

PRAKTIKUM PEMROGRAMAN MOBILE
Pengenalan Aplikasi Mobile
Pertemuan Ke-1



Disusun Oleh :

NAMA : MARIA DEFLORA MEO

NIM : 195410088

KELAS : TI 2

PRODI : INFORMATIKA

A. TUJUAN

1. Mahasiswa diharapkan dapat memahami dan mengetahui tentang aplikasi mobile, cara penginstallan Android Studio, debug dengan virtual device dan debug dengan physical device.
2. Mahasiswa Mampu memahami tentang Pengembangan Aplikasi Android

B. DASAR TEORI

1. Pendahuluan.

Selamat datang di Praktikum Pemrograman Mobile. Kita akan menggunakan Android Kotlin untuk praktikum ini. Praktikum ini menggunakan codelab yang menuntun kita membangun aplikasi Android menggunakan Kotlin. Prasyarat untuk dapat mengikuti praktikum ini dengan baik adalah memiliki pengetahuan dalam bahasa pemrograman berorientasi objek penuh seperti Java, C ++. Untuk pengembangan, disarankan untuk menggunakan sumber referensi selain modul praktikum ini. Kita juga harus terbiasa menavigasi GitHub dan terbiasa dengan konsep-konsep: Algoritma pemrograman, penanganan error, bagaimana kode dibangun, dikompilasi, dan dieksekusi, secara umum.

Kita akan menggunakan sumber referensi utama pada link android developer dan codelab berikut:

1. <https://kotlinlang.org/docs/reference/> untuk belajar dasar pemrograman kotlin.
2. <https://developer.android.com/kotlin> untuk pemahaman kenapa android menggunakan kotlin.
3. <https://developer.android.com/courses/kotlin-android-fundamentals/toc> untuk codelab review untuk dasar pemrograman kotlin.
4. <https://codelabs.developers.google.com/android-kotlin-fundamentals/> untuk belajar koding project kotlin.
5. <https://developer.android.com/kotlin/learn> untuk belajar tentang pemrograman kotlin untuk android.

2. Android dan Pemrograman Mobile.

Android adalah sistem operasi mobile yang open source. Tahun 2005, Google mengakusisi perusahaan Android Inc. untuk memulai mengembangkan platform Android. Tahun 2007, sekelompok pemimpin industri datang bersama membentuk Open Handset Alliance (<http://www.openhandsetalliance.com>). November 2007, Android SDK dirilis pertama kali dengan “tampilan awal”

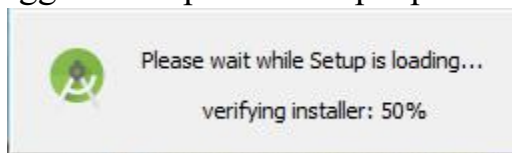
(early look). September 2008, T-Mobile mengumumkan ketersediaan HTC Dream G1, smartphone pertama yang berbasis platform Android. •Beberapa hari berikutnya Google mengumumkan ketersediaan Android SDK Release Candidate 1.0. Oktober 2008, Google membuat kode program dari platform Android tersedia di bawah "Apache's open source license".

C. PRAKTIK

Install Android Studio

1. Android studio didapatkan melalui website resmi <https://developer.android.com/studio>, jika Android Studio telah didownload maka anda dapat langsung menginstall dengan cara, klik 2x pada android-studio-ide183.5452501-windows.exe

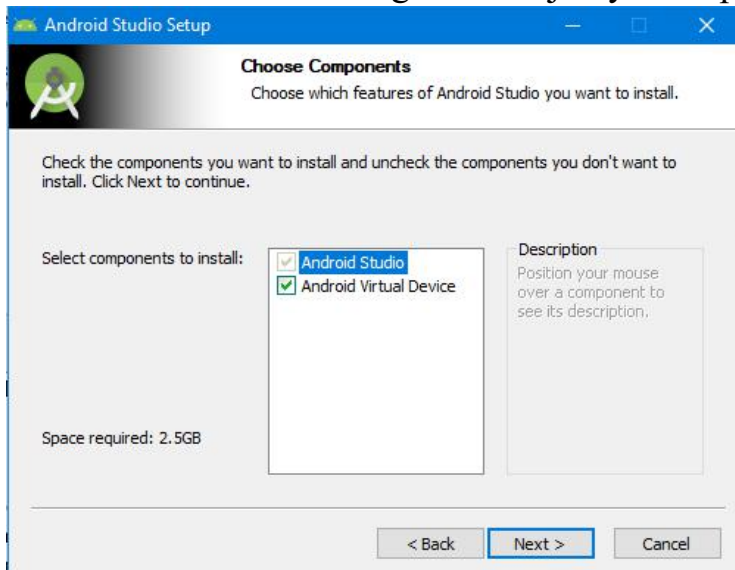
2. Tunggu beberapa saat sampai proses loading verifying installer selesai.



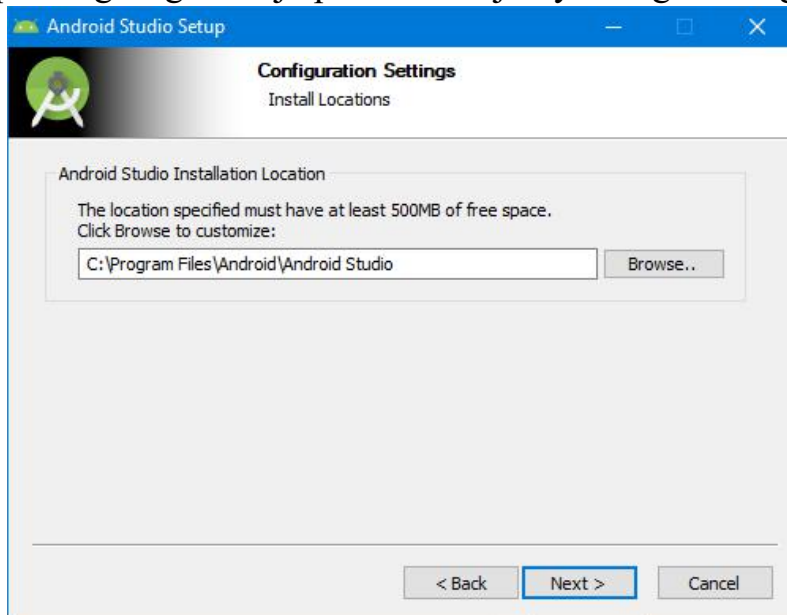
3. Setelah proses loading verifying installer selesai, akan terlihat jendela Android Studio Setup, klik button Next > untuk melanjutkan proses penginstallan.



