JURNAL EMACS
[Engineering, MAthematics and Computer Science] Vol.5 No.1 January 2023: 25-32
e-ISSN: 2686-2573
DOI: 10.21512/emacsjournal.v5i1.9206

Operasi Dasar Baris/Kolom Matriks Secara Interaktif Dengan Menggunakan R

I Gusti Agung Anom Yudistira[1*], Rinda Nariswari[2]

Statistics Department, School of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia 11480
i.yudistira@binus.ac.id; rinda.nariswari@binus.ac.id

*Correspondence: i.yudistira@binus.ac.id

Abstract *- The linear algebra applications available today usually only provide the result. So, it is a challenge to overcome this, and innovation is needed in the computing aspect. One of the popular and open-source programming languages is R. The computational innovation in R needs to be explored further, to explore the R programming logic. The creation of a function environment with the list function and the involvement of local and global variables/objects has received little attention. Based on the problems formulated, this study proposes two objectives, namely (1) developing an R program that is able to provide interactive and step-by-step solutions, to obtain a solution of a system of linear equations, and (2) to explore R's ability to create and handle global variables. An R program is created, starting with creating a function environment. This function environment is filled with four related functions, namely "exchange", "multiply", "fold", and "yield". These four functions are connected to each other through a global object. Users can type in each function to perform row/column operations, interactively and step by step. The environmental function in this program, is named OBE. The OBE function accepts input in the form of a coupling matrix derived from a system of linear equations. The final result of this interactive process chain is given by the "result" function. The result function will display two matrices, namely the Original Matrix which is the input and the Equivalent Matrix.*

Keywords: *Systems of Linear Equations; R Programming; Numerical Methods; Elementary Row Operations; Gauss-Jordan Method*

Abstrak *– Aplikasi-aplikasi aljabar linear yang tersedia saat ini biasanya hanya menyuguhkan hasil akhir. Sehingga merupakan tantangan untuk mengatasi hal tersebut, dan diperlukan inovasi dalam aspek komputasi. Salah satu bahasa pemrograman yang populer dan merupakan sumber terbuka adalah R. Inovasi komputasi pada R perlu dieksplorasi lebih jauh, untuk mendalami logika pemrograman R. Penciptaan lingkungan fungsi dengan fungsi list dan pelibatan variabel / objek lokal serta global, masih sedikit mendapatkan perhatian. Berdasarkan masalah yang dirumuskan tersebut, penelitian ini mengajukan dua tujuannya yaitu (1) mengem-bangkan suatu program R yang mampu memberikan solusi interaktif dan langkah demi langkah, untuk mendapatkan solusi dari suatu sistem persamaan linear, dan (2) melakukan eksplorasi atas kemampuan R dalam membuat dan menangani variabel yang bersifat global. Program R dibuat, mulai dengan menciptakan lingkungan fungsi. Lingkungan fungsi ini diisi oleh empat fungsi terkait, yaitu "tukar", "kali", "lipat", dan "hasil". Kempat fungsi ini terhubung satu sama lain melalui suatu objek global. Pengguna dapat mengetikkan setiap fungsi untuk melakukan operasi baris / kolom, secara interaktif dan tahap demi tahap. Fungsi lingkungan dalam program ini, diberi nama OBE. Fungsi OBE menerima input berupa matriks gandengan yang diturunkan dari suatu sistem persamaan linear. Hasil akhir dari rangkain proses interaktif ini, diberikan oleh fungsi "hasil". Fungsi hasil akan menampilkan dua matriks, yaitu Matriks Asal yang merupakan input dan Matriks Setara, yang merupakan hasil dari serangkaian operasi baris / kolom.*

Kata Kunci: *Sistem Persamaan Linear; Pemrograman R; Metode Numerik; Operasi Dasar Baris; Metode Gauss-Jordan*

I. PENDAHULUAN

Operasi dasar baris / kolom merupakan topik penting yang diajarkan dalam Aljabar Linear, metodenya dikenal dengan eliminasi Gauss-Jordan. Operasi dasar baris / kolom ini digunakan untuk mencari solusi dari suatu sistem persamaan linear. Persoalan yang umum dihadapi oleh siswa / mahasiswa dalam menye-lesaikan operasi dasar baris / kolom adalah persoalan komputasinya. Saat ini memang banyak tersedia banyak aplikasi untuk melakukan komputasi tersebut. Program R pun menyediakan perintah / fungsi untuk mendapat-kan solusi dari suatu sistem persamaan linear, yaitu solve(). Hanya saja pada umumnya solusi yang diberikan adalah solusi akhir, secara langsung, bukan lengkah demi langkah. Contoh berikut ini adalah bagaimana fungsi solve(), memberikan solusi langsung untuk sistem persamaan linear berikut (persamaan (1)):

$$\begin{aligned} 2x_1 + 3x_2 - x_3 &= -4 \\ 3x_1 - 9x_2 - 3x_3 &= -12 \\ -x_1 - 3x_2 + 2x_3 &= 4 \end{aligned} \tag{1}$$

