

Organisasi Sistem Komputer

Bab 7. Subprogram

7.1 Subprogram 101

7.2 Variabel Lokal dari Subprogram



Skeleton Subprogram

func:

```
    push    ebp                ; simpan EBP dari pemanggil
    mov     ebp, esp          ; set EBP = ESP

    ...                       ; kode-kode subprogram

    pop     ebp                ; kembalikan EBP pemanggil
    ret
```



Memanggil Subprogram

```
; memanggil subprogram func
```

```
push    dword    23        ; parameter ke-dua  
push    dword    104       ; parameter ke-satu  
call    func        ; panggil subprogram  
add     esp, 8         ; pop kedua parameter
```



Stack pada Subprogram

```
...
; push parameter-parameter
● push dword 23    ; parameter 2
  push dword 104   ; parameter 1

  call f_add       ; panggil subprogram f_add

  add esp, 8       ; bersihkan stack dari parameter

; nilai kembali di EAX
...
;; subprogram f_add
f_add:
  push ebp         ; simpan nilai ebp lama ke stack
  mov  ebp, esp    ; set ebp = esp

; Kode utama subprogram
  mov  ebx, [ebp + 8]
  mov  eax, [ebp + 12]
  add  eax, ebx

  pop  ebp        ; kembalikan ebp pemanggil
  ret             ; kembali ke pemanggil
```

ESP →

23

Stack pada Subprogram

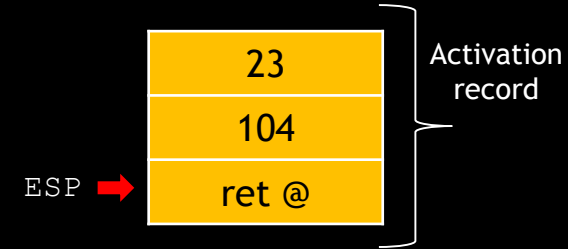
```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
● push dword 104  ; parameter 1  
  
call f_add       ; panggil subprogram f_add  
  
add esp, 8       ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
    push ebp     ; simpan nilai ebp lama ke stack  
    mov  ebp, esp ; set ebp = esp  
  
; Kode utama subprogram  
mov  ebx, [ebp + 8]  
mov  eax, [ebp + 12]  
add  eax, ebx  
  
pop  ebp        ; kembalikan ebp pemanggil  
ret             ; kembali ke pemanggil
```

ESP →

23
104

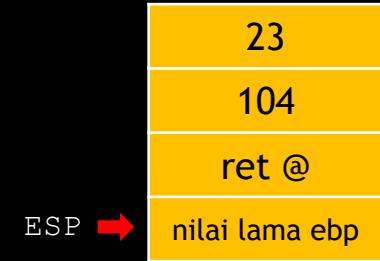
Stack pada Subprogram

```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
push dword 104   ; parameter 1  
  
● call f_add      ; panggil subprogram f_add  
  
add esp, 8       ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
    push ebp      ; simpan nilai ebp lama ke stack  
    mov  ebp, esp ; set ebp = esp  
  
; Kode utama subprogram  
mov  ebx, [ebp + 8]  
mov  eax, [ebp + 12]  
add  eax, ebx  
  
pop  ebp      ; kembalikan ebp pemanggil  
ret          ; kembali ke pemanggil
```



Stack pada Subprogram

```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
push dword 104   ; parameter 1  
  
call f_add       ; panggil subprogram f_add  
  
add esp, 8       ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
● push ebp       ; simpan nilai ebp lama ke stack  
mov  ebp, esp    ; set ebp = esp  
  
; Kode utama subprogram  
mov  ebx, [ebp + 8]  
mov  eax, [ebp + 12]  
add  eax, ebx  
  
pop  ebp        ; kembalikan ebp pemanggil  
ret           ; kembali ke pemanggil
```



Stack pada Subprogram

```
...
; push parameter-parameter
push dword 23    ; parameter 2
push dword 104   ; parameter 1

call f_add       ; panggil subprogram f_add

add esp, 8       ; bersihkan stack dari parameter

; nilai kembali di EAX
...
;; subprogram f_add
f_add:
  push ebp       ; simpan nilai ebp lama ke stack
  mov  ebp, esp  ; set ebp = esp

  ; Kode utama subprogram
  mov  ebx, [ebp + 8]
  mov  eax, [ebp + 12]
  add  eax, ebx

  pop  ebp       ; kembalikan ebp pemanggil
  ret           ; kembali ke pemanggil
```

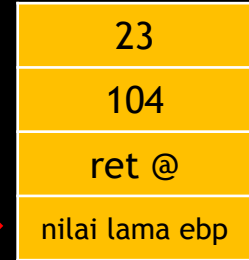
EBP + 12

EBP + 8

EBP + 4

EBP

ESP →



← EBP

Stack pada Subprogram

```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
push dword 104   ; parameter 1  
  
call f_add       ; panggil subprogram f_add  
  
add esp, 8       ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
    push ebp     ; simpan nilai ebp lama ke stack  
    mov  ebp, esp ; set ebp = esp  
  
; Kode utama subprogram  
● mov  ebx, [ebp + 8]  
  mov  eax, [ebp + 12]  
  add  eax, ebx  
  
pop  ebp     ; kembalikan ebp pemanggil  
ret         ; kembali ke pemanggil
```

