanjar dalam Bidang xy.
Sumber : referensi [1]

Pada gambar 2.1 (a) terlihat bahwa kedua garis tidak saling berpotongan, di sini diartikan sebagai bahwa sistem persamaan lanjar tersebut tidak memiliki pemecahan. Sedangkan pada gambar 2.1 (b) terlihat bahwa kedua garis memiliki tepat satu titik perpotongan yang memiliki arti sistem persamaan lanjar memiliki pemecahan yang unik dalam hal ini didapatkan pemecahan tunggal (x_0, y_0).

Begitu juga dengan gambar 2.1 (c) kedua garis saling berimpit yang berarti memiliki banyak pemecahan atau tak hingga.

Beberapa hal yang perlu diketahui yaitu sebuah sistem persamaan yang tidak mempunyai pemecahan dikatakan takkonsisten. Namun jika ada setidak-tidaknya satu pemecahan, maka sistem persamaan tersebut dinamakan konsisten. [1]

Tiga kemungkinan yang sudah ditunjukan juga berlaku untuk sebarang sistem. Tiga kemungkinan tersebut yaitu **sistem persamaan lanjar tidak mempunyai pemecahan**, atau **mempunyai satu pemecahan**, atau **mempunyai banyak atau tak hingga pemecahan**.

B. Sistem Persamaan Lanjar dan Matriks

Dalam sistem persamaan lanjar, dapat dibentuk menjadi sebuah matriks sehingga mempersingkat penulisan. Berikut ini diberikan sebuah sistem persamaan lanjar,

$$\begin{matrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{matrix}$$

sistem persamaa lanjar tersebut dapat dipersingkat sehingga menjadi matriks sebagai berikut.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} & b_m \end{bmatrix}$$

Sebagai contoh berikut ini diberikan suatu sistem persamaan lanjar sebagai berikut.

$$\begin{matrix} x_1 + 5x_2 + 2x_3 = 3 \\ 2x_1 + 4x_2 + x_3 = 1 \\ 3x_1 + 6x_2 = 10 \end{matrix}$$

Dari sistem persamaan lanjar tersebut matriksnya adalah sebagai berikut.

$$\begin{bmatrix} 1 & 5 & 2 & 3 \\ 2 & 4 & 1 & 1 \\ 3 & 6 & 0 & 10 \end{bmatrix}$$

Matriks tersebut menunjukan konstanta-konstanta riil dalam setiap persamaan lanjar. Matriks ini dinamakan **matriks yang diperbesar** (*augmented matrix*). [1]

C. Operasi Baris Elementer

Operasi baris elementer merupakan suatu metode dasar yang digunakan untuk memecahkan sistem persamaan lanjar yaitu dengan mengganti sistem yang diberikan dengan sistem baru yang mempunyai himpunan yang sama dengan pemecahan yang lebih mudah. Sistem baru ini umumnya didapatkan dalam suatu tahapan dengan menerapkan ketiga tipe operasi berikut. [1]

1. Mengalikan persamaan dengan konstanta yang tidak sama dengan nol.
2. Mempertukarkan dua persamaan.
3. Menambahkan kelipatan dari satu persamaan bagi yang lainnya.

D. Eliminasi Gauss dan Eliminasi Gauss-Jordan

Eliminasi Gauss merupakan suatu metode yang menggunakan operasi baris elementer untuk mendapatkan suatu pemecahan dari suatu sistem persamaan lanjar. Berikut ini contoh matriks hasil operasi baris elementer.

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Hasil dari matriks tersebut merupakan hasil eliminasi Gauss. Matriks tersebut disebut **bentuk eselon baris** (*row-echelon form*). Jika operasi baris elementer dilanjutkan sehingga menghasilkan matriks sebagai berikut,

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

Matriks tersebut dikatakan **bentuk eselon baris ter-reduksi** (*reduced row-echelon form*). Metode ini dikatakan eliminasi Gauss-Jordan. Dari kedua matriks tersebut, supaya mendapatkan kedua bentuk tersebut, makan matriks tersebut akan memiliki sifat sebagai berikut.