Matriks koefisien A dan vektor ruas kanan **b** disajikan oleh ekpressi (2) berikut,

$$A = \begin{pmatrix} 2 & 3 & -1 \\ 3 & -9 & -3 \\ -1 & -3 & 2 \end{pmatrix}, \qquad \boldsymbol{b} = \begin{pmatrix} -4 \\ -12 \\ 4 \end{pmatrix} \tag{2}$$

Script R untuk untuk menyelesaikan persamaan tersebut adalah,

```
A = matrix(c(2,3,-1,
             1,-9,-3,
            -1,-3,2), nrow=3,
             byrow = TRUE)
b = matrix(c(-4,-12,4), ncol=1)
solve(A,b)
```

Hasil komputasi akan ditandai oleh *highlight* berwarna abu-abu terang, untuk membedakan dengan perintah-perintah R.

```
           [,1]
[1,] -1.8461538
[2,]  0.5128205
[3,]  1.8461538
```

Jadi fungsi solve memberikan solusi untuk x_1, x_2, dan x_3 secara langsung. Hasil seperti ini untuk keperluan belajar mahasiswa, kurang diharapkan. Kebanyakan aplikasi-aplikasi matriks yang beredar dipasaran juga akan memberikan hasil langsung seperti itu. Hidayati (2019), menemukan bahwa penguasaan siswa atas topik aljabar linear adalah sangat rendah. Teknik pembelajaran interaktif, dapat membantu mengatasi hal itu. Berdasarkan masalah tersebut, penelitian ini akan merumuskan tujuannya yaitu sebagai berikut:

1. Mengembangkan program R yang mampu memberikan solusi interaktif dan langkah demi langkah, untuk mendapatkan solusi suatu sistem persamaan linear.
2. Melakukan eksplorasi atas kemampuan R dalam membuat dan menangani variabel yang berisifat global.

II. METODOLOGI

Metode yang digunakan dalam penelitian ini meliputi metode kepustakaan, dengan pende-katan deskriptif dan eksploratif, Yudistira. (2021).

Langkah-langkah dalam penelitian ini adalah:

1. Telaah pustaka untuk mendapatkan pemahaman yang lengkap terhadap penyelesaian sistem persamaan linear.
2. Telaah pustaka untuk mendapatkan pemahaman yang lengkap terhadap pemrograman R untuk mendapatkan output yang interaktif
3. Membangun algoritma atau flowchart yang menghasilkan langkah-langkah program yang sederhana
4. Membangun script R dalam bentuk yang paling sederhana
5. Memeriksa dan memvalidasi hasilnya
6. Apabila hasil verifikasi dan validasi tidak memuaskan kembali ke langkah 3)
7. Selesai

Penelitian ini membatasi komputasinya hanya untuk mendapatkan solusi numerik dari suatu sistem persamaan linear. Komputasinya sendiri tidak membatasi apakah suatu sistem linear memiliki solusi atau tidak, atau memiliki solusi banyak. Karena script R yang digunakan nanti menghasilkan komputasi yang sifatnya interaktif dan langkah demi langkah, maka pengguna dapat menghentikan langkahnya kapanpun, bila terlihat bahwa solusi yang dihasilkan adalah solusi banyak, atau bahkan tidak mempunyai solusi. Jadi hasil dari penelitian ini adalah berupa skrip R, sehingga siapapun kemudian dapat mengembangkan lebih lanjut kemampuannya.

III. HASIL DAN PEMBAHASAN

Anton, H. dan Chris Rorres (2019), membahas mengenai operasi dasar baris dari suatu matriks adalah sebagai berikut:

1. Mengalikan baris suatu matriks dengan konstanta tidak nol k
2. Menukar dua baris
3. Menambahkan suatu baris dengan kelipatan k baris lain, (k tidak sama dengan nol).

Selanjutnya ketiga langkah tersebut direpresentasikan oleh perkalian matriks elementer E. Matriks elementer ini diperoleh dengan melakukan operasi dasar baris yang bersesuaian terhadap matriks identitas I. Matriks E sebagai pengganda awal dari matriks A yang akan dikenakan operasi dasar terhadap baris-baris, yaitu berbentuk EA. Sedangkan

apabila yang dikenakan adalah operasi dasar terhadap kolom-kolom, maka E merupakan matriks pengganda akhir, yaitu bebentuk AE. Jadi tiga operasi dasar baris yang dikemukakan sebelumnya, dapat disederhanakan sebagai perkalian matriks.

