

E-Book

# Membangun Aplikasi Berbasis Android Menggunakan Kotlin Bagi Software Developer

I Putu Arik Sasmita

# Daftar Isi

<b>Daftar Isi .....</b>	<b>2</b>
<b>Mengapa Harus Belajar Kotlin .....</b>	<b>3</b>
<b>Mengaktifkan Mode Developer di Device Android.....</b>	<b>6</b>
Mengaktifkan opsi developer dan proses debug USB.....	6
Ops umum.....	7
Proses Debug.....	8
Jaringan .....	9
Input .....	10
Rendering yang dipercepat hardware .....	11
Media .....	12
Pemantauan.....	13
Aplikasi.....	13
<b>Menjalankan Aplikasi di Perangkat Hardware.....</b>	<b>14</b>
<b>Informasi Tambahan .....</b>	<b>17</b>
<b>Referensi .....</b>	<b>18</b>

# Mengapa Harus Belajar Kotlin

Selain bahasa pemrograman Java, ada satu lagi bahasa pemrograman yang didukung secara resmi oleh Google yang bisa dipakai bersama dengan Android Studio. Bahasa tersebut adalah bahasa pemrograman Kotlin.

Sejak diluncurkan pada pagelaran Google I/O pada bulan Mei 2017, Kotlin menjadi bahasa kelas utama (*first class language*) dalam pembuatan aplikasi Android. Hal tersebut merupakan alasan mengapa para android developer sangat wajib untuk mempelajari Kotlin dalam meningkatkan skillnya.

Kotlin adalah bahasa pemrograman modern, disajikan secara statis yang berjalan pada *platform Java Virtual Machine* (JVM). Kotlin menggunakan *compiler* LLVM yang artinya, dapat dikompilasi ke dalam kode JavaScript. Inilah beberapa alasan mengapa kamu wajib belajar Kotlin:

## 1. Memiliki Ekosistem yang Sudah Matang

Kotlin sudah melalui tahapan panjang sebelum memasuki versi stabil 1.0 sejak tahun 2016 yang lalu. Itu artinya, bahasa pemrograman Kotlin adalah bahasa yang sudah sangat matang. Bahasa ini tidak lagi dalam tahap beta apalagi alpha, tapi sudah stable sehingga cocok dipakai untuk production.

Setiap hari selalu ada library baru untuk membantu proses pembuatan aplikasi Android dengan Kotlin. Ditambah lagi, kita masih tetap bisa memakai library Java yang sudah ada tanpa masalah.

## 2. Terintegrasi dengan Android studio

Kotlin dikembangkan oleh JetBrains, perusahaan yang mengembangkan IntelliJ, basis awal Android Studio. Tidak mengherankan jika Kotlin terintegrasi dengan sempurna di dalam Android Studio karena memang bisa dibilang kedua nya dikembangkan oleh perusahaan yang sama.

Di Android Studio versi 2.3 kita masih harus memasang plugin tambahan untuk Android Studio agar bisa memakai Kotlin. Tetapi, setelah versi 3.0 Kotlin sudah bisa langsung digunakan tanpa harus memasang plugin tambahan.

### **3. Update Java di Android Relatif Lebih Lambat**

Kotlin memiliki konsep lambda, extension function, functional operation on collection, sealed classes, data classes, dan konsep-konsep modern yang lain yang sudah dibungkus siap pakai saat ini juga. Bandingkan dengan Java, dimana kita harus ngoprek sendiri agar project Android bisa memakai Java 8 dan menggunakan konsep-konsep modern tadi. Dengan Kotlin, semua itu sudah ada tanpa susah payah.

### **4. Aman dari NullPointerException**

Jika pembaca sudah pernah membuat aplikasi Android dengan Java, pasti sudah bersahabat dengan yang namanya NullPointerException (NPE). Exception ini bisa membuat developer frustrasi karena membuat aplikasi crash. NPE memang masih bisa diakali dengan menggunakan blok try catch atau menggunakan annotations, tapi dengan Kotlin kita mendapatkan pengamanan secara otomatis dari sananya.

