

- Sejarah singkat C++
 - Dikembangkan oleh Bjarne Stroustrup di Bell Labs (1979)
 - Awalnya bernama "C with Classes"
 - C++ pertama kali dirilis tahun 1985
 - Merupakan pengembangan dari bahasa C dengan tambahan Object-Oriented Programming
- Mengapa dinamakan "C++"?
 - "++" adalah operator increment dalam C
 - C++ = C + fitur tambahan (enhanced version of C)
 - Simbolis: evolusi dari bahasa C
- Karakteristik utama C++:

- **Compiled language:** Kode diterjemahkan ke machine code sebelum dijalankan
- **Case-sensitive:** Membedakan huruf besar dan kecil
- **Strongly typed:** Setiap variabel harus memiliki tipe data yang jelas
- **Multi-paradigm:** Mendukung procedural, object-oriented, dan generic programming

```
// Contoh case-sensitive
int Nama; // berbeda dengan
int nama; // variable ini

// Contoh strongly typed
int angka = 10; // Harus integer
string teks = "Hello"; // Harus string
// angka = "Hello"; // ERROR! Tidak bisa assign string ke int
```

Setup Development Environment - Part 1

Slide 2

- Apa yang kita butuhkan untuk coding C++?
 - **Text Editor/IDE:** Tempat menulis kode
 - **Compiler:** Penerjemah kode C++ ke machine code
 - **Debugger:** Tool untuk mencari bug/error
- Step 1: Download dan Install VS Code
 - Kunjungi: <https://code.visualstudio.com/>
 - Download sesuai OS (Windows/Mac/Linux)
 - Install dengan setting default
 - Buka VS Code setelah instalasi selesai
- Step 2: Install Extension C/C++

- Buka VS Code
- Klik icon Extensions (Ctrl+Shift+X)
- Cari "C/C++" dari Microsoft
- Klik "Install"
- Extension ini memberikan:
 - Syntax highlighting
 - IntelliSense (auto-complete)
 - Debugging capabilities
 - Code formatting

Setup Development Environment - Part 2

Slide 3

- Step 3: Install Compiler

Untuk Windows:

- Download MinGW-w64 dari: <https://www.mingw-w64.org/downloads/>
- Atau gunakan MSYS2: <https://www.msys2.org/>
- Tambahkan ke PATH environment variable
- Test dengan: `gcc --version` di Command Prompt

Untuk macOS:

- Install Xcode Command Line Tools:

```
xcode-select --install
```

- Atau install via Homebrew: `brew install gcc`

Untuk Linux:

```
sudo apt-get update  
sudo apt-get install build-essential
```

- Step 4: Test Installation

- Buat folder project baru: `cpp-learning`
 - Buat file `hello.cpp`
 - Compile dan jalankan program pertama

```
// File: hello.cpp  
#include <iostream>  
using namespace std;  
  
int main() {  
    cout << "Hello, C++!" << endl;  
    return 0;  
}
```

- Step 5: Compile dan Run

```
g++ hello.cpp -o hello
./hello          # Mac/Linux
hello.exe       # Windows
```

- Output yang diharapkan: Hello, C++!

- Mari kita bedah program C++ paling sederhana:

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello World!" << endl;
    return 0;
}
```

- **Line 1:** `#include <iostream>`

- Preprocessor directive untuk menyertakan header file
- `iostream` = Input Output Stream
- Berisi definisi `cout`, `cin`, `endl`
- Harus ada jika kita ingin melakukan input/output

- **Line 2:** `using namespace std;`

- Namespace adalah cara mengelompokkan kode
- `std` = standard library namespace
- Tanpa ini, kita harus menulis `std::cout` dan `std::endl`
- Mempermudah penulisan kode

