



Program Studi
Sistem Komputer

Modul Pembelajaran

PEMROGRAMAN BERORIENTASI OBJEK

Kode Matakuliah: SKO25211



Bayu Nugroho, S.Kom., M.Eng
NIDN 0218037701

PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
2026

HALAMAN JUDUL

Nama Matakuliah : Pemrograman Berorientasi Objek
Kode Matakuliah : SKO25211
SKS : 2
Program Studi : Sistem Komputer
Tempat Kegiatan : Laboratorium Pemrograman Gedung G Lantai 3
Kelas : 3SK1
Tahun : 2026

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga Modul Pembelajaran Mata Kuliah Pemrograman Berorientasi Objek (Object-Oriented Programming/OOP) yang memuat materi teori dan praktikum ini dapat disusun dengan baik.

Modul pembelajaran ini disusun sebagai salah satu bahan ajar untuk mendukung proses pembelajaran pada mata kuliah Pemrograman Berorientasi Objek. Penyusunan modul bertujuan untuk memberikan panduan pembelajaran yang sistematis, terstruktur, dan mudah dipahami oleh mahasiswa dalam mempelajari konsep-konsep dasar hingga penerapan pemrograman berorientasi objek.

Materi dalam modul ini mencakup pembahasan teori mengenai konsep dasar pemrograman berorientasi objek, class dan object, atribut dan method, constructor, encapsulation, inheritance, polymorphism, abstraction, serta konsep-konsep lain yang relevan dengan pengembangan perangkat lunak berbasis objek. Selain materi teori, modul ini juga dilengkapi dengan kegiatan praktikum yang disusun secara bertahap, mulai dari pengenalan lingkungan pemrograman, penulisan kode program, implementasi konsep OOP, hingga pengembangan aplikasi sederhana. Setiap kegiatan praktikum dilengkapi dengan langkah-langkah pengerjaan, contoh program, serta latihan yang diharapkan dapat membantu mahasiswa memahami dan menerapkan materi yang telah dipelajari.

Melalui penggunaan modul ini, mahasiswa diharapkan tidak hanya mampu memahami teori dan prinsip pemrograman berorientasi objek, tetapi juga memiliki keterampilan dalam merancang, menulis, menguji, dan mengembangkan program secara terstruktur. Modul ini juga diharapkan dapat mendorong mahasiswa untuk berpikir logis, sistematis, kreatif, serta mampu menyelesaikan permasalahan melalui penerapan konsep pemrograman dalam berbagai studi kasus.

Penulis menyadari bahwa modul pembelajaran ini masih memiliki keterbatasan dan memerlukan penyempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan dan pengembangan modul pada masa mendatang.

Akhir kata, semoga modul pembelajaran ini dapat memberikan manfaat bagi dosen, mahasiswa, dan semua pihak yang terlibat dalam proses pembelajaran, khususnya dalam meningkatkan pemahaman dan keterampilan di bidang pemrograman berorientasi objek.

Bandar Lampung, September 2026

Penyusun

DAFTAR ISI

COVER	i
HALAMAN JUDUL	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
Modul 1	1
1. Capaian Pembelajaran Matakuliah (CPMK)	1
2. Dasar Teori	1
3. Alat dan Bahan	2
4. Langkah Kerja / Prosedur Praktikum.....	3
5. Lembar Kerja	7
Modul 2	9
1. Capaian Pembelajaran Matakuliah (CPMK)	9
2. Dasar Teori	9
3. Alat dan Bahan	10
4. Langkah Kerja / Prosedur Praktikum.....	10
5. Lembar Kerja	12
Modul 3	16
1. Capaian Pembelajaran Matakuliah (CPMK)	16
2. Dasar Teori	16
3. Alat dan Bahan	17
4. Langkah Kerja / Prosedur Praktikum.....	17
5. Lembar Kerja	18
Modul 4	20
1. Capaian Pembelajaran Matakuliah (CPMK)	20
2. Dasar Teori	20
3. Alat dan Bahan	21
4. Langkah Kerja / Prosedur Praktikum.....	21
5. Lembar Kerja	23