Saat kita membuat aplikasi Android dengan Kotlin, compiler akan mendeteksi adanya kemungkinan error sebelum membuat APK (atau saat di run). Dengan begini kita bisa langsung tahu dimana sumber yang nantinya bisa bermasalah. Jika bagian yang memiliki kemungkinan error tidak diatasi, maka aplikasi tidak akan bisa di run. Berbeda dengan Java, dimana tidak ada pemeriksaan seperti itu dari sananya. Kita akan tahu bahwa suatu bagian menyebabkan NPE saat aplikasi sudah berjalan dan crash.

### **5. Kotlin Tidak Hanya di Android**

Meski pemakaiannya yang paling booming adalah dilingkungan Android, Kotlin nyatanya bisa dipakai diluar itu. Kotlin bisa dipakai untuk membuat aplikasi iOS (melalui Kotlin Native), juga bisa dipakai untuk menulis JavaScript (melalui Kotlin JS). Intinya, karena Kotlin memiliki interoperability 100% dengan Java, maka apapun yang bisa menggunakan Java bisa menggunakan Kotlin. Dengan demikian, jika bisa menggunakan Kotlin untuk Android, maka kita juga bisa memanfaatkannya untuk bidang lain.

## **6. Perusahaan Mulai Mengimplementasi Kotlin**

Perusahaan-perusahaan besar seperti Gojek sudah mulai mengimplementasi Kotlin di project Java yang sudah ada. Langkah ini tentu akan diikuti oleh perusahaan-perusahaan lainnya. Artinya apa? Semakin banyak perusahaan yang mengimplementasi Kotlin, maka kebutuhan akan developer Android yang memahami bahasa ini akan terus meningkat.

## **7. Tidak Sulit Untuk Dipelajari**

Kenapa bahasa pemrograman Kotlin begitu dipuja dan dipuji developer Android? Jawabannya simpel saja, kesederhanaan Kotlin yang membuat developer Android jatuh cinta. Jika pembaca sudah bisa menggunakan bahasa Java untuk membuat Aplikasi Android, pasti tidak akan mengalami permasalahan yang berarti saat belajar Kotlin.

Kotlin didesain untuk memakai elemen-elemen yang sudah ada dengan perbaikan di bagian tertentu sehingga proses ngoding menjadi lebih menyenangkan. Karena kemudahannya ini ditambah lagi dengan kecanggihan Android Studio, hanya dalam hitungan hari pembaca pasti sudah bisa membuat sendiri aplikasi Android dengan Kotlin.

# Mengaktifkan Mode Developer di Device Android

## Mengaktifkan opsi developer dan proses debug USB

Pada Android 4.1 dan yang lebih rendah, layar Opsi developer sudah tersedia secara default. Pada Android 4.2 dan yang lebih tinggi, Anda harus mengaktifkan layar ini. Untuk mengaktifkan opsi developer, ketuk opsi Nomor Versi 7 kali. Opsi ini dapat Anda temukan di salah satu lokasi berikut, bergantung pada versi Android Anda:

- Android 9 (API level 28) dan yang lebih tinggi: Setelan > Tentang Ponsel > Nomor Versi
- Android 8.0.0 (API level 26) dan Android 8.1.0 (API level 26): Setelan > Sistem > Tentang Ponsel > Nomor Versi
- Android 7.1 (API level 25) dan yang lebih rendah: Setelan > Tentang Ponsel > Nomor Versi

Di bagian atas layar Opsi developer, Anda dapat mengaktifkan dan menonaktifkan opsi. Mungkin sebaiknya ini dibiarkan aktif. Saat nonaktif, sebagian besar opsi dinonaktifkan kecuali opsi yang tidak memerlukan komunikasi antara perangkat dan komputer pengembangan Anda.

