



**Nama** Edgrant Henderson Suryajaya  
**NPM** 2206025016

**Kode Asisten** .....  
**Jenis Tugas** TP

1. Procedure, function, dan impure function dalam VHDL merupakan subprogram yang dapat digunakan untuk mempermudah pemrograman dan memudahkan untuk membaca program. Procedure digunakan untuk melakukan serangkaian operasi yang tidak mengembalikan nilai, tetapi dapat mengubah beberapa signal/variabel. Procedure dapat menggunakan *keyword* seperti wait dan after.

Function digunakan untuk menghitung dan mengembalikan suatu nilai berdasarkan input yang diberikan. Function tidak dapat menggunakan *keyword* seperti wait dan after dan harus *pure*, yaitu nilai kembali hanya bergantung pada input, tanpa efek samping mengubah signal/variabel. Namun, terdapat impure Function, yaitu function yang mengembalikan nilai yang tidak hanya bergantung pada input, tetapi juga pada signal/variabel lain dan juga dapat memiliki efek samping

#### **Referensi:**

John, "Using Procedures, Functions and packages in VHDL - FPGA tutorial," *FPGA Tutorial*, Nov. 29, 2021. Available: <https://fpgatutorial.com/vhdl-function-procedure-package/>

J. J. Jensen, "How to use a function in VHDL," *VHDLwhiz*, Jul. 08, 2023. Available: <https://vhdlwhiz.com/function/>

2. Function dan procedure dapat disintesis jika bersifat deterministik dan synchronous, yaitu hasilnya hanya bergantung pada argumen masukan dan tidak berubah seiring waktu. Sedangkan, impure function adalah jenis subprogram yang menghasilkan nilai yang tidak dapat diprediksi atau bergantung pada faktor-faktor eksternal, seperti waktu, nilai acak, atau variabel global. Jenis subprogram ini biasanya tidak cocok untuk sintesis hardware yang bersifat deterministik dan synchronous.

**Referensi:**

FTUI, Digital Laboratory “Dasar Teori Modul 6: PROCEDURE, FUNCTION AND IMPURE FUNCTION”, 2023. [online]. [Accessed: 05-NOV-2023].

3. Dalam VHDL, fungsi, prosedur, dan fungsi impure memiliki peran yang berbeda dan biasanya tidak dapat saling menggantikan. Fungsi berfungsi untuk mengambil input dan mengembalikan output yang dihitung. Tidak dapat menggunakan konstruksi yang menghabiskan waktu. Prosedur berfungsi untuk melakukan proses sekuensial dan mengembalikan nilai melalui objek global atau parameter formal. Fungsi Impure dapat mengembalikan nilai yang berbeda walaupun parameter yang diberikan sama.

Jadi, meskipun mereka semua adalah subprogram, mereka memiliki peran yang berbeda dan tidak dapat saling menggantikan. Namun, dalam beberapa kasus tertentu, mungkin ada beberapa saat fungsionalitas mereka dapat digantikan dengan yang lain, tetapi secara umum, mereka digunakan untuk tujuan yang berbeda.

**Referensi:**

John, “Using Procedures, Functions and packages in VHDL - FPGA tutorial,” *FPGA Tutorial*, Nov. 29, 2021. Available: <https://fpgatutorial.com/vhdl-function-procedure-package/>

4. Dalam VHDL, subprogram dan structural programming memiliki perbedaan dalam cara mereka digunakan. Subprogram biasanya merujuk pada fungsi dan prosedur yang digunakan untuk mengelompokkan sekelompok pernyataan menjadi satu unit. Subprogram ini dapat dipanggil berulang kali agar kode lebih mudah dibaca dan dipelihara. Structural programming adalah metode untuk membuat desain besar dengan menggabungkan beberapa entitas yang lebih kecil. Entitas yang lebih kecil dalam file terpisah dan dapat menggunakannya dalam entitas yang lebih besar sebagai komponen.

Secara umum, perbedaan antara subprogram dan structural programming tergantung pada kebutuhan spesifik desain. Keduanya memiliki kelebihan dan kekurangan masing-masing dan dapat digunakan secara efektif dalam berbagai skenario.