```
// Tanpa using namespace std
#include <iostream>

int main() {
    std::cout << "Hello World!" << std::endl;
    return 0;
}
```

- **Line 4:** `int main() {`

- `main()` = entry point program (titik mulai eksekusi)
- Setiap program C++ HARUS memiliki fungsi main()
- `int` = main() mengembalikan nilai integer ke sistem operasi
- `{}` = curly braces menandai blok kode

- **Line 5:** `cout << "Hello World!" << endl;`

- `cout` = Console Output (untuk menampilkan ke layar)
- `<<` = insertion operator (memasukkan data ke output stream)
- `"Hello World!"` = string literal yang akan ditampilkan
- `endl` = end line (pindah baris + flush buffer)
- `;` = semicolon mengakhiri setiap statement

- **Line 6:** `return 0;`

- Mengembalikan nilai 0 ke sistem operasi
- 0 = program berhasil dijalankan
- Non-zero = ada error dalam program
- Wajib ada karena main() bertipe `int` (akan ditambahkan otomatis oleh compiler mulai versi C++ 2003)

```
// Contoh program dengan multiple output
#include <iostream>
using namespace std;

int main() {
    cout << "Baris pertama" << endl;
    cout << "Baris kedua" << endl;
    cout << "Angka: " << 123 << endl;
    return 0;
}
```

Output:

```
Baris pertama
Baris kedua
```

Angka: 123

Input Output - cout (Console Output)

- **cout = Console Output**

- Digunakan untuk menampilkan data ke layar
- Bagian dari namespace `std`
- Menggunakan operator `<<` (insertion operator)

- **Cara penggunaan cout:**

```
cout << "Teks yang ingin ditampilkan";
cout << variabel_name;
cout << "Teks" << variabel << "Teks lagi";
```

- **Contoh penggunaan cout:**

```
#include <iostream>
using namespace std;
```

```
int main() {
    int umur = 20;
    string nama = "Budi";

    cout << "Nama: " << nama << endl;
    cout << "Umur: " << umur << " tahun" << endl;
    cout << "Selamat datang di kuliah C++!" << endl;

    return 0;
}
```

Output:

```
Nama: Budi
Umur: 20 tahun
Selamat datang di kuliah C++!
```

- `endl` **vs** `\n` :
 - `endl` = end line + flush buffer (memastikan output langsung tampil)
 - `\n` = new line saja (lebih cepat)
 - Untuk pemula, gunakan `endl` untuk memastikan output tampil

- **cin = Console Input**

- Digunakan untuk menerima input dari user
- Menggunakan operator `>>` (extraction operator)
- Input akan disimpan ke dalam variabel

- **Cara penggunaan cin:**

```
cin >> nama_variabel;  
cin >> var1 >> var2 >> var3; // Multiple input
```

- **Contoh program interaktif:**

```
#include <iostream>  
#include <string>  
using namespace std;
```

```
int main() {
    string nama;
    int umur;
    float tinggi;

    cout << "Masukkan nama: ";
    cin >> nama;

    cout << "Masukkan umur: ";
    cin >> umur;

    cout << "Masukkan tinggi (cm): ";
    cin >> tinggi;

    cout << "\n==== BIODATA ===" << endl;
    cout << "Nama: " << nama << endl;
    cout << "Umur: " << umur << " tahun" << endl;
    cout << "Tinggi: " << tinggi << " cm" << endl;

    return 0;
}
```

- **Penting untuk diingat:**

- `cin` berhenti membaca saat bertemu spasi atau enter
- Untuk input string dengan spasi, gunakan `getline(cin, variabel)`

- Selalu deklarasikan variabel sebelum menggunakan `cin`

- **Apa itu variabel?**

- Tempat penyimpanan data di memory
- Memiliki nama untuk memudahkan akses
- Seperti kotak dengan label yang berisi sesuatu

- **Deklarasi dan inisialisasi variabel:**

```
// Deklarasi (membuat variabel)
int umur;

// Inisialisasi (memberikan nilai)
umur = 20;

