Mobile Programming

Catatan Kuliah #3

Alauddin Maulana Hirzan, M. Kom
0607069401

Arsitektur **Android**

Arsitektur **Android**

Arsitektur Android #1

Arsitektur Android didasarkan pada prinsip-prinsip pola Model-View-Controller (MVC), yang memisahkan logika, presentasi, dan data aplikasi. Arsitektur Android terdiri dari empat lapisan utama:

1. Lapisan kernel Linux
2. Lapisan abstraksi perangkat keras (HAL)
3. Lapisan runtime Android
4. Lapisan kerangka kerja aplikasi

Arsitektur **Android**

Arsitektur Android #2

Arsitektur Android juga mencakup berbagai komponen dan kerangka kerja yang dapat digunakan oleh pengembang untuk membangun aplikasi mereka.

- Sebagai contoh, **Android Architecture Components (AAC)** menyediakan seperangkat pustaka yang membantu pengembang mendesain aplikasi yang tangguh, dapat diuji, dan mudah dipelihara.

Lapisan Linux Kernel

Lapisan Linux Kernel

Arsitektur Android - Lapisan Linux Kernel #1

Lapisan ini menyediakan layanan dan fungsionalitas inti yang diperlukan untuk menjalankan sistem operasi Android pada perangkat. Lapisan kernel Linux bertanggung jawab untuk mengelola sumber daya perangkat keras.

Kernel Linux yang digunakan di Android didasarkan pada kernel Linux versi 2.6 atau yang lebih baru. Kernel ini telah dimodifikasi dan dioptimalkan untuk digunakan pada perangkat seluler, seperti smartphone dan tablet.

Lapisan Linux Kernel

Arsitektur Android - Lapisan Linux Kernel #2

Beberapa fitur utama dari kernel Linux yang digunakan di Android antara lain:

- Manajemen proses
- Manajemen memori
- Kerangka kerja driver perangkat
- Fitur keamanan
- Jaringan

Selain fitur-fitur ini, kernel Linux yang digunakan di Android telah dioptimalkan untuk konsumsi daya yang rendah dan penggunaan sumber daya yang efisien, yang sangat penting untuk perangkat seluler dengan daya tahan baterai yang terbatas.

Lapisan Linux Kernel

Arsitektur Android - Lapisan Linux Kernel #3

Perkembangan Kernel Android

- Sistem operasi Android didasarkan pada kernel Linux, dan secara historis, Google telah mempertahankan fork-nya sendiri dari kernel Linux. Namun, dalam beberapa tahun terakhir, telah ada upaya untuk beralih ke penggunaan **Kernel Linux Mainline** untuk perangkat Android.
- **Kernel Linux Mainline** adalah versi kernel Linux yang dikelola oleh komunitas Linux dan digunakan oleh sebagian besar sistem operasi berbasis Linux. Dengan menggunakan kernel Linux mainline, Android dapat memanfaatkan fitur-fitur dan peningkatan terbaru dalam kernel Linux, dan mengurangi beban pemeliharaan untuk memelihara fork-nya sendiri.

Lapisan Linux Kernel

Arsitektur Android - Lapisan Linux Kernel #4

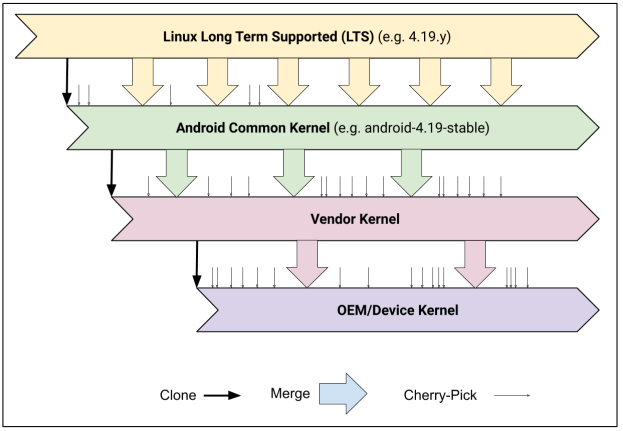
- Perubahan **Kernel Linux Mainline** untuk perangkat Android diatur dalam bentuk **Android Mainline Kernel Project**, yang bertujuan untuk melakukan perubahan khusus Android pada **Kernel Linux Mainline**. Proyek ini telah mengerjakan perubahan hulu yang terkait dengan manajemen daya, keamanan, dan area lainnya
- Selain **Android Mainline Kernel Project**, berbagai vendor perangkat keras dan produsen perangkat telah bekerja untuk meningkatkan dukungan untuk perangkat mereka ke **Kernel Linux Mainline**.

