



PETUNJUK PRAKTIKUM EDISI KURIKULUM OBE

PEMROGRAMAN MOBILE



Penyusun:

Herman Yuliansyah, S.T., M.Eng.

Ika Arfiani, S.T., M.Cs.

Guntur Maulana Zamroni, B.Sc., M.T

Dinan Yulianto, S.T., M.Eng

Muhammad Althoofandy Suprayogi, S.Kom.

2023

HAK CIPTA

PETUNJUK PRAKTIKUM NAMA PRAKTIKUM

Copyright© 2023,

Herman Yuliansyah, S.T., M.Eng.

Ika Arfiani, S.T., M.Cs.

Guntur Maulana Zamroni, B.Sc., M.T

Dinan Yulianto, S.T., M.Eng

Muhammad Althoofandy Suprayogi, S.Kom.

Hak Cipta dilindungi Undang-Undang

Dilarang mengutip, memperbanyak atau mengedarkan isi buku ini, baik sebagian maupun seluruhnya, dalam bentuk apapun, tanpa izin tertulis dari pemilik hak cipta dan penerbit.

Diterbitkan oleh:

Program Studi Informatika

Fakultas Teknologi Industri

Universitas Ahmad Dahlan

Jalan Ring Road Selatan, Tamanan, Banguntapan, Bantul Yogyakarta 55166

Penulis

: Herman Yuliansyah, S.T., M.Eng.

Ika Arfiani, S.T., M.Cs.

Guntur Maulana Zamroni, B.Sc., M.T

Dinan Yulianto, S.T., M.Eng

Muhammad Althoofandy Suprayogi, S.Kom.

Editor

: Laboratorium S1 Informatika, Universitas Ahmad Dahlan

Desain sampul

: Laboratorium S1 Informatika, Universitas Ahmad Dahlan

Tata letak

: Laboratorium S1 Informatika, Universitas Ahmad Dahlan

Ukuran/Halaman

: 21 x 29,7 cm / 133 halaman

Didistribusikan oleh:



Laboratorium S1 Informatika

Universitas Ahmad Dahlan

Jalan Ring Road Selatan, Tamanan, Banguntapan, Bantul Yogyakarta 55166

Indonesia

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan segala rahmat dan hidayah kepada penulis sehingga, modul praktikum Pemrograman Mobile berbasis kurikulum OBE ini dapat diselesaikan. Mata kuliah Pemrograman Mobile merupakan salah satu mata kuliah yang memberikan pemahaman dan panduan menguasai aplikasi teknologi Mobile.

Modul ini merupakan panduan praktikum matakuliah Pemrograman Mobile, modul disusun secara sekuensial, setiap pertemuan praktikum terdapat tugas yang hasilnya akan digunakan pada praktikum berikutnya. Sehingga mahasiswa harus selalu mengikuti praktikum dan menyelesaikan tugasnya serta memperkaya kreatifitas untuk menghasilkan karya yang baik.

Yogyakarta, 1 Agustus 2023

Penyusun

DAFTAR PENYUSUN

Herman Yuliansyah, S.T., M.Eng.

Foto Penyusun 1



Tuliskan biografi penyusun 1 disini

Ika Arfiani, S.T., M.Cs.

Foto Penyusun 2



Tuliskan biografi penyusun 1 disini

Guntur Maulana Zamroni, B.Sc., M.T

Foto Penyusun 3



Tuliskan biografi penyusun 2 disini

Dinan Yulianto, S.T., M.Eng

Foto Penyusun 4



Muhammad Althoofandy Suprayogi, S.Kom.

Foto Penyusun 5



HALAMAN REVISI

Yang bertanda tangan di bawah ini:

Nama : Ika Arfiani, S.T., M.Cs.

NIP/NIPM : 19870920 201606 011 1095511

Jabatan : Dosen Pengampu Mata Kuliah **Pemrograman Mobile**

Dengan ini menyatakan pelaksanaan Revisi Petunjuk Praktikum **Pemrograman Mobile** untuk Program Studi Informatika telah dilaksanakan dengan penjelasan sebagai berikut:

No	Keterangan Revisi	Tanggal Revisi	Nomor Modul
1	Menambahkan teori tentang struktur proyek di Android padan Praktikum 1 di bagian Teori Pendukung	25 Juli 2015	PP/018/V/R1
2	Menambahkan praktik penggunaan database eksternal pada Praktikum 10	12 Mei 2017	PP/018/V/R2
3	Memperbaharui format layout Modul Praktikum Pemrograman Mobile	15 Agustus 2019	PP/018/V/R3
4	Merubah modul ke template OBE & Merubah materi sesuai KKNi/ACM	4 Agustus 2021	PP/018/V/R4
5	Merubah materi praktikum 1-10 dan Bahasa pemrograman Kotlin	10 Agustus 2023	PP/018/V/R5

Yogyakarta, 1 Agustus 2023

Penyusun



Ika Arfiani, S.T., M.Cs.