// Deklarasi + Inisialisasi sekaligus
int umur = 20;
string nama = "Budi";
```

- **Aturan penamaan variabel:**

- Harus diawali huruf atau underscore (_)
- Tidak boleh diawali angka
- Tidak boleh menggunakan spasi
- Case-sensitive (umur ≠ Umur)
- Tidak boleh menggunakan reserved words (int, for, if, dll)

```
// ? Valid
int umur;
string nama_lengkap;
float nilaiUTS;
char _grade;

// ? Invalid
int lnilai;      // Diawali angka
string nama lengkap; // Ada spasi
int for;         // Reserved word
```

- **Mengapa perlu tipe data?**

- Menentukan ukuran memory yang digunakan
- Menentukan operasi yang bisa dilakukan
- Mencegah error saat compile time

- **Integer Types (Bilangan Bulat):**

```
int angka = 42;           // -2,147,483,648 to 2,147,483,647
short kecil = 100;         // -32,768 to 32,767
long besar = 1000000L;     // Lebih besar dari int
long long sangat_besar = 9223372036854775807LL;
```

- **Floating Point (Bilangan Desimal):**

```
float nilai = 3.14f;      // 6-7 digit precision
double presisi = 3.14159265; // 15-17 digit precision
long double extra = 3.141592653589793238L;
```

- **Character dan String:**

```
char huruf = 'A';        // Satu karakter (gunakan single quote)
string kata = "Hello";    // Kumpulan karakter (gunakan double quote)
string kalimat = "Selamat pagi semua!";
```

- **Boolean:**

```
bool benar = true;           // Nilai true atau false
bool salah = false;          // Untuk kondisi logis
```

- **Contoh program lengkap:**

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    // Deklarasi semua tipe data
    int umur = 20;
    float tinggi = 170.5f;
    double berat = 65.75;
    char grade = 'A';
    string nama = "Ahmad";
    bool lulus = true;

    cout << "Nama: " << nama << endl;
    cout << "Umur: " << umur << " tahun" << endl;
    cout << "Tinggi: " << tinggi << " cm" << endl;
    cout << "Berat: " << berat << " kg" << endl;
    cout << "Grade: " << grade << endl;
    cout << "Lulus: " << lulus << endl; // 1 untuk true, 0 untuk false
```

```
    return 0;  
}
```

- **Operator Aritmatika:**

```
int a = 10, b = 3;  
int hasil;  
  
hasil = a + b; // 13 (Penjumlahan)  
hasil = a - b; // 7 (Pengurangan)  
hasil = a * b; // 30 (Perkalian)  
hasil = a / b; // 3 (Pembagian integer)  
hasil = a % b; // 1 (Modulus/sisa bagi)
```

- **Penting tentang operator / dan % :**

- Pembagian integer menghasilkan integer ($10/3 = 3$, bukan 3.33)
- Untuk hasil desimal, gunakan float/double: `10.0/3 = 3.33333`
- Modulus hanya untuk integer: `10 % 3 = 1` (sisa pembagian)

- **Operator Assignment:**

```
int x = 5;

x = 10;      // Assignment biasa
x += 3;      // x = x + 3 ? x = 13
x -= 2;      // x = x - 2 ? x = 11
x *= 4;      // x = x * 4 ? x = 44
x /= 2;      // x = x / 2 ? x = 22
x %= 5;      // x = x % 5 ? x = 2
```

- **Operator Increment/Decrement:**

```
int counter = 10;

counter++; // Post-increment: counter = 11
++counter; // Pre-increment: counter = 12
counter--; // Post-decrement: counter = 11
--counter; // Pre-decrement: counter = 10

// Untuk pemula, gunakan yang sederhana dulu
counter = counter + 1; // Sama dengan counter++
counter = counter - 1; // Sama dengan counter--
```

