

Sistem Kebut Semalam

Yunus Febriansyah

EPISODE 2

BELAJAR DASAR C++

Sejarah Singkat ?

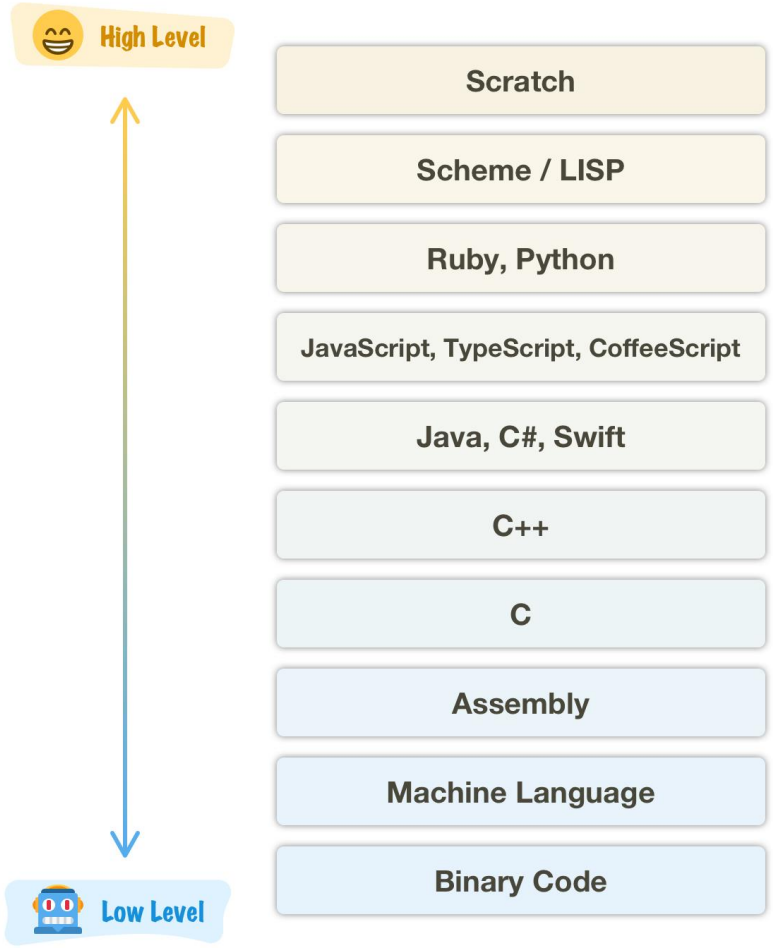
- Bahasa C++ lahir pada tahun 1980, yang dibuat oleh Bjarne Stroustrup di AT&T Bell Laboratories awal tahun 1980-an berdasarkan C ANSI (American National Standard Institute). Pertama kali, prototype C++ muncul sebagai C yang dipercanggih dengan fasilitas kelas, bahasa tersebut disebut “C dengan kelas” (C with Class).

- Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading.[butuh rujukan] Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (object-oriented programming).