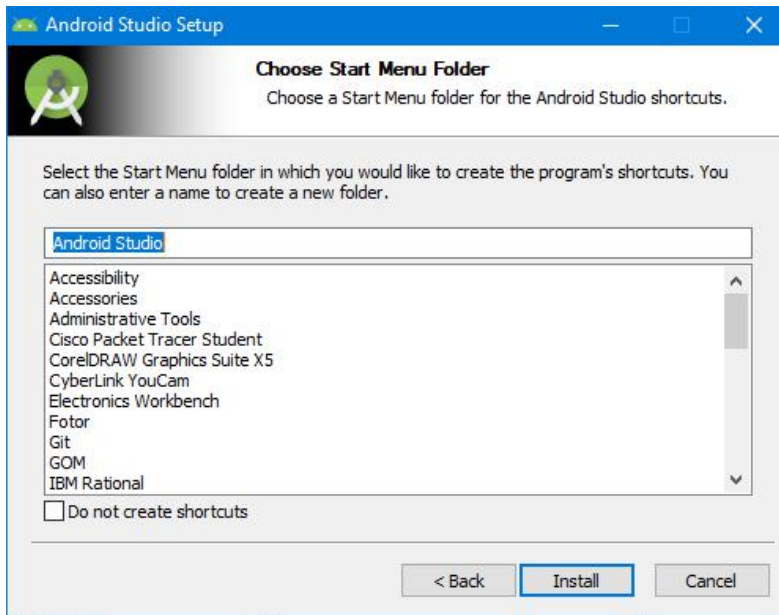
4. Beri tanda checklist atau centang bagian Android Virtual Device. Fungsi dari komponen Android Virtual Device yaitu untuk menampilkan interface android dalam bentuk virtual. Langkah selanjutnya klik pada button Next >



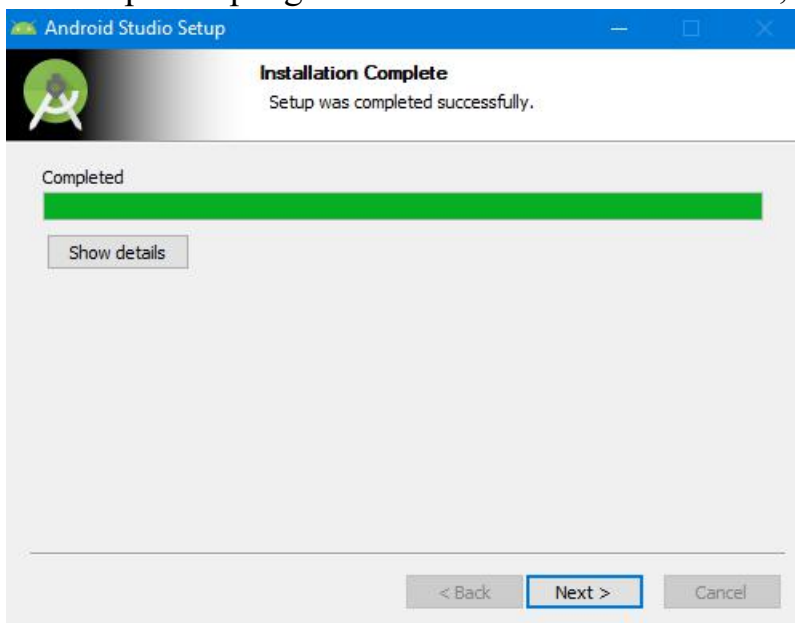
5. Pada bagian Configuration Settings akan diminta untuk memilih lokasi penginstallan, secara default lokasi penginstallan pada direktori C:\Program Files\Android\Android Studio. Jika ingin menggunakan lokasi default maka dapat langsung menuju proses selanjutnya dengan meng-klik button Next >



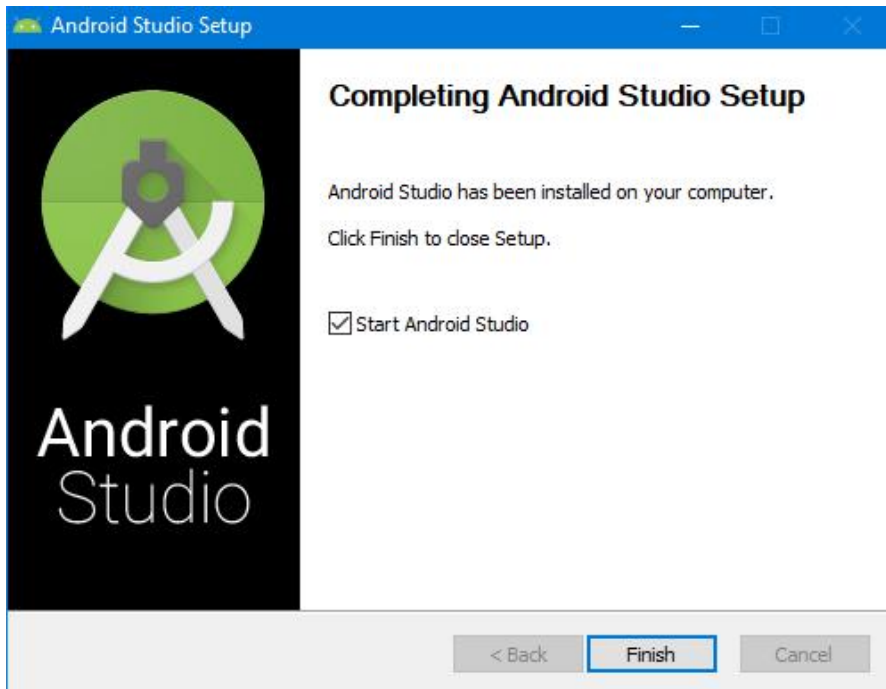
6. Pada bagian Choose Start Menu Folder, klik button Install



7. Tunggulah beberapa saat sampai proses penginstallan Android Studio selesai. Setelah proses penginstallan Android Studio selesai, klik button Next >



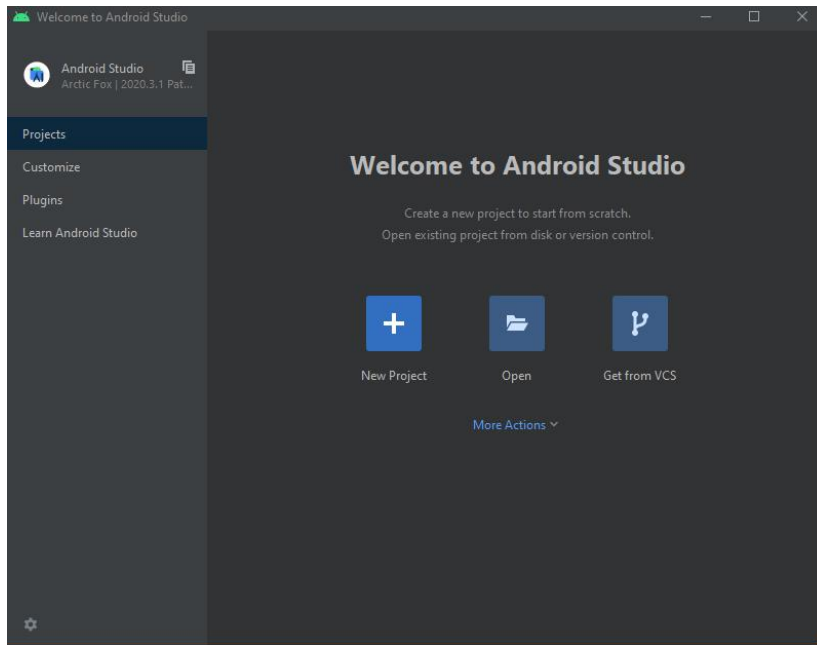
8. Proses penginstallan Android Studio telah selesai, klik button Finish untuk menutup jendela Android Studio Setup. Jika ingin menjalankan langsung program Android Studio maka beri tanda checklist atau centang pada bagian Start Android Studio sebelum menekan button Finish.