1. Mengalikan baris ke-*i* suatu matriks dengan konstanta tidak nol k. Matriks E tersebut diperoleh dengan mengalikan baris ke-*i* matriks identitas (I) dengan jumlah baris sama dengan jumlah baris matriks A. Kemudian dilakukan perkalian matriks EA.
2. Menukar baris ke-i dengan baris ke-j matriks A, yaitu diperoleh dengan perkalian matriks EA, yang mana matriks E diperoleh dari matriks I, dengan baris ke-i dan baris ke-j ditukar.
3. Menambahkan baris ke-j dengan kelipatan k dari baris ke-i. Matriks E diperoleh dari matriks identitas I, yaitu dengan menam-bahkan baris ke-j dengan baris ke-i yang telah dikalikan dengan k.

Berikut ini adalah ilustrasi pada matriks E yang berukuran 3 x 3.

Operasi mengalikan baris ke-2 dengan 3, maka matriks E nya adalah:

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Operasi menukar baris ke-1 dengan baris ke-2

$$E = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

Operasi menambah baris ke-3 dengan kelipatan 2 dari baris ke-1

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix} \quad (5)$$

Matriks E yang berukuran lebih besar diperoleh dengan cara yng serupa.

Venebles, W.N., at al (2022) menjelaskan tentang perintah <<− untuk memberi nilai suatu variabel global. Variabel global dapat diubah nilainya dimanapun didalam fungsi yang ditentukan pengguna. Perintah list akan menghimpun fungsi-fungsi yang saling terkait dan terdapat dalam suatu lingkungan (environment). Ilustrasi berikut memperlihat-kan bagaimana suatu variabel diperlakukan secara global dan lokal.

1) Ciptakan suatu fungsi sebagai wadah untuk mengumpulkan beberapa fungsi dengan menggunakan perintah list.

```
g = function(x){
      list(..
```

2) Membuat fungsi-fungsi yang salaing terkait dan setiap fungsi dipisahkan dengan tanda koma.

```
    f1 = function(y){
      x <<- x + y
      cat("x= ",x,"y= ",y,"\n")
      print(y)
    },
    f2 = function(z){
      x <<- x - z
      cat("x= ",x,"z= ",z,"\n")
      print(y)
    },
    f3 = function(w) {
      x <<- x*w
      cat("x= ",x,"w= ",w,"\n")
    }
  )
}
```

3) Sekarang tercipta tiga fungsi terkait f1, f2, dan f3 dengan x sebagai variabel global. Variabel y, z, dan w merupakan variabel lokal. Oleh karena x diciptakan sebagai variabel global dengan perintah penugasan <<-, maka x dapat dipanggil dan diubah nilainya disetiap fungsi yang ada (f1, f2, atau f3).

4) Jalankan fungsi g dengan memberikan nama lingkungan fungsinya dengan nama "nilai". Ilustrasinya adalah sebagai berikut

```
nilai <- g(100)
```

fungsi f1 dipanggil pada lingkungan "nilai", yaitu sebagai berikut

```
nilai$f1(3)
```

outputnya adalah :

```
x =  103 y =  3
     [1] 3
```

fungsi f3 dipanggil pada lingkungan "nilai", yaitu sebagai berikut

```
nilai$f3(3)
```

outputnya adalah :

```
x =  309 w =  3
```

fungsi f2 dipanggil pada lingkungan "nilai", yaitu sebagai berikut

```
nilai$f2(109)
```

ouputnya adalah

```
x =  200 z =  109
Error in print(y) : object 'y' not found
```

Fungsi f2 memanggil nilai y, melalui perintah print(y), nilai y walaupun ada pada fungsi f1, tetapi tidak dikenal oleh fungsi f2, karena y bersifat lokal. Berbeda dengan x yang merupakan variabel global, akan dikenal oleh fungsi manapun dalam lingkungan "nilai".

3.1 Proses Pembuatan Program

Berdasarkan ilustrasi di atas, akan dikembang-kan, program R untuk pengolahan dasar baris / kolom yang interaktif. Tahapam-tahapannya adalah sebagai berikut:

1. Buat fungsi untuk membangun lingkungan fungsi-fungsi pengolahan dasar baris / kolom. Fungsi ini diberi nama OBE. Fungsi tersebut adalah sebagai berikut:

```
OBE<-function(x, baris=TRUE, r=5)
{  # r pembulatan
list(
x.asal<<-x,
```

   Fungsi OBE mempunyai tiga argumen, yaitu "x" untuk matriks yang akan dikena-kan operasi dasar baris / kolom. Argumen "baris" diberi nilai terpasang TRUE, yang artinya, operasi ini dikenakan untuk baris-baris, dan FALSE sebaliknya untuk kolom-kolom. Argumen r dengan nilai terpasang sama dengan 5, disediakan untuk pembulatan sampai 5 digit.

