

BAHASA PEMROGRAMAN JAVA

Pertemuan X

SUB PROSES

(Metode / Method)

Oleh
Achmad Arrosyidi



TUJUAN PEMBELAJARAN

Mahasiswa dapat menunjukkan hubungan antara konsep Fungsi dan Prosedur menggunakan program.



MATERI

Metode:

1. Pengertian metode (Fungsi & Prosedur) dan nilai balik dari metode.
2. Parameter metode.
3. Method overloading.



1. PENGANTAR SUB PROSES - **PERMASALAHAN**

- Programmer pemula menulis program secara utuh.
- Terdapat beberapa blok program yang ditulis secara berulang kali.
- Kesulitan dalam proses pelacakan program (tracing) untuk pengetesan atau terjadi kesalahan program.
- Tracing mulai awal sampai dengan akhir program.

Tracing ini sering menyebabkan:

Kurang efektif:

- Frustasi karena sulit untuk menemukan kesalahan jika terjadi error.
 - Teknik penulisan yang berulang-ulang pada bagian yang berbeda dapat mengakibatkan kesalahan dalam penulisan kode program sehingga dapat menjadikan error pada program.
- Kurang efisien:
 - Butuh waktu lama karena harus menuliskan blok program yang ditulis berulang-ulang yang sama.
 - Program harus melakukan perbaikan yang disebabkan oleh kesalahan kode program walaupun pernah dilakukan sebelumnya untuk blok program yang berbeda.