Modul 5	25
1. Capaian Pembelajaran Matakuliah (CPMK)	25
2. Dasar Teori	25
3. Alat dan Bahan	26
4. Langkah Kerja / Prosedur Praktikum.....	26
5. Lembar Kerja	28
Modul 6	30
1. Capaian Pembelajaran Matakuliah (CPMK)	30
2. Dasar Teori	30
3. Alat dan Bahan	31
4. Langkah Kerja / Prosedur Praktikum.....	31
5. Lembar Kerja	34
Modul 7	36
1. Capaian Pembelajaran Matakuliah (CPMK)	36
2. Dasar Teori	36
3. Alat dan Bahan	37
4. Langkah Kerja / Prosedur Praktikum.....	37
5. Lembar Kerja	39
Modul 8 (UTS).....	41
1. Capaian Pembelajaran Matakuliah (CPMK)	41

Modul 1

1. Capaian Pembelajaran Matakuliah (CPMK)

CPMK142:

Mampu menerapkan teknik pemrograman terstruktur, modular, dan berorientasi objek untuk menyelesaikan permasalahan komputasi sesuai kebutuhan aplikasi sistem komputer.

Indikator Ketercapaian:

Mahasiswa memahami konsep dasar Pemrograman Berorientasi Objek.

2. Dasar Teori

Pengantar

Pemrograman Berorientasi Objek

1. Konsep Dasar OOP

Pemrograman Berorientasi Objek adalah paradigma pemrograman yang berorientasi pada objek. Pemrograman ini mengorganisasi desain perangkat lunak di sekitar data atau objek, bukan fungsi dan logika.

- **Paradigma Tradisional (Prosedural):** Berfokus pada urutan langkah atau instruksi (apa yang harus dilakukan).
- **Paradigma OOP:** Berfokus pada komponen atau objek yang saling berinteraksi (siapa yang melakukan dan data apa yang dimiliki).

2. Empat Pilar Utama OOP

a) Encapsulation (Pengapsulan)

Proses membungkus data (atribut) dan metode (fungsi) menjadi satu kesatuan (kelas), serta menyembunyikan detail implementasi internal dari luar.

- **Tujuan:** Keamanan data dan mencegah akses langsung yang tidak sah.
- **Konsep Penting:** Access Modifier (private, protected, public).

b) Abstraction (Abstraksi)

Penyederhanaan objek kompleks dengan hanya menampilkan fitur-fitur penting dan menyembunyikan detail kerumitannya.

- **Tujuan:** Mengurangi kompleksitas sistem dan mempermudah fokus pada fungsionalitas utama.
- **Konsep Penting:** Abstract Class dan Interface.

c) Inheritance (Pewarisan)

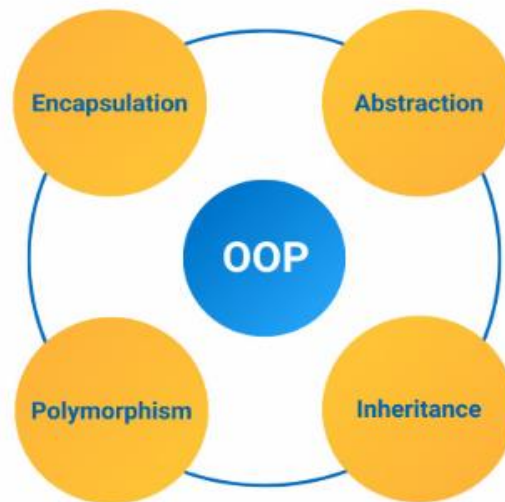
Mekanisme yang memungkinkan suatu kelas (*child class/subclass*) mewarisi atribut dan metode dari kelas lain (*parent class/superclass*).

- **Tujuan:** Reusabilitas kode (*code reusability*) dan efisiensi pengembangan.
- **Konsep Penting:** Superclass, Subclass, Keyword extends / implements.

d) Polymorphism (Polimorfisme)

Kemampuan suatu objek atau metode untuk memiliki banyak bentuk atau perilaku yang berbeda tergantung konteksnya.

- **Compile-time Polymorphism:** *Method Overloading* (nama metode sama, parameter beda).
- **Runtime Polymorphism:** *Method Overriding* (menimpa fungsi kelas induk pada kelas anak).



