

# Mobile Programming

## Pertemuan 05

Alauddin Maulana Hirzan, S.Kom., M.Kom.  
NIDN. 0607069401

Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang



1 Software Development Kit

2 Android SDK & NDK

3 Swift SDK

4 Cross Platform SDK

5 Memilih SDK

# Software Development Kit

## Definisi Software Development Kit

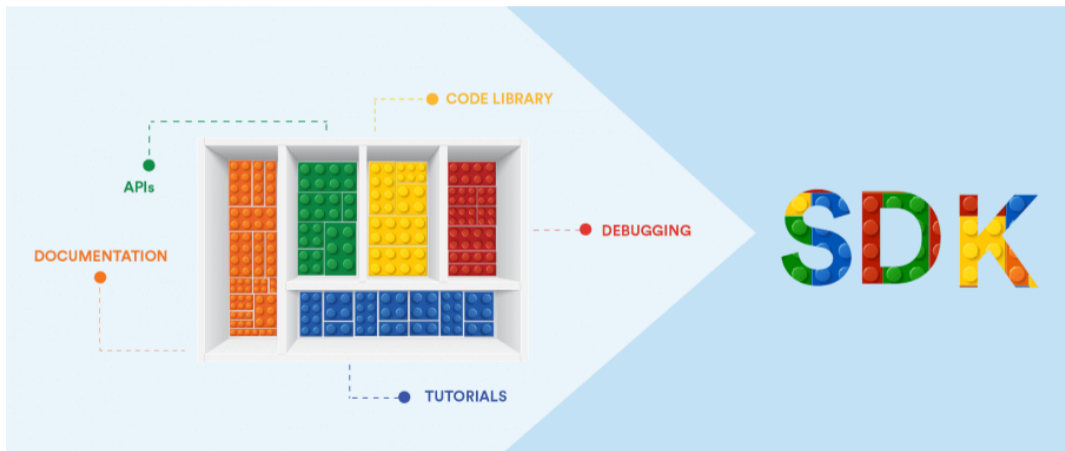
### Definisi:

Software Development Kit (SDK) adalah sekumpulan alat, pustaka, dokumentasi, dan contoh kode yang disediakan oleh pengembang perangkat lunak untuk memfasilitasi pembuatan aplikasi pada platform tertentu atau untuk tujuan tertentu.

Pada dasarnya, SDK berfungsi sebagai paket sumber daya yang dapat digunakan pengembang untuk membangun aplikasi perangkat lunak untuk lingkungan atau kerangka kerja tertentu. Sumber daya ini biasanya mencakup API (Antarmuka Pemrograman Aplikasi), kode sampel, perpustakaan, dan dokumentasi.

# Software Development Kit

## Definisi Software Development Kit



# Software Development Kit

## Pentingnya Software Development Kit

### Manfaat SDK:

- **Mempercepat Pengembangan:** SDK menyediakan komponen dan alat bantu siap pakai yang secara signifikan dapat mempercepat proses pengembangan
- **Meningkatkan Fungsionalitas:** SDK sering kali menawarkan fitur dan kemampuan canggih yang mungkin menantang atau memakan waktu untuk dikembangkan dari awal.
- **Memastikan Kompatibilitas:** SDK dirancang untuk bekerja dengan mulus dalam lingkungan atau platform tertentu.

# Software Development Kit

## Pentingnya Software Development Kit

- **Mempromosikan Praktik Terbaik:** SDK sering kali dilengkapi dengan dokumentasi dan contoh kode yang mengilustrasikan praktik terbaik dan standar pengkodean.
- **Mendukung Ekosistem:** SDK berkontribusi pada ekosistem pengembang yang dinamis dengan memungkinkan pengembang pihak ketiga untuk memperluas dan menyesuaikan platform atau layanan.

# Software Development Kit

## Jenis Software Development Kit

### Jenis-jenis SDK:

1. **SDK Seluler:** SDK Seluler dirancang secara khusus untuk memfasilitasi pengembangan aplikasi untuk perangkat seluler seperti ponsel cerdas dan tablet.
2. **SDK Web:** SDK Web digunakan untuk membangun aplikasi yang berjalan di browser web. SDK ini biasanya menyertakan pustaka JavaScript dan API untuk berinteraksi dengan teknologi web seperti HTML, CSS, dan DOM (Document Object Model)

# Software Development Kit

## Jenis Software Development Kit

3. **SDK Desktop:** SDK Desktop dirancang untuk mengembangkan aplikasi yang berjalan di sistem operasi desktop seperti Windows, macOS, dan Linux.
4. **SDK IoT:** SDK IoT (Internet of Things) digunakan untuk membangun aplikasi untuk perangkat yang terhubung dan ekosistem IoT.