2. CONTOH PROGRAM TANPA SUB PROSES

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_1.java
2 import java.util.Scanner;
3
4 public class Program_5_1 {
5     public static void main (String[] arg) {
6         //Deklarasi variabel
7         int panjang, lebar, luas;
8         Scanner sc = new Scanner(System.in);
9
10        //Menampilkan judul program
11        System.out.println("=====");
12        System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
13        System.out.println("=====");
14
15        //Tahap input
16        System.out.print("Masukkan panjang persegi panjang: ");
17        panjang=sc.nextInt();
18        System.out.print("Masukkan lebar persegi panjang: ");
19        lebar=sc.nextInt();
20
21        //Tahap proses
22        luas=panjang*lebar;
23
24        //Tahap output
25        System.out.print("Luas persegi panjang: " + luas);
26    }
27 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=15156). (Exit code 0)
===== READY =====
```



3. PENGANTAR SUB PROSES - **SOLUSI**

- Teknik pembuatan program yang ditulis secara terpisah.
- Program yang sebelumnya ditulis secara utuh dipecah-pecah menjadi beberapa bagian (Program Induk/Utama/Main, dan anak/sub proses)
- Setiap sub proses → berbeda dengan sub-proses lainnya → tidak boleh sampai terjadi overlapping tugas.
- sub proses dapat memanggil sub proses lain
- Setelah menyelesaikan kegiatan di dalam sub-proses, alur proses akan melanjutkan instruksi pada lokasi yang telah ditinggalkan. Address memory yang ditinggalkan tadi disimpan di dalam stack.
- Sehingga: Lebih efektif, dan lebih efisien

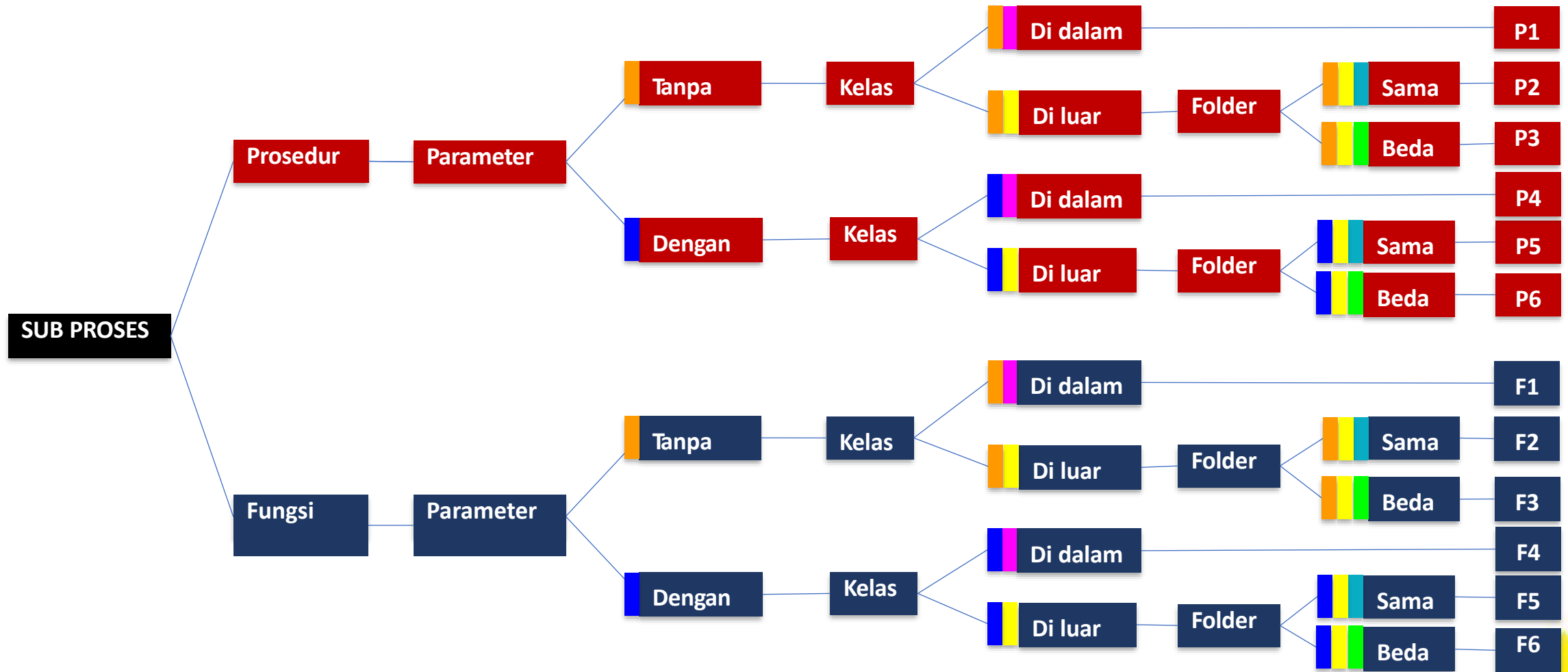


4. MACAM SUB PROSES

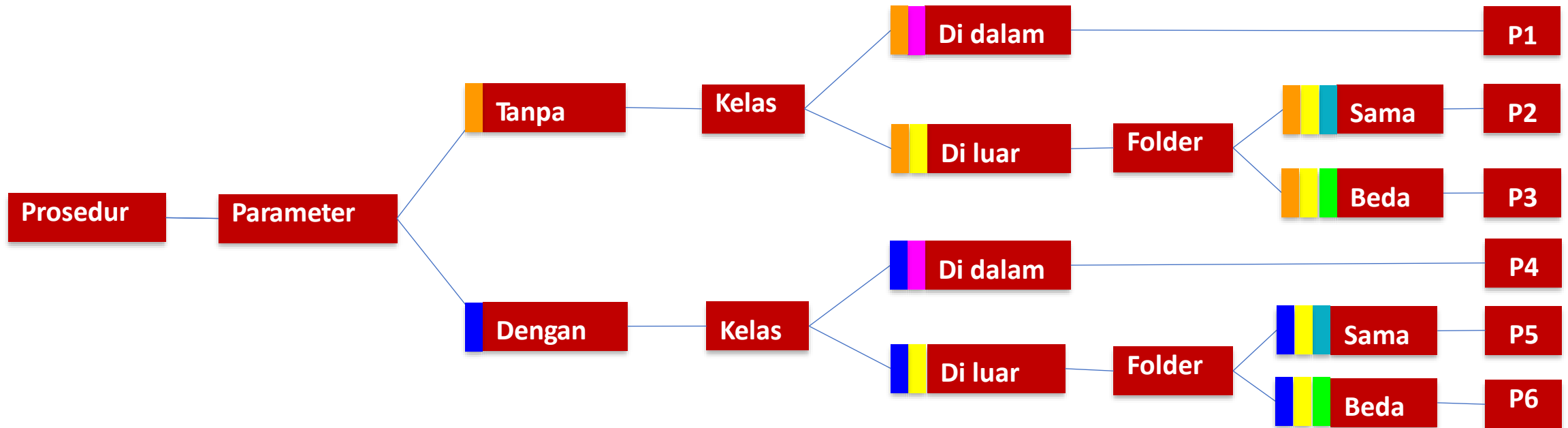
- Di dalam bahasa pemrograman java sub proses disebut juga sebagai method atau metode.
- Metode menurut nilai baliknya / return value-nya terbagi menjadi dua yaitu prosedur (procedure) dan fungsi (function), namun di bahasa pemrograman java sebagian orang yang mengatakan bahwa semuanya dianggap sebagai fungsi.
- Pembedanya adalah fungsi yang mempunyai nilai balik dan fungsi yang tidak mempunyai nilai balik.



5. MACAM PROSEDUR DAN FUNGSI

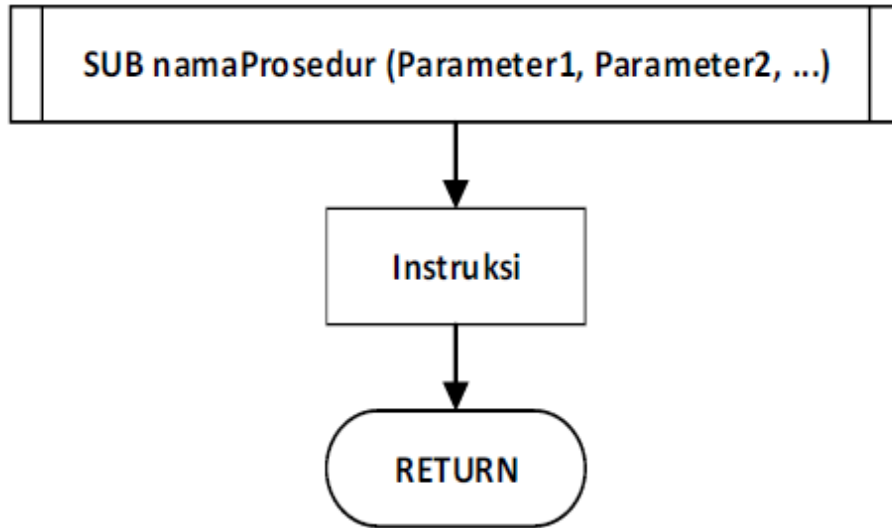


6. PROSEDUR

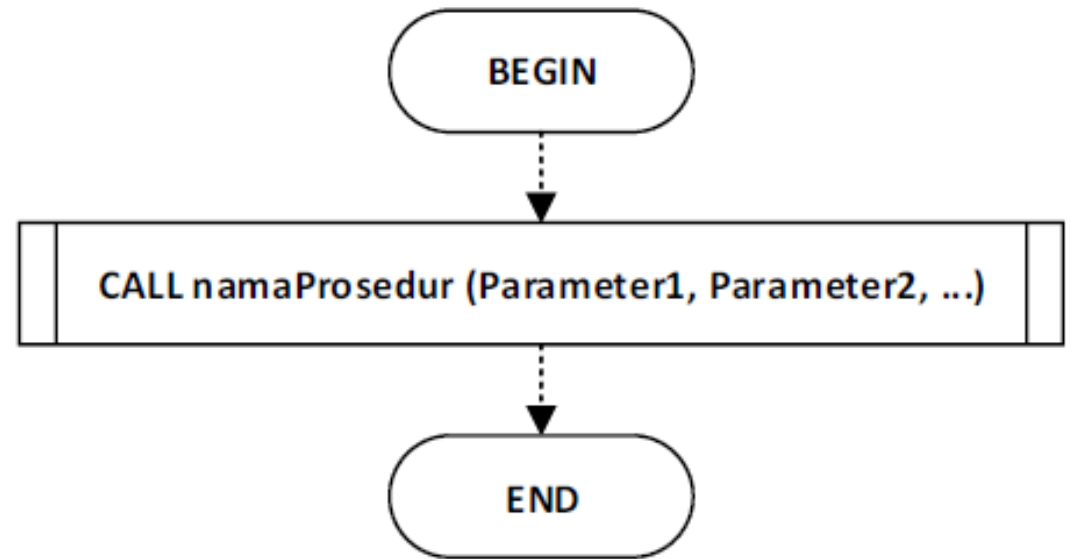


6.1. PROSEDUR SECARA UMUM

GAMBAR FLOWCHART



Prosedur



Memanggil Prosedur



6.1. PROSEDUR SECARA UMUM SYNTAX / BENTUK PENULISAN

Syntax procedure:

```
Modifier static void procedureName (type argumen1, type argumen2, ....) {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    instruction  
}
```

Syntax procedure caller:

```
procedureName (parameter1, parameter2, ...)
```





6.2.1. PROSEDUR TANPA PARAMETER PASSING DIDALAM KELAS PEMANGGIL YANG SAMA – SYNTAX

Syntax procedure:

```
Modifier static void procedureName () {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    instruction  
}
```

Syntax procedure caller:

```
procedureName ()
```





6.2.1. PROSEDUR TANPA PARAMETER PASSING DIDALAM KELAS PEMANGGIL YANG SAMA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_2.java
2 import java.util.Scanner;
3
4 public class Program_5_2 {
5     public static void judulProgram() {
6         //Menampilkan judul program
7         System.out.println("=====");
8         System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
9         System.out.println("=====");
10    }
11    public static void main (String[] arg) {
12        //Deklarasi variabel
13        int panjang, lebar, luas;
14        Scanner sc = new Scanner(System.in);
15
16        judulProgram(); //Memanggil prosedur judulProgram didalam kelas pemanggil pada file yang sama
17
18        //Tahap input
19        System.out.print("Masukkan panjang persegi panjang: ");
20        panjang=sc.nextInt();
21        System.out.print("Masukkan lebar persegi panjang: ");
22        lebar=sc.nextInt();
23
24        //Tahap proses
25        luas=panjang*lebar;
26
27        //Tahap output
28        System.out.print("Luas persegi panjang: " + luas);
29    }
30 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=11776). (Exit code 0)
===== READY =====
```



6.2.2. PROSEDUR TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – SYNTAX

Syntax procedure:

```
Modifier static void procedureName () {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    instruction  
}
```

Syntax procedure caller:

```
className.procedureName ()
```



6.2.2. PROSEDUR TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_3.java
2 import java.util.Scanner;
3
4 class AnakProgram {
5     public static void judulProgram() {
6         //Menampilkan judul program
7         System.out.println("=====");
8         System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
9         System.out.println("=====");
10    }
11 }
12
13 public class Program_5_3 {
14     public static void main (String[] arg) {
15         //Deklarasi variabel
16         int panjang, lebar, luas;
17         Scanner sc = new Scanner(System.in);
18
19         AnakProgram.judulProgram(); //Memanggil prosedur judulProgram diluar kelas pemanggil pada file yang sama
20
21         //Tahap input
22         System.out.print("Masukkan panjang persegi panjang: ");
23         panjang=sc.nextInt();
24         System.out.print("Masukkan lebar persegi panjang: ");
25         lebar=sc.nextInt();
26
27         //Tahap proses
28         luas=panjang*lebar;
29
30         //Tahap output
31         System.out.print("Luas persegi panjang: " + luas);
32     }
33 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=18224). (Exit code 0)
===== READY =====
```

6.2.3. PROSEDUR TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG BERBEDA – SYNTAX

Syntax procedure:

```
Modifier static void procedureName () {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    instruction  
}
```

Syntax procedure caller:

```
className.procedureName ()
```



6.2.3. PROSEDUR TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG BERBEDA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_4.java
2 import java.util.Scanner;
3
4 public class Program_5_4 {
5     public static void main (String[] arg) {
6         //Deklarasi variabel
7         int panjang, lebar, luas;
8         Scanner sc = new Scanner(System.in);
9
10        Program_5_5.judulProgram(); //Memanggil prosedur judulProgram Diluar Kelas Pemanggil Pada File yang Berbeda
11
12        //Tahap input
13        System.out.print("Masukkan panjang persegi panjang: ");
14        panjang=sc.nextInt();
15        System.out.print("Masukkan lebar persegi panjang: ");
16        lebar=sc.nextInt();
17
18        //Tahap proses
19        luas=panjang*lebar;
20
21        //Tahap output
22        System.out.print("Luas persegi panjang: " + luas);
23    }
24 }
```