2. Membuat fungsi-fungsi yang saling terkait, pada lingkungan OBE. Ada tiga fungsi yang akan dibuat pada pengolahan dasar baris. Fungsi pertama adalah fungsi "tukar". Fungsi ini untuk menukar baris ke-i dan ke-j dari matriks yang disimpan oleh x. Fungsinya itu adalah sebagai berikut:

```
## Fungsi pertukaran baris (kolom) ## i
dan j, i <-> j
    tukar = function(i, j) {
        if (is.matrix(x)){
          if (baris) {
          ind <- diag(nrow(x))
          }
          else ind <- diag(ncol(x))
        }
        else stop("Objek harus
           dalam mode matriks")

        te<-ind[i,]
        ind[i,]<-ind[j,]
        ind[j,] <- te

       if (baris) {
         x <<- ind%*%x
       }
       else x <<- x%*%ind
       return(x)
   },
```

   Fungsi "tukar" mempunyai dua agumen yaitu i dan j, yang mangacu pada nomor baris / kolom yang akan dipertukarkan. Pertama-tama fungsi akan memeriksa apakah objek "x" merupakan matriks, dengan menggunakan perintah is.matrix(), jika "x" bukan matriks, maka proses aka dihentikan, dengan menggunakan fungsi stop(). Apabila objek "x" adalah matriks, maka proses dilanjutkan dengan memeriksa apakah nilai "baris" sama dengan TRUE atau FALSE. Nilai TRUE / FALSE, maka ukuran matriks identitasnya berdasarkan ukuran / jumlah baris / kolom dari matriks pada objek "x". Perintah diag(nrow(x)), akan menciptakan matriks identitas berukuran sama dengan jumlah baris matriks "x", sedangkan diag(ncol(x)) ukuran matriks identitasnya sama dengan jumlah kolom matriks "x". Hal ini dilakukan sesuai dengan, apakah yang diinginkan adalah operasi dasar baris atau kolom. Proses pertukaran itu dilakukan oleh serangkain perintah

```
te <- ind[i,]

ind[i,] <- ind[j,]

ind[j,] <- te.
```

   Hasil akhir dari proses ini adalah matriks elementer E, yang mana dalam proses ini diberi nama "ind". Kemudian Proses selan-jutnya adalah kembali memeriksa nilai "baris" apakah TRUE / FALSE, yang akan menentukan apakah "ind" sebagai peng-ganda awal / akhir, yaitu `ind%*%x` atau `x%*%ind`. Hasil perkalian matriks ini dismpan dalam objek "x" secara global, yaitu dengan menggunakan operator penugasan <<-. Penyimpanan secara global dimaksudkan agar matriks "x" ini dapat digunakan kembali oleh fungsi-fungsi lain dalam lingkungan OBE. Hasil akhir dari fungsi "tukar" ini adalah matriks "x" yang baris / kolom ke-i dan ke-j telah dipertukarkan dan bersifat global.

3. Membuat fungsi saling terkait yang kedua. Fungsi ini diberi nama "kali", yang digunakan untuk mengalikan sebuah baris / kolom ke-i dengan skalar k yang tidak nol. Fungsinya adalah sebagai berikut:

```
## Fungsi perkalian baris/kolom
## dengan skalar k. i <- (k)i
    kali = function(i, k) {
      if (is.matrix(x)){
        if (baris) {
          ind <- diag(nrow(x))
        }
        else ind <- diag(ncol(x))
      }
      else stop("Objek harus dalam
                 mode matriks")

      ind[i,] <- k*ind[i,]
      if (baris) x <<- ind %*% x
      else x <<- x %*% ind
      return(x)
    },
```

   Fungsi "kali" memiliki argumen i dan k. Argumen i untuk nomor baris / kolom yang akan dikalikan oleh skalar k yang tidak nol. Sebagaimana pada fungsi "tukar", juga dilakukan pemeriksaan apakah objek "x" berkelas matriks. Kemudian dilanjutkan dengan proses membuat matriks identitas berdasarkan jumlah baris / kolom, tergantung apakah operasi dasar yang diinginkan, apakah berdasarkan baris / kolom. Proses inti pada fungsi ini adalah perintah `ind[i,]<-k*ind[i,]`, yaitu proses menciptakan matriks

elementer, dengan mengalikan baris ke-i dari matriks identitas dengan skalar k. Variabel global "x" yang dihasilkan adalah matriks "x" yang baris ke-i nya sudah dikalikan k. Karena bersifat global, maka matriks "x" ini dapat dipanggil kembali oleh fungsi lainnya, dalam lingkungan OBE.