Running Program Android Studio

Dalam menjalankan program program Android Studio sama halnya dalam menjalankan program-program lainnya, agar mudah dalam memahami cara menjalankan Android Studio dapat dengan mengikuti langkah-langkah berikut:

1. Buka Android Studio dari komputer yang dipakai.
2. Maka akan muncul jendela Welcome to Android Studio.

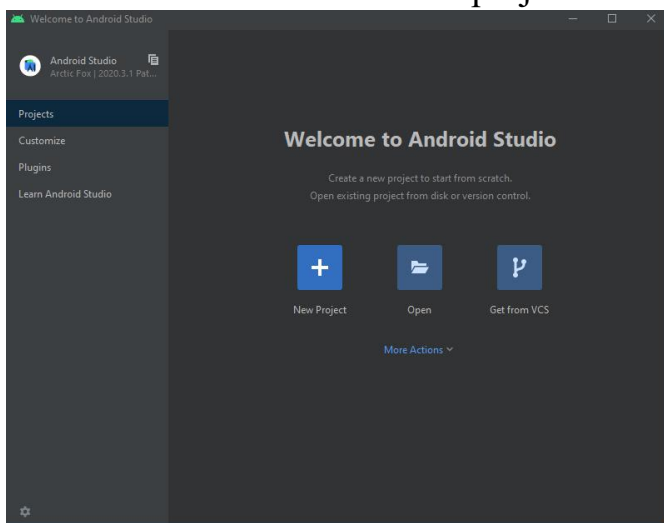


Membuat Project Dengan Android Studio

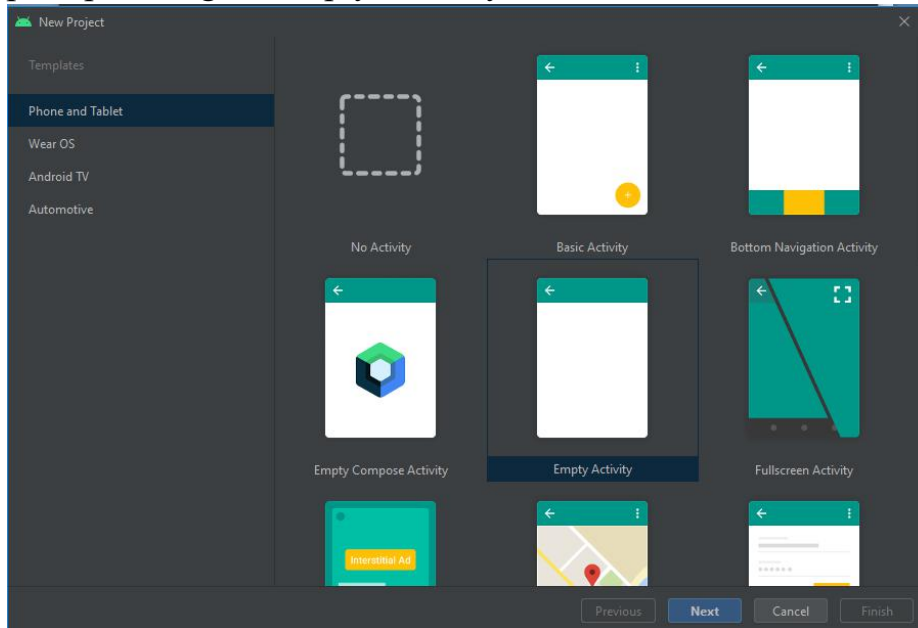
Pembuatan project dengan Android Studio dapat dilakukan dengan langkah-langkah

sebagai berikut :

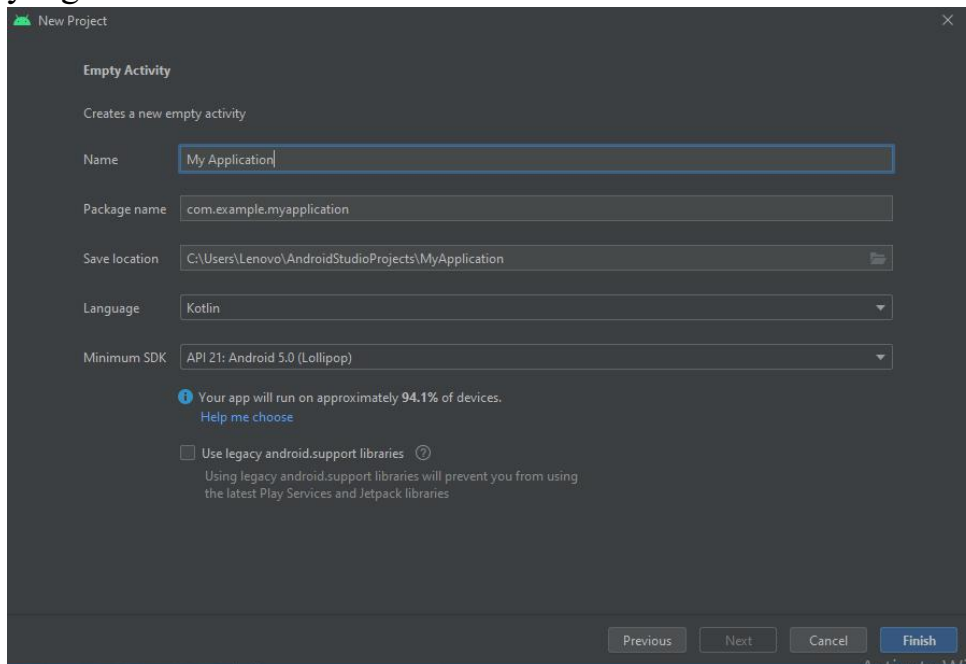
1. Dalam membuat project awal dengan Android Studio dengan cara klik pada
+
Start a new Android Studio project