Precedence dan Contoh Program Kalkulator

Slide 11

- **Operator Precedence (Urutan Prioritas):**

```
int hasil;  
hasil = 2 + 3 * 4;      // 14, bukan 20 (perkalian dulu)  
hasil = (2 + 3) * 4;    // 20 (tanda kurung mengubah prioritas)  
hasil = 10 / 2 + 3;     // 8  (pembagian dulu, lalu penjumlahan)  
hasil = 10 / (2 + 3);   // 2  (kurung dulu, lalu pembagian)
```

Prioritas	Operator	Keterangan
Tertinggi	()	Tanda kurung
	++ , --	Increment/Decrement
	* , / , %	Perkalian, Pembagian, Modulus
	+ , -	Penjumlahan, Pengurangan

Prioritas	Operator	Keterangan
Terendah	= , += , -= , dll	Assignment

- **Contoh Program: Konversi Mata Uang**

```
#include <iostream>
using namespace std;

int main() {
    double rupiah, dollar, euro;
    double kurs_dollar = 15000.0; // 1 USD = 15,000 IDR
    double kurs_euro = 17000.0; // 1 EUR = 17,000 IDR

    cout << "Masukkan jumlah Rupiah: ";
    cin >> rupiah;

    // Konversi menggunakan operator aritmatika
    dollar = rupiah / kurs_dollar;
    euro = rupiah / kurs_euro;

    cout << "\n==== HASIL KONVERSI ===" << endl;
    cout << "Rupiah: Rp " << rupiah << endl;
    cout << "Dollar: $" << dollar << endl;
    cout << "Euro: €" << euro << endl;
```

```
// Bonus: Hitung total dalam dollar jika ada fee 2%
double total_dollar = dollar * 0.98; // Dikurangi fee 2%
cout << "Setelah fee 2%: $" << total_dollar << endl;

return 0;
}
```

- **Naming Conventions (Penamaan yang Baik):**

```
// ? Good naming
int umurMahasiswa;           // camelCase
string nama_lengkap;         // snake_case
const double PI = 3.14159;    // CONSTANTS dalam UPPERCASE

// ? Bad naming
int a, b, c;                 // Tidak deskriptif
string n;                     // Terlalu singkat
int UmUrMaHaSiSwA;           // Mixed case sembarangan
```

- **Komentar yang Efektif:**

```
// Single line comment untuk penjelasan singkat
int umur = 20; // Umur mahasiswa dalam tahun

/*
Multi-line comment untuk penjelasan panjang
Program ini menghitung gaji karyawan
```

```
berdasarkan gaji pokok dan tunjangan
*/
// ? Good comment - Menjelaskan MENGAPA
double pajak = gaji * 0.1; // Pajak 10% sesuai aturan pemerintah

// ? Bad comment - Menjelaskan apa yang sudah jelas
int umur = 20; // Membuat variabel umur dengan nilai 20
```

- **Indentasi dan Formatting:**

```
// ? Good formatting
#include <iostream>
using namespace std;

int main() {
    int gaji_pokok;
    double tunjangan, total_gaji;

    cout << "Masukkan gaji pokok: ";
    cin >> gaji_pokok;

    tunjangan = gaji_pokok * 0.2;
    total_gaji = gaji_pokok + tunjangan;

    cout << "Total gaji: " << total_gaji << endl;
    return 0;
```

```
}
```

- **Tips Menghindari Error Umum:**

- Selalu inisialisasi variabel sebelum digunakan
- Jangan lupa semicolon (;) di akhir statement
- Pastikan tipe data sesuai dengan nilai yang disimpan
- Gunakan `#include <string>` jika menggunakan tipe string