Console

```
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=10976). (Exit code 0)
===== READY =====
```

```
1 //File: Program_5_5.java
2 class Program_5_5 {
3     public static void judulProgram() {
4         //Menampilkan judul program
5         System.out.println("=====");
6         System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
7         System.out.println("=====");
8     }
9 }
```



6.2.4. PROSEDUR DENGAN PARAMETER PASSING DIDALAM KELAS PEMANGGIL YANG SAMA – SYNTAX

Syntax procedure:

```
Modifier static void procedureName (type argument1, type argument2, ...) {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    instruction  
}
```

Syntax procedure caller:

```
procedureName (parameter1, parameter2, ...)
```



6.2.4. PROSEDUR DENGAN PARAMETER PASSING DIDALAM KELAS PEMANGGIL YANG SAMA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_6.java
2 import java.util.Scanner;
3
4 public class Program_5_6 {
5     //Deklarasi Global
6     public static int panjang, lebar, luas;
7
8     public static void proses(int pj, int lbr) {
9         //Tahap proses
10        luas=pj*lbr;
11    }
12
13    public static void main (String[] arg) {
14        //Deklarasi variabel
15
16        Scanner sc = new Scanner(System.in);
17
18        //Menampilkan judul program
19        System.out.println("=====");
20        System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
21        System.out.println("=====");
22
23        //Tahap input
24        System.out.print("Masukkan panjang persegi panjang: ");
25        panjang=sc.nextInt();
26        System.out.print("Masukkan lebar persegi panjang: ");
27        lebar=sc.nextInt();
28
29        proses(panjang, lebar); //Memanggil prosedur proses Didalam Kelas Pemanggil Pada File yang Sama
30
31        //Tahap output
32        System.out.print("Luas persegi panjang: " + luas);
33    }
34 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=15024). (Exit code 0)
===== READY =====
```



6.2.5. PROSEDUR DENGAN PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – SYNTAX

Syntax procedure:

```
Modifier static void procedureName (type argument1, type argument2, ...) ) {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    instruction  
}
```

Syntax procedure caller:

```
className.procedureName (parameter1, parameter2, ...) )
```



6.2.5. PROSEDUR DENGAN PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_7.java
2 import java.util.Scanner;
3 class AnakProgram {
4     public static void output(int ls) {
5         //Tahap output
6         System.out.print("Luas persegi panjang: " + ls);
7     }
8 }
9
10 public class Program_5_7 {
11     public static void main (String[] arg) {
12         //Deklarasi variabel
13         int panjang, lebar, luas;
14         Scanner sc = new Scanner(System.in);
15
16         //Menampilkan judul program
17         System.out.println("=====");
18         System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ===");
19         System.out.println("=====");
20
21         //Tahap input
22         System.out.print("Masukkan panjang persegi panjang: ");
23         panjang=sc.nextInt();
24         System.out.print("Masukkan lebar persegi panjang: ");
25         lebar=sc.nextInt();
26
27         //Tahap proses
28         luas=panjang*lebar;
29
30         AnakProgram.output(luas); //Memanggil prosedur output Diluar Kelas Pemanggil Pada File yang Sama
31     }
32 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ===
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=17044). (Exit code 0)
===== READY =====
```



Syntax procedure:

```
Modifier static void procedureName (type argument1, type argument2, ...) ) {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    instruction  
}
```

Syntax procedure caller:

```
className.procedureName (parameter1, parameter2, ...) )
```



6.2.6. PROSEDUR DENGAN PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG BERBEDA – CONTOH PROGRAM

```
1 //File: Program_5_8.java
2 import java.util.Scanner;
3
4 public class Program_5_8 {
5     public static void main (String[] arg) {
6         //Deklarasi variabel
7         int panjang, lebar, luas;
8         Scanner sc = new Scanner(System.in);
9
10        //Menampilkan judul program
11        System.out.println("=====");
12        System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
13        System.out.println("=====");
14
15        //Tahap input
16        System.out.print("Masukkan panjang persegi panjang: ");
17        panjang=sc.nextInt();
18        System.out.print("Masukkan lebar persegi panjang: ");
19        lebar=sc.nextInt();
20
21        //Tahap proses
22        luas=panjang*lebar;
23
24        Program_5_9.output(luas); //Memanggil prosedur output Diluar Kelas Pemanggil Pada File yang berbeda
25    }
26 }
```