- Biodata biografi.
  https://en.wikipedia.org/wiki/Bjarne_Stroustrup

Source :https://www.stroustrup.com/

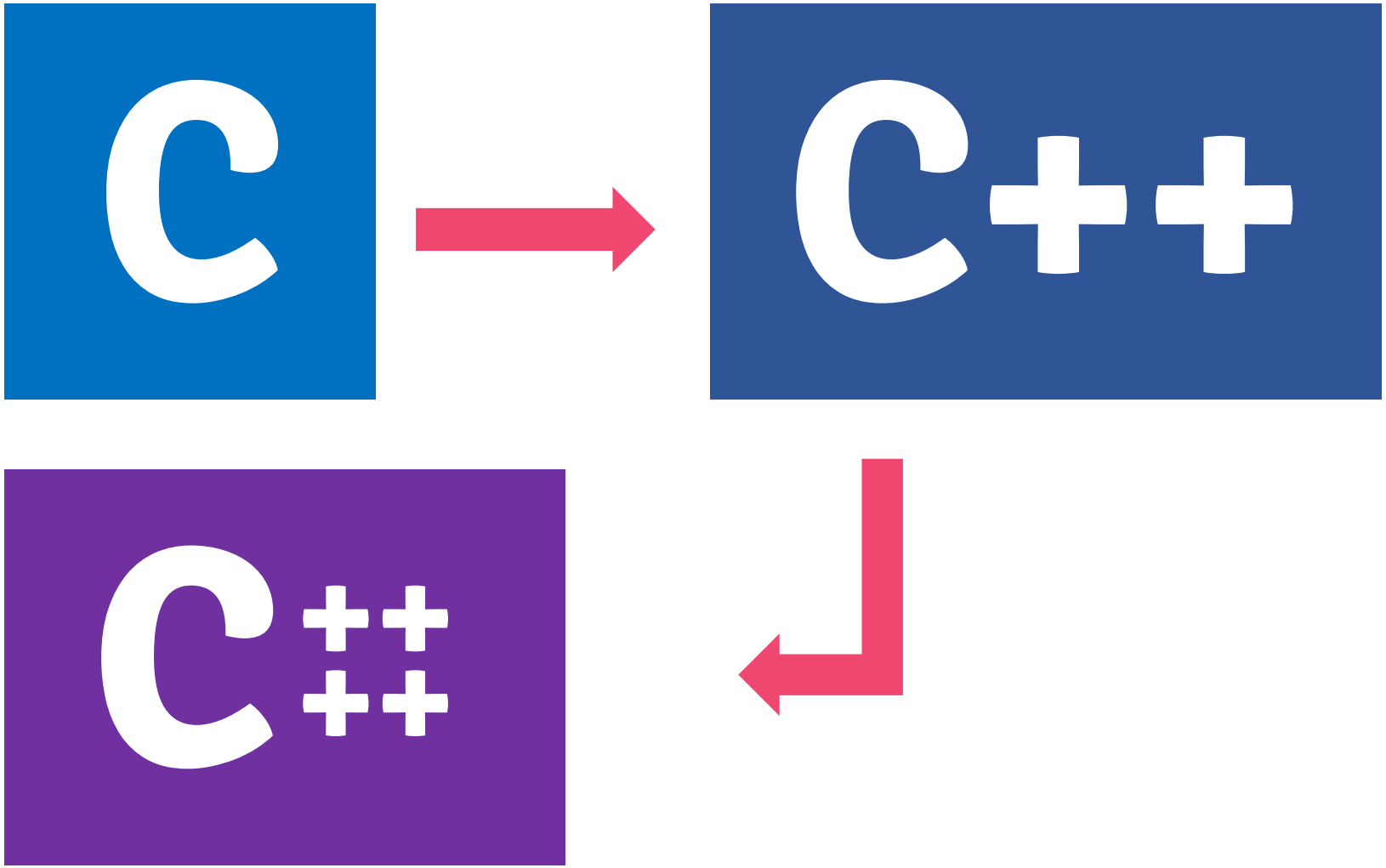
Low & High Level Programming Lang ?



Standarisasi C++ ?

Tahun	C++ Standar	Nama Informal
1998	ISO/IEC 14882:1998	C++98
2003	ISO/IEC 14882:2003	C++03
2011	ISO/IEC 14882:2011	C++11, C++0x
2014	ISO/IEC 14882:2014	C++14, C++1y
2017	ISO/IEC 14882:2017	C++17, C++1z
2020	Belum nemu	C++20,[17] C++2a

C, C++, C# ?



Contoh Aplikasi dengan C++ ?

- Winamp Media Player
- MySQL Server
- Mozilla Firefox
- Google Chrome
- Microsoft Office
- Adobe Photoshop & Illustrator
- Inkscape
- dan masih banyak lagi.

Kenapa C++ ?

- Bahasa C++ memiliki kapabilitas yang sangat baik sehingga programmer dapat memperoleh seluruh tenaga yang dimiliki komputer.

- Dapat dikembangkan di berbagai platform sehingga aplikasi yang dibangun dapat berjalan di sistem operasi yang berbeda.

- Compiler C++ yang sangat baik sehingga dapat mempercepat proses kompilasi.

