**LAPORAN IMPLEMENTASI OOP DI C#**



Anggota kelompok:

Andika Romario (F1E121071)

Putri Anggellina Khairunisa (F1E121088)

Dinda Fatimah Sarah (F1E121090)

Cepi Ramdan (F1E121097)

Yosika Dian Saputri (F1E121111)

Dosen Pengampu:

Andre Rabiula, S.Kom., M.Eng.

**FAKULTAS SAINS DAN TEKNOLGI**

**UNIVERSITAS JAMBI**

**TAHUN AJARAN 2023**

# Kata Pengantar

Puji dan Syukur kami panjatkan kehadirat kepada Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya sehingga kami dapat menyelesaikan laporan ini dengan baik dan tepat pada waktunya. Adapun tujuan laporan ini adalah untuk memenuhi tugas mata kuliah “Pemrograman Visual”

Dalam penyusunan laporan ini, kami banyak mengalami kesulitan, terutama disebabkan oleh kurangnya ilmu pengetahuan yang menunjang. Namun, berkan bimbingan dan bantuan, akhirnya laporan ini dapat terselesaikan dengan cukup baik. Oleh karena itu, sudah sepantasnya jika kami mengucapkan terima kasih kepada Andre Rabiula, S.Kom., M.Eng. dan bapak Edi Saputra, S.T., M.Sc. selaku dosen pembimbing pada mata kuliah Pemrograman Visual.

Kami menyadari laporan ini masih banyak kekurangannya. Oleh karena itu, kami sangat mengharapkan adanya kritik dan saran yang bersifat positif, guna penyusunan laporan yang lebih baik lagi kedepannya. Untuk itu kami mengucapkan banyak terima kasih dan semoga laporan ini bermanfaat bagi para pembaca.

# Bab I

## Latar Belakang

OOP adalah singkatan dari Object Oriented Programming, yaitu suatu metode pemrograman yang fokus atau berorientasi pada objek. Tujuan dari dirancangnya OOP adalah membantu para developer dalam mengembangkan model yang sudah ada di kehidupan sehari-hari. Sehingga satu bagian dari suatu permasalahan dalam program tersebut adalah objek. Objek itu juga merupakan komposis dari komponen-komponen objek yang lebih kecil lagi.

OOP pertama kali dikembangkan pada tahun 1960-an sebagai alternatif untuk pemrograman prosedural yang dominan pada saat itu. Bahasa pemrograman Simula, yang dikembangkan oleh Ole-Johan Dahl dan Kristen Nygaard pada tahun 1967, dianggap sebagai bahasa pemrograman OOP pertama. Selanjutnya, pada tahun 1980-an bahasa pemrograman seperti Smalltalk dan C++ menjadi populer dan digunakan secara luas dalam pengembangan perangkat lunak.

Pada tahun 1990-an, Java muncul sebagai bahasa pemrograman OOP yang paling populer dan banyak digunakan dalam pengembangan perangkat lunak. Kelebihan Java adalah kemampuannya untuk berjalan pada berbagai platform, serta memiliki fitur keamanan dan modularitas yang baik. Pentingnya OOP dalam Pengembangan Perangkat Lunak Modern

OOP menjadi penting dalam pengembangan perangkat lunak modern karena paradigma OOP memungkinkan pengembang untuk memodelkan dunia nyata dalam program komputer. Dalam OOP, program dibagi menjadi objek-objek yang saling berinteraksi, sehingga memudahkan pengembangan dan pemeliharaan aplikasi.