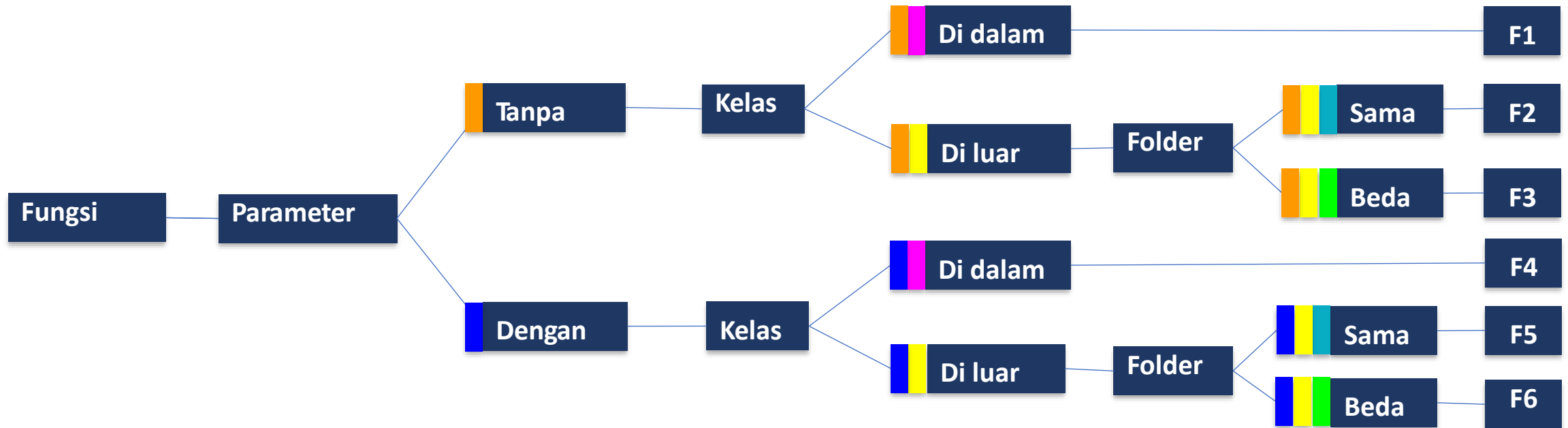
```
1 //File: Program_5_9.java
2 class Program_5_9 {
3     public static void output(int ls) {
4         //Tahap output
5         System.out.print("Luas persegi panjang: " + ls);
6     }
7 }
```

CONTOH:
Menghitung Luas
Persegi Panjang

Console

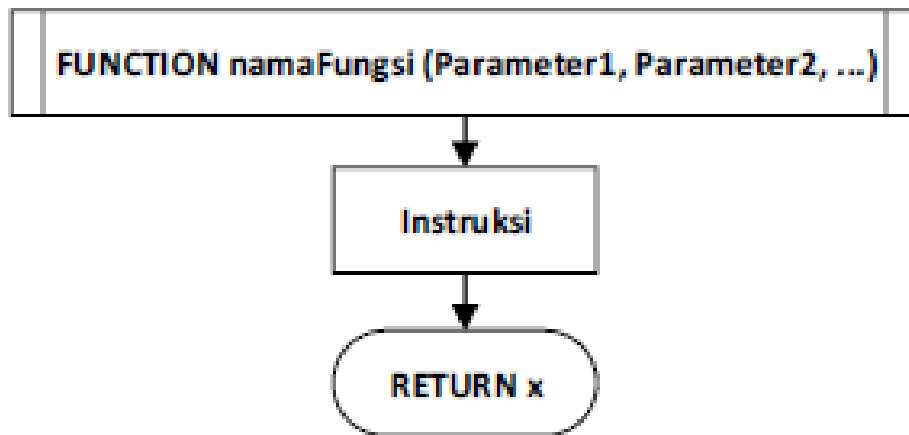
```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30 <<< Process finished (PID=10644). (Exit code 0)
===== READY =====
```

7. FUNGSI

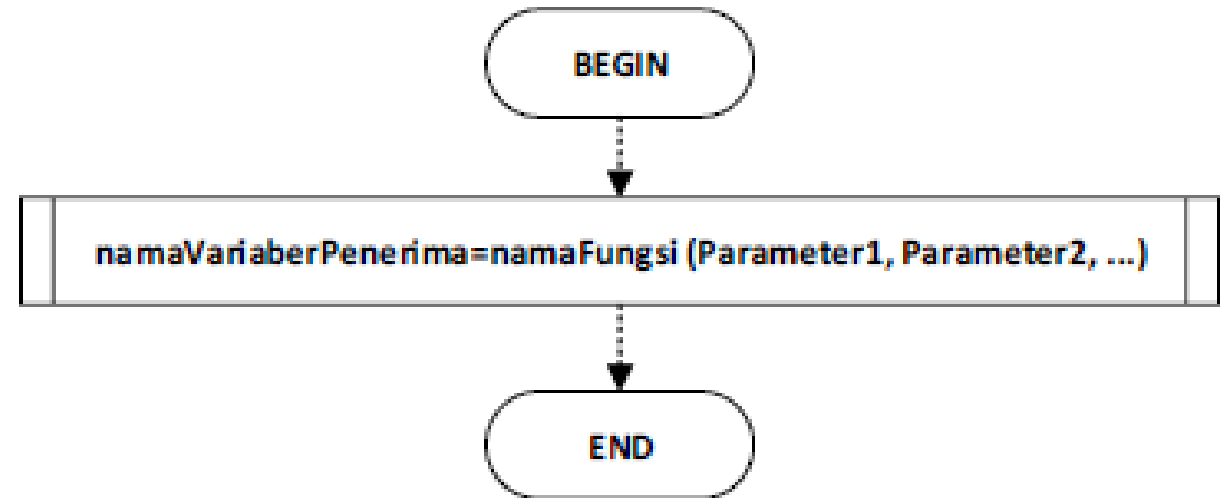


7.1. FUNGSI SECARA UMUM

GAMBAR FLOWCHART



Fungsi



Memanggil Fungsi



7.1. FUNGSI SECARA UMUM SYNTAX / BENTUK PENULISAN

Syntax function:

```
Modifier static type functionName (type argumen1, type argumen2, ....) {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    Instruction  
    Return(value)  
}
```

Syntax function caller:

```
variableName=functionName (parameter1, parameter2, ...)
```





