

## 2.3 Fungsi eksternal perkalian matrik

Mari kita beralih ke perkalian matrik. Kita akan membuat fungsi eksternal untuk perkalian matrik. Berikut ini adalah *source code* perkalian matrik hasil akhir optimasi yang telah ditulis panjang lebar pada bab sebelumnya

```
clear all
clc

A = [3 8 5; 6 4 7]; % inisialisasi matrik A
B = [1 3; 5 9; 2 4]; % inisialisasi matrik B

% ---proses perkalian matrik---
for i=1:2
    for j=1:2
        E(i,j)=0;
    end
end

for i=1:2
    for j=1:2
        for k=1:3
            E(i,j)=E(i,j)+A(i,k)*B(k,j);
        end
    end
end

% ---menampilkan matrik A, B dan E---
A
B
E
```

*Source code* tersebut digunakan untuk menghitung perkalian matrik berikut

$$E_{2 \times 2} = A_{2 \times 3} \cdot B_{3 \times 2}$$

Dan kita bisa sepakati simbol indeks  $m$ ,  $n$ , dan  $p$  untuk men-generalisir dimensi matrik

$$E_{m \times n} = A_{m \times p} \cdot B_{p \times n}$$

Dengan demikian, *source code* tersebut dapat dioptimasi menjadi

```
clear all
clc

A = [3 8 5; 6 4 7; 3 8 5]; % inisialisasi matrik A
B = [1 3 1; 5 9 2; 2 4 3]; % inisialisasi matrik B

% ---proses perkalian matrik---
dim=size(A);
m=dim(1);
p=dim(2);
dim=size(B);
n=dim(2);
for i=1:m
    for j=1:n
        E(i,j)=0;
    end
end

for i=1:m
    for j=1:n
```

```

                for k=1:p
                    E(i,j)=E(i,j)+A(i,k)*B(k,j);
                end
            end
        end

% ---menampilkan matrik A, B dan E----
A
B
E

```

Selanjutnya kita ambil bagian *proses perkalian matrik* nya untuk dibuat fungsi eksternal

```

function E=kali(A,B)
dim=size(A);
m=dim(1);
p=dim(2);
dim=size(B);
n=dim(2);
for i=1:m
    for j=1:n
        E(i,j)=0;
    end
end

for i=1:m
    for j=1:n
        for k=1:p
            E(i,j)=E(i,j)+A(i,k)*B(k,j);
        end
    end
end
end

```

lalu di-save dengan nama *kali.m*, maka terciptalah fungsi eksternal yang bernama fungsi *kali*. Kemudian coba anda uji fungsi *kali* tersebut dengan menjalankan *source code* berikut

```

clear all
clc

A = [3 8 5; 6 4 7]; % inisialisasi matrik A
B = [1 3; 5 9; 2 4]; % inisialisasi matrik B

% ---proses perkalian matrik----
E = kali(A,B)

% ---menampilkan matrik A, B dan E----
A
B
E

```

Silakan anda periksa hasil perhitungannya. Pasti betul! Anda bisa mencoba perkalian matrik lainnya dengan menggunakan *source code* tersebut. Bahkan anda bisa mengganti namamatriknya untuk selain **A**, **B** dan **E**.

## 2.4 Fungsi eksternal perkalian matrik dan vektor-kolom

Mari kita beralih ke perkalian matrik dan vektor-kolom. Kita akan membuat fungsi eksternal untuk perkalian matrik dan vektor-kolom. Berikut ini adalah *source code* perkalian matrik dan vektor-kolom hasil akhir optimasi yang telah ditulis panjang lebar pada bab sebelumnya

```
clear all
clc

A = [3 8 5; 6 4 7]; % inisialisasi matrik A
x = [2; 3; 4]; % inisialisasi vektor x

% ---proses perkalian matrik dan vektor---
for i=1:2
    y(i,1)=0;
end

for i=1:2
    for j=1:3
        y(i,1)=y(i,1)+A(i,j)*x(j,1);
    end
end

% ---menampilkan matrik A, B dan E---
A
x
y
```

| Tes Membuat Fungsi |          |                  |
|--------------------|----------|------------------|
| No                 | NIM      | Nama             |
| 1                  | 13113004 | Dimas Trilaksono |
| 2                  | 13113015 | Aditya M N       |
| 3                  | 13113029 | Ricko            |
| 4                  | 13113022 | Canis            |
| 5                  | 13113030 | Zenaldi          |