- C++ merupakan bahasa terstruktur yang mendukung OOP(Object Oriented Programming).

- C++ dikategorikan sebagai bahasa tingkat menengah sehingga mendekati bahasa mesin.

- Dengan bahasa C++ kita dapat membangun aplikasi graphic processor menggunakan librari OpenGL.

- Kemudahan dalam memanipulasi data seperti merubah alamat dari suatu variabel menggunakan pointer

Dengan C++ ?

- Mempelajari Kernel dari Sistem Operasi
- Membangun aplikasi desktop
- Membuat aplikasi mikrokontroler
- Ikut mengembangkan teknologi open source
- Membuat library untuk bahasa pemrograman lain
- Membuat aplikasi perangkat mobile
- Membuat Game

Notes ?

- Compiler C++ yang banyak

- Banyak Bahasa pemrograman keturunan C

studywithstudent

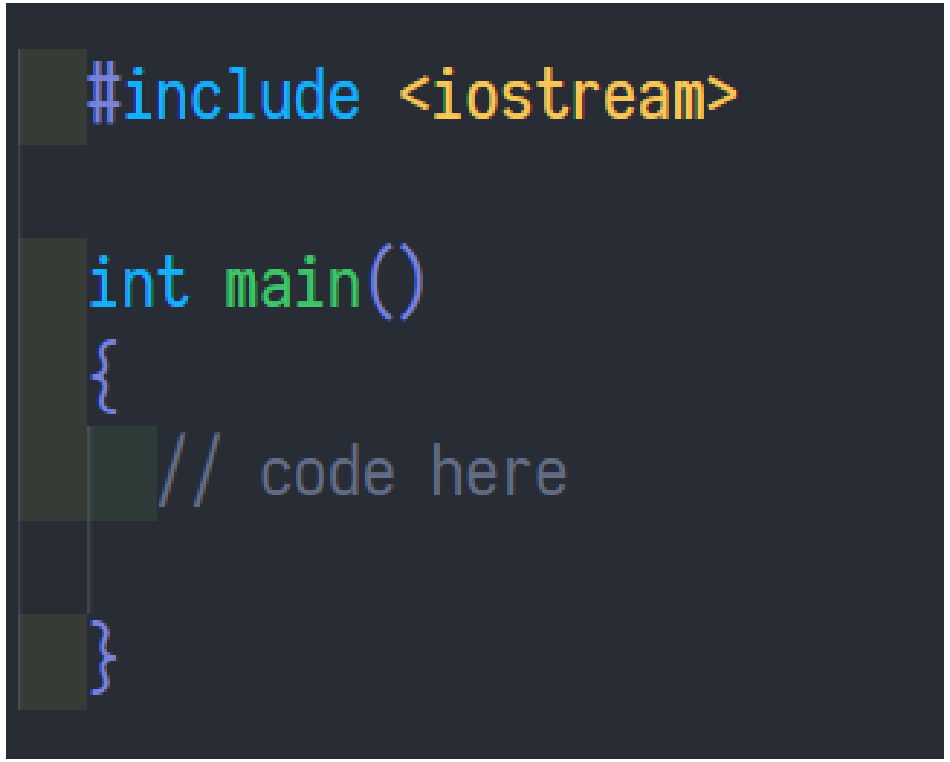
Materi ?

1. Pengantar
2. Persiapan
3. Pendahuluan
4. Comment
5. Variabel, Konstanta & Tipe Data
6. User Input
7. Operator
8. Casting
9. Percabangan IF
10. Percabangan Ternary
11. Percabangan Switch Case
12. For Loop
13. While Loop
14. Do.. While Loop
15. Array
16. Void Function
17. Return Function
18. Recursive Function
19. Variable Scope (Local, Global & Block)