7.1.1. FUNGSI TANPA PARAMETER **PASSING** **DIDALAM** KELAS **PEMANGGIL YANG SAMA** – **SYNTAX**

Syntax function:

```
Modifier static type functionName () {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    Instruction  
    Return(value)  
}
```

Syntax function caller:

```
variableName=functionName ()
```





7.1.1. FUNGSI TANPA PARAMETER PASSING DIDALAM KELAS PEMANGGIL YANG SAMA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_10.java
2 import java.util.Scanner;
3
4 public class Program_5_10 {
5     //Deklarasi variabel
6     public static int panjang, lebar, luas;
7     public static Scanner sc = new Scanner(System.in);
8
9     public static int proses() {
10         int ls; //Deklarasi variable lokal
11         ls = panjang * lebar;
12         return(ls);
13     }
14
15     public static void main (String[] arg) {
16         //Tahap input
17         System.out.print("Masukkan panjang persegi panjang: ");
18         panjang=sc.nextInt();
19         System.out.print("Masukkan lebar persegi panjang: ");
20         lebar=sc.nextInt();
21
22         //Tahap proses
23         //Memanggil fungsi proses Didalam Kelas Pemanggil Pada File yang Sama
24         luas=proses();
25
26         //Tahap output
27         System.out.print("Luas persegi panjang: " + luas);
28     }
29 }
```

Console

Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process
===== READY ==:



7.1.2. FUNGSI TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – SYNTAX

Syntax function:

```
Modifier static type functionName () {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    Instruction  
    Return(value)  
}
```

Syntax function caller:

```
variableName= className.functionName ()
```



7.1.2. FUNGSI TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_11.java
2 import java.util.Scanner;
3
4 class AnakProgram {
5     public static String garis() {
6         //Deklarasi variable lokal
7         String str1;
8         str1="===== ";
9         return(str1);
10    }
11
12    public static String judul() {
13        //Deklarasi variable lokal
14        String str2;
15        str2 = "==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ===="
16        return(str2);
17    }
18 }
19
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=17564). (Exit code 0)
===== READY =====
```



7.1.2. FUNGSI TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
20 public class Program_5_11 {
21     public static void main (String[] arg) {
22         //Deklarasi variabel
23         int panjang, lebar, luas;
24         String Garis, Judul;
25         Scanner sc = new Scanner(System.in);
26
27
28         Garis=AnakProgram.garis(); //Memanggil fungsi garis Diluar Kelas Pemanggil Pada File yang Sama
29         System.out.println(Garis);
30
31         Judul=AnakProgram.judul(); //Memanggil fungsi judul Diluar Kelas Pemanggil Pada File yang Sama
32         System.out.println(Judul);
33
34         Garis=AnakProgram.garis(); //Memanggil fungsi garis Diluar Kelas Pemanggil Pada File yang Sama
35         System.out.println(Garis);
36
37         //Tahap input
38         System.out.print("Masukkan panjang persegi panjang: ");
39         panjang=sc.nextInt();
40         System.out.print("Masukkan lebar persegi panjang: ");
41         lebar=sc.nextInt();
42
43         //Tahap proses
44         luas=panjang*lebar;
45
46         //Tahap output
47         System.out.print("Luas persegi panjang: " + luas);
48     }
49 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=17564). (Exit code 0)
===== READY =====
```



7.1.3. FUNGSI TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG BERBEDA – SYNTAX

Syntax function:

```
Modifier static type functionName () {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    Instruction  
    Return(value)  
}
```



7.1.3. FUNGSI TANPA PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG BERBEDA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_12.java
2 import java.util.Scanner;
3
4 public class Program_5_12 {
5     public static void main (String[] arg) {
6         //Deklarasi variabel
7         int panjang, lebar, luas;
8         String Garis, Judul;
9         Scanner sc = new Scanner(System.in);
10
11
12         Garis=Program_5_13.garis(); //Memanggil fungsi garis Diluar Kelas Pemanggil Pada File yang Sama
13         System.out.println(Garis);
14
15         Judul=Program_5_13.judul(); //Memanggil fungsi judul Diluar Kelas Pemanggil Pada File yang Sama
16         System.out.println(Judul);
17
18         Garis=Program_5_13.garis(); //Memanggil fungsi garis Diluar Kelas Pemanggil Pada File yang Sama
19         System.out.println(Garis);
20
21         //Tahap input
22         System.out.print("Masukkan panjang persegi panjang: ");
23         panjang=sc.nextInt();
24         System.out.print("Masukkan lebar persegi panjang: ");
25         lebar=sc.nextInt();
26
27         //Tahap proses
28         luas=panjang*lebar;
29
30         //Tahap output
31         System.out.print("Luas persegi panjang: " + luas);
32     }
33 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=6416). (Exit code 0)
===== READY =====
```



7.1.4. FUNGSI DENGAN PARAMETER PASSING DIDALAM KELAS PEMANGGIL YANG SAMA – SYNTAX

Syntax function:

```
Modifier static type functionName (argument1, argument2, ...) {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    Instruction  
    Return(value)  
}
```

Syntax function caller:

```
variableName=functionName (parameter1, parameter2, ...)
```



7.1.4. FUNGSI DENGAN PARAMETER PASSING DIDALAM KELAS PEMANGGIL YANG SAMA – CONTOH PROGRAM

CONTOH:
Menghitung Luas
Persegi Panjang

```
1 //File: Program_5_13.java
2 class Program_5_13 {
3     public static String garis() {
4         //Deklarasi variable lokal
5         String str1;
6         str1="===== ";
7         return(str1);
8     }
9
10    public static String judul() {
11        //Deklarasi variable lokal
12        String str2;
13        str2 = "=== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ===";
14        return(str2);
15    }
16 }
```