Gambar 1. Empat Pilar Utama dalam PBO

3. Komponen Utama OOP

- **Class (Kelas):** Cetak biru (*blueprint*) atau *template* untuk membuat objek. Kelas mendefinisikan atribut dan perilaku.
- **Object (Objek):** Instansi nyata (*instance*) yang dibuat berdasarkan kelas. Objek memiliki *state* (nilai atribut) dan *behavior* (metode).
- **Attribute / Property:** Variabel yang menyimpan data atau karakteristik objek.
- **Method:** Fungsi di dalam kelas yang mendefinisikan tindakan atau perilaku objek.
- **Constructor:** Metode khusus yang dipanggil pertama kali saat sebuah objek diinstansiasi untuk menginisialisasi nilai awal

3. Alat dan Bahan

Hardware :

1. Laptop/PC
2. LCD/proyektor
3. Jaringan internet

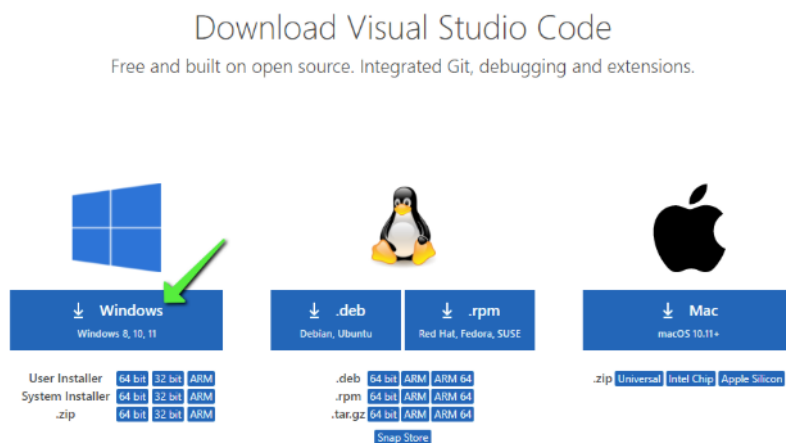
Software :

1. Sistem operasi Windows/Linux/macOS
2. Java Development Kit (JDK)
3. IDE pemrograman: NetBeans / IntelliJ IDEA / Visual Studio Code
4. LMS untuk materi dan aktivitas pembelajaran