Sebelum Anda dapat menggunakan debugger dan alat lainnya, Anda harus mengaktifkan proses debug USB, yang memungkinkan Android Studio dan alat SDK lainnya untuk mengenali perangkat Anda saat terhubung melalui USB. Untuk mengaktifkan proses debug USB, aktifkan opsi proses debug USB pada menu Opsi Developer. Anda dapat menemukan opsi ini di salah satu lokasi berikut, bergantung pada versi Android Anda:

- Android 9 (API level 28) dan yang lebih tinggi: Setelan > Sistem > Lanjutan > Opsi Developer > Proses debug USB
- Android 8.0.0 (API level 26) dan Android 8.1.0 (API level 26): Setelan > Sistem > Opsi Developer > Proses debug USB
- Android 7.1 (API level 25) dan yang lebih rendah: Setelan > Opsi Developer > Proses debug USB

## Opsi umum

Di Android 8.0 dan yang lebih tinggi, Anda dapat mengetuk Kotak setelan cepat untuk developer untuk menambahkan opsi developer yang dipilih ke panel Setelan Cepat Anda. Setelah memilih satu atau beberapa kotak yang tersedia (gambar 2), buka panel Setelan Cepat, lalu ketuk ikon pensil untuk masuk ke mode edit. Kemudian, tarik kotak developer dari panel kotak ke panel Setelan cepat, lalu ketuk ikon pensil kembali untuk keluar dari mode edit. Opsi umum lain di antaranya sebagai berikut:

- **Memori:** (Pada Android 8.0 dan yang lebih tinggi) Menampilkan statistik memori, seperti penggunaan rata-rata memori, performa memori, total memori yang tersedia, rata-rata memori yang digunakan, jumlah memori bebas yang tersedia, dan jumlah memori yang digunakan oleh aplikasi.
- **Ambil laporan bug:** Mendapatkan salinan file log perangkat saat ini untuk dibagikan dengan orang lain. Saat Anda mendapatkan notifikasi bahwa laporan bug sudah siap, ketuk notifikasi untuk membagikannya.
- **Mode demo UI sistem:** Mempermudah pengambilan screenshot bersih dengan menampilkan baris notifikasi preset umum dengan tidak menampilkan notifikasi atau peringatan daya baterai lemah. Aktifkan Mode Demo memungkinkan Anda mengubah tampilan status bar menggunakan perintah mode demo adb. Atau, Anda dapat menggunakan Tampilkan Mode Demo untuk menyembunyikan notifikasi dan menampilkan status bar preset.
- **Sandi cadangan desktop:** Menyetel sandi cadangan sehingga Anda dapat menggunakan perintah adb untuk mencadangkan serta memulihkan aplikasi dan data perangkat dengan perlindungan sandi.
- **Tetap terjaga:** Mengatur layar agar tetap menyala setiap kali Anda mengisi daya.
- **Aktifkan log snoop Host Controller Interface (HCI) Bluetooth:** Mengambil semua paket HCI Bluetooth dalam file yang disimpan di /sdcard/btsnoop\_hci.log. Anda dapat mengambil paketnya, kemudian menggunakan program seperti Wireshark untuk menganalisis dan memecahkan masalah informasi tersebut.

*Catatan: Perintah mode demo adb mungkin tidak berfungsi pada semua perangkat karena tidak diverifikasi selama pengujian sertifikasi Android. Perintah hanya diverifikasi agar berfungsi pada perangkat Nexus dan Pixel.*

## Proses Debug

Opsi proses debug menyediakan cara untuk mengonfigurasi proses debug pada perangkat, dan untuk menjalin komunikasi antara perangkat dengan komputer pengembangan Anda.

Aktifkan Proses debug USB sehingga perangkat Android Anda dapat berkomunikasi dengan mesin pengembangan melalui Android Debug Bridge (adb). Opsi Tunggu Debugger tidak tersedia sebelum Anda menggunakan Pilih aplikasi debug untuk memilih aplikasi yang akan di-debug. Jika Anda mengaktifkan Tunggu Debugger, aplikasi yang dipilih akan menunggu debugger disematkan sebelum berjalan. Berikut adalah opsi proses debug lainnya:

- Terus simpan data pencatat log di perangkat: Memilih jenis pesan log yang ingin Anda simpan secara persisten di perangkat. Opsinya adalah nonaktif, semua, semua kecuali radio, atau hanya kernel.
- Pilih aplikasi lokasi palsu: Menggunakan opsi ini untuk memalsukan lokasi GPS perangkat guna menguji apakah aplikasi Anda berperilaku yang sama dengan yang ada di lokasi lain. Untuk menggunakan opsi ini, download dan instal aplikasi lokasi palsu GPS.
- Aktifkan inspeksi atribut tampilan: Menyimpan informasi atribut tampilan dalam variabel anggota mAttributes instance View sehingga dapat digunakan untuk proses debug. Anda dapat mengakses informasi atribut melalui antarmuka pengguna Layout Inspector, seperti ditunjukkan dalam Gambar 4 (tanpa mengaktifkan opsi ini, item "Attributes" tidak akan tersedia).
- Aktifkan lapisan debug GPU: Tersedia di perangkat yang menjalankan Android 9 (API level 28) dan yang lebih baru. Aktifkan opsi ini untuk memungkinkan pemuatan lapisan validasi Vulkan dari penyimpanan perangkat lokal. Untuk mempelajari lebih lanjut, baca Lapisan validasi Vulkan di Android.



## Jaringan

Opsi jaringan menyediakan cara untuk mengkonfigurasi setelan Wi-Fi dan DHCP.

Ketuk Select USB Configuration untuk menentukan cara komputer mengidentifikasi perangkat seperti yang Anda inginkan. Sebagaimana ditunjukkan dalam gambar 5, Anda dapat mengkonfigurasi perangkat hanya untuk mengisi daya, mentransfer file (MTP), mentransfer gambar (PTP), menggunakan internet seluler di PC (RNDIS), atau mentransfer audio atau file MIDI.

Ketuk Bluetooth AVRCP version dan pilih versi profil yang ingin Anda gunakan untuk mengontrol semua peralatan Bluetooth A/V yang aksesnya dimiliki oleh perangkat Anda. Selain itu, untuk menyempurnakan pemutaran audio pada perangkat, ketuk dan setel opsi berikut:

- Codec Audio Bluetooth: Menyesuaikan kualitas suara (codec) perangkat Anda dengan memilih salah satu codec berikut:
  1. SBC: Mentransfer data ke perangkat output audio Bluetooth seperti headphone dan speaker.
  2. AAC: Kualitas suara yang lebih baik dari perangkat berkabel daripada MP3 dengan kecepatan bit yang serupa.
  3. aptX: Suara nirkabel jernih di smartphone, speaker, soundbar, headphone, dan tablet berkualitas tinggi.
  4. aptX HD: Streaming dengan resolusi tinggi ke perangkat Bluetooth.
  5. LDAC: Menyediakan musik berkualitas tinggi melalui koneksi nirkabel.
  6. Aktifkan Codec Opsional dan Nonaktifkan Codec Opsional: Jika Anda sudah menginstal implementasi codec tambahan, gunakan opsi ini untuk mengaktifkan dan menonaktifkannya.
- Rentang Sampel Audio Bluetooth: Menyesuaikan jumlah sampel audio per detik dengan memilih frekuensi sampel codec audio. Frekuensi sampel yang lebih tinggi menggunakan lebih banyak resource.
- Bit Audio Bluetooth per sampel: Menentukan jumlah bit informasi dalam setiap sampel audio. Dengan kecepatan bit yang lebih tinggi, suaranya akan berkualitas lebih baik tetapi file sampelnya berukuran lebih besar.
- Mode Saluran Audio Bluetooth: Memilih mono atau stereo.

- Codec LDAC Audio Bluetooth: Mengoptimalkan suara untuk meningkatkan kualitas audio, menyeimbangkan audio dan kualitas koneksi, meningkatkan kualitas koneksi, atau menggunakan kecepatan bit adaptif untuk menyeimbangkan kualitas audio dan koneksi.