1. Jika baris tidak terdiri seluruhnya dari nol, maka bilangan bukan nol pertama dalam baris tersebut adalah 1. 1 tersebut disebut sebagai 1 utama. [1]
2. Jika terdapat baris yang seluruhnya terdiri dari nol, makan semua baris seperti itu dikelompokan bersama-sama di bawah matriks. [1]
3. Dalam sebarang dua baris yang berurutan yang seluruhnya tidak terdiri dari nol, maka 1 utama dalam baris yang lebih rendah terdapat lebih jauh ke kanan dari 1 utama dalam baris yang lebih tinggi. [1]
4. Masing-masing kolom yang mengandung 1 utama mempunyai nol di tempat lain. [1]

Sifat-sifat pada nomor 1-3 dimiliki oleh bentuk eselon baris dan seluruhnya dimiliki oleh bentuk eselon tereduksi.

E. Metode Numerik

Metode numerik merupakan cara sistematis untuk menyelesaikan persoalan matematika dengan operasi angka (+,-,*,/). [2] Cara penyelesaian persoalan matematika ada dua yaitu secara analitik dan secara numerik. Secara analitik merupakan suatu cara untuk mendapatkan suatu pemecahan secara eksak atau tepat sedangkan numerik suatu cara untuk mendapatkan suatu pemecahan secara hampiran atau aproksimasi. Secara analitik atau metode analitik merupakan metode yang menggunakan rumus dan teorema yang sudah baku di dalam matematika sedangkan numerik menggunakan pendekatan aproksimasi untuk mencari pemecahan hanya dengan operasi aritmetika biasa. [2]

Oleh karena metode numerik merupakan pemecahan hampiran, maka akan terdapat suatu galat. Galat merupakan perbedaan antara pemecahan hampiran dengan pemecahan eksak. Salah satu sumber galat merupakan galat pembulatan. [2] Salah satu contoh galat pembulatan yaitu terjadi dalam komputer, dikarenakan komputer memiliki keterbatasan dalam merepresentasikan bilangan riil sehingga terjadi pembulatan.

III. Contoh Matriks Hasil Operasi Baris Elementer dengan Pemecahannya

Untuk menunjukan ketiga pemecahan sistem persamaan lanjar tersebut, berikut ini diberikan tiga contoh matriks dengan pemecahan tersebut.

(a) $\begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$

(b) $\begin{bmatrix} 1 & 1 & 0 & -2 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

(c) $\begin{bmatrix} 1 & 1 & 0 & -2 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Pemecahan untuk matriks (a). Sistem persamaan yang bersesuaian sebagai berikut

$$x_1 = -2,\ x_2 = 1,\ x_3 = 1$$

Persamaan tersebut merupakan suatu pemecahan dari suatu sistem yang sebelumnya yang sudah dioperasikan dengan operasi baris elementer. Persamaan tersebut memiliki pemecahan tunggal.

Sedangkan untuk matriks (b). Sistem persamaan yang bersesuaian sebagai berikut

$$x_1 + x_2 = -2$$
$$x_2 + x_3 = 1$$

Pemecahan yang didapatkan yang seperti diatas merupakan pemecahan banyak atau pemecahan tak-hingga sehingga perlu dibentuk sebagai berikut

$$x_1 = -x_2 - 2$$
$$x_2 = -x_3 + 1$$

Subsitusikan x_2 ke dalam persamaan x_1 menjadi seperti berikut.

$$x_1 = x_3 - 3$$

Persamaan tersebut diubah seperti tersebut dikarenakan $x_1$dan x_2 bersesuaian dengan 1 utama sehingga dapat dinamakan menjadi **peubah-peubah utama** (*leading variables*). Setelah itu x_3 dijadikan suatu paramater atau sebagai sembarang nilai, misalkan u, sehingga mendapatkan himpunan pemecahan tak-berhingga sebagai berikut.

$$x_1 = u - 3$$
$$x_2 = -u + 1$$
$$x_3 = u$$

Dengan memasukan u dengan sembarang nilai, akan didapatkan suatu pemecahan dari sistem persamaan lanjar yang telah diberikan. Dalam hal ini pemecahan akan banyak tergantung pada nilai u.