4. Membuat fungsi terkait yang ketiga dan diberi nama "lipat". Fungsi ini untuk melakukan operasi dasar yang ketiga, yaitu menjumlahkan baris / kolom ke-j dengan kelipatan k dari baris ke-i suatu matriks. Adapun script R untutk fungsi ini adalah:

```
## Fungsi untuk menjumlahan baris
## (kolom) j dengan kelipatan k dari
## baris (kolom) i, j <- j + (k)i
    lipat = function(i, j, k) {
      if (is.matrix(x)){
        if (baris) {
          ind <- diag(nrow(x))
        }
        else ind <- diag(ncol(x))
      }
      else stop("Objek harus dalam
                 mode matriks")
      if (baris) {
        ind[j,i] <- k
        x <<- ind %*% x}
      else {
        ind[i,j] <- k
        x <<- x %*% ind}
      return(x)
    },
```

5. Merancang output dari fungsi OBE, yang terdiri dari matriks asal / awal, dan matriks setara yang merupakan hasil operasi baris / kolom. Mariks setara yang dihasilkan dibulatkan sampai r tempat desimal (nilai terpasang r adalah 5). Pada bagaian awal dari fungsi OBE, telah ditetapkan objek / variabel "x.asal", yang menyimpan matriks pada objek "x" untuk pertama kali, yaitu sebagai berikut:

```
OBE<-function(x, baris=TRUE, r=5)
{  # r pembulatan
List( x.asal<<-x,
```

Jadi objek "x.asal" dibuat bersifat global, sehingga dapat dipanggil oleh fungsi yang akam memberikan output untuk OBE ini. Fungsi output ini diberi nama "hasil", yang secara lengkap ditulis sebagai berikut:

```
## Hasil pengolahan dasar baris
## (kolom)
hasil = function(){
  x.setara <- x
  has<-list(Matriks.Asal=x.asal,
   Matriks.Setara=round(x.setara,
      r))
   return(has)
    }
  )
}
```

Pada baris-baris awal dari fungsi "hasil", objek global "x", yang merupakan hasil akhir operasi dasar, disimpan sebagai objek lokal dengan nama "x.setara". Objek "x.setara" ini tidak perlu dibuat bersifat global, karena tidak akan dilakukan pengolahan oleh fungsi lain diluar fungsi "hasil". Kemudian diciptakan sebuah objek baru dengan nama "has", yang akan mengumpulkan objek-objek "x.asal" dan "x.setara". Nilai objek "x.asal" disimpan dengan nama "Matriks.Asal", sedangkan "x.setara" setelah dibulatkan sampai "r" tempat desimal (r nilai terpasangnya dibuat sama dengan 5), disimpan dengan nama "Matriks. Setara". Objek "has" ini dibuat berkelas list. Perintah return akan mengembalikan nilai kedua objek, yaitu "Matrisk. Asal" dan "Matriks.Setara".

3.2 Proses Pemeriksaan Programn (Verifikasi)

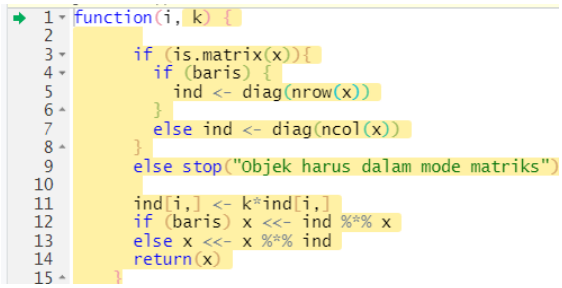
Karena R adalah bahasa pemrograman yang berbasis *interpreter*, maka proses akan berjalan langkah demi langkah, dari atas ke bawah dan dari kiri ke kanan. Sifat ini tentunya akan memudahkan dalam melakukan verifikasi atas jalannya proses. Proses verifikasi sudah dimulai sejak skrip R mulai dibuat. Agar memudahkan penelusuran, R menyediakan fungsi debug() yang dapat digunakan untuk menelusuri setiap baris perintah langkah demi langkah. Apabila timbul "error", maka fungsi ini akan memberitahu posisi baris perintah, dimana "error" tersebut terjadi. Pada R-Studio dutandai dengan "highlight" berwarna. Ilustrasi berikut memberikan gambaran bagaimana fungsi "kali" dijalankan secara langkah demi langkah.

```
# menyiapkan input untuk argumen x pada
OBE

A = matrix(c(2, 3,-1,
             1,-9,-3,
            -1,-3,2),nrow=3,
                  byrow=TRUE)

# operasi terhadap kolom
OprKolom = OBE(A, baris = FALSE)
debug(OprKolom$kali)
# kolom 2 matriks A dikalikan 10
OprKolom$kali(2, 10)
```

Gambar 1 di bawah ini menampilkan penampakan pada R-Studio, dimana perintah-perintah fungsi "kali" ditampilkan.