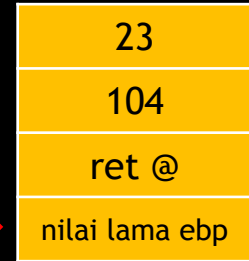
EBP + 12

EBP + 8

EBP + 4

EBP

ESP →



← EBP

Stack pada Subprogram

```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
push dword 104   ; parameter 1  
  
call f_add       ; panggil subprogram f_add  
  
add esp, 8       ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
    push ebp     ; simpan nilai ebp lama ke stack  
    mov  ebp, esp ; set ebp = esp  
  
; Kode utama subprogram  
mov  ebx, [ebp + 8]  
● mov  eax, [ebp + 12]  
add  eax, ebx  
  
pop  ebp        ; kembalikan ebp pemanggil  
ret            ; kembali ke pemanggil
```

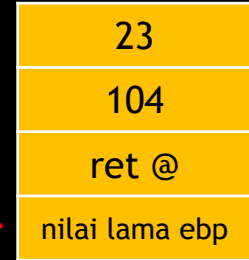
EBP + 12

EBP + 8

EBP + 4

EBP

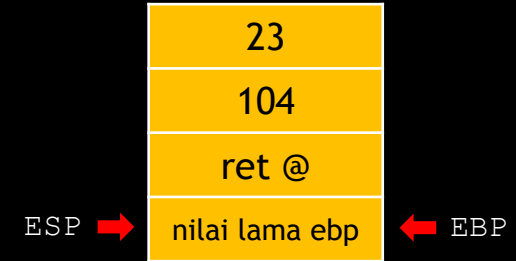
ESP →



← EBP

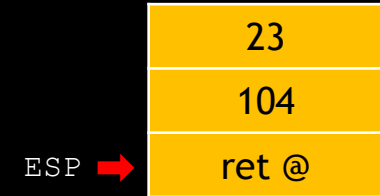
Stack pada Subprogram

```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
push dword 104   ; parameter 1  
  
call f_add       ; panggil subprogram f_add  
  
add esp, 8       ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
    push ebp     ; simpan nilai ebp lama ke stack  
    mov  ebp, esp ; set ebp = esp  
  
    ; Kode utama subprogram  
    mov  ebx, [ebp + 8]  
    mov  eax, [ebp + 12]  
    add  eax, ebx  
  
    pop  ebp     ; kembalikan ebp pemanggil  
    ret         ; kembali ke pemanggil
```



Stack pada Subprogram

```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
push dword 104   ; parameter 1  
  
call f_add       ; panggil subprogram f_add  
  
add esp, 8       ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
    push ebp     ; simpan nilai ebp lama ke stack  
    mov  ebp, esp ; set ebp = esp  
  
    ; Kode utama subprogram  
    mov  ebx, [ebp + 8]  
    mov  eax, [ebp + 12]  
    add  eax, ebx  
  
    pop  ebp     ; kembalikan ebp pemanggil  
    ret         ; kembali ke pemanggil
```



Stack pada Subprogram

```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
push dword 104   ; parameter 1  
  
call f_add       ; panggil subprogram f_add  
  
add esp, 8       ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
    push ebp     ; simpan nilai ebp lama ke stack  
    mov  ebp, esp ; set ebp = esp  
  
; Kode utama subprogram  
mov  ebx, [ebp + 8]  
mov  eax, [ebp + 12]  
add  eax, ebx  
  
pop  ebp        ; kembalikan ebp pemanggil  
ret            ; kembali ke pemanggil
```



Stack pada Subprogram

```
...  
; push parameter-parameter  
push dword 23    ; parameter 2  
push dword 104   ; parameter 1  
  
call f_add       ; panggil subprogram f_add  
  
● add esp, 8      ; bersihkan stack dari parameter  
  
; nilai kembali di EAX  
...  
;; subprogram f_add  
f_add:  
    push ebp      ; simpan nilai ebp lama ke stack  
    mov  ebp, esp ; set ebp = esp  
  
; Kode utama subprogram  
mov  ebx, [ebp + 8]  
mov  eax, [ebp + 12]  
add  eax, ebx  
  
pop  ebp          ; kembalikan ebp pemanggil  
ret              ; kembali ke pemanggil
```

ESP →

23
104

Variabel Lokal dalam Subprogram

- Pada contoh subprogram sebelumnya, subpgrogram dapat bekerja dengan menggunakan register-register yang tersedia
- Namun terkadang, sebuah subprogram membutuhkan tempat penyimpanan sementara (variabel lokal) yang lebih banyak dari jumlah register yang tersedia sehingga beberapa data harus disimpan dalam memori
- Salah satu cara untuk mengatasinya adalah dengan mendeklarasikan sebuah variabel pada segment `.bss` untuk semua variabel lokal yang dibutuhkan setiap subprogram
 - ❑ Kekurangan #1: Pemborosan memori
 - Pencadangan memori ini akan menggunakan ruang memori sepanjang durasi dari jalannya program meskipun jika subprogram hanya aktif pada sebagian kecil durasi program (atau mungkin tidak pernah aktif)
 - ❑ Kekurangan #2: Subprogram tidak *re-entrant*