Selanjutnya dalam matriks (c), persamaan terakhir yaitu

$$0x_1 + 0x_2 + 0x_3 = 1$$

tidak akan mendapatkan pemecahan atau tidak ada pemecahan yang dapat memenuhi persamaan tersebut.

IV. Tatancang Pemorosan

Tatancang pemorosan (*pivoting strategy*) merupakan suatu metode untuk mendapatkan pemecahan dari sistem persamaan lanjar dengan galat yang minimal akibat pembulatan. Tatancang pemorosan ini dengan cara memilih suatu *pivot* (poros) dari semua elemen pada kolom j yang mempunyai nilai mutlak terbesar, lalu pertukarkan baris yang memiliki nilai mutlak terbesar dengan baris ke i. Sebagai contoh yaitu sebagai berikut.

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

Dimulai dari kolom 1, mutlak terbesar yaitu baris 3, lalu tukarkan baris 1 dengan baris 3. Sehingga seperti berikut.

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

Lalu perhatikan kolom 2, mencari mutlak terbesar juga dimulai dari baris kedua. Didapatkan baris kedua. Sehingga tidak perlu ditukarkan. Lalu maju ke kolom 3 dan baris 3. Karena sudah pada poros dan tidak ada yang perlu dibandingkan. Sehingga hasilnya yaitu seperti yang di atas. Yang perlu diperhatikan, yang dibandingkan hanya bagian bawah dengan dirinya sendiri. Serta sebelum berpindah kolom, akan dilakukan operasi baris elementer yang akan dibahas selanjutnya.

V. Membentuk Algoritma untuk Pemecahan Sistem Persamaan Lanjar secara Numerik

Agar memudahkan pembentukan algoritma untuk pemecahan sistem persamaan lanjar secara numerika akan dibagi dengan beberapa modul atau prosedur seperti modul khusus tatancang pemorosan dan khusus operasi baris elementer dengan metode eliminasi Gauss atau eliminasi Gauss-Jordan. Struktur data yang digunakan merupakan matriks yang dapat digunakan dalam komputer dengan mudah. Adapun struktur data Matriks sebagai berikut, dengan IdxBrsMax dan IdxKolMax merupakan baris dan kolom maksimum yang dapat diubah sesuai kapasitas yang diinginkan.

```
Matriks : < BrsMax : integer,
            KolMax : integer,
            Mat : array [1...IdxBrsMax] of array
            [1...IdxKolMax] of real >
```

Berikut ini prosedur tatancang pemorosan, pada dasarnya merupakan algoritma mencari bilangan terbesar namun ini dengan bilangan mutlak terbesar.

```
procedure pivotingstrategy (input/output M : Matriks,
input k : integer, input j : integer)
{ Masukan M berupa matriks yang siap diubah, k
merupakan baris awal yang siap dicek, dan j merupakan
kolom yang ingin dicari. }
KAMUS LOKAL
   i,jt : integer
   imax : integer
   Temp : real
ALGORITMA
   imax ← k
   i traversal [k+1...M.BrsMax]
      if ( abs(M.Mat[i][j]) > abs(M.Mat[imax][j]) ) then
         imax ← i
      i ← i + 1
```

```
    if (imax <> k) then
    { Jika tidak sama, pertukarkan baris }
        jt traversal [j...M.KolMax]
            Temp ← M.Mat[k][jt]
            M.Mat[k][jt] ← M.Mat[imax][jt]
            M.Mat[imax][jt] ← Temp
    { Jika imax sama dengan j tidak perlu melakukan
    apa-apa atau tidak ditukar. }
```