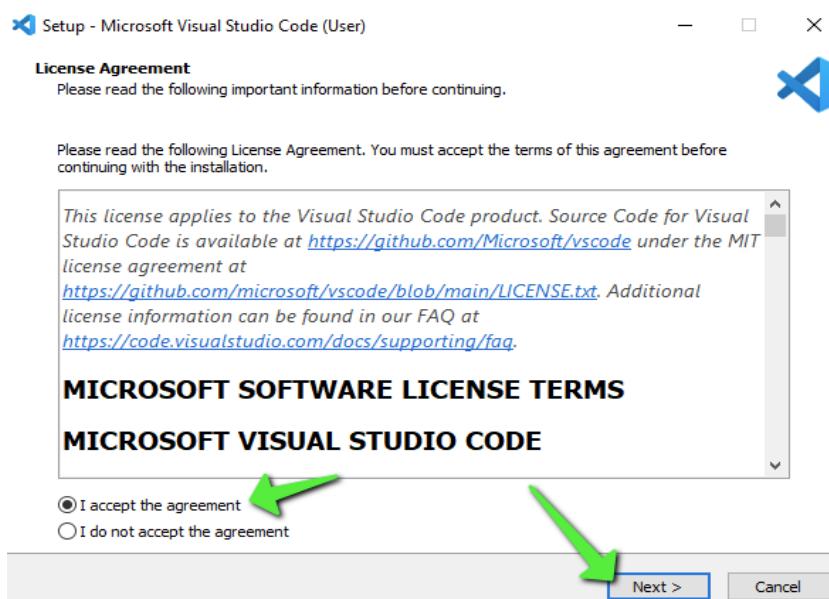
4. Langkah Kerja / Prosedur Praktikum

Instalasi Visual Studio Code

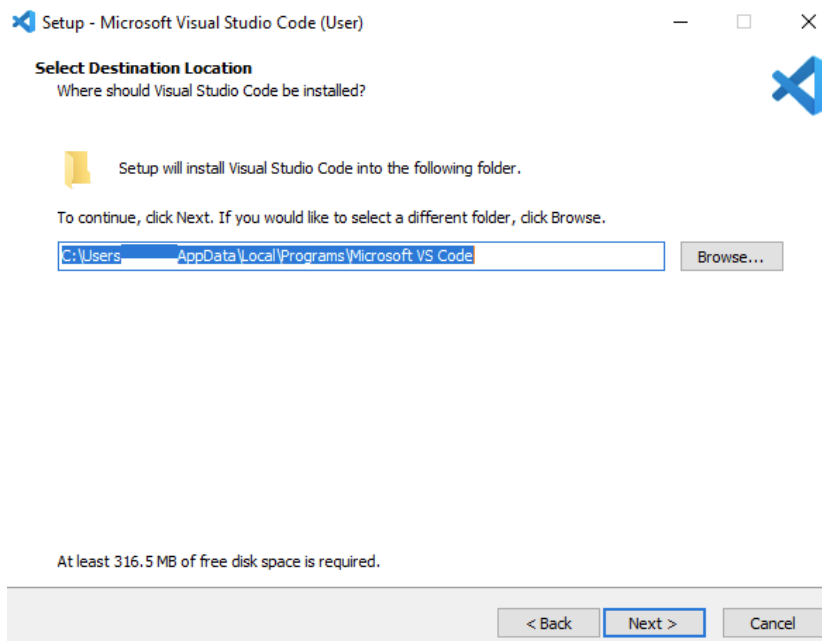
1. Kunjungi tautan berikut: <https://code.visualstudio.com/download>
Klik tombol unduh berwarna biru yang bertuliskan Windows di bawah Logo Windows:



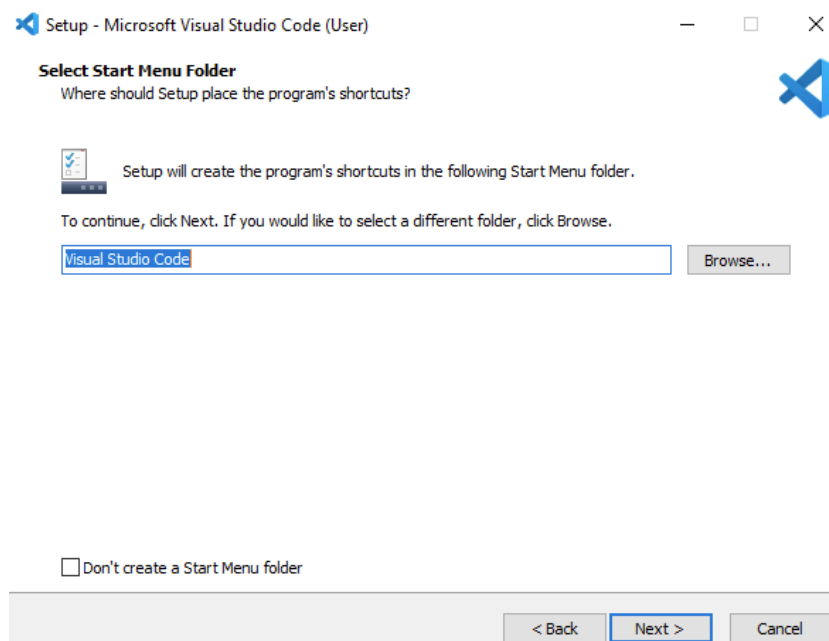
2. Pada jendela File Explorer, temukan dan klik Unduhan. Klik dua kali pada file 'VSCodeUserSetup-x64-1.xxexe' yang telah diunduh
3. Ikuti langkah-langkah instalasi.
Pertama, cek list Perjanjian Lisensi, lalu klik Next >.