NIPM. 19870920 201606 011 1095511

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Murein Miksa Mardhia, S.T., M.T.

NIPM : 19891019 201606 011 1236278

Jabatan : Kepala Laboratorium S1 Informatika

Menerangkan dengan sesungguhnya bahwa Petunjuk Praktikum ini telah direview dan akan digunakan untuk pelaksanaan praktikum di Semester Gasal Tahun Akademik 2023/2024 di Laboratorium Praktikum S1 Informatika, Program Studi Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

Yogyakarta, 1 Agustus 2023

Mengetahui,
Ketua Kelompok Keilmuan

Kepala Laboratorium S1 Informatika



Nama Ketua KK
NIP/NIY.

Murein Miksa Mardhia, S.T., M.T.
NIPM. 19891019 201606 011 1236278

VISI DAN MISI PRODI INFORMATIKA

VISI

Menjadi Program Studi Informatika yang diakui secara internasional dan unggul dalam bidang Informatika serta berbasis nilai-nilai Islam.

MISI

1. Menjalankan pendidikan sesuai dengan kompetensi bidang Informatika yang diakui nasional dan internasional
2. Meningkatkan penelitian dosen dan mahasiswa dalam bidang Informatika yang kreatif, inovatif dan tepat guna.
3. Meningkatkan kuantitas dan kualitas publikasi ilmiah tingkat nasional dan internasional
4. Melaksanakan dan meningkatkan kegiatan pengabdian masyarakat oleh dosen dan mahasiswa dalam bidang Informatika.
5. Menyelenggarakan aktivitas yang mendukung pengembangan program studi dengan melibatkan dosen dan mahasiswa.
6. Menyelenggarakan kerja sama dengan lembaga tingkat nasional dan internasional.
7. Menciptakan kehidupan Islami di lingkungan program studi.

TATA TERTIB LABORATORIUM S1 INFORMATIKA

DOSEN/KOORDINATOR PRAKTIKUM

1. Dosen harus hadir saat praktikum minimal 15 menit di awal kegiatan praktikum dan menandatangani presensi kehadiran praktikum.
2. Dosen membuat modul praktikum, soal seleksi asisten, pre-test, post-test, dan responsi dengan berkoordinasi dengan asisten dan pengampu mata praktikum.
3. Dosen berkoordinasi dengan koordinator asisten praktikum untuk evaluasi praktikum setiap minggu.
4. Dosen menandatangani surat kontrak asisten praktikum dan koordinator asisten praktikum.
5. Dosen yang tidak hadir pada slot praktikum tertentu tanpa pemberitahuan selama 2 minggu berturut-turut mendapat teguran dari Kepala Laboratorium, apabila masih berlanjut 2 minggu berikutnya maka Kepala Laboratorium berhak mengganti koordinator praktikum pada slot tersebut.

PRAKTIKAN

1. Praktikan harus hadir 15 menit sebelum kegiatan praktikum dimulai, dan dispensasi terlambat 15 menit dengan alasan yang jelas (kecuali asisten menentukan lain dan patokan jam adalah jam yang ada di Laboratorium, terlambat lebih dari 15 menit tidak boleh masuk praktikum & dianggap INHAL).
2. Praktikan yang tidak mengikuti praktikum dengan alasan apapun, wajib mengikuti INHAL, maksimal 4 kali praktikum dan jika lebih dari 4 kali maka praktikum dianggap GAGAL.
3. Praktikan harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Praktikan tidak boleh makan dan minum selama kegiatan praktikum berlangsung, harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di dalam laboratorium (tidak boleh membuang sampah sembarangan baik kertas, potongan kertas, bungkus permen baik di lantai karpet maupun di dalam ruang CPU).
5. Praktikan dilarang meninggalkan kegiatan praktikum tanpa seizin Asisten atau Laboran.
6. Praktikan harus meletakkan sepatu dan tas pada rak/loker yang telah disediakan.
7. Selama praktikum dilarang NGENET/NGE-GAME, kecuali mata praktikum yang membutuhkan atau menggunakan fasilitas Internet.
8. Praktikan dilarang melepas kabel jaringan atau kabel power praktikum tanpa sepengetahuan laboran
9. Praktikan harus memiliki FILE Petunjuk praktikum dan digunakan pada saat praktikum dan harus siap sebelum praktikum berlangsung.
10. Praktikan dilarang melakukan kecurangan seperti mencontek atau menyalin pekerjaan praktikan yang lain saat praktikum berlangsung atau post-test yang menjadi tugas praktikum.
11. Praktikan dilarang mengubah setting software/hardware komputer baik menambah atau mengurangi tanpa permintaan asisten atau laboran dan melakukan sesuatu yang dapat merugikan laboratorium atau praktikum lain.

