

Subrutin adalah prosedur yang dapat didefinisikan dalam program ABAP dan dapat disebut bentuk dalam program ABAP. Subrutin digunakan secara lokal, subrutin ini umumnya dipanggil dari program di mana mereka didefinisikan.

### Praktikum.

#### 1. Program pertama.

Buat program dengan nama = Z\_GLOBAL\_NPM. Hasil Execute Programnya?

Dengan ketentuan

Komponen	Soal	Diganti
Variable	v_int1	v_int1_NPM
Variable	v_int2	v_int2_NPM
Variable	v_sum	v_sum_NPM
Value	v_int1	2 npm dari belakang
Value	v_int2	2 npm dari depan

```

*&-----*
*& Report  ZLOCAL_SUBROUTIN_02
*&
*&-----*
*&
*&
*&-----*

report zlocal_subroutin_02.
data: v_int1 type i,
      v_int2 type i,
      v_sum type i.
perform sum1.
write :/.
write: 'hasilnya adalah:', v_sum.
write :/.
perform sum2.
write :/.
write: 'hasilnya adalah:', v_sum.
*&-----*
*&      Form  SUM1
*&-----*
*      text
*&-----*
* --> p1      text
* <-- p2      text
*&-----*
form sum1 .
  v_int1 = 10.
  v_int2 = 20.
  v_sum = v_int1 + v_int2 .
  write: 'hasilnya adalah:', v_sum.
endform.
*&-----*
*&      Form  SUM2
*&-----*
*      text
*&-----*
* --> p1      text
* <-- p2      text
*&-----*
form sum2 .
  local: v_int1,
         v_int2,
         v_sum .
  v_int1 = 20.
  v_int2 = 15.
  v_sum = v_int1 + v_int2.
  write: 'hasilnya adalah:', v_sum.
endform.

```

## 2. Program kedua.

Buat program dengan nama = Z\_LOCAL\_NPM. Hasil Execute Programnya?

Dengan ketentuan

Komponen	Soal	Diganti
<b>Sroutine1</b>		
Variable	lv_text	lv_text_npm
Value	xxxx	Nama Depan
<b>Sroutine2</b>		
Variable	lv_text	lv_text_npm
Value	xxxx	Nama Belakang

```

*&-----*
*& Report  ZLOCAL_SUBROUTIN_DATA
*&
*&-----*
*&
*&-----*

report zlocal_subroutin_data.

perform sroutine1.
perform sroutine1.
skip.
perform sroutine2.
perform sroutine2.
*&-----*
*&      Form  SRROUTINE1
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
form sroutine1 .

    data lv_text type string value 'INIT'.
    write: lv_text.
    lv_text = 'XXXX'.
    write: lv_text.
endform.          " SRROUTINE1
*&-----*
*&      Form  SRROUTINE2
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
form sroutine2 .

    statics lv_text type string value 'INIT'.
    write : lv_text.
    lv_text = '0000'.
    write : lv_text.
endform.          " SRROUTINE2

```

## 3. Program ketiga

Buat program dengan nama = Z\_BYREFERENCE\_NPM. Hasil Execute Programnya?

Dengan ketentuan

Komponen	Soal	Diganti
Variable	v_sum	v_sum_NPM

Variable	v_num1	v_num1_NPM
Variable	v_num2	v_num2_NPM
Variable	p_v_num1	p_v_num1_NPM
Variable	p_v_num2	p_v_num2_NPM
Value	v_num1	2 npm dari belakang + 5
Value	v_num2	2 npm dari depan + 10
Value	p_v_num1	tetap
Value	p_v_num2	tetap

```

*&-----*
*& Report  ZSUBROUTINES_CALLBYREF_DOSEN
*&
*&-----*
*&
*&
*&-----*

report zsubroutines_callbyref_dosen.

data: v_sum type i,
      v_num1 type i,
      v_num2 type i.
v_num1 = 5.
v_num2 = 10.

write: / 'Variable 1 :',v_num1,
        'Variable 2 :',v_num2.

perform pass using v_num1
                  v_num2.
skip.
write: / 'Variable 1 :',v_num1,
        'Variable 2 :',v_num2.

*&-----*
*&      Form  PASS
*&-----*
*      text
*&-----*
*      -->P_V_NUM1  text
*      -->P_V_NUM2  text
*&-----*
form pass using  p_v_num1 type i
                p_v_num2 type i.

      p_v_num1 = 100.
      p_v_num2 = 111.

endform.          " PASS

```

#### 4. Program keempat

Buat program dengan nama = Z\_BYVALUE\_NPM. Hasil Execute Programnya?

**Dengan ketentuan**

Komponen	Soal	Diganti
Variable	v_sum	v_sum_NPM
Variable	v_num1	v_num1_NPM
Variable	v_num2	v_num2_NPM
Variable	p_v_num1	p_v_num1_NPM
Variable	p_v_num2	p_v_num2_NPM
Value	v_num1	2 npm dari belakang + 5

Value	v_num2	2 npm dari depan + 10
Value	p_v_num1	tetap
Value	p_v_num2	tetap

```

*&-----*
*& Report  ZSUBROUTINES_CALLBYVALUE_D05EN
*&
*&-----*
*&
*&-----*

report zsubroutines_callbyvalue_dosen.

data: v_sum type i,
      v_num1 type i,
      v_num2 type i.
v_num1 = 5.
v_num2 = 10.
write: / 'Variable 1 :',v_num1,
       'Variable 2 :',v_num2.
perform pass using v_num1
                  v_num2 .

skip.
write: / 'Variable 1 :',v_num1,
       'Variable 2 :',v_num2.

*&-----*
*&      Form  PASS
*&-----*
*      text
*-----*
*      -->P_V_NUM1  text
*      -->P_V_NUM2  text
*-----*
form pass  using value(p_v_num1)
                  value(p_v_num2).

      p_v_num1 = 100.
      p_v_num2 = 111.
endform.          " PASS

```

##### 5. Program kelima.

Buat program dengan nama = Z\_BYVALUEREFF\_NPM. Hasil Execute Programnya?

**Dengan ketentuan**

Komponen	Soal	Diganti
Variable	v_sum	v_sum_NPM
Variable	v_num1	v_num1_NPM
Variable	v_num2	v_num2_NPM
Variable	p_v_num1	p_v_num1_NPM
Variable	p_v_num2	p_v_num2_NPM
Value	v_num1	2 npm dari belakang + 5
Value	v_num2	2 npm dari depan + 10
Value	p_v_num1	tetap
Value	p_v_num2	tetap

```
*&-----*
*& Report  ZSUBROUTINES_BYVALUENREF_D0SEN
*&
*&-----*
*&
*&-----*

report zsubroutines_byvaluenref_dosen.
data: v_sum type i,
      v_num1 type i,
      v_num2 type i.
v_num1 = 5.
v_num2 = 10.
write: / 'Variable 1 :',v_num1,
        'Variable 2 :',v_num2.
perform pass using v_num1
              changing v_num2.
skip.
write: / 'Variable 1 :',v_num1,
        'Variable 2 :',v_num2.

*&-----*
*&      Form  PASS
*&-----*
*      text
*-----*
*      -->P_V_NUM1  text
*      <--P_V_NUM2  text
*-----*
form pass  using value(p_v_num1) type i
              changing value(p_v_num2) type i.

      p_v_num1 = 100.
      p_v_num2 = 111.

endform.                                " PASS
```