Re-entrant dari Subprogram

- Subprogram disebut **aktif** jika subprogram tersebut telah dipanggil namun instruksi `RET` belum dieksekusi
- Subprogram disebut **re-entrant** jika ia dapat dipanggil darimana saja dalam program
- Subprogram yang re-entrant mengimplikasikan bahwa subprogram tersebut dapat memanggil dirinya sendiri, secara langsung maupun tidak langsung, yang memungkinkan rekursi
 - Misal: f memanggil g, g memanggil h, dan h memanggil f
- Subprogram yang re-entrant memungkinkan dalam satu waktu terdapat dua atau lebih *instance* dari sebuah subprogram yang aktif
 - Ini berarti terdapat dua atau lebih activation record untuk subprogram ini pada stack

Re-entrant dari Subprogram

- Jika kita menyimpan variabel lokal pada segment .bss, maka hanya bisa terdapat satu activation record
 - Jika lebih dari satu activation record, maka activation #2 dapat mengubah variable lokal dari activation #1
- Maka, dengan menyimpan variabel local pada segment .bss, subprogram tidak re-entrant

Variabel Lokal pada Stack

- Teknik yang umum untuk menggunakan variabel lokal pada subprogram adalah dengan menyimpan variabel lokal tersebut dalam stack
- Variabel lokal umumnya disimpan tepat setelah nilai `EBP` yang lama
 - Ini dilakukan dengan mengurangi pointer `ESP` dengan suatu nilai
- Variabel lokal dapat diakses dengan `[EBP-4]`, `[EBP-8]`, dan seterusnya



Contoh Variabel Lokal

- Misal, sebuah subprogram menerima 2 parameter dan menggunakan 3 variabel lokal
- Kerangka kode untuk subpgogram tersebut:

Cadangkan ruang 12 byte (3 x 4 byte) dalam stack

func:

```
push    ebp        ; simpan nilai EBP lama
mov     ebp, esp    ; set nilai ebp = esp
sub     esp, 12     ; cadangkan ruang untuk
                    ; 3 variabel lokal

; mulai subprogram body
...
; akhir subprogram body

mov     esp, ebp    ; dealokasi variabel lokal
pop     ebp        ; kembalikan nilai EBP lama
ret
```

parameter ke-2
parameter ke -1
alamat kembali
nilai lama EBP
variabel lokal ke-1
variabel lokal ke-2
variabel lokal ke-3

Contoh Variabel Lokal

- Dalam subprogram, parameter-parameter direferensikan dengan:
 - $[EBP + 8]$: parameter ke-1
 - $[EBP + 12]$: parameter ke-2
- Dalam subprogram, variabel lokal direferensikan dengan:
 - $[EBP - 4]$: variabel lokal ke-1
 - $[EBP - 8]$: variabel lokal ke-2
 - $[EBP - 12]$: variabel lokal ke-3

$EBP + 12$

$EBP + 8$

$EBP + 4$

EBP

$EBP - 4$

$EBP - 8$

$EBP - 12$

parameter ke-2

parameter ke -1

alamat kembali

nilai lama EBP

variabel lokal ke-1

variabel lokal ke-2

variabel lokal ke-3

Penting untuk dapat membayangkan stack ini dalam menulis subprogram

ENTER dan LEAVE

- Kita selalu mempunyai *prolog* dan *epilog* yang sama

prolog

```
push    ebp          ; simpan nilai EBP lama
mov     ebp, esp      ; set nilai ebp = esp
sub     esp, X         ; cadangkan X = 4*N bytes untuk variabel lokal sebanyak N
```

ekuivalen dengan

```
enter   X, 0
```

epilog

```
mov     esp, ebp      ; buang ruang yang dicadangkan untuk variabel lokal
pop     ebp           ; kembalikan nilai lama dari EBP
ret
```

ekuivalen dengan

```
leave
ret
```

Prolog dan Epilog pada Skeleton

```
segment .data  
; directive Dx
```

```
segment .bss  
; directive RESx
```

```
segment .text  
global _main  
_main:
```

```
; Routine "setup"
```

```
enter 0, 0
```

```
pusha
```

```
; Program Anda di bawah
```

```
; Routine "cleanup"
```

```
popa
```

```
mov eax, 0
```

```
leave  
ret
```

Prolog dari _main

Epilog dari _main

Menulis Ulang Subprogram

Contoh

```
...  
;; subprogram f_add  
f_add:
```

```
push ebp  
mov  ebp, esp
```

```
; Kode utama subprogram  
mov  ebx, [ebp + 8]  
mov  eax, [ebp + 12]  
add  eax, ebx
```

```
pop  ebp  
ret
```

```
...  
;; subprogram f_add  
f_add:
```

```
enter 0,0      ; prolog
```

```
; Kode utama subprogram  
mov  ebx, [ebp + 8]  
mov  eax, [ebp + 12]  
add  eax, ebx
```

```
leave          ; epilog  
ret
```