12. Asisten, Koordinator Praktikum, Kepala laboratorium dan Laboran mempunyai hak untuk menegur, memperingatkan bahkan meminta praktikan keluar ruang praktikum apabila dirasa anda mengganggu praktikan lain atau tidak melaksanakan kegiatan praktikum sebagaimana mestinya dan atau tidak mematuhi aturan lab yang berlaku.
13. Pelanggaran terhadap salah satu atau lebih dari aturan diatas maka Nilai praktikum pada pertemuan tersebut dianggap 0 (NOL) dengan status INHAL.

ASISTEN PRAKTIKUM

1. Asisten harus hadir 15 Menit sebelum praktikum dimulai (konfirmasi ke koordinator bila mengalami keterlambatan atau berhalangan hadir).
2. Asisten yang tidak bisa hadir WAJIB mencari pengganti, dan melaporkan kepada Koordinator Asisten.
3. Asisten harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Asisten harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di laboratorium, menegur atau mengingatkan jika ada praktikan yang tidak dapat menjaga kebersihan, ketertiban atau kesopanan.
5. Asisten harus dapat merapikan dan mengamankan presensi praktikum, Kartu Nilai serta tertib dalam memasukan/Input nilai secara Online/Offline.
6. Asisten harus dapat bertindak secara profesional sebagai seorang asisten praktikum dan dapat menjadi teladan bagi praktikan.
7. Asisten harus dapat memberikan penjelasan/pemahaman yang dibutuhkan oleh praktikan berkenaan dengan materi praktikum yang diasistensi sehingga praktikan dapat melaksanakan dan mengerjakan tugas praktikum dengan baik dan jelas.
8. Asisten tidak diperkenankan mengobrol sendiri apalagi sampai membuat gaduh.
9. Asisten dimohon mengkoordinasikan untuk meminta praktikan agar mematikan komputer untuk jadwal terakhir dan sudah dilakukan penilaian terhadap hasil kerja praktikan.
10. Asisten wajib untuk mematikan LCD Projector dan komputer asisten/praktikan apabila tidak digunakan.
11. Asisten tidak diperkenankan menggunakan akses internet selain untuk kegiatan praktikum, seperti Youtube/Game/Medsos/Streaming Film di komputer praktikan.

LAIN-LAIN

1. Pada Saat Responsi Harus menggunakan Baju Kemeja untuk Laki-laki dan Perempuan untuk Praktikan dan Asisten.
2. Ketidakhadiran praktikum dengan alasan apapun dianggap INHAL.
3. Izin praktikum mengikuti aturan izin SIMERU/KULIAH.
4. Yang tidak berkepentingan dengan praktikum dilarang mengganggu praktikan atau membuat keributan/kegaduhan.
5. Penggunaan lab diluar jam praktikum maksimal sampai pukul 21.00 dengan menunjukkan surat ijin dari Kepala Laboratorium S1 Informatika.

Yogyakarta, 1 Agustus 2023

Kepala Laboratorium S1 Informatika



Murein Miksa Mardhia, S.T., M.T.