## Tujuan

Tujuan dari laporan implementasi OOP adalah untuk memberikan gambaran tentang bagaimana paradigma OOP diterapkan dalam pembuatan program. Laporan ini berisi penjelasan tentang konsep-konsep dasar OOP, seperti class, object, inheritance, encapsulation, dan polymorphism. Selain itu, laporan implementasi OOP juga memberikan contoh penggunaan OOP dalam kode program dengan menggunakan bahasa pemrograman tertentu seperti java, C# atau python

Tujuan dari laporan implementasi OOP adalah untuk memberikan gambaran tentang bagaimana paradigma OOP diterapkan dalam pembuatan program. Laporan ini berisi penjelasan tentang konsep-konsep dasar OOP, seperti class, object, inheritance, encapsulation, dan polymorphism. Selain itu, laporan implementasi OOP juga memberikan contoh penggunaan OOP dalam kode program dengan menggunakan bahasa pemrograman tertentu seperti java, C# atau python

# Bab II

## Pengenalan OOP

OOP (Object Oriented Programming) adalah paradigma pemrograman yang didasarkan pada konsep objek. OOP memungkinkan pengembang untuk memecah program menjadi objek-objek yang saling berinteraksi. Konsep objek memungkinkan pengembang untuk memodelkan dunia nyata dalam program komputer. Dalam OOP, suatu objek memiliki atribut (data) dan metode (fungsi) yang mengendalikan perilaku dari objek tersebut. Atribut dapat disebut juga sebagai variabel anggota, sedangkan metode dapat disebut juga sebagai fungsi anggota.

Ada empat konsep dasar dalam OOP, yaitu :

1. Encapsulation 🡺 Encapsulation adalah konsep untuk menyembunyikan detail implementasi objek dan hanya mengekspos interface publiknya. Dengan cara ini, data dalam objek terlindungi dari akses yang tidak sah dan hanya dapat diakses melalui interface yang diberikan. Dalam OOP, setiap objek memiliki beberapa akses modifier seperti public, private, dan protected. Akses modifier ini menentukan apakah suatu atribut atau metode dapat diakses dari luar objek atau tidak.
2. Inheritance 🡺 Inheritance adalah konsep dimana sebuah objek baru dapat diwarisi sifat dan fitur dari objek lama yang menjadi induknya. Dalam OOP, objek yang menjadi induk disebut sebagai superclass atau parent class, sedangkan objek baru yang diwarisi sifat dan fiturnya disebut sebagai subclass atau child class. Konsep inheritance memungkinkan pengembang untuk membuat objek baru dengan sifat dan fitur yang mirip dengan objek lama, tetapi dengan tambahan atau perubahan tertentu.
3. Polymorphism 🡺 Polymorphism adalah konsep dimana sebuah objek dapat memiliki banyak bentuk yang berbeda. Dalam OOP, polymorphism dapat dicapai melalui inheritance atau melalui konsep interface. Dalam inheritance, sebuah subclass dapat memiliki metode dengan nama yang sama dengan metode dari superclass. Dalam konsep interface, sebuah kelas dapat mengimplementasikan banyak interface sehingga memiliki banyak kemampuan.
4. Abstraction 🡺 Abstraction adalah konsep dimana pengembang mengisolasi bagian-bagian tertentu dari suatu sistem sehingga hanya bagian-bagian tertentu yang terlihat oleh pengguna. Konsep ini berguna dalam pengembangan aplikasi yang kompleks karena dapat memudahkan pemeliharaan aplikasi dan meningkatkan keamanan dari aplikasi tersebut. Dalam OOP, abstraction dapat dicapai melalui konsep interface atau abstract class. Interface dan abstract class adalah template untuk membuat objek, tetapi tidak memiliki implementasi yang sebenarnya.

OOP memiliki beberapa kelebihan yakni :