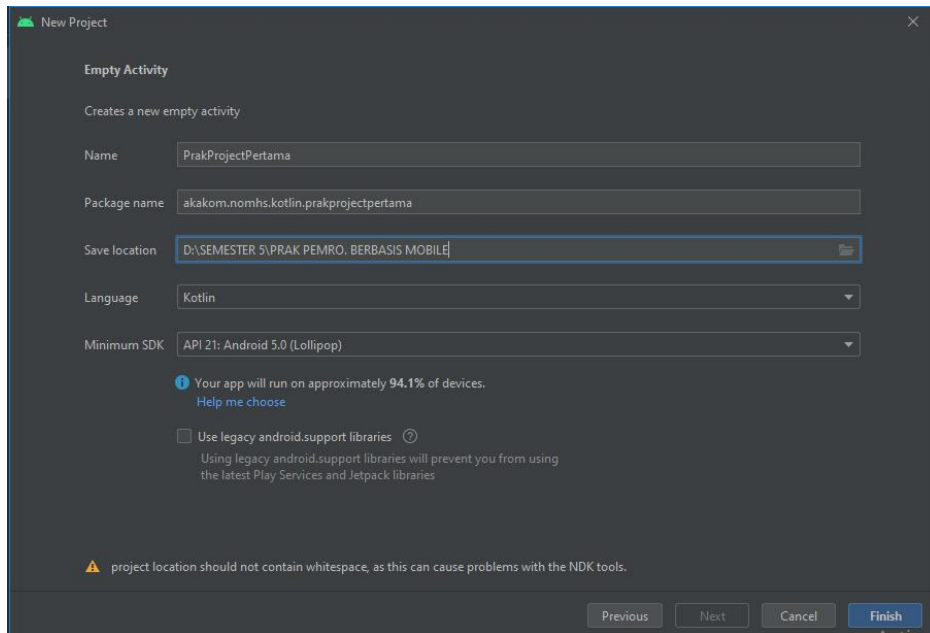
2. Kemudian akan terlihat jendela Create New Project, pada bagian ini programmer dapat memilih bentuk dari project yang akan dibuat (hal ini menyesuaikan dengan project yang akan dibuat). Sebagai latihan awal maka pilih pada bagian Empty Activity, kemudian klik button Next.



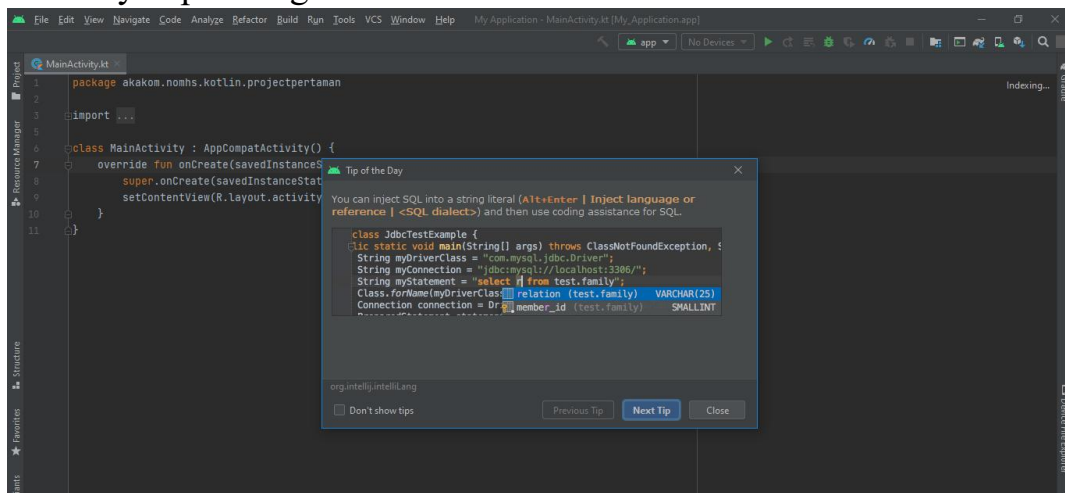
3. Pada jendela Create New Project, maka secara default Name project yaitu My Application, penamaan ini dapat dirubah sesuai dengan nama project yang dibuat.



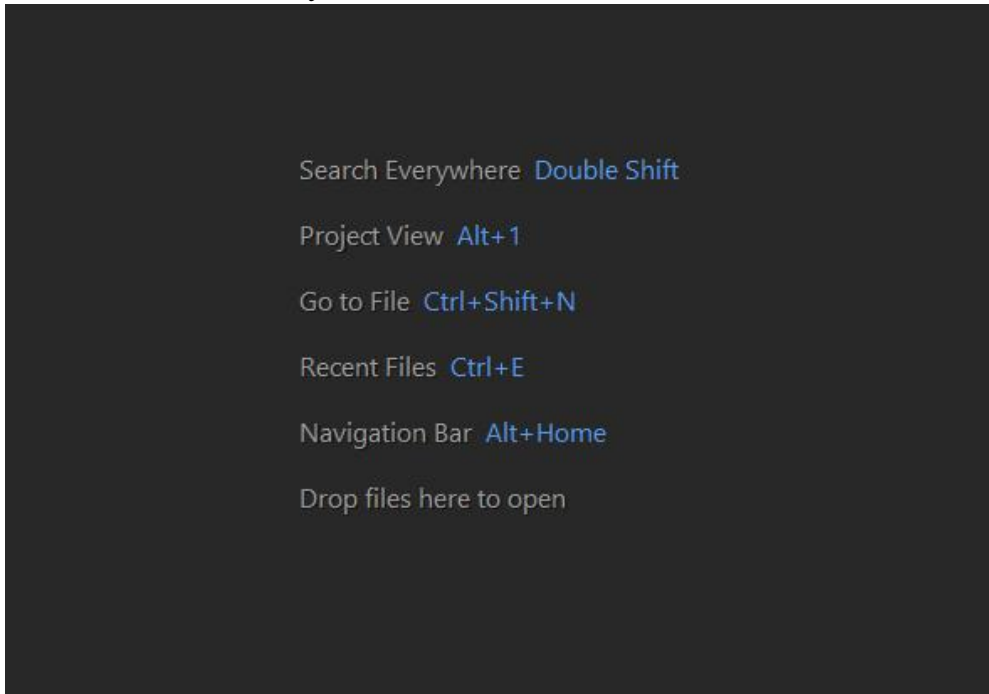
4. Sebagai bahan latihan, beri nama pada bagian Name : ProjectPertama kemudian pada bagian Save location : (gunakan folder kerja anda), kemudian klik button Finish.



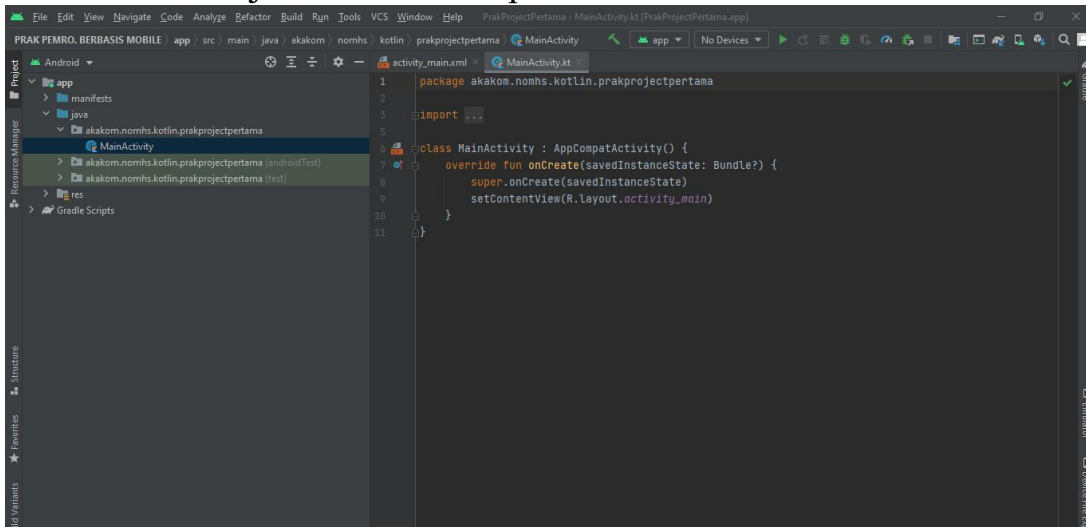
5. Pada saat awal atau pertamakali membuat project maka akan muncul sebuah jendela dengan nama Tips of The Day yaitu informasi tentang pemrograman yang dapat dijadikan acuan para programmer, untuk menutup jendela Tips of The Day dapat dengan cara klik button Close.