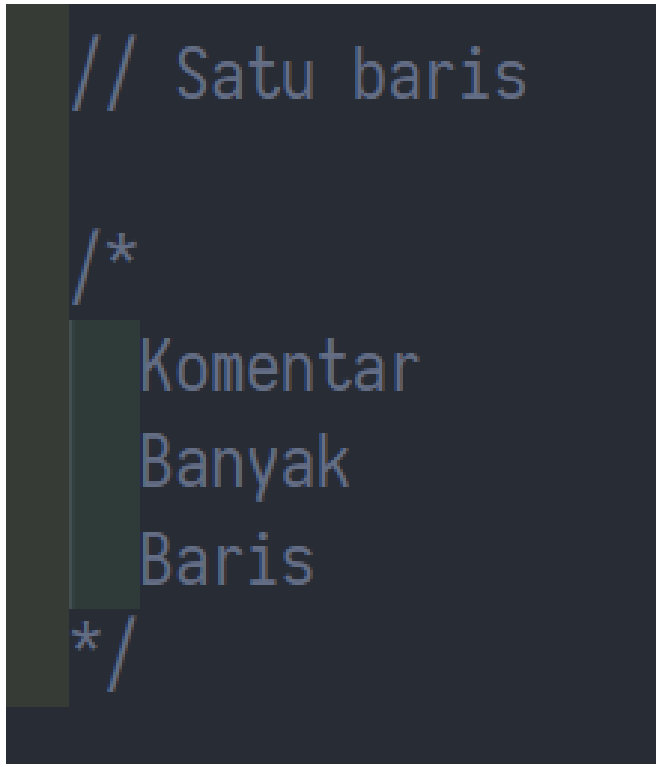
Persiapan ?

* Text Editor

Pendahuluan ?



Comment ?



Variabel, Konstanta & Tipe Data ?

```
// Declare Variable
/*
  dataType varName;
  dataType varName = varValue;
  dataType varName1, varName2, varNameN;
  dataType varName1 = varVal1, varName2
  = varVal2;
*/

// Declare Constant
/*
  const dataType constName = constValue;
  const dataType constName1 =
  constVal1, constName2 = constVal2;
*/
```

```
// Bilangan Bulat
  int Integer;
// Bilangan Pecahan
  float Float;
// Bilangan Pecahan
  double Double;
// Boolean
  bool Boolean;
// String
  string String;
```

User Input ?

```
// Declare Variable
    // dataType varName;

// User Input
    // cin >> varName;

// for String
    // getline(cin, varName)
```

Operator ?

```
// Aritmatika
    // +, -, *, /, %, ++, --

// Pengisian / Assignment
    // =, +=, -=, *=, /=, %=,
    // ^=, &=, |=, >>=, <<=

// Pembanding
    // ==, !=, >, <, >=, <=

// Logika
    // &&, ||, !
```

Casting ?

```
// Berlaku untuk bilangan
// (dataType) varName or Number
```

studywithstudent

Percabangan IF ?

```
// IF Tunggal
/*
  if (condition) {
    // block of code to be executed if the
    condition is true
  }
*/

// IF Ganda
/*
  if (condition) {
    // block of code to be executed if the
    condition is true
  }else{
    // block of code to be executed if the
    condition is false
  }
*/
```