NIPM. 19891019 201606 011 1236278

DAFTAR ISI

HAK CIPTA	1
KATA PENGANTAR.....	2
DAFTAR PENYUSUN.....	3
HALAMAN REVISI.....	4
HALAMAN PERNYATAAN.....	5
VISI DAN MISI PRODI INFORMATIKA.....	6
TATA TERTIB LABORATORIUM S1 INFORMATIKA	7
DAFTAR ISI	10
DAFTAR GAMBAR.....	11
DAFTAR TABEL.....	12
SKENARIO PRAKTIKUM SECARA DARING	13
PRAKTIKUM 1: STRUKTUR PROYEK ANDROID	21
PRAKTIKUM 2: LAYOUT	35
PRAKTIKUM 3: PENGENALAN WIDGET FORM	46
PRAKTIKUM 4: ROOM DATABASE.....	57
PRAKTIKUM 5: RECYCLERVIEW	67
PRAKTIKUM 6: SERVICE.....	74
PRAKTIKUM 7: KONSUMSI REST API.....	82
PRAKTIKUM 8: FIREBASE REALTIME DATABASE	95
PRAKTIKUM 9: FIREBASE STORAGE	109
PRAKTIKUM 10: DEBUGGING.....	120
PRAKTIKUM 11: PETA.....	126
DAFTAR PUSTAKA.....	133

DAFTAR GAMBAR

Gambar 1.1 Tampilan Masuk Android Studio	25
Gambar 1.2 Menu Awal	26
Gambar 1.3 Konfigurasi Android Studio 1.....	27
Gambar 1.4 Konfigurasi Android Studio 2.....	27
Gambar 1.5 Membuat AVD 1	28
Gambar 1.6 Membuat AVD 2	28
Gambar 1.7 Pilih Jenis Perangkat	29
Gambar 1.8 Setting Versi Android AVD.....	29
Gambar 1.9 Setting nama AVD.....	30
Gambar 1.10 AVD Siap Digunakan	30
Gambar 1.11 Tombol Run App Menggunakan Smartphone	31
Gambar 1.12 Siklus Activity.....	31
Gambar 2.1 Linear Layout	36
Gambar 2.2 Table Layout	36
Gambar 2.3 Frame Layout.....	37
Gambar 2.4 Constraint Layout	37
Gambar 2.5 Percobaan Linear Layout	39
Gambar 2.6 Percobaan Relative Layout.....	41
Gambar 2.7 Percobaan Constraint Layout	42
Gambar 3.1 Tampilan TextView	47
Gambar 3.2 Tampilan EditText.....	47
Gambar 3.3 Tampilan button	47
Gambar 3.4 Tampilan ImageButton	47
Gambar 3.5 Tampilan Checkbox	48
Gambar 3.6 Tampilan ToggleButton	48
Gambar 3.7 Tampilan RadioButton.....	48
Gambar 5.1 Tampilan RecyclerView	72
Gambar 6.1 Menambahkan Postman Pada Google Chrome	78
Gambar 6.2 Akses Localhost	78
Gambar 7.1 Mengubah dari Android menjadi Project.....	92
Gambar 7.2 Penambahan repo jcenter()	92
Gambar 8.1 Firebase tools	99
Gambar 8.2 Membuat project baru pada Firebase Console	100
Gambar 8.3 Pop up untuk koneksi prproject	100
Gambar 8.4 Gambar berhasil koneksikan Firebase.....	100
Gambar 8.5 Aktifkan Realtime Database	101
Gambar 9.1 aktifkan fitur storage	111
Gambar 10.1 Tools membuat deebug	123
Gambar 10.2 popup setelah berhasil build file APK.....	123
Gambar 10.3 Bentuk file APK hasil debugging.	123

DAFTAR TABEL

SKENARIO PRAKTIKUM SECARA DARING

Nama Mata Praktikum : Pemrograman Mobile

Jumlah Pertemuan : 11

TABEL SKENARIO PRAKTIKUM DARING

Pertemuan ke	Judul Materi	Waktu (Lama praktikum sampai pengumpulan posttest)	Skenario Praktikum dari pemberian pre-test, post-test dan pengumpulannya serta mencantumkan metode yang digunakan misal video, whatsapp group, Google meet atau lainnya
1	Struktur Proyek Android	Minggu 1 praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan pertama dan menjelaskan alat(<i>software</i>) yang digunakan dalam pelaksanaan praktikum. Praktikan mengerjakan pre-test melalui elearning menu PRETEST 1
		Minggu 1 praktikum Durasi : 70 menit	Praktikum: <ul style="list-style-type: none"> Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. Praktikan melakukan uji coba membuat <i>project</i> baru menggunakan aplikasi Android Studio sesuai buku petunjuk praktikum. Praktikan melakukan uji coba terhadap fitur yang terdapat pada aplikasi Android Studio. Praktikan melakukan uji coba membuat <i>android virtual device</i> menggunakan aplikasi Android Studio sesuai buku petunjuk praktikum.
		Minggu 1 Praktikum Durasi : 6 hari	Post-test: <ul style="list-style-type: none"> Asisten membuat media google form untuk pengumpulan post-test. Praktikan membuat laporan yang menjelaskan hasil aktivitas praktikum pada pertemuan pertama.
2	Layout	Minggu 2 Praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> Dosen/Asisten praktikum menjelaskan(diskusi) materi yang