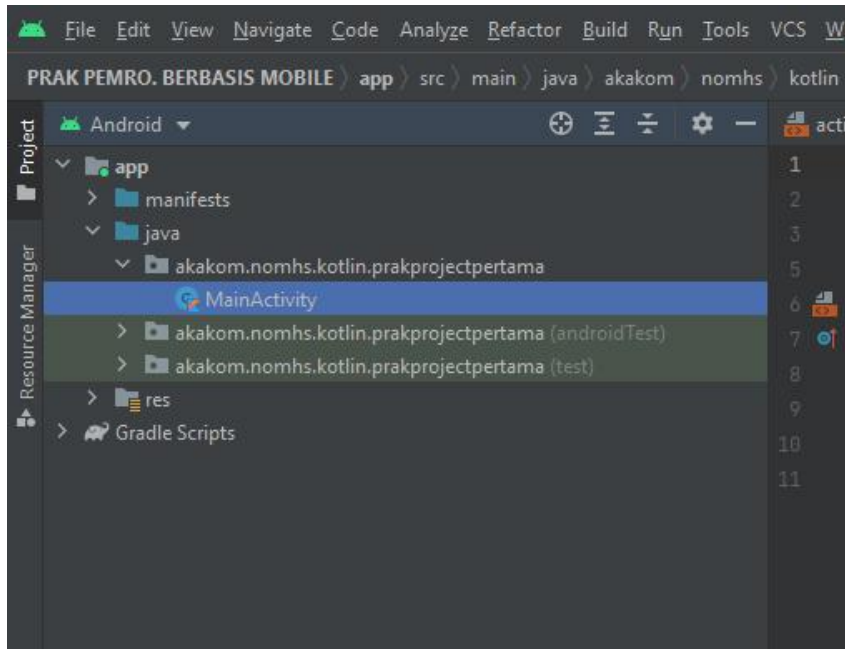
6. Jendela ProjectPertama akan muncul seperti pada gambar dibawah ini, dan tentunya belum secara utuh. Tunggulah beberapa saat sampai proses loading fitur-fitur atau library dan tools muncul.



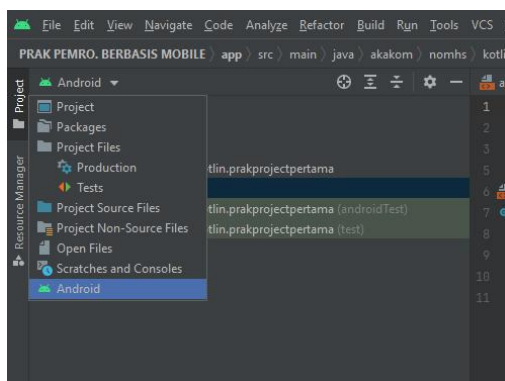
7. Jika proses loading fitur-fitur atau library selesai dan tools telah muncul, maka akan terlihat pada jendela IDE seperti pada gambar dibawah ini. Terdapat 3 bagian jendela pada Android Studio yaitu jendela Project, jendela Sourcecode dan jendela Built Output.