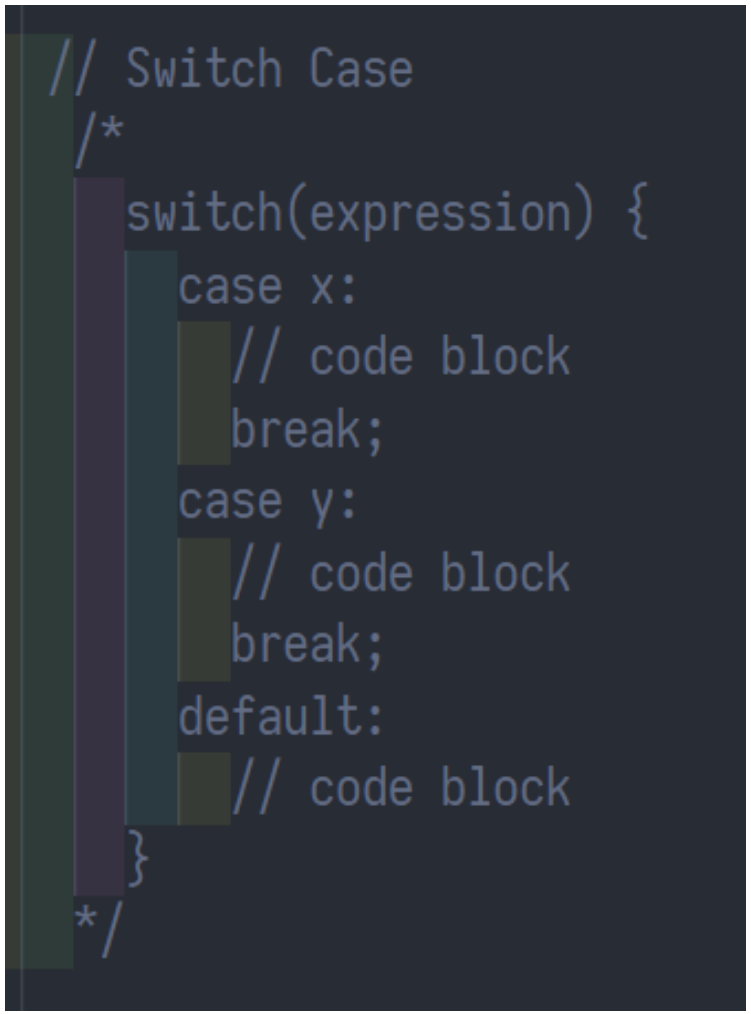
```
// IF Majemuk
/*
  if (condition1) {
    // block of code to be executed if the
    condition1 is true
  }else if (condition2) {
    // block of code to be executed if the
    condition2 is true
  }else if (conditionN) {
    // block of code to be executed if the
    conditionN is true
  }else{
    // block of code to be executed if all
    condition is false
  }
*/
```

Percabangan Ternary ?

```
// variable = (condition) ? expressionTrue : expressionFalse;
```

studywithstudent

Percabangan Switch Case ?



```
// Switch Case
/*
  switch(expression) {
    case x:
      // code block
      break;
    case y:
      // code block
      break;
    default:
      // code block
  }
*/
```

For Loop ?

```
// For Loop
/*
    for (nilaiAwal; kondisi; step) {
        // Eksekusi
    }
*/
```

While Loop ?

```
// While Loop
/*
    while (condition) {
        // code block to be executed
    }
*/
```

Do..While Loop ?

```
// Do..While Loop
/*
    do {
    // code block to be executed
    }
    while (condition);
*/
```

Array ?

```
// Declare
/*
  dataType arrName[arrSize];
  dataType arrName[arrSize] = {val1, valN};
*/

// Assignment
/*
  arrName[index] = value;
*/
```

Void Function ?

```
// Declare
  /*
    void funcName(){
      // code to be executed
    }
  */
  /*
    void funcName(dataType par1, dataType parN){
      // code to be executed
    }
  */



// Call
  /*
    funcName();
  */
  /*
    funcName(arg1, argN);
  */
```

Return Function ?

```
// Declare
  /*
    dataType funcName(){
      // code to be executed
      return (value must match data type)
    }
  */
  /*
    void funcName(dataType par1, dataType parN){
      // code to be executed
      return (value must match data type)
    }
  */

// Call
  /*
    funcName();
  */
  /*
    funcName(arg1, argN);
  */
```

Recursive Function ?

```
// Declare
/*
  dataType funcName(){
    // code to be executed
    if(condition){
      funcName();
    }
    return (value must match data type)
  }
*/
/*
  void funcName(dataType par1, dataType parN){
    // code to be executed
    if(condition){
      funcName(arg1, argN);
    }
    return (value must match data type)
  }
*/
```

```
// Call
/*
  funcName();
*/
/*
  funcName(arg1, argN);
*/
```

Variable Scope (Local, Global & Block) ?

```cpp
#include <iostream>

// global variable
/*
    dataType varName;
    dataType varName = varVal;
*/

```

```cpp
int main()
{

    // Local Variable
    /*
        dataType varName;
        dataType varName = varVal;
    */

    {
        // Block Variable
        /*
            dataType varName;
            dataType varName = varVal;
        */
    }

}
```

Live Streaming

ENDING...

Thank you

#KEEPLEARNING
#KEEPSPIRITS