Sedangkan prosedur untuk operasi baris elementer sebagai berikut.

```
procedure OBE (input/output M : Matriks, input l : integer, input j : integer)
{ Masukan M berupa Matriks yang siap diubah, l merupakan baris awal dan j kolom awal yang akal di OBE. }
KAMUS LOKAL
  i,k : integer
  Pembagi,temp : real
ALGORITMA
  {Melakukan pembagian hingga menjadikan satu
  utama}
  Pembagi ← M.Mat[l][j]
  i traversal [j...M.KolMax]
    M.Mat[l][i] ← M.Mat[l][i] / Pembagi
  {Melakukan menjadikan nol pada baris bagian bawah
  satu utama}
  i traversal [l+1...M.BrsMax]
    temp ← M.Mat[i][j]
    k traversal [j...M.KolMax]
        M.Mat[i][k] ← M.Mat[i][k] – M.Mat[l][k] * temp
```

Setelah itu perlu membentuk suatu algoritma utama untuk menggunakan prosedur tersebut sehingga menjadi metode eliminasi Gauss. Berikut ini algoritmanya.

```
KAMUS
    M : Matriks
    p,q : integer
ALGORITMA
  {Melakukan Eliminasi Gauss dengan bantuan tatancang
  pemorosan}
  p ← 1
  q ← 1
  repeat
      pivotingstrategy(M,p,q)
      if (M.Mat[p][q]<>0) then
          OBE(M,p,q)
          p ← p + 1
          q ← q + 1
      else
          q ← q + 1
  until (q<=M.KolMax)
```

Secara sederhana algoritma utamanya hanya pada algoritma di atas yaitu melakukan tatancang pemorosan dan operasi baris elementer sehingga menghasilkan **bentuk eselon baris**. Selanjutnya untuk mendapatkan hasilnya dalam algoritma utama perlu ditambahkan prosedur lain. Dapat dilanjutkan dengan eliminasi Gauss-Jordan atau sulih mundur (subsitusi mundur). Saat ini akan digunakan prosedur eliminasi Gauss-Jordan.

Prosedur Gauss-Jordan tidak akan diberikan secara lengkap, namun akan diberikan ‘ide’ pembentuknya. Langkah pertama, mencari 1 utama, lalu membuat 0 setiap baris selain baris tersebut, lalu lakukan hal yang sama pada baris berikutnya, hingga baris terakhir.

Lalu untuk indikator jika pemecahan tidak konsisten dapat menggunakan rank. Diketahui bahwa jika sebuah rank matriks A (matriks variabel) kurang dari rank matriks [A|b] (matriks *augmented*) maka sistem persamaan lanjar tersebut tidak konsisten. Dengan begitu, diperlukan untuk menghitung *rank*. Pencarian *rank* hanya dapat dilakukan setelah eliminasi Gauss-Jordan.

```
procedure rank (input M : Matriks , input j : integer)
{ Masukan M berupa matriks, j berupa batasan kolom yang
diinginkan }
KAMUS LOKAL
  i,k : integer
  sum : integer
ALGORITMA
  sum ← 0
  i traversal [1...M.BrsMax]
    k ← i
    while ( (k<=j) AND (M.Mat[i][k] <> 1) )
    { Mencari satu utama }
        k ← k + 1
    if ( (k<>j) AND (M.Mat[i][k]=1) ) then
        sum ← sum + 1
```

Sangat sederhana untuk pencarian jumlah *rank,* parameter j digunakan agar mempermudah pencarian dengan batasan kolom tertentu. Sehingga algoritma utama akan bertambah menjadi sebagai berikut.

```
KAMUS
    M : Matriks
    p,q : integer
ALGORITMA
  {Melakukan Eliminasi Gauss dengan bantuan tatancang
  pemorosan}
  p ← 1
  q ← 1
  repeat
      pivotingstrategy(M,p,q)
      if (M.Mat[p][q]<>0) then
          OBE(M,p,q)
          p ← p + 1
          q ← q + 1
      else
          q ← q + 1
  until (q<=M.KolMax)
  GaussJordan(M)
  if (rank(M,M.KolMax-1) <> rank(M,M.KolMax)) then
    output("Tidak ada pemecahan yang memenuhi.")
  else
    {Pemecahan konsisten atau ada}
```