- **Latihan: Konversi Suhu**

```
#include <iostream>
using namespace std;

int main() {
    double celsius, fahrenheit, kelvin;

    cout << "==== KONVERSI SUHU ===" << endl;
    cout << "Masukkan suhu dalam Celsius: ";
    cin >> celsius;

    // Rumus konversi
    fahrenheit = (celsius * 9.0 / 5.0) + 32;
    kelvin = celsius + 273.15;

    cout << "\n==== HASIL KONVERSI ===" << endl;
    cout << "Celsius: " << celsius << "°C" << endl;
    cout << "Fahrenheit: " << fahrenheit << "°F" << endl;
    cout << "Kelvin: " << kelvin << " K" << endl;

    return 0;
}
```

Soal Latihan Dasar Level 1 - Input Output

Slide 14

- **Soal 1: Program Perkenalan**

- Buat program yang meminta user memasukkan nama, umur, dan asal kota
- Tampilkan kembali informasi tersebut dengan format yang rapi
- Gunakan tipe data yang sesuai untuk setiap informasi

Contoh Output:

```
==== PROGRAM PERKENALAN ====
Masukkan nama: Budi
Masukkan umur: 19
Masukkan asal kota: Bandung

==== INFORMASI ANDA ====
Nama: Budi
Umur: 19 tahun
Asal: Bandung
```

- **Soal 2: Kalkulator Luas Persegi Panjang**

- Input: panjang dan lebar (boleh desimal)
- Output: luas dan keliling
- Rumus: Luas = panjang × lebar, Keliling = 2 × (panjang + lebar)

Contoh Output:

```
==== KALKULATOR PERSEGI PANJANG ====
Masukkan panjang: 10.5
Masukkan lebar: 7.2

==== HASIL PERHITUNGAN ====
Panjang: 10.5 cm
Lebar: 7.2 cm
Luas: 75.6 cm2
Keliling: 35.4 cm
```

- **Tips Penggerjaan:**

- Perhatikan tipe data yang digunakan (int, double, string)

- Jangan lupa include yang diperlukan
- Buat output yang user-friendly dengan label yang jelas
- Test program dengan berbagai input

- **Soal 3: Program Penghitung Nilai Rata-rata**

- Input: 3 buah nilai ujian (float/double)
- Output: Nilai total, rata-rata, dan keterangan
- Hitung total dan rata-rata dari 3 nilai tersebut

Contoh Output:

```
==== PENGHITUNG RATA-RATA ====
Masukkan nilai ujian 1: 85.5
Masukkan nilai ujian 2: 78.0
Masukkan nilai ujian 3: 92.5

==== HASIL PERHITUNGAN ====
Nilai 1: 85.5
Nilai 2: 78.0
Nilai 3: 92.5
Total: 256.0
Rata-rata: 85.33
```

- **Soal 4: Konverter Detik ke Jam-Menit-Detik**

- Input: total detik (integer)
- Output: konversi ke jam, menit, dan sisa detik
- Gunakan operator modulus (%) untuk menghitung sisa

Hint Rumus:

```
jam = total_detik / 3600  
menit = (total_detik % 3600) / 60  
detik = total_detik % 60
```

Contoh Output:

```
==== KONVERTER WAKTU ===  
Masukkan total detik: 7265  
  
==== HASIL KONVERSI ===  
7265 detik = 2 jam, 1 menit, 5 detik
```

- **Soal 5: Kalkulator Diskon**

- Input: harga barang dan persentase diskon
- Output: harga sebelum diskon, besar diskon, dan harga setelah diskon

Contoh Output:

```
==== KALKULATOR DISKON ====
Masukkan harga barang: 150000
Masukkan persentase diskon: 20

==== DETAIL PEMBELIAN ====
Harga asli: Rp 150000
Diskon (20%): Rp 30000
Harga setelah diskon: Rp 120000
Penghematan: Rp 30000
```

- **Soal 6: Kalkulator Lingkaran**

- Input: jari-jari lingkaran
- Output: diameter, keliling, dan luas lingkaran
- Gunakan konstanta PI = 3.14159

Rumus:

```
diameter = 2 * jari_jari
keliling = 2 * PI * jari_jari
luas = PI * jari_jari * jari_jari
```