Gambar 1. Hasil debugging dengan fungsi debug(OprKali)

Untuk mengakhiri proses debugging, dilakukan dengan perintah,

```
undebug(OprKolom$kali)
```

Setelah dilakukan proses verifikasi dengan menggunakan fungsi debug(), program OBE ini memberikan hasil yang baik, dan berjalan sesuai dengan yang diharapkan.

3.3 Proses Validasi

Validasi dilakukan untuk melihat apakah proses komputasi bekerja sesuai dengan yang diingin-kan, yang mana ditunjukkan oleh output fungsi OBE yang benar. Proses ini dilakukan dengan mengambil teladan soal dari Anton, H. (2019). Teladan soal tentunya sudah meyediakan jawaban dari pertanyaan yang diajukan, yang telah diselesaikan secara manual. Kemudian jawaban tersebut, dibandingkan dengan hasil yang diperoleh dari program R interaktif yang dikembangkan dalam penenlitian ini. Output program ini, diharapkan mendapatkan hasil yang sama dari penyelesaian manualnya.

Sistem persamaan linear yang akan dicari solusinya, dalam bentuk matriks adalah sebagai berikut:

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$$A = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 4 & -3 \\ 3 & 6 & -5 \end{pmatrix}, \text{dan } \boldsymbol{b} = \begin{pmatrix} 9 \\ 1 \\ 0 \end{pmatrix} \quad (6)$$

Matriks gandengan yang akan dikenakan operasi dasar baris adalah,

$(A|\boldsymbol{b}) = \left(\begin{array}{ccc|c} 1 & 1 & 2 & 9 \\ 2 & 4 & -3 & 1 \\ 3 & 6 & -5 & 0 \end{array}\right)$, matriks ini berukur-an 3×4.

Matriks gandengan ini selajutnya, dipakai sebagai input dari fungsi OBE untuk diolah lebih lanjut, dan diberi nama *Ab*.

```
Ab = matrix(c(1, 1, 2, 9,
              2, 4,-3, 1,
              3, 6,-5, 0), nrow=3,
                    byrow=T )
Ab
     [,1] [,2] [,3] [,4]
[1,]    1    1    2    9
[2,]    2    4   -3    1
[3,]    3    6   -5    0
```

1) Tambahkan baris 2 (j) dengan kelipatan -2 (k) dari baris 1 (i) matriks **Ab**.

```
OprBaris = OBE(Ab, baris=TRUE)

## i = 1, j = 2, dan k = -2
i = 1; j = 2; k = -2
OprBaris$lipat(i, j, k)
```

Hasilnya adalah,

```
     [,1] [,2] [,3] [,4]
[1,]    1    1    2    9
[2,]    0    2   -7  -17
[3,]    3    6   -5    0
```

2) Tambahkan baris 3 (j) dengan kelipatan -3 (k) dari baris 1 (i) matriks **Ab**.

```
## i = 1, j = 3, dan k = -3
i = 1; j = 3; k = -3
OprBaris$lipat(i, j, k)
```

Hasilnya adalah,

```
     [,1] [,2] [,3] [,4]
[1,]    1    1    2    9
[2,]    0    2   -7  -17
[3,]    0    3  -11  -27
```

3) Kalikan baris 2 (i) dengan ½ (k)

```
i = 2; k = 1/2
OprBaris$kali(i, k)
```

Hasilnya adalah,

```
     [,1] [,2]  [,3]  [,4]
[1,]    1    1   2.0   9.0
[2,]    0    1  -3.5  -8.5
[3,]    0    3 -11.0 -27.0
```

4) Tambahkan baris 3 (j) dengan kelipatan -3 (k) dari baris 2 (i) matriks **Ab**

```
i = 2; j = 3; k = -3
OprBaris$lipat(i, j, k)
```

Hasilnya adalah,

```
     [,1] [,2] [,3] [,4]
[1,]    1    1  2.0  9.0
[2,]    0    1 -3.5 -8.5
[3,]    0    0 -0.5 -1.5
```