Lapisan Linux Kernel

Dari **Linux LTS** ke **Vendor OEM**

Singkatnya:

1. Melakukan Kloning **Kernel Linux Long Term Service (LTS)**
2. Ditambal menjadi **Android Common Kernel**
3. Vendor memodifikasi
4. Jadilah Kernel khusus OEM



Lapisan Linux Kernel

Arsitektur Android - **Android Common Kernel** #1

Android Common Kernel adalah versi kernel Linux yang digunakan oleh banyak perangkat Android. Android Common Kernel adalah satu versi kernel yang digunakan bersama di beberapa perangkat, bukannya setiap perangkat yang unik.

Android Common Kernel dikelola oleh Android Open Source Project (AOSP) dan didasarkan pada kernel Linux utama. Namun, kernel ini menyertakan beberapa perubahan dan pengoptimalan khusus Android untuk mendukung fitur-fitur seperti akselerasi perangkat keras, manajemen daya, dan keamanan.

Lapisan Linux Kernel

Arsitektur Android - **Android Common Kernel** #2

Perubahan / Tambalan ini dapat mencakup:

- Dukungan dan pilihan fungsionalitas hulu yang diperlukan untuk fitur Android
- Fitur yang siap untuk perangkat Android tetapi masih dalam pengembangan di bagian hulu (misalnya, pengoptimalan penempatan tugas Penjadwal Sadar Energi).
- Fitur vendor/OEM yang berguna bagi mitra ekosistem lain (misalnya, sdcardfs).

Lapisan Linux Kernel

Arsitektur Android - **Android Common Kernel** #3

Dengan menggunakan kernel yang sama di berbagai perangkat, produsen perangkat dapat memperoleh manfaat dari berkurangnya waktu pengembangan dan pengujian, serta pemeliharaan dan dukungan yang lebih mudah. Selain itu, pengembang dapat memperoleh manfaat dari lingkungan perangkat keras dan perangkat lunak yang lebih konsisten, yang dapat mempermudah pengembangan dan pengujian aplikasi.

Android Common Kernel juga dirancang agar kompatibel dengan **Android Compatibility Definition Document (CDD)**, yang merupakan seperangkat persyaratan yang harus dipenuhi oleh perangkat agar dianggap kompatibel dengan sistem operasi Android.

Lapisan Linux Kernel

Arsitektur Android - **Android Common Kernel** - **android-mainline** #1

android-mainline adalah cabang pengembangan utama untuk fitur-fitur Android. **Linux Mainline** digabungkan ke dalam android-mainline setiap kali Linus Torvalds memposting rilis atau kandidat rilis.

Sebelum tahun 2019, **Android Common Kernel** dibuat dengan mengkloning kernel LTS yang baru saja dideklarasikan dan menambahkan patch khusus Android.

Proses ini berubah pada tahun 2019 dengan mencabang **Android Common Kernel**. Model baru ini menghindari upaya yang signifikan untuk meneruskan port dan menguji patch Android dengan mencapai hasil yang sama secara bertahap.

Lapisan Linux Kernel

Arsitektur Android - **Android Common Kernel** - **android-mainline** #2

Ketika LTS baru dideklarasikan di bagian hulu, kernel umum yang sesuai akan bercabang dari **android-mainline**. Hal ini memungkinkan para mitra untuk memulai sebuah proyek sebelum deklarasi versi LTS, dengan menggabungkannya dari **android-mainline**. Setelah cabang kernel umum yang baru dibuat, para mitra dapat dengan mudah mengubah sumber penggabungan ke cabang yang baru.

Lapisan Linux Kernel

Arsitektur Android - **Generic Kernel Image (GKI)** #1

Jika Android Common Kernel (ACK) adalah dasar untuk semua kernel produk Android, maka **Generic Kernel Image (GKI)** adalah kernel yang telah dibuat sebelumnya yang dapat digunakan di berbagai perangkat dengan komponen perangkat keras yang serupa.