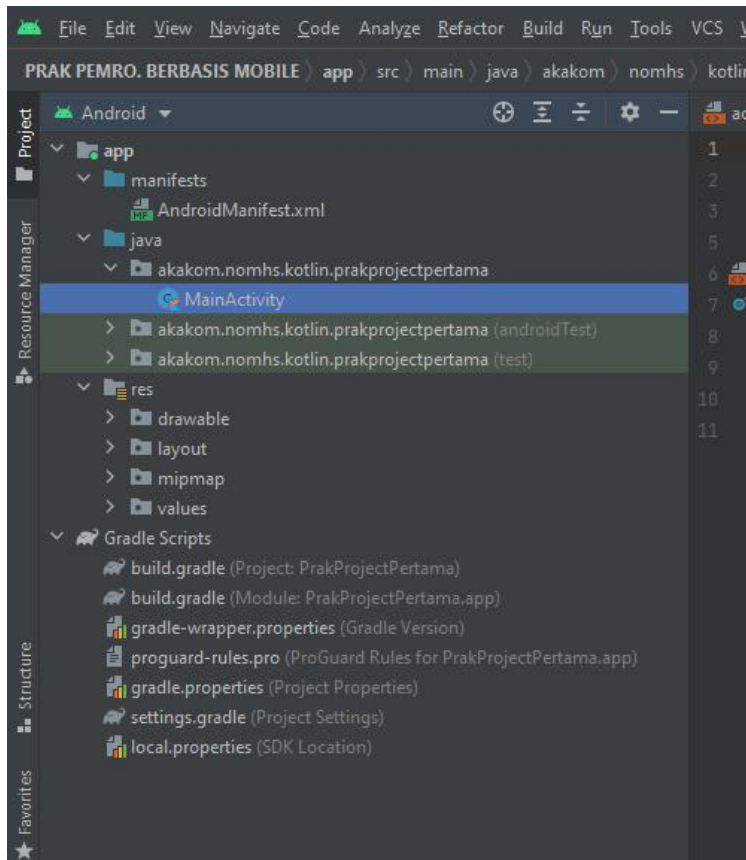


8. Pada jendela kiri atas, jika tab Project belum dipilih, klik tab Project.

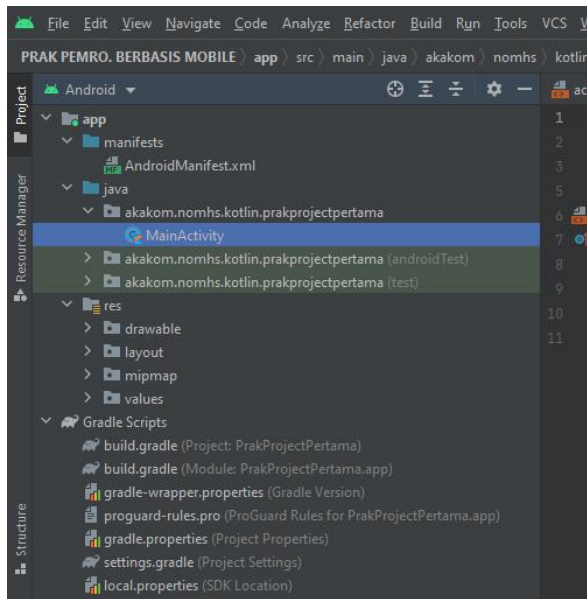


9. Hirarki Project dapat dipilih dengan menggunakan drop down menu. Hirarki standar adalah Android.





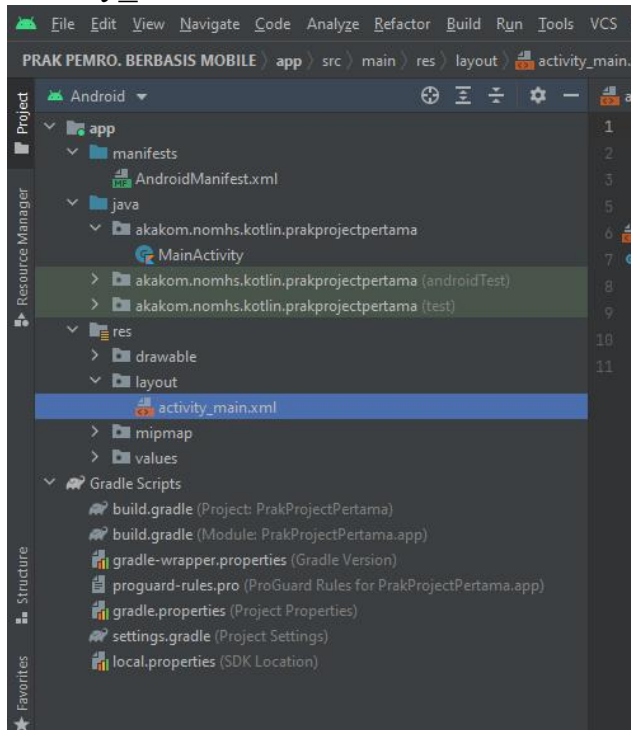
- 10.. Untuk selanjutnya kita akan menggunakan hirarki Android.
- 11.Sekarang akan kita lihat panel Project. Yang pertama kita mengeksplorasi folder app.
- 12.Di panel Project> Android, eksplora folder app. Di dalam folder app ada empat subfolder: manifest, java, generateJava, dan res.
13. Buka folder java, dan kemudian ekspans folder akakom.nomhs.android.ProjectPertama untuk melihat file MainActivity Kotlin.



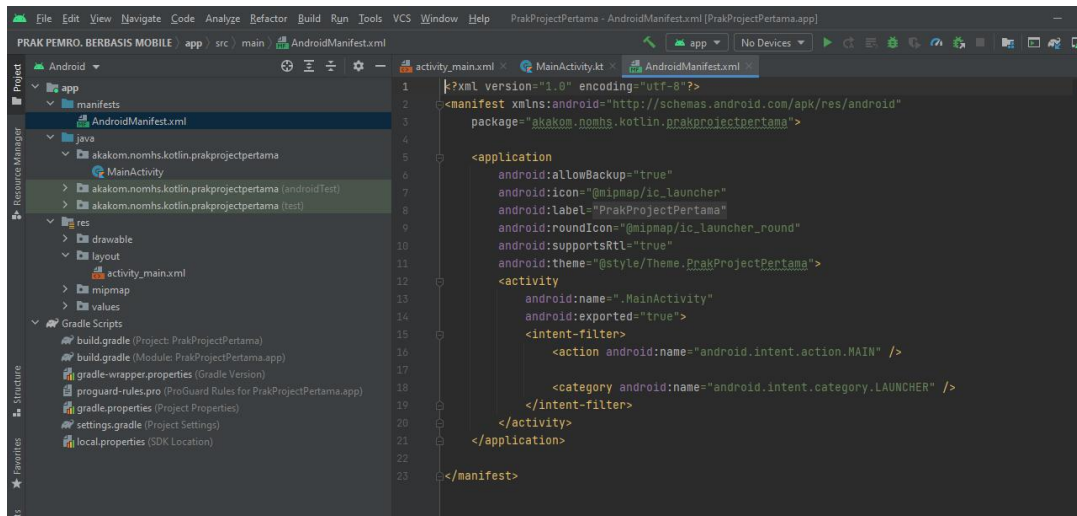
14. Folder java berisi semua kode Kotlin utama untuk aplikasi Android. Ada alasan historis mengapa kode Kotlin Anda muncul di folder java. Konvensi itu memungkinkan Kotlin untuk beroperasi tanpa hambatan dengan kode yang ditulis dalam bahasa pemrograman Java, bahkan dalam proyek dan aplikasi yang sama.
15. File kelas aplikasi kita terkandung dalam tiga subfolder, seperti yang ditunjukkan pada gambar di atas. Folder `akakom.nomhs.android.ProjectPertama` berisi semua file untuk paket app. Secara khusus, kelas `MainActivity` adalah titik masuk utama untuk app kita. Dua folder lain di folder java digunakan untuk kode yang terkait dengan pengujian, seperti tes unit.
16. Dalam sistem file, file Kotlin memiliki ekstensi `.kt` dan ikon K. Pada tampilan Proyek, Android Studio menunjukkan nama kelas (`MainActivity`) tanpa ekstensi.
17. Catat folder `generatedJava`. Folder ini berisi file yang dihasilkan Android Studio saat membangun aplikasi. Jangan edit apa pun di folder ini, karena perubahan yang dilakukan, mungkin ditimpa ketika kita membangun kembali app. Tetapi penting untuk mengetahui tentang folder ini ketika kita perlu melihat file-file ini selama debugging.
18. Kemudian kita akan melihat folder berikutnya, yaitu `res`. Di panel `Project > Android`, expand folder `res`.
19. Folder `res` menyimpan sumber daya. Sumber daya di Android adalah konten statis yang digunakan dalam aplikasi kita. Sumber daya termasuk gambar,

string teks, tata letak layar, gaya, dan nilai-nilai seperti warna heksadesimal atau dimensi standar.

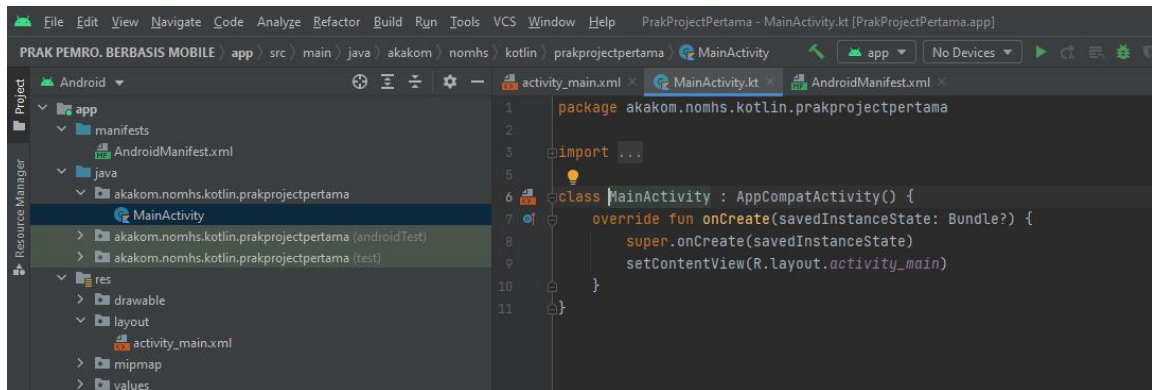
20. Aplikasi Android memisahkan kode dan sumber daya Kotlin sebanyak mungkin. Itu membuatnya lebih mudah untuk menemukan semua string atau ikon yang digunakan di UI app. Juga, ketika kita mengubah salah satu file sumber daya ini, perubahan itu berlaku di mana-mana file tersebut digunakan dalam aplikasi.
21. Di dalam folder res, expand folder layout untuk melihat file `activity_main.xml`.