Contoh Subprogram Rekursif

- Membuat program assembly factorial menggunakan subprogram

```
int factorial(int num) {  
    int x;  
    if ( num == 0) return 1;  
    x = num;  
    return x * factorial(num-1);  
}
```


Contoh Subprogram Rekursif

```
• push    dword 2          ; simpan parameter 1 ke stack
  call    factorial        ; panggil subprogram factorial
  add     esp, 4           ; bersihkan parameter 1
  ...
;; Subprogram Factorial
factorial:
  ;; prolog
  enter 4, 0               ; reservasi 1 variabel lokal
  mov     eax, [ebp + 8]    ; ambil parameter 1: ebx = num
  ; jika num == 0, return result * 1
  cmp     eax, 0
  jg      next             ; jika > 0; goto next
  mov     eax, 1
  jmp     end
next:
  mov     [ebp - 4], eax    ; simpan num di variabel lokal ke-1
  dec     eax              ; num - 1
  push    eax              ; push num - 1 sbg parameter
  call    factorial        ; rekursif; eax = factorial(num-1)
  add     esp, 4           ; bersihkan parameter num-1
  imul    eax, [ebp - 4]    ; eax = factorial(num-1) * num
end:
  ;; epilog
  leave
  ret
```

ESP →

2

Contoh Subprogram Rekursif

```
    push    dword 2          ; simpan parameter 1 ke stack
    • call   factorial       ; panggil subprogram factorial
    add     esp, 4           ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0             ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]    ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next             ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax    ; simpan num di variabel lokal ke-1
    dec     eax              ; num - 1
    push    eax              ; push num - 1 sbg parameter
    call    factorial        ; rekursif; eax = factorial(num-1)
    add     esp, 4           ; bersihkan parameter num-1
    imul    eax, [ebp - 4]    ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

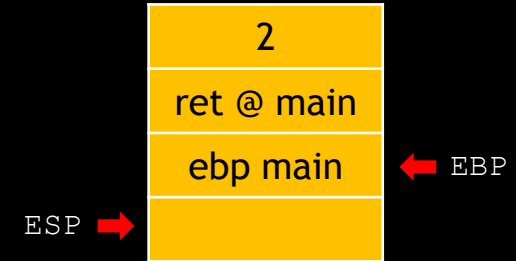
ESP →

2

ret @ main

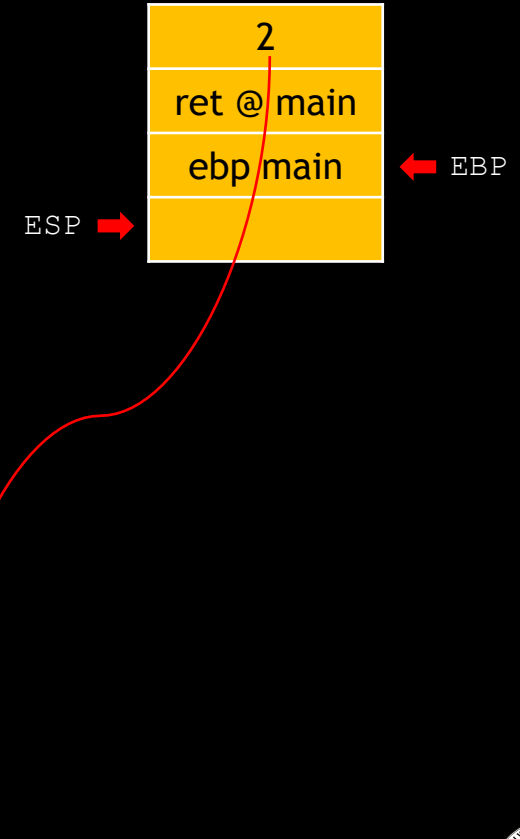
Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    • enter 4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    • mov    eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

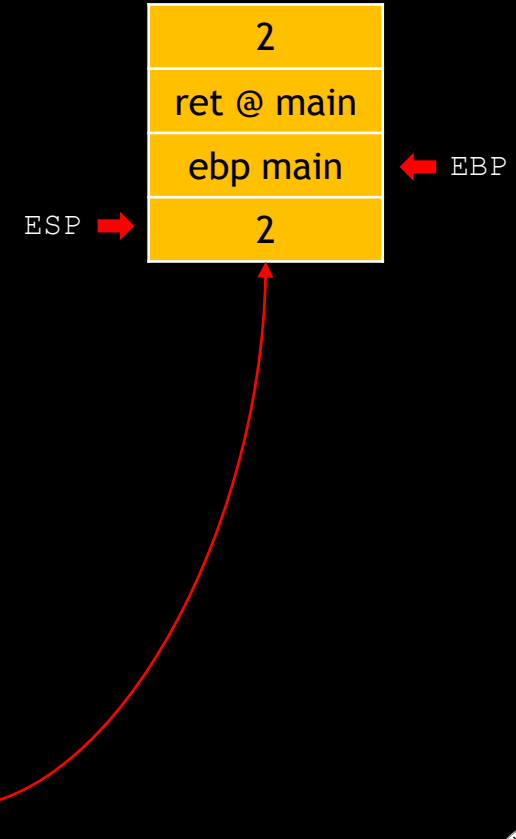


Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                 ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                  ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]         ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                  ; jika > 0; goto next
    mov     eax, 1
    jmp     end

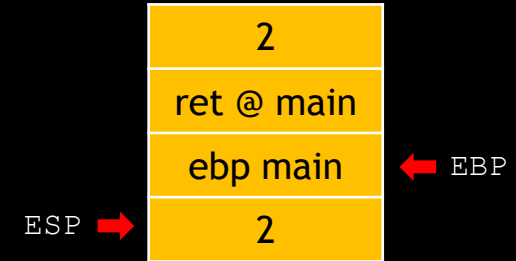
next:
    ● mov    [ebp - 4], eax        ; simpan num di variabel lokal ke-1
    dec     eax                   ; num - 1
    push    eax                   ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                 ; bersihkan parameter num-1
    imul    eax, [ebp - 4]         ; eax = factorial(num-1) * num

end:
    ;; epilog
    leave
    ret
```