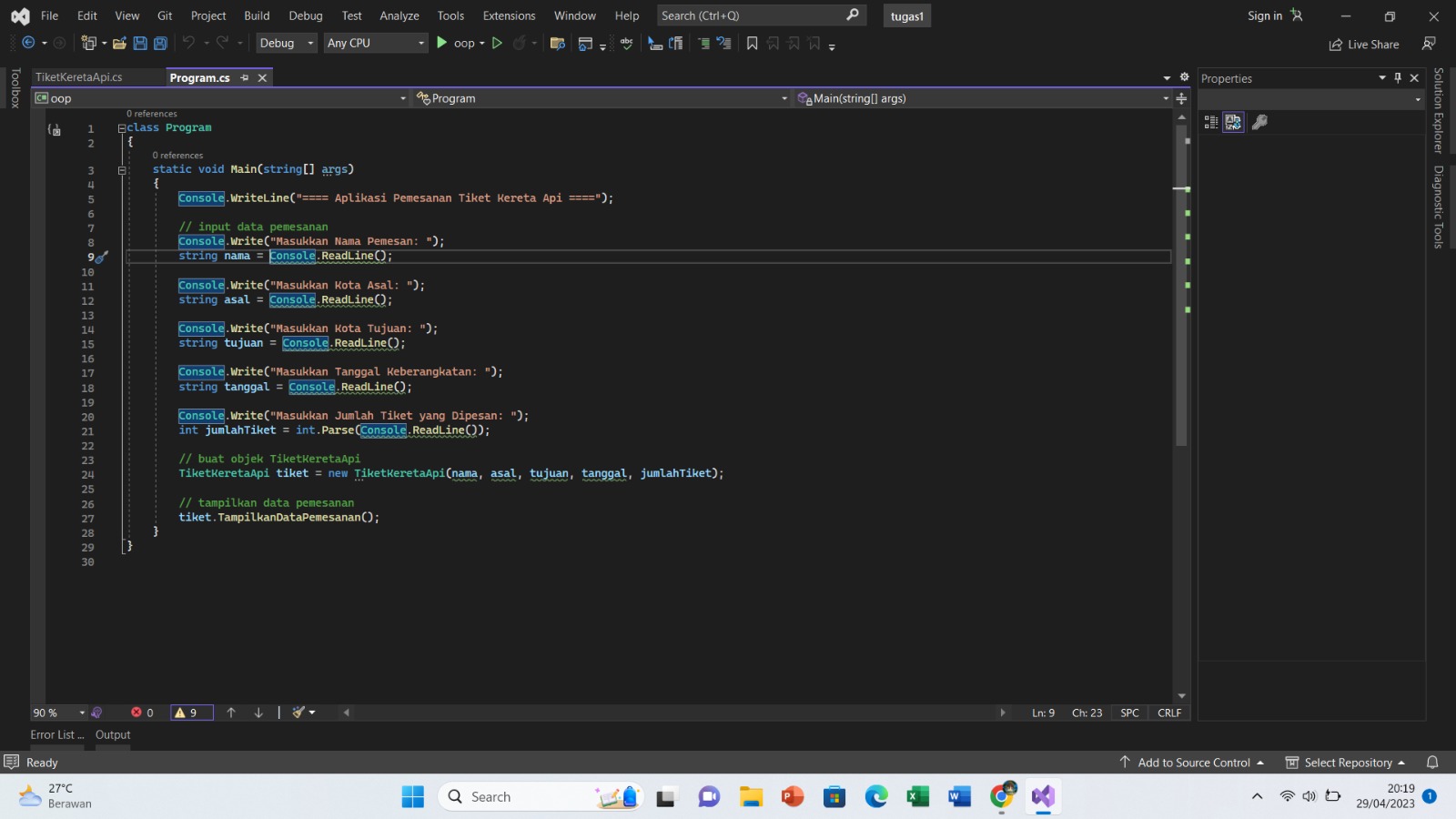
1. Modularitas : OOP memungkinkan pengembangan program dalam bentuk modul-modul yang terpisah. Hal ini memudahkan pengembang untuk memelihara dan mengembangkan program secara terpisah tanpa menggaggu modul lainnya.
2. Reusabilitas : Dalam OOP, objek dapat diwarisi sifat dan fiturnya dari objek lain. Hal ini memungkinkan pengembang untuk membuat objek baru dengan sifat dan fitur yang mirip dengan objek lama, tetapi dengan tambahan atau perubahan tertentu. Dengan demikian, pengembang dapat menggunakan kembali objek yang sudah ada dalam pengembangan program baru.
3. Keterbacaan kode: OOP memungkinkan kode program ditulis dalam bentuk yang lebih mudah dipahami dan keterbacaannya lebih baik. Hal ini karena kode program OOP dibagi menjadi objek-objek yang saling berinteraksi, sehingga memudahkan pengembang untuk memahami alur program.
4. Fleksibilitas: OOP memungkinkan pengembang untuk menyesuaikan program dengan kebutuhan bisnis yang berubah-ubah. Hal ini karena OOP memungkinkan pengembang untuk menambah atau mengubah objek dan metode dalam program tanpa mengganggu objek dan metode lainnya.
5. Keamanan: Dalam OOP, akses terhadap atribut dan metode objek dapat dikontrol dengan menggunakan akses modifier seperti public, private, dan protected. Dengan demikian, data dalam objek terlindungi dari akses yang tidak sah dan hanya dapat diakses melalui interface yang diberikan.