5) Kalikan baris 3 (i) dengan -2 (k)

```
i = 3; k = -2
OprBaris$kali(i, k)
```

Hasilnya adalah,

```
     [,1] [,2] [,3] [,4]
[1,]    1    1  2.0  9.0
[2,]    0    1 -3.5 -8.5
[3,]    0    0  1.0  3.0
```

6) Langkah selanjutnya adalah penyapuan ke atas, yaitu menambahkan baris 2 (j) dengan kelipatan 3.5 (k) dari baris 3 (i) matriks **Ab**.

```
i = 3; j = 2; k = 3.5
OprBaris$lipat(i, j, k)
```

Hasilnya adalah,

```
     [,1] [,2] [,3] [,4]
[1,]    1    1    2    9
[2,]    0    1    0    2
[3,]    0    0    1    3
```

7) Tambahkan baris 1 (j) dengan kelipatan -2 (k) dari baris 3 (i) matriks **Ab**

```
i = 3; j = 1; k = -2
OprBaris$lipat(i, j, k)
```

Hasilnya adalah,

```
     [,1] [,2] [,3] [,4]
[1,]    1    1    0    3
[2,]    0    1    0    2
[3,]    0    0    1    3
```

8) Tambahkan baris 1 (j) dengan kelipatan -1 (k) dari baris 2 (i) matriks **Ab.**

```
i = 2; j = 1; k = -1
OprBaris$lipat(i, j, k)
```

Hasilnya adalah,

```
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    1
[2,]    0    1    0    2
[3,]    0    0    1    3
```

Jadi diperoleh . Hasil ini sesuai dengan perhitungan manual.

Fungsi solve() juga memberikan hasil yang sama, sebagai berikut:

```
A <- Ab[,1:3]; b <- Ab[,4]
solve(A,b)
[1] 1 2 3
```

Percobaan dilakukan untuk berbagai macam matriks gandengan, dengan berbagai kemung-kinan solusi, yaitu yang tanpa solusi, solusi banyak, maupun solusi unik seperti pada kasus di atas, Yoshida (2021). Rangkaian proses interaktif berikut menghasilkan jawaban tanpa solusi, untuk suatu sistem linear

```
Av = matrix(c(1, 1, 1, 2,
              0, 1,-3, 1,
              2, 1, 5, 0), nrow=3,
                  byrow=TRUE)
OprBaris2 = OBE(Av, baris=TRUE)
## i = 1, j = 3, dan k = -2
i = 1; j = 3; k = -2
OprBaris2$lipat(i, j, k)
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    2
[2,]    0    1   -3    1
[3,]    0   -1    3   -4
```

```
## i = 2, j = 3, dan k = 1
i = 2; j = 3; k = 1
OprBaris2$lipat(i, j, k)
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    2
[2,]    0    1   -3    1
[3,]    0    0    0   -3
```

Apabila menggunakan fungsi solve, hasilnya adalah:

```
A = Av[, 1:3]; v = Av[, 4]
solve(A, v)
Error in solve.default(A, v) :
  Lapack routine dgesv: system is exactly
singular: U[3,3] = 0
```

Fungsi solve tidak dapat digunakan bila matriks koefisiennya singular. Oleh karena itu fungsi interkatif OBE ini dapat sebagai alternatif, untuk menentukan solusi suatu sistem persamaan linear.

IV. KESIMPULAN

Program R yang dihasilkan untuk melakukan komputasi matriks, khususnya dalam menjalan-kan operasi dasar baris / kolom, berhasil dibuat dengan memanfaatkan fungsi list(), yang dipakai untuk menciptakan suatu lingkungan fungsi-fungsi. Agar suatu objek dapat dibaca oleh fungsi-fungsi yang selingkungan, maka digunakan perintah penugasan <<- yang menciptakan objek global. Lingkungan yang diciptakan mengandung fungsi-fungsi terkait, yang melakukan operasi dasar baris / kolom.

Dengan tersedianya suatu aplikasi interaktif untuk mendapatkan solusi dari suatu sistem persamaan linear, akan membantu mahasiswa dalam memahami subjek tentang sistem persamaan linear.

Program R interaktif untuk operasi ini, juga mampu menentukan apakah suatu sistem linear mempunyai solusi atau tidak, solusi unik atau solusi banyak. Fungsi terpasang solve pada base, akan menghasilkan “error” apabila sistem tidak mempunyai solusi.