22. Activity kita biasanya dikaitkan dengan file tata letak UI, yang didefinisikan sebagai file XML di direktori `res/layout`. File tata letak itu biasanya dinamai berdasarkan aktivitasnya. Dalam hal ini, nama aktivitas adalah `MainActivity`, jadi layout yang terkait adalah `activity_main`.
23. Sekarang kita akan menjelajahi folder manifest dan `AndroidManifest.xml`
24. Folder manifest berisi file yang memberikan informasi penting tentang app kita ke sistem Android.
25. Buka folder manifest dan klik dua kali `AndroidManifest.xml` untuk membukanya. File `AndroidManifest.xml` mencakup detail yang dibutuhkan sistem Android untuk menjalankan app kita, termasuk aktivitas apa yang merupakan bagian dari app.



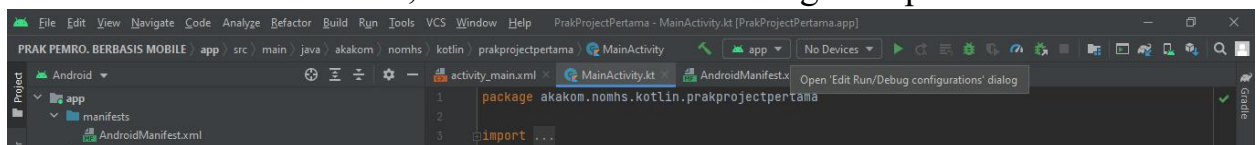
- 26.. Perhatikan bahwa MainActivity direferensikan di elemen . Aktivitas apa pun di app kita harus dinyatakan dalam manifest. Contoh manifest untuk MainActivity.
- 27.Catat elemen Elemen dan dalam filter maksud ini memberi tahu Android tempat memulai app ketika pengguna mengklik ikon run. 14
- 28.File AndroidManifest.xml juga merupakan tempat kita menentukan izin apa pun yang dibutuhkan aplikasi kita. Izin mencakup kemampuan aplikasi kita untuk membaca kontak telepon, mengirim data melalui internet, atau mengakses perangkat keras seperti kamera perangkat.
- 29.Terakhir, kita akan menjelajahi folder Script Gradle.
- 30.Gradle adalah sistem otomasi bangunan yang menggunakan bahasa khusus domain untuk menggambarkan struktur, konfigurasi, dan dependensi project app. Ketika kita mengkompilasi dan menjalankan aplikasi kita, kita melihat informasi tentang Gradle build running. kita juga melihat informasi tentang Android Package Kit (APK) yang diinstal. (APK adalah format file paket yang digunakan sistem operasi Android untuk mendistribusikan dan menginstal aplikasi seluler.)
- 31.Kita akan fokus pada dua bagian pada bagian script program pada project Android yaitu .kt dan .xml. Untuk .kt dapat dibaca kotlin yaitu bahasa pemrograman pada Android Studio untuk membangun sistem. Sedangkan .xml dapat dibaca Extensible Markup Language yaitu bahasa pemrograman pada Android Studio untuk membangun user interface. Klik pada bagian activity_main.xml untuk melihat design atau atampilan project ProjectPertama.



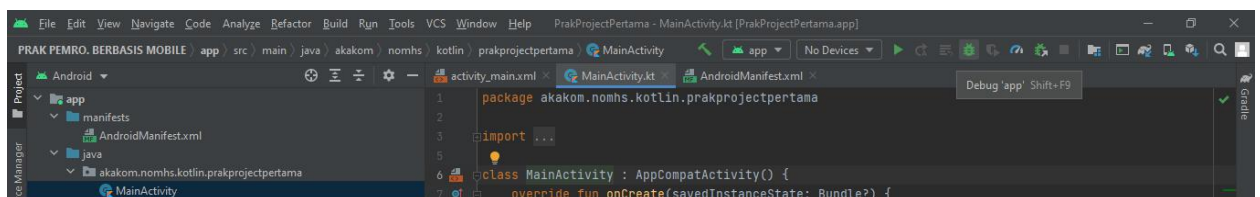
Pengenalan Aplikasi Android dan IDE Android Studio

Untuk dapat membuat aplikasi mobile dengan Android Studio maka perlu mengenali lingkungan Aplikasi Android dan IDE Android Studio. Berikut beberapa IDE pada Android Studio yang nantinya akan digunakan selama proses pembuatan project :

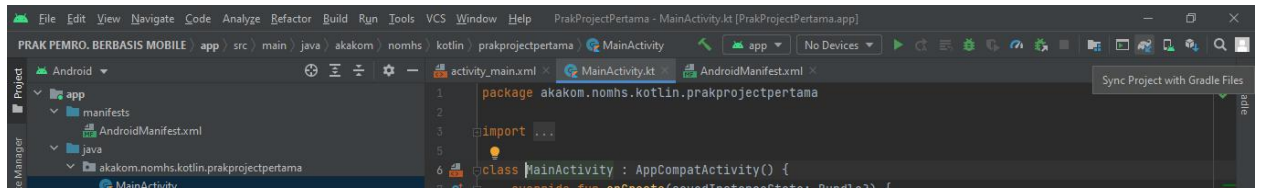
1. Open edit Run/Debug configurations dialog Bagian ini berfungsi untuk memilih atau membuka, edit dan melakukan debug/run aplikasi



2. Run up
Pada bagian ini berfungsi untuk menjalankan aplikasi mobile yang telah dibuat.

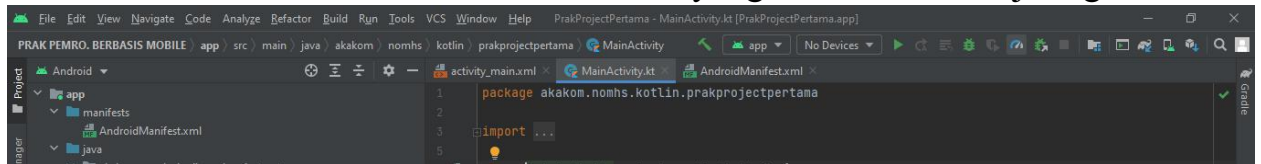


3. Debug App
Debug App berfungsi untuk melakukan compile script dan melakukan pengecekan apakah terdapat script yang error atau tidak, jika script tidak mengalami error maka Android Studio akan membuat APK (Application Package File).