OOP juga memiliki beberapa kekurangan, yakni :

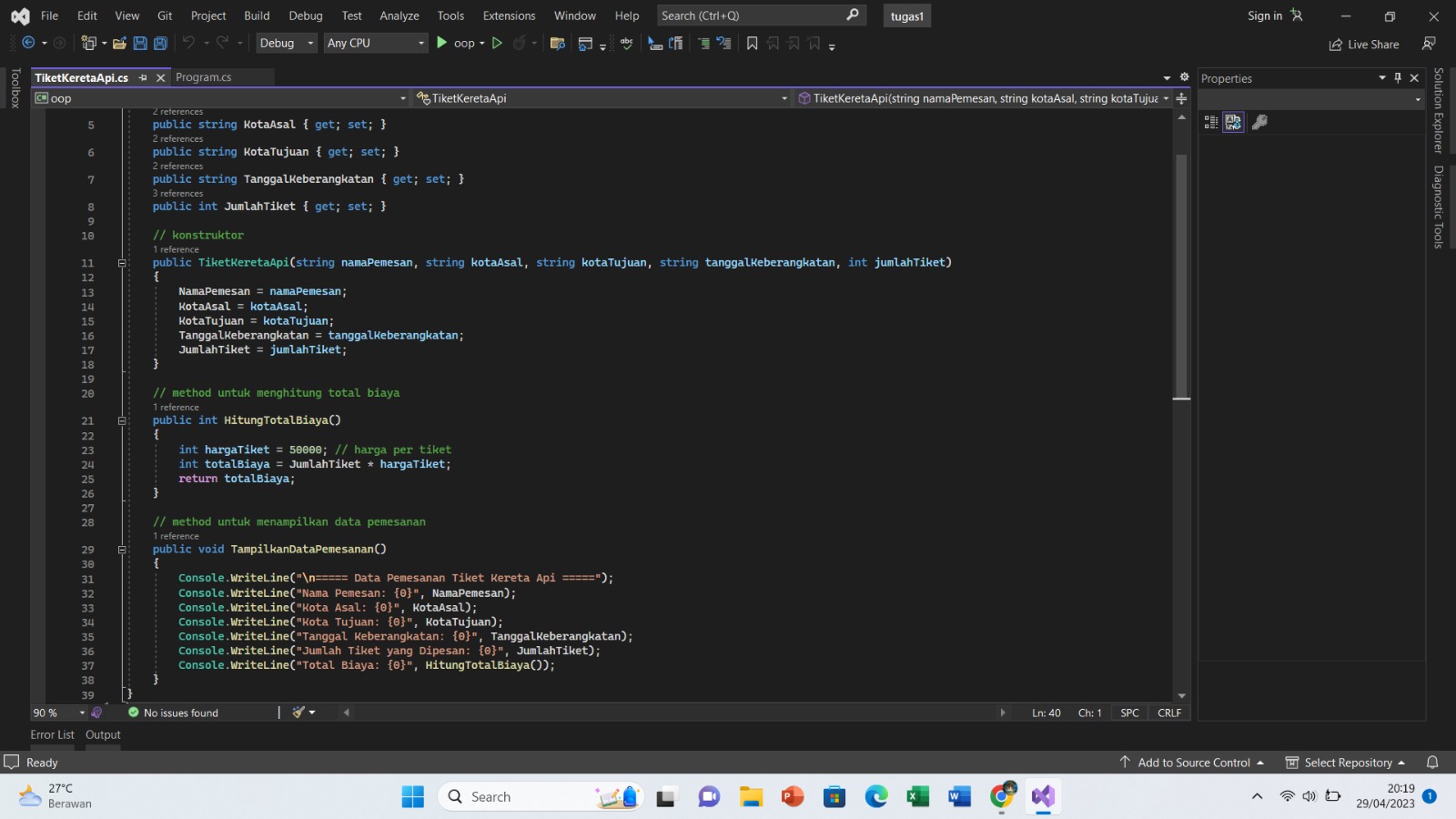
1. Kompleksitas: OOP memerlukan kemampuan pemrograman yang lebih tinggi karena penggunaannya yang kompleks. Selain itu, OOP juga memerlukan waktu yang lebih lama untuk dipelajari dan dipahami oleh pengembang pemula.
2. Kinerja: OOP membutuhkan memori dan prosesor yang lebih besar daripada pemrograman prosedural. Hal ini karena OOP menggunakan banyak objek yang harus dikelola oleh sistem.
3. Kesalahan desain: Kesalahan desain dalam OOP dapat menyebabkan perangkat lunak yang dibuat menjadi sulit untuk diubah dan dipelihara.
4. Biaya: Implementasi OOP memerlukan biaya yang lebih tinggi karena memerlukan sumber daya komputer yang lebih besar dan kebutuhan pelatihan yang lebih tinggi bagi para pengembang.
5. Keterbatasan pada beberapa aplikasi: OOP mungkin tidak cocok untuk beberapa aplikasi yang bersifat prosedural atau bersifat matematika.