Dalam kasus Android, GKI adalah kernel yang telah dikompilasi untuk mendukung berbagai komponen perangkat keras, dan bukannya secara khusus disesuaikan untuk perangkat tertentu.

GKI diperkenalkan sejak Android 12 untuk mengurangi fragmentasi dan meningkatkan kompatibilitas antar perangkat

Lapisan Linux Kernel

Arsitektur Android - **Generic Kernel Image (GKI)** #2

Di Android, sebagian besar perangkat menggunakan **Device-Specified Kernel** daripada **Generic Kernel Image**. Namun, beberapa perangkat dengan komponen perangkat keras yang serupa, seperti yang berbasis chipset Qualcomm Snapdragon, mungkin dapat menggunakan kernel yang sama di beberapa perangkat.

Dengan demikian, kernel perangkat terdiri dari:

- Hulu: Kernel Linux dari kernel.org
- AOSP: Patch khusus Android tambahan dari kernel umum AOSP
- Vendor: Patch pengoptimalan dan pemberdayaan SoC dan periferal dari vendor
- OEM/perangkat: Driver dan penyesuaian perangkat tambahan

Lapisan Abstraksi Hardware

Lapisan Abstraksi Hardware

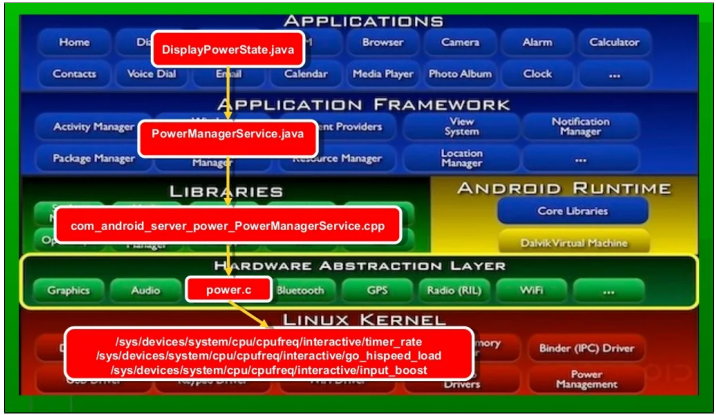
Arsitektur Android - **Hardware Abstraction Layer (HAL)** #1

Di Android, Hardware Abstraction Layer (HAL) adalah lapisan perangkat lunak yang berada di antara kerangka kerja Android dan driver perangkat khusus perangkat keras. HAL menyediakan antarmuka standar untuk mengakses komponen perangkat keras, terlepas dari implementasi spesifik komponen tersebut pada perangkat tertentu.

HAL biasanya diimplementasikan oleh produsen perangkat, dan dapat disesuaikan untuk mendukung komponen perangkat keras tertentu yang digunakan pada perangkat tertentu. Dengan menyediakan antarmuka standar untuk mengakses komponen perangkat keras, HAL dapat membantu mengurangi fragmentasi dan meningkatkan kompatibilitas di berbagai perangkat.

Lapisan Abstraksi Hardware

Arsitektur Android - **Hardware Abstraction Layer (HAL)** #2



Lapisan Abstraksi Hardware

Arsitektur Android - **Hardware Abstraction Layer (HAL)** #3

Contoh Lain:

Ketika aplikasi Android perlu mengakses komponen perangkat keras, seperti kamera atau sensor, aplikasi tersebut mengirimkan permintaan ke kerangka kerja Android. Kerangka kerja kemudian meneruskan permintaan ke modul HAL yang sesuai, yang bertanggung jawab untuk berkomunikasi dengan driver perangkat untuk komponen perangkat keras tersebut.

Modul HAL berkomunikasi dengan driver perangkat menggunakan perintah dan protokol khusus perangkat keras, yang tidak terpapar pada kerangka kerja Android yang lebih tinggi. Modul HAL menerjemahkan perintah khusus perangkat keras ini ke dalam antarmuka standar yang dapat digunakan oleh kerangka kerja Android.

