

Bab 2

Fungsi

Objektif :

- <Mengenalkan fungsi internal.
- <Membuat fungsi eksternal.
- <Membuat fungsi eksternal untuk penjumlahan matrik.
- <Membuat fungsi eksternal untuk perkalian matrik.

2.1 Fungsi internal

Pada bab terdahulu kita sudah melakukan proses optimasi penjumlahan matrik dengan *sourcecode* akhir seperti ini

```
clear all
clc

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

% ---proses penjumlahan matrik---
for i=1:2
    for j=1:3
        D(i,j)=A(i,j)+C(i,j);
    end
end

% ---menampilkan matrik A, C dan D---
A
C
D
```

Pertanyaan yang segera muncul adalah apakah *source code* tersebut bisa digunakan untuk menyelesaikan penjumlahan matrik yang dimensinya bukan 2x3 ? Misalnya

$$\mathbf{D} = \mathbf{A} + \mathbf{C}$$
$$\mathbf{D} = \begin{bmatrix} 4 & 3 & 8 & 6 \\ 5 & 1 & 2 & 3 \\ 6 & 7 & 9 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 6 & 7 & 2 \\ 9 & 1 & 3 & 8 \\ 5 & 8 & 4 & 7 \end{bmatrix}$$

Tentu saja bisa, asal indeks *i* bergerak dari 1 sampai 3 dan indeks *j* bergerak dari 1 sampai 4.

Lihat *source code* berikut

```
clear all
clc

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

% ---proses penjumlahan matrik---
for i=1:3
    for j=1:4
        D(i,j)=A(i,j)+C(i,j);
    end
end

% ---menampilkan matrik A, C dan D---
A
```

C
D

Walaupun bisa digunakan, namun cara modifikasi seperti itu sangat tidak fleksibel dan beresikosalah jika kurang teliti. Untuk menghindari resiko kesalahan dan agar lebih fleksibel, *source code* tersebut perlu dioptimasi sedikit lagi menjadi

```
clear all
clc

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

% ---proses penjumlahan matrik----
dim=size(A);
n=dim(1);
m=dim(2);
for i=1:n
    for j=1:m
        D(i,j)=A(i,j)+C(i,j);
    end
end

% ---menampilkan matrik A, C dan D----
A
C
D
```

Perhatikan, ada tambahan 3 statemen yaitu mulai dari baris ke-8 sampai ke-10. Sementarabaris ke-11 dan ke-12 hanya mengalami sedikit perubahan. Statemen di baris ke-8 bermaksudmendeklarasikan variabel *dim* untuk diisi oleh hasil perhitungan fungsi internal yang bernama *size*. Matrik **A** dijadikan parameter input fungsi *size*. Fungsi *size* berguna untuk menghitungjumlah baris dan jumlah kolom dari matrik **A**. Hasilnya adalah *dim(1)* untuk jumlah barisdan *dim(2)* untuk jumlah kolom. Pada baris ke-9, variabel *n* dideklarasikan untuk menerimainformasi jumlah baris dari *dim(1)*, sementara variabel*m* diisi dengan informasi jumlah kolomdari *dim(2)* pada baris ke-10. Adapun baris ke-11 dan ke-12 hanya mengubah angka indeksbatas atas, masing-masing menjadi *n* dan *m*.Sekarang kalau kita balik lagi menghitung penjumlahan matrik dari contoh sebelumnya yang berukuran 2x3, maka *source code* akan seperti ini

```
clear all
clc

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

% ---proses penjumlahan matrik----
dim=size(A);
n=dim(1);
m=dim(2);
for i=1:n
    for j=1:m
        D(i,j)=A(i,j)+C(i,j);
    end
end

% ---menampilkan matrik A, C dan D----
A
C
```

D

Ajaib bukan!? Tidak ada statemen yang berubah kecuali hanya pada baris ke-4 dan ke-5. Perubahan itu tidak bisa dihindari karena memang di kedua baris itulah deklarasi elemen-elemen matrik **A** dan matrik **C** dilakukan.

2.2 Fungsi eksternal penjumlahan matrik

Saatnya kita memasuki topik tentang pembuatan fungsi eksternal. Dari *source code* yang terakhirtadi, mari kita ambil bagian proses penjumlahan matrik-nya saja

```
dim=size(A);
n=dim(1);
m=dim(2);
for i=1:n
    for j=1:m
        D(i,j)=A(i,j)+C(i,j);
    end
end
```

Kita akan jadikan potongan *source code* ini menjadi fungsi eksternal, dengan menambahkan statemen *function* seperti ini

```
function D=jumlah(A,C)
dim=size(A);
n=dim(1);
m=dim(2);
for i=1:n
    for j=1:m
        D(i,j)=A(i,j)+C(i,j);
    end
end
```

kemudian ia harus di-save dengan nama *jumlah.m*. Sampai dengan langkah ini kita telah membuat fungsi eksternal dan diberi nama fungsi *jumlah*. Sederhana sekali bukan? Untuk mengujikerja fungsi eksternal tersebut, coba jalankan *source code* berikut ini

```
clear all
clc

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

% ---proses penjumlahan matrik---
D=jumlah(A,C)

% ---menampilkan matrik A, C dan D---
A
C
D
```

atau anda jalankan *source code* yang berikut ini

```
clear all
clc

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

% ---proses penjumlahan matrik---
D=jumlah(A,C)

% ---menampilkan matrik A, C dan D---
```

A
C
D

atau coba iseng-iseng anda ganti matrik-nya menjadi

```
clear all
clc

V=[4 3; 5 1]; % inisialisasi matrik V
W=[2 6; 9 3]; % inisialisasi matrik W

% ---proses penjumlahan matrik---
U=jumlah(V,W)

% ---menampilkan matrik V, W dan U---
W
V
U
```

Periksa hasilnya, betul atau salah? Pasti betul! Kesimpulannya adalah setelah fungsi eksternal berhasil anda dapatkan, maka seketika itu pula anda tidak perlu menggubrisnya lagi. Bahkan anda tidak perlu mengingat nama matrik aslinya yang tertulis di fungsi *jumlah* yaitu matrik **A**, matrik **C** dan matrik **D**. Ditambah lagi, *source code* anda menjadi terlihat lebih singkat dan elegan. Dan kini, perhatian anda bisa lebih difokuskan pada deklarasi matrik-nya saja.