## Implementasi OOP

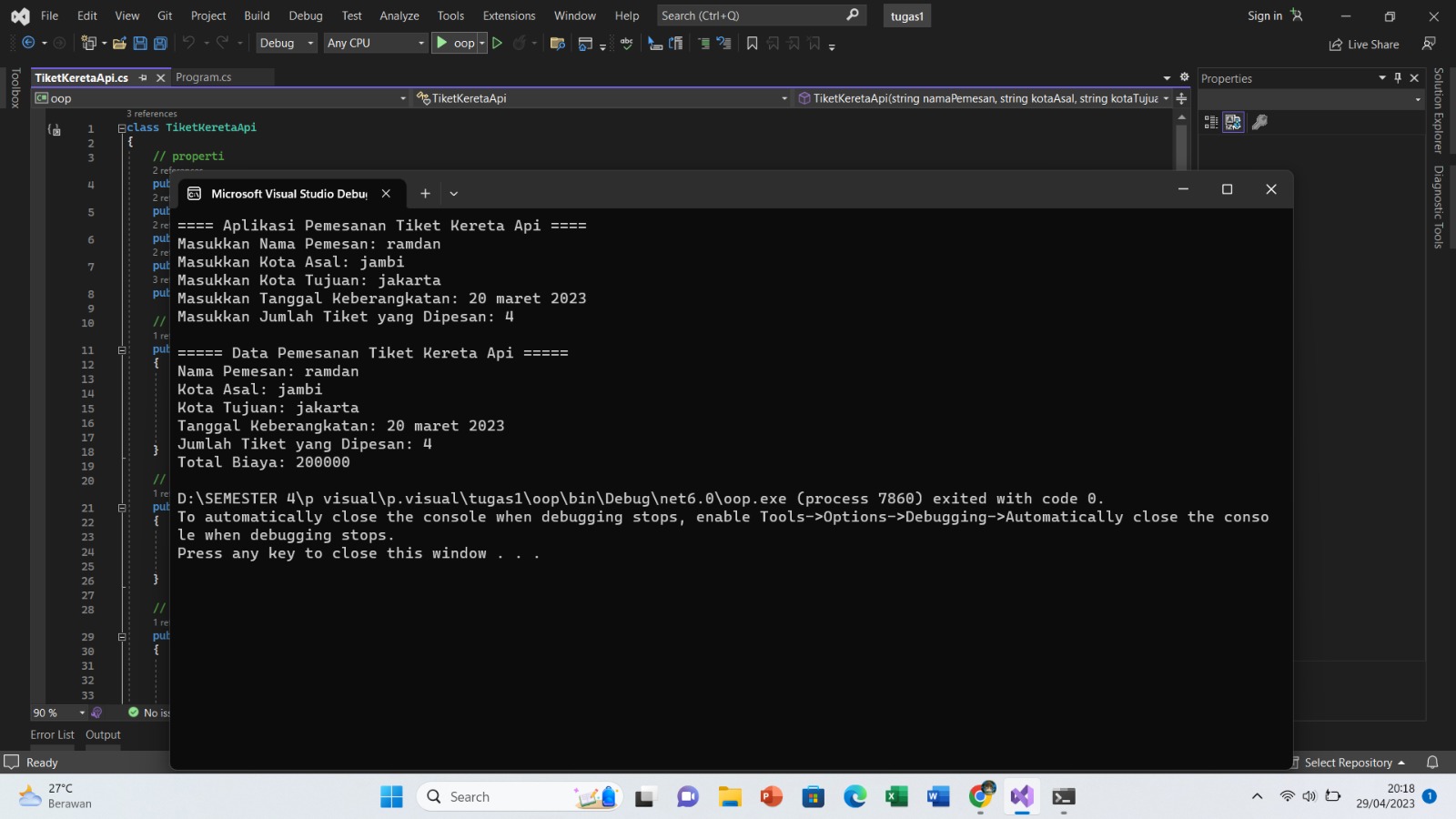
**Class TiketKeretaApi**



**Class Program**



**Hasil ketika program di run**



Dalam kode di atas, telah dibuat sebuah class dengan nama TiketKeretaApi yang memiliki beberapa properti seperti NamaPemesan, KotaAsal, KotaTujuan, TanggalKeberangkatan, dan JumlahTiket. Class TiketKeretaApi juga memiliki konstruktor untuk menginisialisasi nilai properti pada saat objek dibuat. Selain itu, terdapat dua method pada class tersebut, yaitu HitungTotalBiaya() yang berfungsi untuk menghitung total biaya pemesanan tiket dan TampilkanDataPemesanan() yang berfungsi untuk menampilkan data pemesanan.

Selain classTiketKeretaApi, juga dibuat sebuah class dengan nama Program yang berisi method Main sebagai entry point dari program. Pada method Main, user diminta untuk memasukkan data pemesanan tiket kereta api seperti nama pemesan, kota asal, kota tujuan, tanggal keberangkatan, dan jumlah tiket. Setelah itu, objek TiketKeretaApi dibuat dan diinisialisasi dengan nilai yang dimasukkan oleh user. Kemudian, data pemesanan tiket ditampilkan menggunakan method TampilkanDataPemesanan() dari objek TiketKeretaApi yang telah dibuat sebelumnya.

# Kesimpulan

Berdasarkan penjelasan diatas, dapat disimpulkan bahwa OOP (Object Oriented Programming) adalah paradigma pemrograman yang memodelkan program berdasarkan objek-objek yang saling berinteraksi. OOP memiliki kelebihan seperti Modularitasm reusabilitas, keterbacaan kode, fleksibilitas, dan keamanan. Namun, Oop juga memiliki kekurangan seperti kompleksitas, kinerja, kesalahan desain, biaya dan keterbatasan pada beberapa aplikasi.

Meskipun memiliki kekurangan, OOP masih menjadi paradigma pemrograman yang sangat populer dan banyak digunakan dalam pengembangan perangkat lunak modern karena kelebihannya yang banyak dan kecenderungannya untuk mempermudah pemeliharaan dan pengembangan program secara terpisah, serta memungkinkan pengembang untuk menyesuaikan program dengan kebutuhan bisnis yang berubah-ubah. Oleh karena itu, pemahaman tentang OOP sangat penting bagi pengembang perangkat lunak untuk mempercepat proses pengembangan program dan menghasilkan program yang berkualitas tinggi.