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                   ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    ● dec     eax                  ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



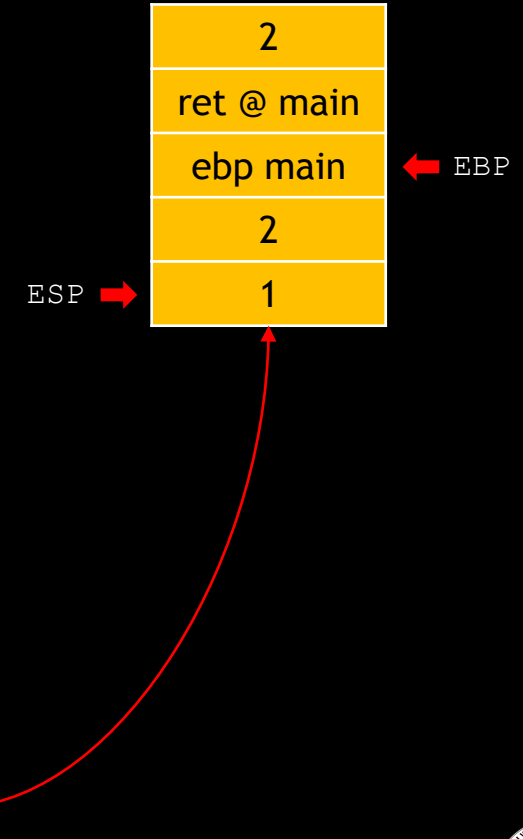
EAX

1



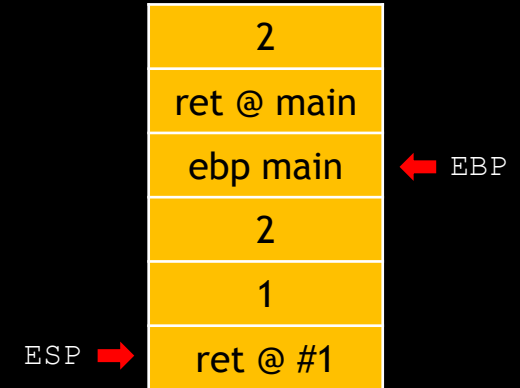
Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                   ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                    ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



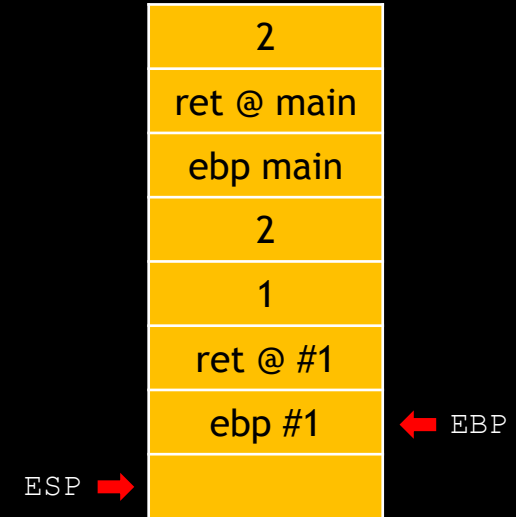
EAX

1



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    • enter 4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