Daftar berikut menjelaskan cara lain untuk mengonfigurasi penyiapan Wi-Fi dan DHCP:

- Sertifikasi layar nirkabel: Mengaktifkan kontrol dan setelan konfigurasi lanjutan untuk sertifikasi Layar Nirkabel sesuai spesifikasi yang ditetapkan dalam Spesifikasi Tampilan Wi-Fi untuk Wi-Fi Alliance. Sertifikasi berlaku untuk Android 4.4 (API level 19) dan yang lebih baru.
- Aktifkan logging panjang Wi-Fi: Meningkatkan level logging Wi-Fi untuk setiap jaringan nirkabel (SSID) yang Anda hubungkan sesuai dengan kekuatan sinyal relatif yang diterima (RSSI). Untuk mengetahui informasi selengkapnya tentang log, lihat Menulis dan menampilkan log dengan Logcat.
- Penerimaan Wi-Fi agresif ke seluler: Ketika sinyal lemah, fitur ini akan membuat Wi-Fi lebih efisien dalam menerima koneksi data ke jaringan seluler.

## Input

Aktifkan Show taps untuk menampilkan ketukan saat Anda menyentuh layar. Lingkaran akan muncul di bawah jari atau stilus dan mengikuti gerakan jari Anda saat menjelajahi layar. Ketukan berfungsi seperti penunjuk saat Anda Merekam video pada perangkat.

Aktifkan Lokasi Penunjuk untuk menampilkan lokasi penunjuk (ketukan) pada perangkat dengan garis-bidik. Sebuah kolom akan muncul di bagian atas layar untuk melacak koordinat garis-bidik. Saat Anda menggerakkan penunjuk, koordinat dalam kolom akan melacak lokasi garis-bidik dan jalur penunjuk tergambar di layar.

Opsi Gambar memberikan isyarat visual tentang antarmuka pengguna aplikasi dan cara pengoperasiannya. Aktifkan Show Layout Bounds untuk menunjukkan garis batas klip, margin, dan konstruksi antarmuka pengguna aplikasi yang lain pada perangkat. Opsi Gambar lainnya meliputi opsi berikut:

- Paksa arah tata letak RTL: Memaksa arah tata letak layar dari kanan ke kiri (RTL) atau dari kiri ke kanan (default).

- Skala animasi jendela: Menyetel kecepatan pemutaran animasi jendela agar Anda dapat memeriksa performanya pada kecepatan yang berbeda. Skala yang lebih kecil akan menghasilkan kecepatan yang lebih tinggi.
- Skala animasi transisi: Menyetel kecepatan pemutaran animasi transisi agar Anda dapat memeriksa performanya pada kecepatan yang berbeda. Skala yang lebih kecil akan menghasilkan kecepatan yang lebih tinggi.
- Simulasikan tampilan sekunder: Membuat tampilan sekunder sebagai overlay di perangkat. Hal ini berguna ketika mendukung tampilan tambahan dengan Presentation API. Lihat Tampilan sekunder.

## Rendering yang dipercepat hardware

Opsi rendering yang dipercepat hardware memberikan cara untuk mengoptimalkan aplikasi Anda bagi platform hardware targetnya dengan memanfaatkan opsi berbasis hardware seperti GPU, lapisan hardware, dan multisample anti-aliasing (MSAA).

Ketuk Simulate color space untuk mengubah skema warna di seluruh UI perangkat. Opsi ini merujuk pada jenis kebutaan warna. Pilihannya adalah Dinonaktifkan (tidak ada skema warna yang disimulasikan), Monokromasi (hitam, putih, dan abu-abu), Deuteranomali (merah-hijau), Protanomali (merah-hijau), dan Tritanomali (biru-kuning). Protanomali mengacu pada kebutaan warna merah-hijau dengan kelemahan pada warna merah, dan Deuteranomali mengacu pada kebutaan warna merah-hijau dengan kelemahan pada warna hijau.