7.1.4. FUNGSI DENGAN PARAMETER PASSING DIDALAM KELAS PEMANGGIL YANG SAMA – CONTOH PROGRAM

```
1 //File: Program_5_14.java
2 import java.util.Scanner;
3
4 public class Program_5_14 {
5     public static int proses(int pj, int lbr) {
6         int ls; //Deklarasi lokal
7         ls=pj*lbr;
8         return(ls);
9     }
10
11     public static void main (String[] arg) {
12         //Deklarasi variabel
13         int panjang, lebar, luas;
14         Scanner sc = new Scanner(System.in);
15
16         //Menampilkan judul program
17         System.out.println("=====");
18         System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
19         System.out.println("=====");
20
21         //Tahap input
22         System.out.print("Masukkan panjang persegi panjang: ");
23         panjang=sc.nextInt();
24         System.out.print("Masukkan lebar persegi panjang: ");
25         lebar=sc.nextInt();
26
27         luas=proses(panjang,lebar); //Memanggil fungsi Didalam Kelas Pemanggil Pada File yang Sama
28
29         //Tahap output
30         System.out.print("Luas persegi panjang: " + luas);
31     }
32 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=18060). (Exit code 0)
===== READY =====
```

CONTOH:
Menghitung Luas
Persegi Panjang



7.1.5. FUNGSI DENGAN PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – SYNTAX

Syntax function:

```
Modifier static type functionName (argument1, argument2, ...) {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    Instruction  
    Return(value)  
}
```

Syntax function caller:

```
variableName=className.functionName (parameter1, parameter2, ...)
```



7.1.5. FUNGSI DENGAN PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG SAMA – CONTOH PROGRAM

```
1 //File: Program_5_15.java
2 import java.util.Scanner;
3
4 class AnakProgram {
5     public static int proses(int pj, int lbr) {
6         int ls; //Deklarasi lokal
7         ls=pj*lbr;
8         return(ls);
9     }
10 }
11
12 public class Program_5_15 {
13     public static void main (String[] arg) {
14         //Deklarasi variabel
15         int panjang, lebar, luas;
16         Scanner sc = new Scanner(System.in);
17
18         //Menampilkan judul program
19         System.out.println("=====");
20         System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
21         System.out.println("=====");
22
23         //Tahap input
24         System.out.print("Masukkan panjang persegi panjang: ");
25         panjang=sc.nextInt();
26         System.out.print("Masukkan lebar persegi panjang: ");
27         lebar=sc.nextInt();
28
29         luas=AnakProgram.proses(panjang,lebar); //Memanggil fungsi Diluar Kelas Pemanggil Pada File yang Sama
30
31         //Tahap output
32         System.out.print("Luas persegi panjang: " + luas);
33     }
34 }
```

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30 <<< Process finished (PID=10488). (Exit code 0)
===== READY =====
```

CONTOH:
Menghitung Luas
Persegi Panjang



Syntax function:

```
Modifier static type functionName (argument1, argument2, ...) {  
    Variabel declaration //Local variabel  
    Constanta declaration //Local constanta  
    Instruction  
    Return(value)  
}
```

Syntax function caller:

```
variableName=className.functionName (parameter1, parameter2, ...)
```



7.1.6. FUNGSI DENGAN PARAMETER PASSING DILUAR KELAS PEMANGGIL PADA FILE YANG BERBEDA – CONTOH PROGRAM

```
1 //File: Program_5_16.java
2 import java.util.Scanner;
3
4 public class Program_5_16 {
5     public static void main (String[] arg) {
6         //Deklarasi variabel
7         int panjang, lebar, luas;
8         Scanner sc = new Scanner(System.in);
9
10        //Menampilkan judul program
11        System.out.println("=====");
12        System.out.println("==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====");
13        System.out.println("=====");
14
15        //Tahap input
16        System.out.print("Masukkan panjang persegi panjang: ");
17        panjang=sc.nextInt();
18        System.out.print("Masukkan lebar persegi panjang: ");
19        lebar=sc.nextInt();
20
21        luas=Program_5_17.proses(panjang,lebar); //Memanggil fungsi Diluar Kelas Pemanggil Pada File berbeda
22
23        //Tahap output
24        System.out.print("Luas persegi panjang: " + luas);
25    }
26 }
```

```
1 //File: Program_5_17.java
2 class Program_5_17 {
3     public static int proses(int pj, int lbr) {
4         int ls; //Deklarasi lokal
5         ls=pj*lbr;
6         return(ls);
7     }
8 }
```

CONTOH:
Menghitung Luas
Persegi Panjang

Console

```
=====
==== PROGRAM PENGHITUNG LUAS PERSEGI PANJANG ====
=====
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Luas persegi panjang: 30<<< Process finished (PID=4464). (Exit code 0)
===== READY =====
```


8.1. METHOD OVERLOADING - PENGANTAR

- Metode overloading = fungsi yang secara otomatis menyesuaikan tipe data dari program pemanggil.
- Jika terdapat lebih dari satu tipe data pada parameter pemanggil → otomatis menyesuaikan pilihannya, sesuai tipe data yang di-passing
- Cara → membuat beberapa fungsi dalam sebuah kelas, setiap fungsi menyediakan tipe data yang berbeda agar dapat dipilih secara otomatis oleh java sesuai tipe data program pemanggilnya.

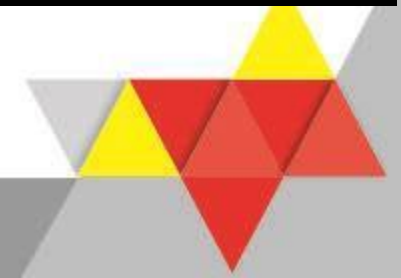


8.2. METHOD OVERLOADING - CONTOH PROGRAM