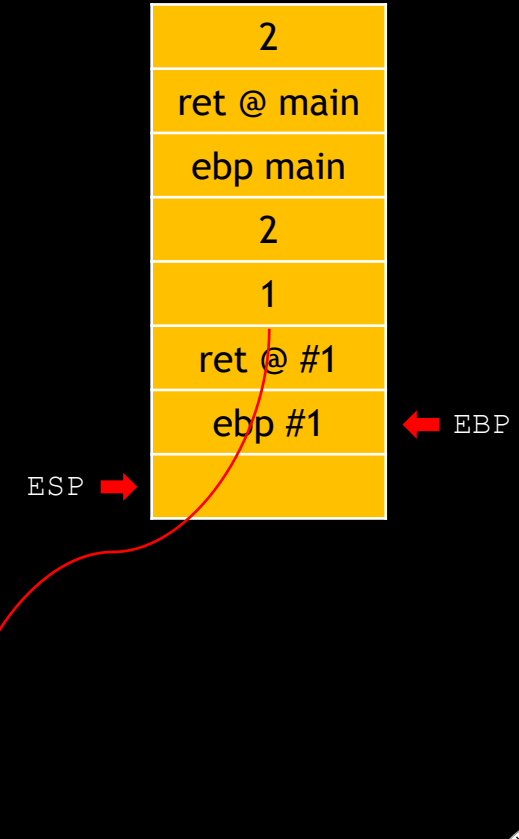
EAX

1



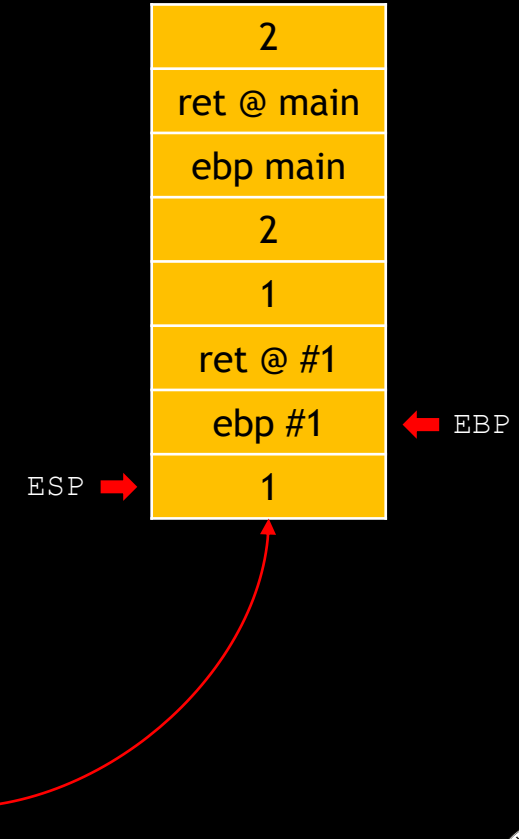
Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                   ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                    ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    • mov    [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    ● dec     eax                    ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

ESP →

2
ret @ main
ebp main
2
1
ret @ #1
ebp #1
1

← EBP

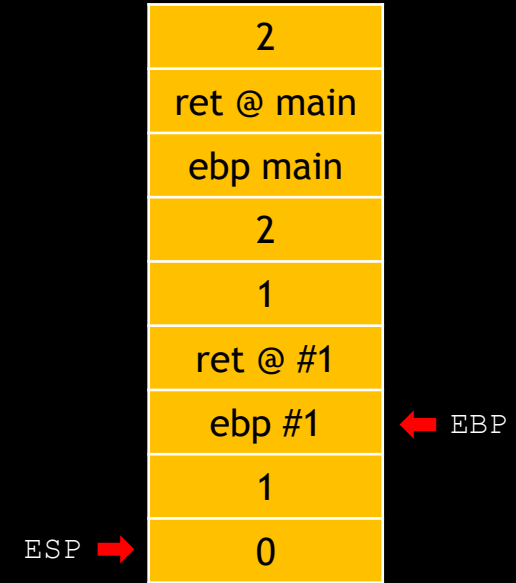
EAX

0



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



EAX

0



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter 4, 0                     ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                    ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    ● call   factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

ESP →

2
ret @ main
ebp main
2
1
ret @ #1
ebp #1
1
0
ret @ #2

← EBP

EAX

0



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    • enter 4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

EAX

0

ESP →

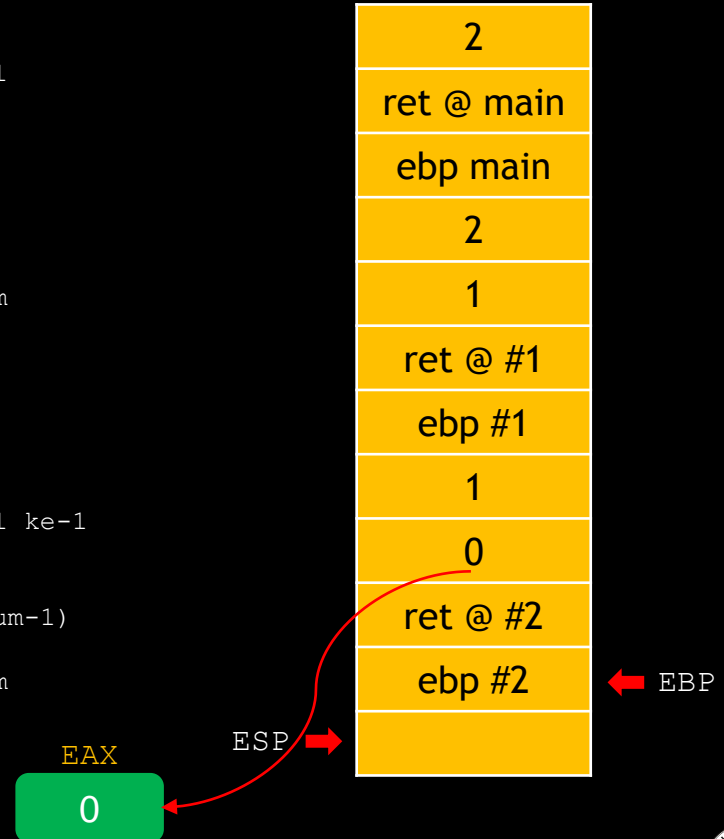
2
ret @ main
ebp main
2
1
ret @ #1
ebp #1
1
0
ret @ #2
ebp #2