Program R yang dihasilkan sudah melalui proses verifikasi dan validasi, yang hasilnya dapat dipertanggung jawabkan. Proses verifikasi dilakukan dengan bantuan fungsi debug. Fungsi ini untuk menelusuri baris-baris perintah R yang telah disusun. Sedangkan proses validasi dilakukan dengan membanding-kan hasil komputasi oleh program R yang dibuat dalam penelitian ini, dengan perhitungan manualnya. Proses perbandingan ini dilakukan berulang-ulang dengan berbagai kemungkinan input matriks gandengan. Semua hasilnya sesuai dengan hasil perhitungan manual. Tentu saja ukuran matriks gandengan bisa diperbesar sesuai dengan masalah sistem linear yang dihadapi.

DAFTAR PUSTAKA

Anton, H., & Anton Kaul. (2019). *Elementary Linear Algebra, 12th Edition*. John Wiley & Sons, Inc. New York, USA.

Berkelaar, M. et al. *lpsolve package*, (2020). *https://cran.r-project.org/web/packages/lpSolve/index.html*

Cut Intan Salasiyah. (2017). *Pengembang-an Modul Aljabar Linear Untuk Mahasiswa Pendidikan Matemati-ka.* Al-Khawarizmi: Jurnal Pendi-dikan dan Pembelajaran Matemati-ka. E-ISSN 2549-3914. Halaman. 146-156. Vol. 1, No. 2, Desember 2017.

Dona Fitriawan. (2020). *Pengembangan Bahan Ajar Aljabar Linear Elementer Berdasarkan Kemam-puan Koneksi Matematis.* Jurnal Pendidikan Matematika dan IPA. Vol. 11, No. 2 (2020) h. 217-229. *https://jurnal.untan.ac.id/index.php/PMP*

Fawzi, Alhussein. at al (2022). *Discover-ing faster matrix multiplication algorithms with reinforcement learning.* Nature. 610, pages 47–53 (2022). *https://www.nature.com/articles/s41586-022-05172-4.*

Fieller, Nick. (2018). *Basics of Matrix Algebra for Statistics with R*. CRC Press, Taylor and Francis Group. New York. *https://xn--webducation-dbb.com/wp-content/uploads/2018/07/Basics-of-Matrix-Algebra-for-Statistics-with-R.pdf*

Friendly, Michael. (2022). *Package 'matlib'*: *Matrix Functions for Teaching and Learning Linear Algebra and Multivariate Statistics*. *https://cran.r-project.org/web/packages/matlib/matlib.pdf*

Gharib, S., Syeda Roshana Ali, Rabia Khan, Nargis Munir & Memoona Khanam. (2015). *System of Linear Equations, Guassian Elimination*. Global Journal of Computer Science and Technology: C Software & Data Engineering. Volume 15 Issue 5 Version 1.0. ISSN: 0975-4172

Hidayati, V R., M A Maulyda, G Gunawan, A N Rahmatih, and M Erfan. (2019). *System of Linear Equation Problem Solving: Descriptive Study about Students' Mathematical Connection Ability*. The 1st Progress in Science and Technology Research Symposium (PSTRS). Journal of Physics: Conference Series.

Ruswana, A. M. (2019). *Analisis Kemam-puan Pemahaman Matematis pada Kuliah Aljabar Linier Elementer*. Journal Cendekia: Jurnal Pendidikan Matematika. Volume 03, No. 02, Agustus 2019, pp. 293-299

Venables, W.N., D.M. Smith, and the R Core Teams. (2022). *An Introduction to R: Notes on R: A Programming Environment for Data Analysis and Graphics Version 4.2.2 (2022-10-31*). *The Comprehensive R Archive Network (r-project.org)*

Wilda Mahmudah. (2020). *Analisis Kesalahan dalam Menyelesaikan Soal Sistem Persamaan Linear pada Aljabar Linier Elementer.* Briliant: Jurnal Riset dan Konseptual. Vol. 5, No. 3 (2020) h. 449-456. *http://jurnal.unublitar.ac.id/index.php/briliant/article/view/463/pdf*

Xingping Sheng. (2014). *Execute Elementary Row and Column Operations on thePartitioned Matrix to Compute M-P Inverse A*. Hindawi Publishing Corporation Abstract and Applied Analysis Vol 2014, Article ID 596049, 6 pages *https://www.researchgate.net/publication/275556465_Execute_Elementary_Row_and_Column_Operations_on_the_Partitioned_Matrix_to_Compute_M-P_Inverse*

Yoshida, Ruriko. (2021). *Linear Algebra and Its Applications with R*. Taylor & Francis Group, LLC.

Yudistira, I G.A. Anom. (2021). *Pengembang-an Simulasi Kejadian Diskret Berbasis Paket Simmer pada R*. Jurnal EMACS (Engineering Mathematics and Computer Science). Vol 3 No. 2 May 2021: 79-85.