# Software Development Kit

## Jenis Software Development Kit



- 1 Software Development Kit
- 2 Android SDK & NDK**
- 3 Swift SDK
- 4 Cross Platform SDK
- 5 Memilih SDK

# Android SDK & NDK

## Definisi Android SDK

### Definisi Android SDK:

Android SDK (Kit Pengembangan Perangkat Lunak): Android SDK adalah seperangkat alat, pustaka, dan sumber daya yang disediakan oleh Google kepada pengembang untuk membangun aplikasi untuk platform Android.

SDK ini mencakup berbagai API dan alat yang diperlukan untuk mengembangkan, menguji, dan men-debug aplikasi Android. SDK terutama terdiri dari Android Studio IDE (Lingkungan Pengembangan Terpadu), yang didasarkan pada IntelliJ IDEA, bersama dengan Android Debug Bridge (ADB), Android Emulator, alat platform, dan berbagai perpustakaan dan API untuk mengakses fitur dan fungsi perangkat.

# Android SDK & NDK

## Komponen Android SDK

**Android SDK terdiri dari:**

1. **Android Studio:** IDE resmi untuk pengembangan aplikasi Android.
2. **Android Debug Bridge (ADB):** Alat bantu baris perintah yang digunakan untuk berkomunikasi dengan perangkat atau emulator Android yang terhubung.
3. **Emulator Android:** Perangkat virtual yang berjalan di komputer Anda untuk mensimulasikan berbagai perangkat Android.
4. **Alat Platform:** Alat-alat seperti AAPT (Alat Pengemasan Aset Android) dan Fastboot untuk mengelola komponen aplikasi Android.

# Android SDK & NDK

## Komponen Android SDK

5. **Alat Pembuatan:** Alat-alat seperti Gradle untuk membangun, menguji, dan menerapkan aplikasi Android.
6. **Alat SDK Android:** Alat tambahan untuk mengelola paket SDK, SDK Manager, AVD Manager, dll.
7. **Sistem ROM Emulator:** ROM sistem Android yang telah dikonfigurasi sebelumnya untuk pengujian dan debugging pada konfigurasi perangkat yang berbeda

# Android SDK & NDK

## Definisi Android NDK

### Definisi Android NDK:

Android Native Development Kit (NDK) adalah perangkat yang disediakan oleh Google untuk mengembangkan bagian yang sangat penting dalam hal performa dari aplikasi Android dengan menggunakan bahasa kode asli seperti C dan C++.

Meskipun sebagian besar aplikasi Android dikembangkan menggunakan Java atau Kotlin dan dijalankan dalam lingkungan Android Runtime (ART), NDK memungkinkan pengembang untuk menulis bagian dari aplikasi mereka dalam kode asli, yang dapat menawarkan kinerja yang lebih baik, akses ke fitur-fitur khusus platform, dan integrasi dengan pustaka asli yang ada.

# Android SDK & NDK

## Tujuan Android NDK

### Tujuan NDK:

- **Optimalisasi Kinerja:** Hal ini memungkinkan pengembang untuk menulis bagian yang sangat penting dari aplikasi mereka dalam kode asli
- **Akses ke Perpustakaan Asli:** Pengembang dapat memanfaatkan pustaka C/C++ yang sudah ada atau berintegrasi dengan fitur-fitur khusus
- **Porting Kode yang Sudah Ada:** Pengembang dapat menggunakan kembali basis kode atau pustaka C/C++ yang sudah ada
- **Interaksi Perangkat Keras:** NDK menyediakan akses ke sumber daya sistem tingkat rendah dan fitur perangkat keras

# Android SDK & NDK

## Perbandingan Android SDK dan NDK

- **Bahasa:** Android SDK terutama menggunakan Java atau Kotlin, sedangkan Android NDK menggunakan C atau C++.
- **Kinerja:** NDK dapat menawarkan kinerja yang lebih baik untuk tugas-tugas tertentu karena eksekusi kode asli, sementara aplikasi SDK berjalan dalam lingkungan runtime yang terkelola.
- **Portabilitas:** Aplikasi SDK biasanya lebih portabel di berbagai perangkat dan versi Android, sedangkan aplikasi NDK mungkin memerlukan lebih banyak pertimbangan untuk pengoptimalan khusus platform.
- **Kompleksitas Pengembangan:** Menulis dan men-debug kode asli bisa lebih kompleks dibandingkan dengan pengembangan Java atau Kotlin di SDK.