← EBP



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    • mov    eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

EAX

1

ESP →

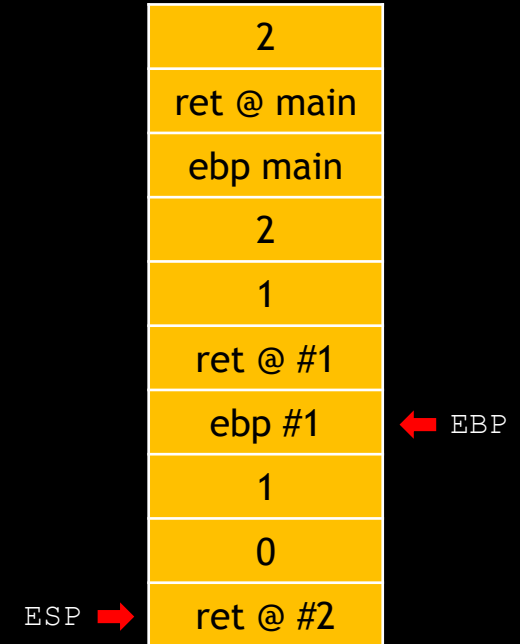
2
ret @ main
ebp main
2
1
ret @ #1
ebp #1
1
0
ret @ #2
ebp #2

← EBP



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



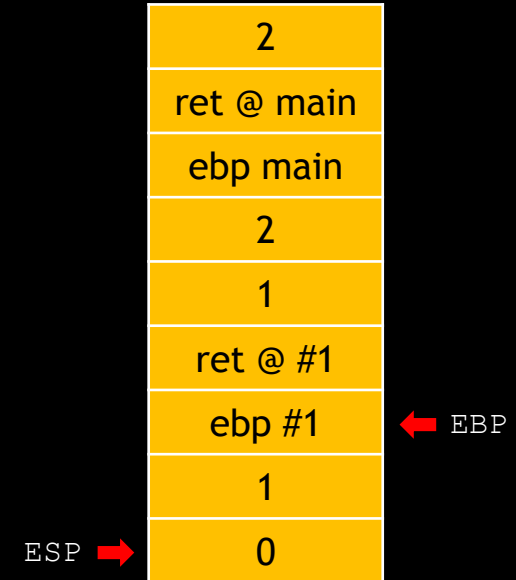
EAX

1



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter 4, 0                     ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                    ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



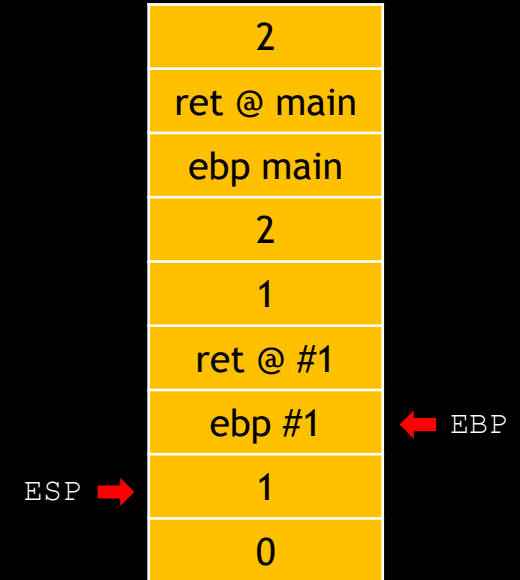
EAX

1



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter 4, 0                     ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                    ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