4. Profile App

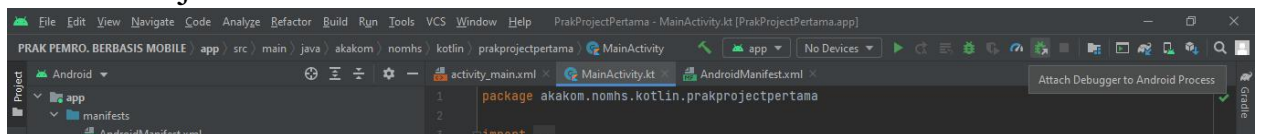
Profile App merupakan alat pembuatan profil baru yang menyediakan data realtime untuk CPU, memori, dan aktivitas jaringan aplikasi Anda. Programmer dapat melakukan pelacakan metode berbasis sampel untuk mengukur waktu eksekusi script, merekam heap-dump, menampilkan alokasi memori, dan memeriksa detail file yang dikirim melalui jaringan.



Dengan debugger Android Studio, maka programmer dapat :

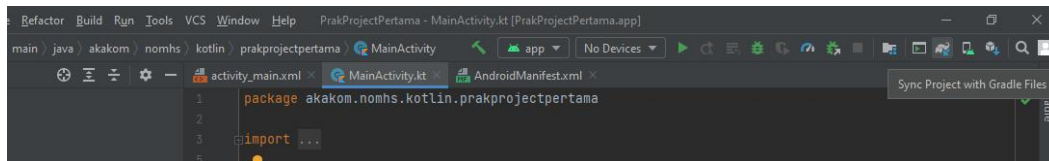
- Memilih perangkat untuk men-debug pada aplikasi yang dibuat.
- Menyetel breakpoint dalam kode Java dan C/C++ pada aplikasi yang dibuat.
- Memeriksa variabel dan mengevaluasi ekspresi pada saat waktu proses.
- Mengambil tangkapan layar dan video dari aplikasi yang dibuat.

5. Attach Debugger to Android Process Fungsi dari Attach Debugger to Android Procces yaitu melakukan debugging dan running proses dari script yang telah diubah atau update programmer dengan syarat emulator sudah dijalankan.

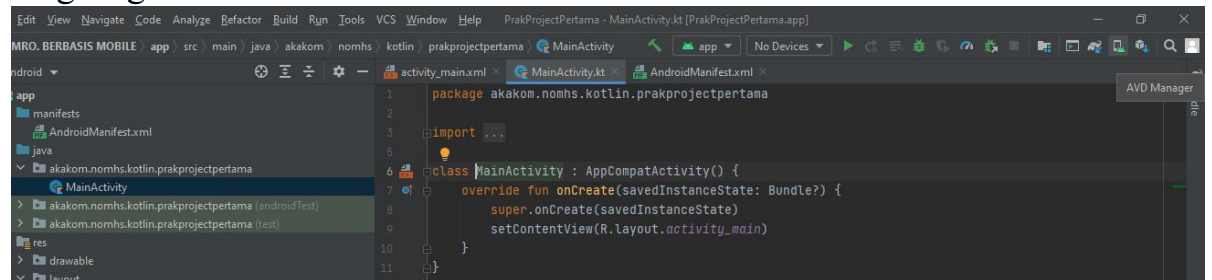


6. Sync Project with Gradle Files Jika programmer membuat perubahan pada file konfigurasi build dalam project Android yang dibuat, maka Android Studio mewajibkan programmer melakukan sinkronisasi file project sehingga sistem dapat mengimpor perubahan konfigurasi 16 build

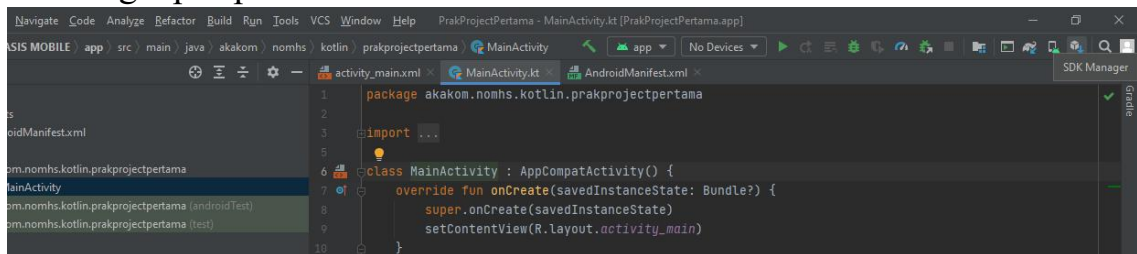
dan menjalankan beberapa pemeriksaan untuk memastikan konfigurasi tidak akan menimbulkan kesalahan build.



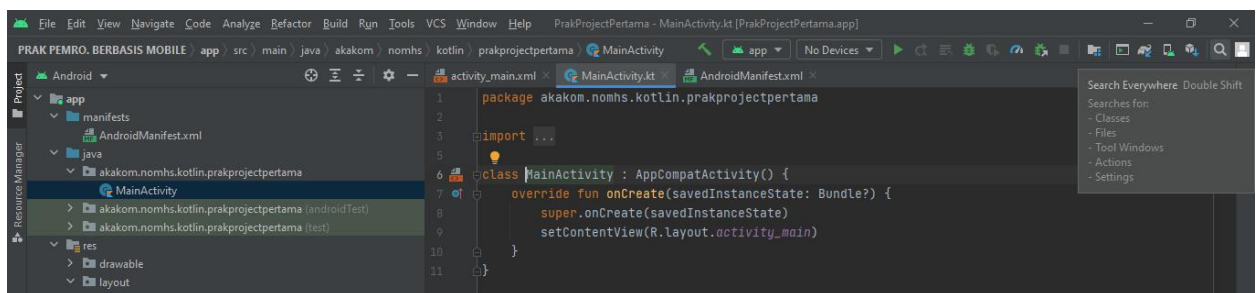
7. AVD Manager AVD atau Android Virtual Device adalah fitur pada Android Studio untuk membuat device dalam bentuk virtual sehingga ketika aplikasi atau project dirunning maka akan muncul di virtual device tersebut. Namun AVD juga dapat untuk membuat physical device yaitu melakukan running aplikasi atau project melalui smartphone secara langsung.



8. SDK Manager SDK Manger adalah fitur untuk melihat, menginstal, memperbarui, dan menghapus paket untuk Android SDK



9. Search Everywhere Search Everywhere adalah sebuah fitur untuk menampilkan daftar Gradle Daemon aktif di Android Studio.



D. LATIHAN

Pelajari dan cobalah IDE Android Studio lebih lanjut dengan membuat project Android dasar.

E. TUGAS

Install Android Studio pada perangkat komputer anda masing-masing di rumah

F. KESIMPULAN

Mahasiswa diharapkan dapat memahami dan mengetahui tentang aplikasi mobile, cara penginstallan Android Studio, debug dengan virtual device dan debug dengan physical device dan mahasiswa mampu memahami tentang Pengembangan Aplikasi Android