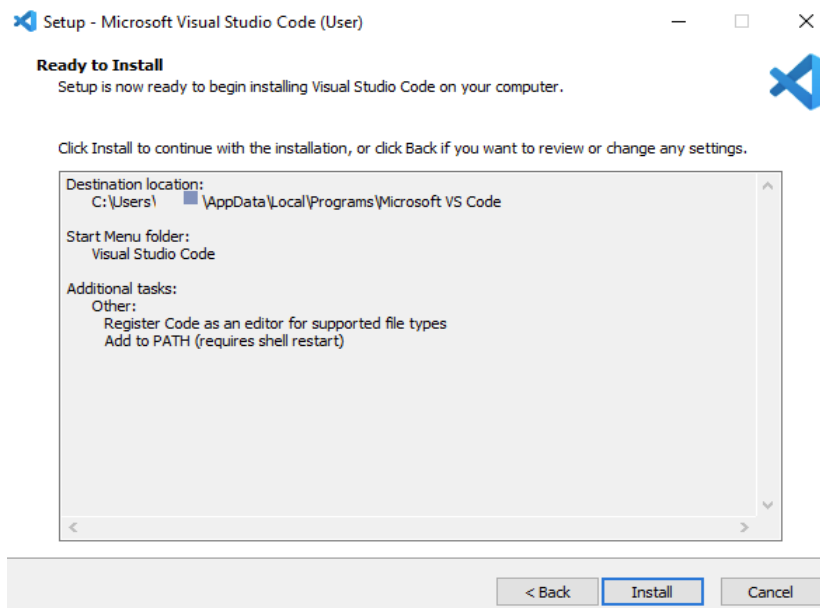
4. Pilih lokasi default untuk instalasi dengan mengklik Next >.



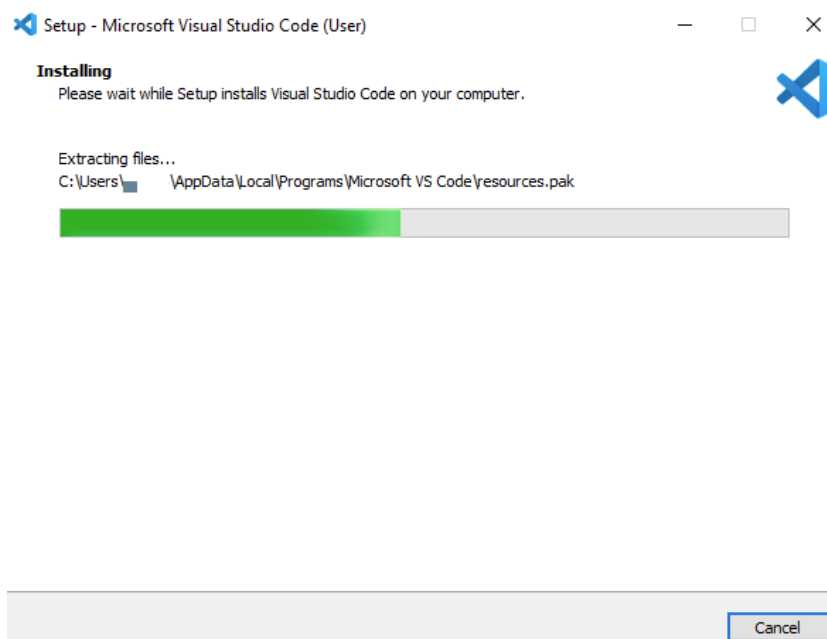
5. Terima nama folder Menu Mulai default dengan mengklik Next >.



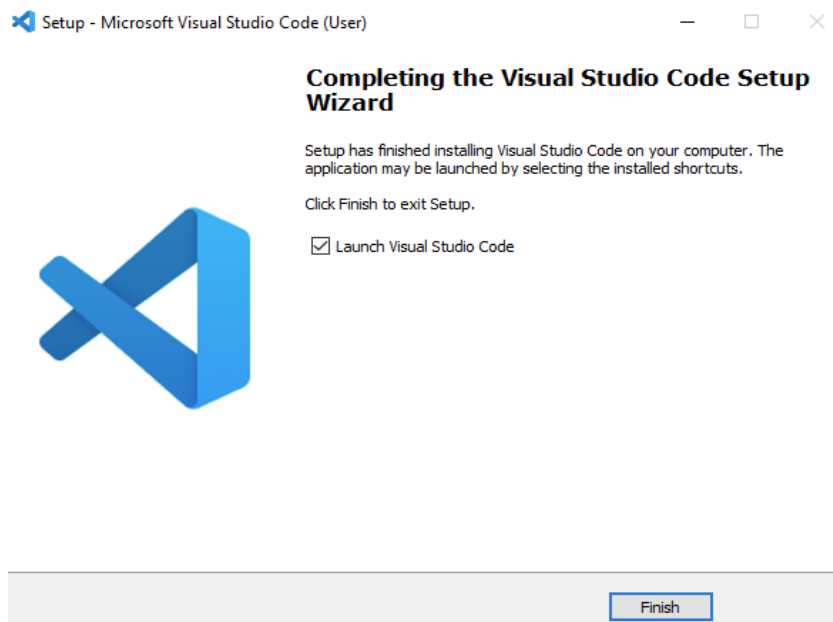
6. Konfirmasikan opsi instalasi, lalu klik Install.