*Source code* tersebut digunakan untuk menghitung perkalian matrik dan vektor-kolom berikut

$$\mathbf{y}_{2 \times 1} = \mathbf{A}_{2 \times 3} \cdot \mathbf{x}_{3 \times 1}$$

Dan kita bisa sepakati simbol indeks  $m$  dan  $n$  untuk men-generalisir dimensi matrik

$$\mathbf{y}_{m \times 1} = \mathbf{A}_{m \times n} \cdot \mathbf{x}_{n \times 1}$$

Dengan demikian, *source code* tersebut dapat dioptimasi menjadi

```
clear all
clc

A = [3 8 5; 6 4 7]; % inisialisasi matrik A
x = [2; 3; 4]; % inisialisasi vektor x

% ---proses perkalian matrik dan vektor---
dim=size(A);
m=dim(1);
n=dim(2);
for i=1:m
    y(i,1)=0;
end

for i=1:m
    for j=1:n
        y(i,1)=y(i,1)+A(i,j)*x(j,1);
    end
end

% ---menampilkan matrik A, B dan E---
A
x
y
```

```
A
x
y
```

Selanjutnya kita ambil bagian *proses perkalian matrik dan vektor* nya untuk dibuat fungsi eksternal

```
function y=kalivektor(A,x)
dim=size(A);
m=dim(1);
n=dim(2);
for i=1:m
    y(i,1)=0;
end

for i=1:m
    for j=1:n
        y(i,1)=y(i,1)+A(i,j)*x(j,1);
    end
end
```

lalu di-save dengan nama *kalivektor.m*, maka terciptalah fungsi eksternal yang bernama *fungsi kalivektor*. Kemudian coba anda uji fungsi *kalivektor* tersebut dengan menjalankan *source code* berikut

```
clear all
clc

A = [3 8 5; 6 4 7]; % inisialisasi matrik A
x = [2; 3; 4]; % inisialisasi vektor x

% ---proses perkalian matrik dan vektor---
y = kalivektor(A,x);

% ---menampilkan matrik A, vektor X dan hasil perkalian Y---
A
x
y
```

Silakan anda periksa hasil perhitungannya. Pasti betul! Anda bisa mencoba perkalian matrik dan vektor-kolom dengan angka elemen yang berbeda menggunakan *source code* tersebut. Bahkan anda bisa mengganti nama matrik dan vektor nya untuk selain **A**, **x** dan **y**.

## 2.5 Penutup

Ada tiga pilar yang harus dikuasai oleh seorang calon *programmer*. Pertama, ia harus tahu bagaimana cara mendeklarasikan data. Kedua, ia harus tahu bagaimana mendayagunakan

*flow-control*, yang dalam bab ini dan bab sebelumnya menggunakan pasangan *for-end*. Dan ketiga, ia harus bisa membuat fungsi eksternal.

Sesungguhnya Matlab memiliki banyak fungsi internal yang bisa langsung dipakai. Anda bisa coba sendiri suatu saat nanti. Kekuatan bahasa pemrograman salah satunya terletak padaseberapa kaya dia memilik banyak fungsi. *Library* adalah kata lain untuk fungsi. Jadi, suatubahasa pemrograman akan semakin unggul bila diamemiliki semakin banyak *library*. Menurut saya, yang terdepan saat ini masih dimenangkan oleh Python. Dengan Python, *source code* anda akan bisa berjalan di Windows, Linux dan Macintosh serta beberapa platform lainnya.