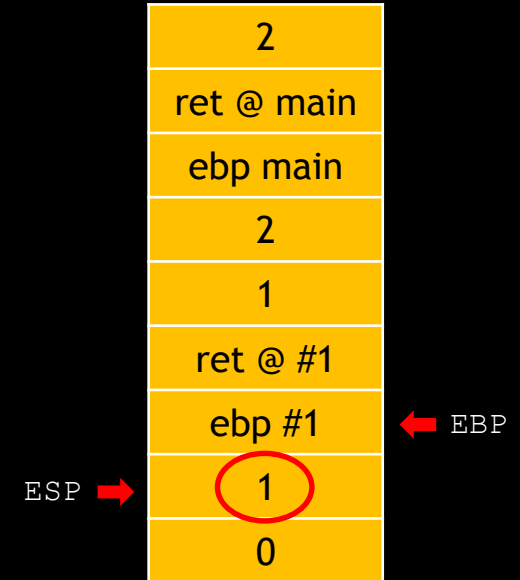
EAX

1



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



EAX

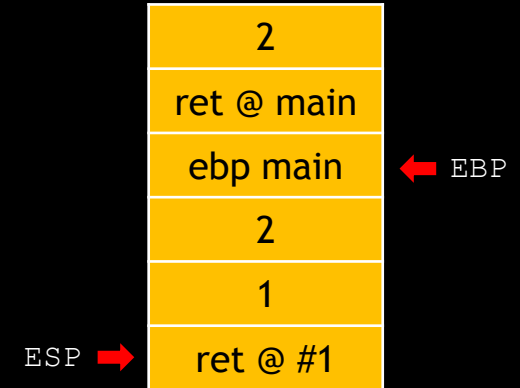
1

EAX := EAX * 1
→ EAX = 1



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                   ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                    ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



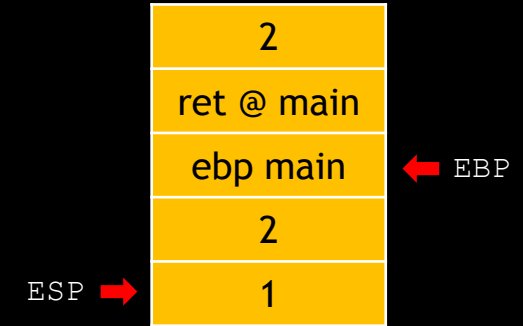
EAX

1



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                   ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                    ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



EAX

1

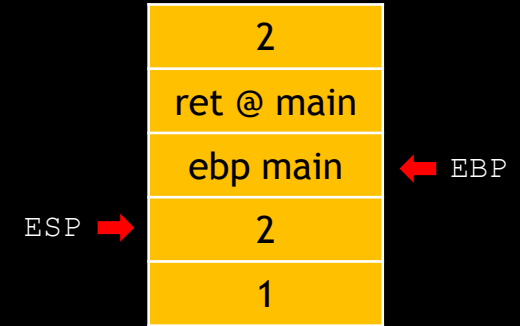


Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end

next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num

end:
    ;; epilog
    leave
    ret
```



EAX

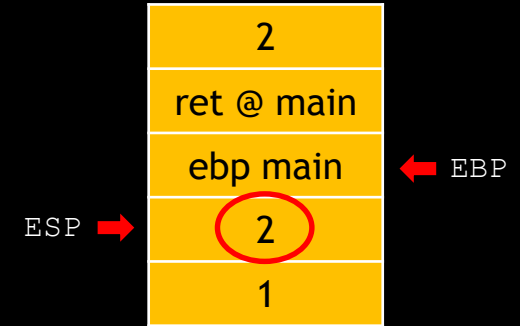
1



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end

next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    ● imul   eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```



EAX

2

EAX := EAX * 2
→ EAX = 2



Contoh Subprogram Rekursif

```
    push    dword 2          ; simpan parameter 1 ke stack
    call    factorial        ; panggil subprogram factorial
    add     esp, 4           ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0             ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]    ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next             ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax    ; simpan num di variabel lokal ke-1
    dec     eax              ; num - 1
    push    eax              ; push num - 1 sbg parameter
    call    factorial        ; rekursif; eax = factorial(num-1)
    add     esp, 4           ; bersihkan parameter num-1
    imul    eax, [ebp - 4]    ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

ESP →

2

ret @ main

EAX

2



Contoh Subprogram Rekursif

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                    ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]           ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                    ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax           ; simpan num di variabel lokal ke-1
    dec     eax                     ; num - 1
    push    eax                     ; push num - 1 sbg parameter
    call    factorial               ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]           ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

ESP →

2

EAX

2



Contoh Subprogram Rekursif ^{ESP} →

2

```
    push    dword 2                ; simpan parameter 1 ke stack
    call    factorial              ; panggil subprogram factorial
    add     esp, 4                  ; bersihkan parameter 1
    ...
;; Subprogram Factorial
factorial:
    ;; prolog
    enter   4, 0                   ; reservasi 1 variabel lokal
    mov     eax, [ebp + 8]          ; ambil parameter 1: ebx = num
    ; jika num == 0, return result * 1
    cmp     eax, 0
    jg      next                   ; jika > 0; goto next
    mov     eax, 1
    jmp     end
next:
    mov     [ebp - 4], eax          ; simpan num di variabel lokal ke-1
    dec     eax                    ; num - 1
    push    eax                    ; push num - 1 sbg parameter
    call    factorial              ; rekursif; eax = factorial(num-1)
    add     esp, 4                  ; bersihkan parameter num-1
    imul    eax, [ebp - 4]          ; eax = factorial(num-1) * num
end:
    ;; epilog
    leave
    ret
```

EAX

2