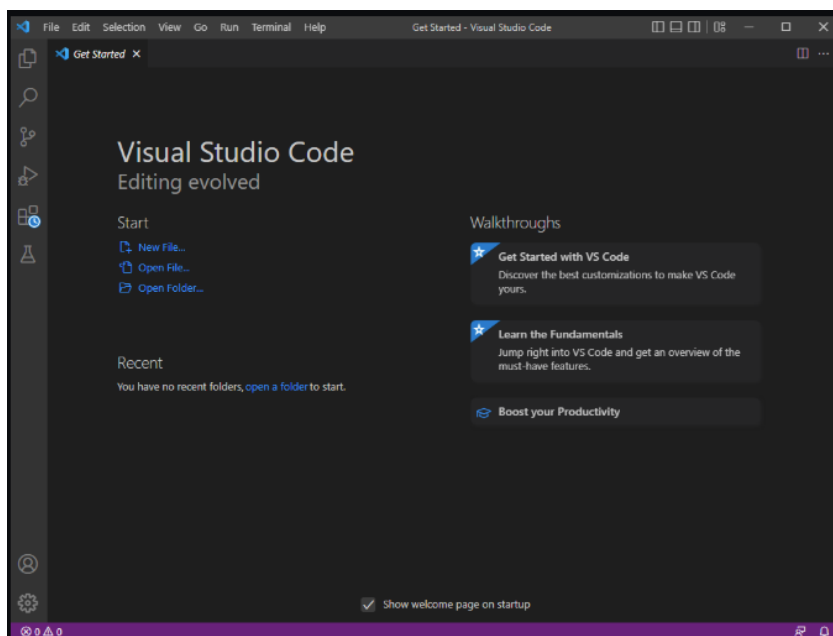
7. Memulai proses instalasi.



8. Klik Finish untuk keluar dari instalasi dan (secara default) meluncurkan Visual Studio Code.



9. Hasil Instalasi VSCODE.



5. Lembar Kerja

Latihan 1

Lakukan Instalasi pada PC / laptop sesuai langkah kerja.

Lakukan Pengujian Program OOP C++ Pertama:

- Buat folder baru di komputer Anda, lalu buka folder tersebut melalui VS Code (File > Open Folder)
- Buat berkas baru bernama main.cpp
- Tuliskan kode dasar berikut:

```
#include <iostream>

class Mobil {
public:
    std::string merk;

    void klakson() {
        std::cout << "Beep beep! Mobil " << merk << " berbunyi." << std::endl;
    }
};

int main() {
    Mobil mobilKu;
    mobilKu.merk = "Toyota";
    mobilKu.klakson();

    return 0;
}
```

- Jalankan program dengan menekan tombol **Run** atau melalui terminal VS Code Analisis Hasil Keluarannya.

HASIL PERCOBAAN:

Screenshoot Hasil LATIHAN 1 dan Berikan Analisis.
(UPLOAD ke LMS → TUGAS PRAKTIKUM 1)

Tugas Mandiri 1

(Kerjakan Tugas Mandiri 1 Upload ke LMS → **TUGAS TEORI 1**)

1. Apakah pengertian tentang CLASS, OBJECT, METHOD ?
2. Jelaskan perbedaan mendasar antara pemrograman prosedural dan pemrograman berorientasi objek (OOP)
3. Sebutkan minimal 5 atribut dan minimal 3 method yang melekat pada diri anda ?
4. Mengapa metodologi berorientasi objek berkembang lebih pesat dibandingkan dengan metode-metode yang lain dalam bahasa pemrograman ?
5. Mengapa kita memerlukan kompilator C++ terpisah (seperti MinGW/G++) ketika menggunakan Editor Visual Studio Code?



Program Studi
SISTEM KOMPUTER