**Referensi:**

Efy, "Subprograms in VHDL," *Electronics for You*, Sep. 07, 2023. Available:

<https://electronicsforyou.in/subprograms-in-vhdl/>

D. Joshi, "Structural modeling architecture in VHDL," *Technobyte*, May 03, 2020.

Available: <https://technobyte.org/vhdl-structural-modeling-style-architecture/>

- a. Lampu merah procedure (soal bonus)

```
library ieee;
use ieee.std_logic_1164.all;

entity TP_PSD7_EdgrantHendersonSuryajaya_2206025016_a is
  port (
    clk : in std_logic;
    aktifkanLampu : in std_logic;
    RedLight : out std_logic;
    YellowLight : out std_logic;
    GreenLight : out std_logic
  );
end entity;
```



```
);  
end TP PSD7 EdgrantHendersonSuryajaya 2206025016_a;  
  
architecture rtl of TP PSD7 EdgrantHendersonSuryajaya 2206025016_a is  
    procedure ChangeLightStatus(signal light : out std_logic; status : in  
std_logic) is  
    begin  
        light <= status;  
    end procedure ChangeLightStatus;  
  
    signal state : integer range 0 to 2;  
begin  
    process(clk)  
    begin  
        if rising_edge(clk) then  
            if aktifkanLampu = '1' then  
                if state = 0 then  
                    ChangeLightStatus(RedLight, '0');  
                    ChangeLightStatus(YellowLight, '1');  
                    state <= 1;  
                elsif state = 1 then  
                    ChangeLightStatus(YellowLight, '0');  
                    ChangeLightStatus(GreenLight, '1');  
                    state <= 2;  
                elsif state = 2 then  
                    ChangeLightStatus(GreenLight, '0');  
                    ChangeLightStatus(RedLight, '1');  
                    state <= 0;  
                end if;  
            end if;  
        end if;  
    end process;  
end rtl;
```





## b. Faktorial function

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity TP_PSD7_EdgranthendersonSuryajaya_2206025016_b is
    port (
        N : in integer range 0 to 16 := 0;
        factorial : out integer := 0
    );
end TP_PSD7_EdgranthendersonSuryajaya_2206025016_b;

architecture rtl of TP_PSD7_EdgranthendersonSuryajaya_2206025016_b is
    function faktorial (n : integer) return integer is
    begin
        if n < 2 then return 1;
        else return n*faktorial(n - 1);
        end if;
    end function;
begin
    factorial <= faktorial(N);
end rtl;
```

Wave - Default							
		Msgs					
	/tp_psd7_edgranthendersonsuryajaya_2206025016_b/N	4	0	1	2	3	4
	/tp_psd7_edgranthendersonsuryajaya_2206025016_b/factorial	24	1	2	6	24	



c. Suhu impure function

A=1      B=6

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
use IEEE.MATH_REAL.ALL;

entity TP_PSD7_EdgrantHendersonSuryajaya_2206025016_c is
    port (
        CLK : in std_logic;
        Suhu : out integer := 0
    );
end TP_PSD7_EdgrantHendersonSuryajaya_2206025016_c;

architecture rtl of TP_PSD7_EdgrantHendersonSuryajaya_2206025016_c is
begin
    process(CLK)
        variable temp : integer;
        variable seed1, seed2: integer := 999;
        impure function suhuRandom(min_val: integer; max_val : integer) return
integer is
            variable r : real;
            begin
                uniform(seed1, seed2, r);
                return integer(
                    round(r * real(max_val - min_val + 1) + real(min_val) - 0.5));
            end function;
        begin
            if rising_edge(CLK) then
                temp := suhuRandom(-56, 81);
                Suhu <= temp;
            end if;
        end process;
    end rtl;
```

		Process									
TP_PSD7_EdgrantHendersonSuryajaya_2206025016_c/CLK	1										
TP_PSD7_EdgrantHendersonSuryajaya_2206025016_c/Suhu	10	8	81	19	98	97	68	19			