Contoh Output:

```
==== KALKULATOR LINGKARAN ====
Masukkan jari-jari: 7
```

```
==== HASIL PERHITUNGAN ===
```

```
Jari-jari: 7 cm
```

```
Diameter: 14 cm
```

```
Keliling: 43.98 cm
```

```
Luas: 153.94 cm2
```

- **Soal 7: Konversi Suhu Lengkap**

- Input: suhu dalam Celsius
- Output: konversi ke Fahrenheit, Kelvin, dan Reamur
- Tampilkan semua hasil konversi dalam satu program

Rumus:

```
fahrenheit = (celsius * 9/5) + 32  
kelvin = celsius + 273.15  
reamur = celsius * 4/5
```

Contoh Output:

```
==== KONVERSI SUHU LENGKAP ====
Masukkan suhu dalam Celsius: 25
```

```
==== HASIL KONVERSI ====
Celsius: 25°C
Fahrenheit: 77°F
Kelvin: 298.15 K
Reamur: 20°R
```

- **Soal 8: Kalkulator BMI (Body Mass Index)**

- Input: berat badan (kg) dan tinggi badan (cm)
- Output: BMI dan keterangan kategori
- Rumus: $BMI = \text{berat} / (\text{tinggi_meter})^2$

Contoh Output:

```
==== KALKULATOR BMI ====
Masukkan berat badan (kg): 70
Masukkan tinggi badan (cm): 175

==== HASIL PERHITUNGAN ====
Berat: 70 kg
```

Tinggi: 175 cm (1.75 m)

BMI: 22.86

Kategori: Normal (tanpa kondisional, hanya tampilkan angka BMI)

- **Soal 9: Program Kasir Sederhana**

- Input: nama barang, harga satuan, dan jumlah barang
- Output: struk pembelian dengan total harga
- Hitung subtotal, pajak (10%), dan total bayar

Contoh Output:

```
==== KASIR SEDERHANA ====
Masukkan nama barang: Pulpen
Masukkan harga satuan: 2500
Masukkan jumlah: 3

==== STRUK PEMBELIAN ====
Barang: Pulpen
Harga satuan: Rp 2500
Jumlah: 3 pcs
Subtotal: Rp 7500
Pajak (10%): Rp 750
```

TOTAL BAYAR: Rp 8250

- **Soal 10: Kalkulator Gaji Karyawan**

- Input: nama karyawan, gaji pokok, jam lembur, tarif lembur per jam
- Output: slip gaji dengan rincian lengkap
- Hitung tunjangan (20% dari gaji pokok), gaji lembur, dan total gaji

Contoh Output:

```
==== SLIP GAJI ====
Nama: Ahmad Budi
Gaji pokok: Rp 5000000
Tunjangan (20%): Rp 1000000
Jam lembur: 10 jam
Tarif lembur: Rp 50000/jam
Gaji lembur: Rp 500000
TOTAL GAJI: Rp 6500000
```

- **Soal 11: Program Parkir**

- Input: jenis kendaraan, jam masuk, jam keluar
- Output: durasi parkir dan tarif yang harus dibayar
- Motor: Rp 2000/jam, Mobil: Rp 5000/jam

Contoh Output:

```
==== SISTEM PARKIR ====
Jenis kendaraan (M/C): M
Jam masuk: 8
Jam keluar: 12

==== TIKET PARKIR ====
Kendaraan: Motor
Jam masuk: 08:00
Jam keluar: 12:00
Durasi: 4 jam
Tarif: Rp 2000/jam
TOTAL BAYAR: Rp 8000
```

• Tips untuk semua soal:

- Gunakan tipe data yang tepat untuk setiap variable

- Buat tampilan output yang rapi dan informatif
- Perhatikan perhitungan matematika dan operator yang digunakan
- Test dengan berbagai input untuk memastikan program benar