# Android SDK & NDK

## Studi Kasus Android NDK

Aplikasi Open Camera memanfaatkan Android NDK untuk tugas-tugas tertentu yang sangat penting bagi performa, khususnya di bidang pemrosesan gambar dan perekaman video.

Dengan memanfaatkan kode asli yang ditulis dalam bahasa C atau C++, aplikasi ini mampu mencapai kinerja dan efisiensi yang lebih baik untuk tugas-tugas seperti pengambilan, pemrosesan, dan penyandian gambar.

- 1 Software Development Kit
- 2 Android SDK & NDK
- 3 Swift SDK**
- 4 Cross Platform SDK
- 5 Memilih SDK

# Swift SDK

## Definisi Swift SDK

### Definisi Swift SDK”

Swift SDK (Software Development Kit) mengacu pada kumpulan alat perangkat lunak, perpustakaan, dokumentasi, dan sumber daya yang memungkinkan pengembang untuk membuat aplikasi yang secara khusus menggunakan bahasa pemrograman Swift.

Swift adalah bahasa pemrograman yang kuat dan intuitif yang dikembangkan oleh Apple untuk pengembangan aplikasi iOS, macOS, watchOS, dan tvOS. SDK Swift biasanya mencakup modul kode yang sudah ditulis sebelumnya, API (Antarmuka Pemrograman Aplikasi), dan komponen lain yang dapat digunakan oleh pengembang untuk membangun aplikasi dengan lebih efisien dan efektif.

# Swift SDK

## Fitur Swift SDK

### Fitur Swift SDK:

- 1 **Modul dan pustaka yang sudah dibuat sebelumnya:** SDK Swift sering kali menyediakan modul dan pustaka kode yang sudah ditulis sebelumnya
- 2 **Dokumentasi yang komprehensif:** SDK yang baik dilengkapi dengan dokumentasi lengkap yang menjelaskan cara menggunakan fitur-fitur yang disediakan
- 3 **Akses API:** SDK Swift biasanya menawarkan akses ke berbagai API yang memungkinkan pengembang untuk berinteraksi dengan fungsi sistem

# Swift SDK

## Fitur Swift SDK

- 1 **Dukungan perkakas:** SDK dapat menyertakan alat bantu, seperti debugger, profiler, dan kerangka kerja pengujian.
- 2 **Kompatibilitas:** SDK Swift dirancang untuk bekerja secara mulus dengan alat dan platform pengembangan Apple

# Swift SDK

## Integrasi ke XCode

Xcode adalah lingkungan pengembangan terintegrasi (IDE) Apple untuk macOS, yang digunakan untuk mengembangkan perangkat lunak untuk macOS, iOS, iPadOS, watchOS, dan tvOS. Xcode menyediakan seperangkat alat yang komprehensif untuk pengembangan Swift, termasuk pengeditan kode, debugging, pengujian, dan desain antarmuka.

Integrasi dengan Xcode memungkinkan para pengembang untuk menggunakan Swift SDK dengan lancar dalam lingkungan pengembangan mereka, sehingga lebih mudah untuk memasukkan fungsionalitas pihak ketiga ke dalam aplikasi mereka.

- 1 Software Development Kit
- 2 Android SDK & NDK
- 3 Swift SDK
- 4 Cross Platform SDK**
- 5 Memilih SDK

# Cross Platform SDK

## Definisi Cross Platform SDK

### **Definisi Cross Platform SDK:**

SDK Lintas Platform, juga dikenal sebagai SDK Multi-platform, adalah kit pengembangan perangkat lunak yang memungkinkan pengembang membangun aplikasi yang dapat berjalan di berbagai sistem operasi atau platform dengan modifikasi minimal.

SDK ini mengabstraksikan perbedaan spesifik platform, sehingga pengembang dapat menulis kode sekali dan menyebarkannya di berbagai platform, seperti iOS, Android, Windows, dan browser web.