Lapisan Runtime Android

Lapisan Runtime Android

Arsitektur Android - **Android Runtime** dan **Dalvik** #1

Android runtime (ART) adalah runtime terkelola yang digunakan oleh aplikasi dan beberapa layanan sistem di Android. ART dan pendahulunya, Dalvik, pada awalnya dibuat khusus untuk proyek Android. ART sebagai runtime mengeksekusi format Dalvik Executable dan spesifikasi **bytecode Dex**.

ART dan Dalvik adalah runtime yang kompatibel yang menjalankan **bytecode Dex**, sehingga aplikasi yang dikembangkan untuk Dalvik akan bekerja ketika dijalankan dengan ART. Namun, beberapa teknik yang bekerja pada Dalvik tidak bekerja pada ART.

Lapisan Runtime Android

Arsitektur Android - **Android Runtime** dan **Dalvik** #2

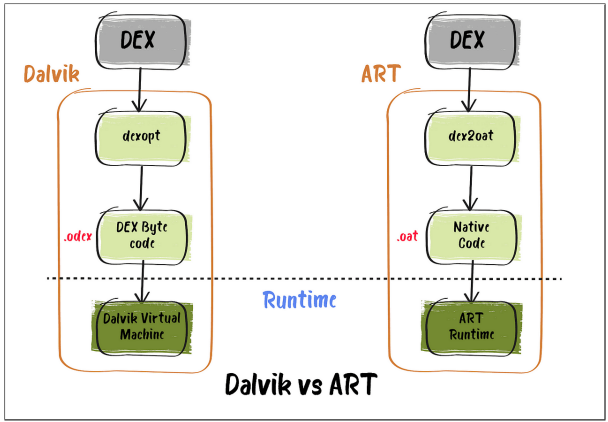
Apa itu **Bytecode Dex**?

Bytecode Dex adalah format kode yang dikompilasi yang berjalan di **Dalvik Virtual Machine (DVM)**, yang merupakan mesin virtual yang digunakan di Android untuk menjalankan aplikasi. Nama "Dex" merupakan kependekan dari **Dalvik Executable.**

Ketika aplikasi Android dikembangkan, kode sumber Java pertama kali dikompilasi ke dalam bytecode Java, yang merupakan format yang tidak bergantung pada platform. Kode bytecode Java kemudian diubah menjadi kode bytecode Dex menggunakan alat "dx", yang merupakan bagian dari Android SDK.

Lapisan Runtime Android

Arsitektur Android - **Android Runtime** dan **Dalvik** #3



Lapisan Runtime Android

Arsitektur Android - Fitur ART #1

Ahead of Time (AOT)

ART memperkenalkan kompilasi ahead-of-time (AOT), yang dapat meningkatkan kinerja aplikasi. ART juga memiliki verifikasi waktu pemasangan yang lebih ketat daripada Dalvik.

Pada saat penginstalan, ART mengkompilasi aplikasi menggunakan alat dex2oat pada perangkat Utilitas ini menerima file DEX sebagai masukan dan menghasilkan aplikasi yang dikompilasi yang dapat dieksekusi untuk perangkat target. Utilitas ini seharusnya dapat mengkompilasi semua file DEX yang valid tanpa kesulitan.

Lapisan Runtime Android

Arsitektur Android - Fitur ART #2

Pengumpulan sampah yang lebih baik

Pengumpulan sampah (GC) sangat boros sumber daya, yang dapat mengganggu kinerja aplikasi, menghasilkan tampilan yang berombak, responsifitas UI yang buruk, dan masalah lainnya.

Peningkatan pengembangan dan debugging

ART menawarkan sejumlah fitur untuk meningkatkan pengembangan dan debugging aplikasi.

Lapisan Framework dan Aplikasi Android

Lapisan Framework dan Aplikasi Android

Arsitektur Android - **Lapisan Framework dan Aplikasi Android** #1

Lapisan kerangka kerja adalah lapisan tingkat bawah dari tumpukan perangkat lunak Android, yang menyediakan komponen dan layanan inti yang diperlukan untuk menjalankan aplikasi Android.

Lapisan aplikasi adalah lapisan tingkat yang lebih tinggi dari tumpukan perangkat lunak Android, yang mencakup semua aplikasi yang berhadapan langsung dengan pengguna yang diinstal pada perangkat Android.

Lapisan Framework dan Aplikasi Android

Arsitektur Android - **Lapisan Framework dan Aplikasi Android** #2


THANK YOU