Jika Anda mengambil screenshot dalam ruang warna yang disimulasikan, warna tersebut akan tampak normal seolah Anda tidak mengubah skema warna. Beberapa cara lain untuk memanfaatkan opsi berbasis hardware adalah sebagai berikut:

- Setel perender GPU: Mengubah mesin grafis Open GL default ke mesin grafis Open GL Skia.
- Paksa rendering GPU: Memaksa aplikasi menggunakan GPU untuk gambar 2D, jika ditulis tanpa rendering GPU secara default.
- Tampilkan pembaruan tampilan GPU: Menampilkan setiap elemen pada layar yang digambar dengan GPU.

- Debug overdraw oleh GPU: Menampilkan pemberian kode warna di perangkat sehingga Anda dapat memvisualisasikan jumlah piksel serupa yang telah digambar dalam frame yang sama. Visualisasi menunjukkan di mana aplikasi Anda mungkin melakukan lebih banyak rendering daripada yang diperlukan. Untuk mengetahui informasi selengkapnya, lihat Memvisualisasikan overdraw GPU.
- Debug operasi klip non-kotak: Menonaktifkan area kliping di kanvas untuk membuat area kanvas yang tidak biasa (non-kotak). Biasanya, area kliping mencegah gambar apa pun di luar batas area kliping lingkaran.
- Paksa 4x MSAA: Mengaktifkan multisample anti-aliasing (MSAA) dalam aplikasi Open GL ES 2.0.
- Nonaktifkan overlay HW: Menggunakan overlay hardware memungkinkan setiap aplikasi yang menampilkan sesuatu pada layar untuk menggunakan daya pemrosesan yang lebih hemat. Tanpa overlay, aplikasi akan berbagi memori video serta harus selalu memeriksa bentrokan dan kliping untuk merender gambar yang tepat. Pemeriksaan ini menggunakan banyak daya pemrosesan.

## Media

Aktifkan Perutean audio USB nonaktif untuk menonaktifkan perutean otomatis ke perangkat audio eksternal yang terhubung ke komputer melalui port USB. Perutean otomatis dapat mengganggu aplikasi yang peka terhadap USB.

Di Android 11 dan yang lebih baru, saat aplikasi tanpa izin `RECORD_AUDIO` menggunakan `UsbManager` untuk meminta akses langsung ke perangkat audio USB dengan kemampuan perekaman audio (seperti headset USB), pesan peringatan yang meminta pengguna untuk mengonfirmasi izin penggunaan perangkat akan ditampilkan. Sistem akan mengabaikan opsi "selalu gunakan", sehingga pengguna harus mengonfirmasi peringatan dan memberikan izin setiap kali aplikasi meminta akses. Untuk menghindari perilaku ini, aplikasi Anda harus meminta izin `RECORD_AUDIO`.

---

## Pemantauan

Opsi pemantauan memberikan informasi visual tentang performa aplikasi, seperti thread yang panjang dan operasi GPU. Ketuk Profile GPU Rendering, lalu On screen as bars untuk menampilkan profil rendering GPU dalam bentuk batang. Untuk mengetahui informasi selengkapnya, lihat Rendering GPU profil.

## Aplikasi

Opsi aplikasi membantu Anda memahami bagaimana aplikasi beroperasi pada perangkat target. Ketuk Background process limit untuk menyetel jumlah proses yang dapat berjalan di latar belakang pada satu waktu.

Ketuk Reset ShortcutManager rate-limiting selama pengujian, sehingga aplikasi latar belakang dapat terus memanggil API pintasan hingga batas kapasitas tercapai lagi. Untuk mengetahui informasi selengkapnya tentang pintasan dan batas kapasitas, lihat ShortcutManager.

Aktifkan Jangan simpan aktivitas untuk meningkatkan masa pakai baterai dengan segera menutup setiap aktivitas setelah pengguna meninggalkan tampilan utama aktivitas.

# Menjalankan Aplikasi di Perangkat Hardware

Ketika membuat aplikasi Android, Anda harus menguji aplikasi tersebut pada perangkat sebenarnya sebelum merilisnya ke pengguna. Halaman ini menjelaskan cara menyiapkan lingkungan pengembangan dan perangkat Android untuk menjalankan pengujian dan proses debug melalui koneksi *Android Debug Bridge* (ADB).

