

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 <u>zlocal subroutin 02</u>.
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. "SUM1
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
Sroutine1		
Variable	lv_text	lv_text_npm
Value	XXXX	Nama Depan
	Sroutine2	!
Variable	lv_text	lv_text_npm
Value	XXXX	Nama Belakang

```
*& Report ZLOCAL SUBROUTIN DATA
*&
*&-
*&
*&
*&-
report zlocal subroutin data.
perform <u>sroutine1</u>.
perform <u>sroutine1</u>.
skip.
perform sroutine2.
perform <u>sroutine2</u>.
*&---
*&
        Form SROUTINE1
*&
      text
             text
* --> p1
* <-- p2
                   text
form sroutine1 .
  data lv_text type string value 'INIT'.
  write: lv_text.
lv_text = 'XXXX'.
  write: lv_text.
endform.
                              " SROUTINE1
*&--
       Form SROUTINE2
*&
*&--
        text
              text
* --> p1
  <-- p2
form <u>sroutine2</u>.
  statics lv_text type string value 'INIT'.
  write : lv_text.
lv_text = '0000'.
write : lv_text.
endform.
                               " SROUTINE2
```

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

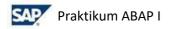
```
*& Report ZSUBROUTINES_CALLBYREF_DOSEN
*&
*&-
*&
*&
*&------*
report zsubroutines callbyref dosen.
data: v_sum type i,
     v_num1 type i,
v_num2 type i.
v_num1 = 5.
v_num2 = 10.
write: / 'Variable 1 :',v_num1,
'Variable 2 :',v_num2.
perform pass using v_num1
                  v_num2.
skip.
*&----
*&
     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



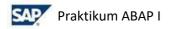
Value	v_ num2	2 npm dari depan + 10
Value	p_v_num1	tetap
Value	p_v_num2	tetap

```
*& Report ZSUBROUTINES_CALLBYVALUE_DOSEN
*&
*&
*&
*&-
report zsubroutines_callbyvalue_dosen.
data: v_sum type i,
    v_num1 type i,
    v_num2 type i.
v_num1 = 5.
v_num2 = 10.
perform pass using v_num1
               v_num2 .
skip.
*&----
    Form PASS
*&
*&----
    -->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.
                   " PASS
endform.
```

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_DOSEN
*&-
*δε
*&
*&-----*
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.
                       " PASS
endform.
```