# Cross Platform SDK

## Contoh Cross Platform SDK

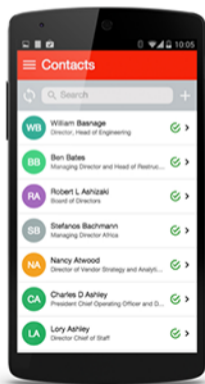
Salah satu contoh SDK lintas platform yang populer adalah Flutter, yang dikembangkan oleh Google. Flutter adalah kit pengembangan perangkat lunak UI sumber terbuka untuk membangun aplikasi yang dikompilasi secara native untuk seluler, web, dan desktop dari satu basis kode.

Dengan Flutter, pengembang dapat menulis kode menggunakan bahasa pemrograman Dart dan menyebarkan aplikasi ke berbagai platform, termasuk iOS, Android, peramban web, dan sistem operasi desktop seperti Windows, macOS, dan Linux.

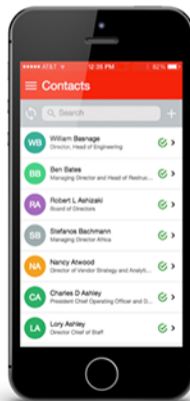
# Cross Platform SDK

## Contoh Cross Platform SDK

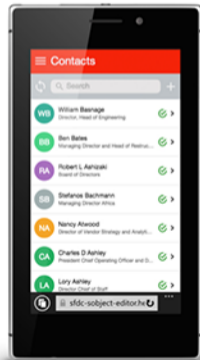
Android



iOS



HTML5



# Cross Platform SDK

## Tantangan Cross Platform SDK

### Tantangan SDK:

- **Pertimbangan kinerja:** SDK lintas platform dapat menimbulkan biaya tambahan dan keterbatasan kinerja dibandingkan dengan pengembangan khusus platform.
- **Fitur khusus platform:** Menerapkan fitur dan pengoptimalan khusus platform mungkin memerlukan upaya dan solusi tambahan.
- **Masalah kompatibilitas:** Mengikuti pembaruan dan perubahan di berbagai platform dapat menjadi tantangan, yang menyebabkan masalah kompatibilitas dan potensi bug.

# Cross Platform SDK

## Tantangan Cross Platform SDK

- **Pengalaman pengguna:** Mencapai pengalaman pengguna yang konsisten dan berkualitas tinggi di berbagai platform mungkin memerlukan perhatian ekstra terhadap detail dan penyesuaian.
- **Kurva pembelajaran:** Pengembang mungkin perlu mempelajari alat, bahasa, dan paradigma pengembangan baru untuk menggunakan SDK lintas platform secara efektif, yang dapat meningkatkan kurva pembelajaran dan waktu pengembangan pada awalnya.

- 1 Software Development Kit
- 2 Android SDK & NDK
- 3 Swift SDK
- 4 Cross Platform SDK
- 5 **Memilih SDK**

# Memilih SDK

## Pertimbangan Memilih SDK

**Ketika memilih SDK, pertimbangkan:**

1. **Kompatibilitas Platform:** Pilih SDK yang mendukung platform yang ingin ditargetkan, seperti iOS, Android, atau keduanya.
2. **Bahasa Pemrograman:** Pertimbangkan bahasa pemrograman yang digunakan oleh SDK.
3. **Kumpulan Fitur:** Nilai fitur yang disediakan oleh SDK
4. **Kinerja:** Evaluasi karakteristik kinerja SDK, termasuk faktor-faktor seperti kecepatan runtime, penggunaan memori, dan konsumsi baterai.
5. **Komunitas dan Dukungan:** Pertimbangkan ukuran dan aktivitas komunitas pengembang SDK.

# Memilih SDK

## Pertimbangan Memilih SDK

- 6. Kemudahan Penggunaan dan Kurva Pembelajaran:** Nilai kemudahan penggunaan SDK dan kurva pembelajaran.
- 7. Integrasi dengan Alat Pengembangan:** Pertimbangkan seberapa baik SDK terintegrasi dengan alat pengembangan dan IDE
- 8. Biaya dan Lisensi:** Evaluasi struktur biaya dan persyaratan lisensi yang terkait dengan SDK.
- 9. Skalabilitas dan Fleksibilitas:** Pertimbangkan skalabilitas dan fleksibilitas SDK
- 10. Keamanan dan Kepatuhan:** Pastikan SDK memprioritaskan praktik terbaik keamanan dan mematuhi standar

*Terima Kasih*