*Catatan: Gunakan Android Emulator untuk menguji aplikasi Anda pada berbagai versi platform Android dan ukuran layar. Pertimbangkan juga untuk menggunakan Firebase Test Lab untuk menjalankan aplikasi Anda di berbagai perangkat sebenarnya yang dihosting di infrastruktur berbasis cloud.*

## 1. Menyiapkan perangkat untuk pengembangan

Sebelum dapat memulai proses debug di perangkat, tentukan apakah Anda ingin terhubung ke perangkat menggunakan kabel USB atau Wi-Fi. Lalu, lakukan langkah berikut:

- 1) Di perangkat, buka aplikasi Setelan, pilih Opsi developer, lalu aktifkan Proses debug USB (jika ada).
- 2) Siapkan sistem agar mendeteksi perangkat Anda.
  - Chrome OS: Konfigurasi tambahan tidak diperlukan.
  - macOS: Konfigurasi tambahan tidak diperlukan.
  - Ubuntu Linux: Ada dua hal yang harus disiapkan dengan benar: setiap pengguna yang ingin menggunakan adb harus termasuk dalam grup plugdev, dan sistem harus menginstal aturan udev yang mencakup perangkat.
  - Windows: Instal driver USB untuk ADB (jika ada). Untuk panduan penginstalan dan link ke driver OEM, lihat dokumen Menginstal driver USB OEM.

## 2. Menghubungkan ke perangkat menggunakan USB

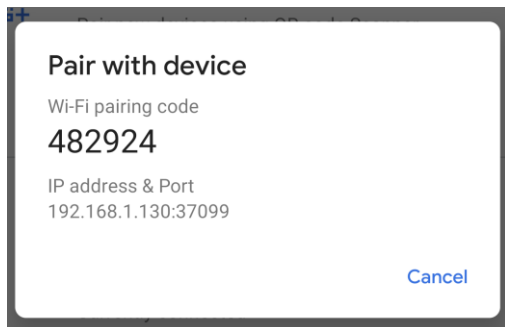
Setelah perangkat Anda siap dan terhubung melalui USB, klik Run pada Android Studio untuk membuat dan menjalankan aplikasi Anda di perangkat. Anda juga dapat menggunakan adb untuk menjalankan perintah, sebagai berikut:

- Pastikan bahwa perangkat Anda terhubung dengan menjalankan perintah `adb devices` dari direktori `android_sdk/platform-tools/` Anda. Jika sudah terhubung, perangkat Anda akan tercantum pada daftar.
- Jalankan perintah `adb` dengan flag `-d` untuk menargetkan perangkat Anda.

### 3. Menghubungkan ke perangkat menggunakan Wi-Fi

Android 11 (dan yang lebih baru) mendukung deployment dan proses debug aplikasi secara nirkabel dari komputer melalui Android Debug Bridge (adb). Misalnya, Anda dapat men-deploy aplikasi yang dapat di-debug ke beberapa perangkat jarak jauh, tanpa harus menghubungkan perangkat secara langsung melalui USB dan mengalami masalah koneksi USB umum, seperti penginstalan driver.

Untuk menggunakan proses debug nirkabel, Anda harus menyambungkan perangkat ke komputer menggunakan kode penghubung. Komputer dan perangkat Anda harus terhubung ke jaringan nirkabel yang sama. Untuk menghubungkan perangkat, ikuti langkah-langkah berikut:



- 1) Di komputer Anda, update **SDK Platform-Tools** ke versi terbaru.
- 2) Aktifkan opsi Proses debug nirkabel di bagian Opsi developer.
- 3) Pada dialog yang menanyakan Izinkan proses debug nirkabel di jaringan ini?, klik Izinkan.
- 4) Pilih Sambungkan perangkat dengan kode penghubung. Catat kode penghubung, alamat IP, dan nomor port yang ditampilkan di perangkat (lihat gambar).
- 5) Di komputer, buka terminal dan buka **`android_sdk/platform-tools`**.
- 6) Jalankan **`adb pair ipaddr:port`**. Gunakan alamat IP dan nomor port dari langkah 3.
- 7) Saat diminta, masukkan kode penghubung yang Anda terima pada langkah 4. Akan muncul pesan yang menunjukkan bahwa perangkat Anda berhasil disambungkan.