Namun tidak berhenti di bagian tersebut, perlu diketahui suatu indikator jika pemecahan banyak atau tak-hingga. Salah satunya dengan jumlah *rank*, jika jumlah *rank* kurang dari jumlah kolom, pemecahan tersebut merupakan pemecahan banyak atau tak-hingga. Sehingga dapat ditambahkan sebagai berikut.

```
KAMUS
    M : Matriks
    p,q : integer
ALGORITMA
  {Melakukan Eliminasi Gauss dengan bantuan tatancang
  pemorosan}
  p ← 1
  q ← 1
  repeat
      pivotingstrategy(M,p,q)
      if (M.Mat[p][q]<>0) then
          OBE(M,p,q)
          p ← p + 1
          q ← q + 1
      else
          q ← q + 1
  until (q<=M.KolMax)
  GaussJordan(M)
  if ( rank(M,M.KolMax-1) <> rank(M,M.KolMax) ) then
      output("Tidak ada pemecahan yang memenuhi.")
  else
      {Pemecahan konsisten atau ada}
      if ( rank(M,M.KolMax) < M.KolMax ) then
          output("Pemecahan banyak atau tak-hingga.")
      else
          pemecahan(M)
```

Prosedur pemecahan(M) merupakan suatu prosedur yang akan menampilkan pemecahan tunggal. Algoritma tersebut sudah memberikan suatu pemecahan dari sistem persamaan lanjar secara sederhana untuk lebih lanjut dapat diteruskan dengan membuat prosedur-prosedur lain seperti membuat prosedur membentuk pemecahan parameter jika pemecahan banyak.

Yang perlu diperhatikan yaitu struktur data yang digunakan dapat diubah sesuai kebutuhan serta penamaan variabel yang digunakan dapat diubah sesuai keinginan asalkan dapat mudah dipahami. Selain itu masih ada kemungkinan bentuk algoritma yang lain, algoritma yang diberikan merupakan salah satu alternatif yang dapat digunakan.

VI. SIMPULAN

Membentuk suatu algoritma untuk pemecahan sistem persamaan lanjar merupakan suatu hal yang sederhana. Dengan membagi-bagi sistem pengerjaan, dapat dilakukan dengan baik. Algoritma dibentuk agar komputer dapat mengerjakannya dengan baik dan cukup efisien. Pemecahan sistem persamaan lanjar menggunakan komputer sama dengan pemecahan secara numerik dikarenakan komputer menggunakan pembulatan-pembulatan terutama dalam representasi bilangan riil. Diharapkan banyak orang mengerti dan memahami suatu algoritma dan menerapkannya dalam menyelesaikan suatu masalah.

VII. UCAPAN TERIMAKASIH

Penulis mengucapkan syukur kepada Tuhan YME untuk segala karunia-Nya sehingga penulis dapat menyelesaikan makalah ini. Terimakasih kepada dosen pengampu IF2123 Aljabar Geometri, Bapak Rinaldi Munir untuk pengajaran yang telah diberikan dalam kuliah Aljabar Geometri secara khusus mengenai sistem persamaan lanjar yang menjadi dasar dalam penulisan makalah ini.

REFERENSI

[1] Anton, Howard, 1997, *Aljabar Linear Elementer*, Edisi Kelima, (diterjemahkan oleh : Pantur Silaban dan I. Nyoman Susila), Penerbit Erlangga, Jakarta.
[2] Slide Kuliah IF2123, *Aplikasi Aljabar Lanjar pada Metode Numerik.pptx*, oleh Rinaldi Munir.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Desember 2015

Bervianto Leo P - 13514047