```
1 //File: Program_5_18.java
2 class Program_5_18 {
3     static int bagi(int pembilang, int pembagi) {
4         int hasil = pembilang / pembagi;
5         return(hasil);
6     }
7
8     static double bagi(double pembilang, double pembagi) {
9         double hasil = pembilang / pembagi;
10        return(hasil);
11    }
12
13    public static void main (String[] arg) {
14        System.out.println("Hasil pembagian ==> 9 dibagi 3 = " + bagi(9,3));
15        System.out.println("Hasil pembagian ==> 9.0 dibagi 3.0 = " + bagi(9.0,3.0));
16    }
17 }
```

Console

Hasil pembagian ==> 9 dibagi 3 = 3
Hasil pembagian ==> 9.0 dibagi 3.0 = 3.0
<<< Process finished (PID=16804). (Exit code 0)
===== READY =====



8.3. METHOD OVERLOADING

CONTOH PROGRAM – DENGAN SCANNER

```
1 //File: Method_Overloading_dengan_Scanner.java
2 import java.util.Scanner;
3
4 class Method_Overloading_dengan_Scanner {
5
6     static int hitungLuasPP(int p, int l) {
7         int L = p * l;
8         return(L);
9     }
10
11     static double hitungLuasPP(double p, double l) {
12         double L = p * l;
13         return(L);
14     }
15
16     public static void main (String[] arg) {
17         //Deklarasi variabel
18         int panjang, lebar, luas;
19         double pj, lbr, ls;
20         Scanner sc = new Scanner(System.in);
21
22         //Tahap input - tipe data integer
23         System.out.print("Masukkan panjang persegi panjang: ");
24         panjang = sc.nextInt();
25         System.out.print("Masukkan lebar persegi panjang: ");
26         lebar = sc.nextInt();
```

```
27
28         //Tahap input - tipe data double
29         System.out.print("Masukkan panjang persegi panjang: ");
30         pj = sc.nextDouble();
31         System.out.print("Masukkan lebar persegi panjang: ");
32         lbr = sc.nextDouble();
33
34         //Memanggil Methode Overloading - tipe data integer
35         luas = hitungLuasPP(panjang,lebar);
36
37         //Memanggil Methode Overloading - tipe data double
38         ls = hitungLuasPP(pj,lbr);
39
40         //Tahap output - tipe data integer
41         System.out.println("Luas PP = " + luas);
42         //Tahap output - tipe data double
43         System.out.println("Luas PP = " + ls);
44     }
45 }
```

```
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Masukkan panjang persegi panjang: 5.0
Masukkan lebar persegi panjang: 6.0
Luas PP = 30
Luas PP = 30.0
<<< Process finished (PID=10464). (Exit code 0)
===== READY =====
```

8.4. METHOD OVERLOADING

CONTOH PROGRAM – DENGAN BUFFERREADER

```
1 //File: Method_Overloading_dengan_BufferReader.java
2 import java.io.*;
3
4 class Method_Overloading_dengan_BufferReader {
5
6     static int hitungLuasPP(int p, int l) {
7         int L = p * l;
8         return(L);
9     }
10
11     static double hitungLuasPP(double p, double l) {
12         double L = p * l;
13         return(L);
14     }
15
16     public static void main (String[] arg) throws IOException {
17         //Deklarasi variabel
18         int panjang, lebar, luas;
19         double pj, lbr, ls;
20         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
21
22         //Tahap input - tipe data integer
23         System.out.print("Masukkan panjang persegi panjang: ");
24         panjang = Integer.parseInt(br.readLine());
25         System.out.print("Masukkan lebar persegi panjang: ");
26         lebar = Integer.parseInt(br.readLine());
```

```
27
28         //Tahap input - tipe data double
29         System.out.print("Masukkan panjang persegi panjang: ");
30         pj = Double.parseDouble(br.readLine());
31         System.out.print("Masukkan lebar persegi panjang: ");
32         lbr = Double.parseDouble(br.readLine());
33
34         //Memanggil Methode Overloading - tipe data integer
35         luas = hitungLuasPP(panjang,lebar);
36
37         //Memanggil Methode Overloading - tipe data double
38         ls = hitungLuasPP(pj,lbr);
39
40         //Tahap output - tipe data integer
41         System.out.println("Luas PP = " + luas);
42         //Tahap output - tipe data double
43         System.out.println("Luas PP = " + ls);
44     }
45 }
46
```

Console

```
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 6
Masukkan panjang persegi panjang: 5.0
Masukkan lebar persegi panjang: 6.0
Luas PP = 30
Luas PP = 30.0
<<< Process finished (PID=3196). (Exit code 0)
===== READY =====
```

9. KESIMPULAN

- Sub proses adalah potongan program digunakan untuk mempermudah pembuatan program yang kompleks karena memindahkan bagian program tertentu didalam program ke luar program kecil.
- Hal ini potongan program tersebut sering digunakan kembali, selain itu untuk memudahkan dalam pelacakan saat debugging.
- Pemanggilan sub proses bisa menggunakan parameter ataupun tanpa parameter untuk keperluan pemberian/passing nilai tertentu ke sub proses.



**SELESAI
TERIMA KASIH**