- 8) Jalankan ***adb connect ipaddr:port***. Gunakan alamat IP dan port pada Proses debug nirkabel (lihat gambar di bawah).

#### 4. Memecahkan masalah koneksi perangkat dengan Connection Assistant

Connection Assistant menyediakan petunjuk langkah demi langkah untuk membantu Anda menyiapkan dan menggunakan perangkat melalui koneksi ADB. Untuk memulai asisten ini, pilih Tools > Connection Assistant.

Connection Assistant menyediakan petunjuk, kontrol sesuai konteks, dan daftar perangkat yang terhubung dalam serangkaian halaman di panel Assistant. Gunakan tombol Next dan Previous di bagian bawah panel Assistant untuk mengerjakan halaman yang diperlukan:

- Hubungkan perangkat Anda melalui USB: Connection Assistant memulai dengan meminta Anda menghubungkan perangkat melalui USB, dan menyediakan tombol Rescan USB devices untuk memulai pemindaian baru dan menemukan perangkat yang terhubung.
- Aktifkan proses debug USB: Selanjutnya, Connection Assistant memberitahu Anda cara mengaktifkan proses debug USB pada opsi developer di perangkat.
- Mulai ulang server ADB: Terakhir, jika perangkat Anda tetap tidak tercantum pada daftar perangkat yang tersedia, Anda dapat menggunakan tombol Restart ADB server di halaman terakhir Connection Assistant. Memulai ulang server ADB juga menyebabkan ADB menjalankan pemindaian untuk menemukan perangkat lagi. Jika perangkat Anda tetap tidak tercantum pada daftar perangkat yang tersedia, coba langkah pemecahan masalah di bagian selanjutnya halaman ini.

#### 5. Kunci keamanan RSA

Jika Anda menghubungkan perangkat yang menjalankan Android 4.2.2 (API level 17) atau yang lebih baru ke komputer, sistem akan menampilkan dialog yang menanyakan apakah Anda bersedia menerima kunci RSA yang memungkinkan proses debug melalui komputer ini. Mekanisme keamanan ini melindungi perangkat pengguna karena memastikan bahwa proses debug USB dan perintah adb lainnya tidak dapat dijalankan kecuali jika Anda dapat membuka kunci perangkat dan mengkonfirmasi dialog.



# Informasi Tambahan

Untuk mengakses informasi mengenai layout (tata letak) di Android, Constraint Layout, RecyclerView, Kotlin, Retrofit, dan Material Design dapat diakses pada situs berikut.

- Icon Resource Dice Roll  
[http://bit.ly/mulaiandroid\\_diceicons](http://bit.ly/mulaiandroid_diceicons)
- Layout (Tata Letak) di Android  
<https://developer.android.com/guide/topics/ui/declaring-layout?hl=id>
- Constraint Layout  
<https://developer.android.com/training/constraint-layout?hl=id>
- RecyclerView  
<https://developer.android.com/guide/topics/ui/layout/recyclerview?hl=id>
- Menu (Navigation Component)  
<https://developer.android.com/guide/topics/ui/menus?hl=id>
- Kotlin untuk Development Android  
<https://developer.android.com/kotlin>
- Kotlin Class  
<https://kotlinlang.org/docs/classes.html>
- Kotlin Class Inheritance  
<https://kotlinlang.org/docs/inheritance.html>
- Retrofit  
<https://square.github.io/retrofit/>
- Material Design  
<https://material.io/>
- Material Color Tool  
<https://material.io/resources/color>

## Referensi

<https://www.dicoding.com/blog/apa-itu-kotlin-kenapa-kita-harus-mempelajari-kotlin/>

<https://developer.android.com/studio/debug/dev-options?hl=id>

<https://developer.android.com/studio/run/device?hl=id>