

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

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

1. Looping construct adalah struktur kontrol yang memungkinkan untuk menentukan urutan perilaku yang berulang dalam kode VHDL. Tujuannya adalah untuk mengeksekusi blok kode yang sama beberapa kali.

**Referensi:**

“Introduction to Digital systems: modeling, synthesis, and simulation using VHDL,”

*O'Reilly Online Learning*. Available:

<https://www.oreilly.com/library/view/introduction-to-digital/9780470900550/chap4-sec014.html>

2. Ada dua jenis utama looping construct dalam VHDL:
  - For loop: Digunakan untuk iterasi melalui rentang nilai tertentu. Misalnya, for i in 0 to 9 loop ... end loop; akan melakukan iterasi dari 0 hingga 9.
  - While loop: Digunakan untuk melakukan iterasi selama kondisi tertentu benar.Perbedaan utamanya adalah for loop digunakan saat ingin loop yang menghitung, dan while loop saat ingin ulang berdasarkan kondisi.

**Referensi:**

“Introduction to Digital systems: modeling, synthesis, and simulation using VHDL,”

*O'Reilly Online Learning*. Available:

<https://www.oreilly.com/library/view/introduction-to-digital/9780470900550/chap4-sec014.html>

FTUI, Digital Laboratory “Dasar Teori Modul 6: STRUCTURAL STYLE

PROGRAMMING IN VHDL”, 2023. [online]. [Accessed: 03-OCT-2023].

3. Next dan Exit statement digunakan untuk mengontrol aliran eksekusi dalam loop
  - Next statement: Digunakan untuk melompati sisa iterasi (seperti continue di bahasa C) dan langsung melanjutkan ke iterasi berikutnya dalam loop. Misalnya, dalam for i in 0 to 9 loop if i=5 then next; end if; ketika i=5, pernyataan ‘next’ akan dipanggil dan semua kode setelahnya dalam loop akan dilewati.
  - Exit statement: Digunakan untuk keluar dari loop sepenuhnya (seperti break di bahasa C). Misalnya, dalam for i in 0 to 9 loop if i=5 then exit; ketika i=5, pernyataan ‘exit’ akan dipanggil dan seluruh loop akan dihentikan.

**Referensi:**

“Next statement.” Available:

[https://www.hdlworks.com/hdl\\_corner/vhdl\\_ref/VHDLContents/NextStatement.htm](https://www.hdlworks.com/hdl_corner/vhdl_ref/VHDLContents/NextStatement.htm)

“Exit statement.” Available:

[https://www.hdlworks.com/hdl\\_corner/vhdl\\_ref/VHDLContents/ExitStatement.htm](https://www.hdlworks.com/hdl_corner/vhdl_ref/VHDLContents/ExitStatement.htm)



4. For loop syntaxnya adalah

```
for loop_var in 0 to 4 loop  
end loop;
```

ini akan loop 0, 1, 2, 3, 4

```
for loop_var in 4 downto 0 loop  
end loop;
```

ini akan loop 4, 3, 2, 1, 0



## Nomor 2

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

entity SReg is
    port (
        dataIn : IN std_logic_vector (7 downto 0);
        K : IN std_logic;
        jumShift : IN std_logic_vector (2 downto 0);
        dataOut : OUT std_logic_vector (7 downto 0)
    );
end entity SReg;

architecture rtl of SReg is
begin
    shift: process(dataIn, K, jumShift)
        variable loopLen : integer := 0;
        variable dataTemp : std_logic_vector (7 downto 0) := (others => '0');
    begin
        dataTemp := dataIn;
        loopLen := to_integer(UNSIGNED(jumShift));

        for i in 1 to loopLen loop
            if k = '0' then
                dataTemp := dataTemp(0) & dataTemp(7 downto 1);
            elsif k = '1' then
                dataTemp := dataTemp(6 downto 0) & dataTemp(7);
            else
                dataTemp := "00000000";
            end if;
        end loop;
        dataOut <= dataTemp;
    end process shift;
end architecture rtl;
```



## Screenshot

Wave - Default		Msgs							
/sreg/dataIn	11001100	11001100							
/sreg/K	1								
/sreg/jumShift	011	011							
/sreg/dataOut	01100110	10011001				01100110			