			<p>akan dipelajari pada pertemuan kedua dan menjelaskan alat(<i>software</i>) yang digunakan dalam pelaksanaan praktikum.</p> <ul style="list-style-type: none"> • Praktikan mengerjakan pre-test melalui elearning menu PRETEST 2
		Minggu 2 Praktikum Durasi : 70 menit	<p>Praktikum:</p> <ul style="list-style-type: none"> • Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. • Praktikan menuliskan kode program <i>linear layout</i> yang terdapat pada buku petunjuk praktikum. • Praktikan menuliskan kode program <i>relative layout</i> yang terdapat pada buku petunjuk praktikum. • Praktikan menuliskan kode program <i>gridview layout</i> yang terdapat pada buku petunjuk praktikum. • Praktikan menjalankan hasil penulisan kode program. • Praktikan menganalisa perbedaan ketiga struktur program pada buku petunjuk praktikum.
		Minggu 2 Praktikum Durasi : 6 hari	<p>Post-test:</p> <ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan post-test. • Praktikan membuat laporan analisis terkait ketiga struktur program pada pertemuan ke-dua.
	3 Pengenalan Widget Form	Minggu 3 Praktikum Durasi : 20 menit	<p>Pre-test:</p> <ul style="list-style-type: none"> • Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan ke-tiga. • Praktikan mengerjakan pre-test melalui elearning menu PRETEST 3
		Minggu 3 Praktikum Durasi : 70 menit	<p>Praktikum:</p> <ul style="list-style-type: none"> • Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. • Praktikan menuliskan kode program untuk membuat <i>basic widget</i> seperti <i>textview</i>, <i>radiobutton</i>, <i>checkbox</i>, dan lain

4	Room Database		<p>yang terdapat pada buku petunjuk praktikum.</p> <ul style="list-style-type: none"> • Praktikan menjalankan hasil penulisan kode program.
		Minggu 3 Praktikum Durasi : 6 hari	<p>Post-test:</p> <ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan post-test. • Praktikan memodifikasi hasil praktikum pertemuan ke-dua dengan menambahkan basic widget(materi pertemuan ke-tiga). • Praktikan membuat laporan yang menjelaskan hasil aktivitas praktikum pada pertemuan ke-tiga.
		Minggu 4 Praktikum Durasi : 20 menit	<p>Pre-test:</p> <ul style="list-style-type: none"> • Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan ke-empat. • Praktikan mengerjakan pre-test melalui elearning menu PRETEST 4
		Minggu 4 Praktikum Durasi : 70 menit	<p>Praktikum:</p> <ul style="list-style-type: none"> • Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. • Praktikan menuliskan kode program untuk aktivitas yang terdapat pada buku petunjuk praktikum. • Praktikan membuat sejumlah folder dan menyertakan <i>file</i> "<i>activity_main.xml</i>" sesuai buku petunjuk praktikum. • Praktikan menjalankan hasil penulisan kode program.
		Minggu 4 Praktikum Durasi : 6 hari	<p>Post-test:</p> <ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan post-test. • Praktikan menjelaskan fungsi kode program keadaan: onCreate(), onResume(), onPause(), onStop(), onDestroy(). • Praktikan membuat laporan yang menjelaskan hasil aktivitas praktikum pada pertemuan ke-empat.

5	Recycler View	Minggu 5 Praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan ke-lima. Praktikan mengerjakan pre-test melalui elearning menu PRETEST 5
		Minggu 5 Praktikum Durasi : 70 menit	Praktikum: <ul style="list-style-type: none"> Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. Praktikan menuliskan kode program untuk <i>select file</i>, <i>open file</i>, dan <i>setting</i> yang terdapat pada buku petunjuk praktikum. Praktikan menjalankan hasil penulisan kode program. Praktikan melakukan uji coba terhadap hasil penulisan kode program.
		Minggu 5 Praktikum Durasi : 6 hari	Post-test: <ul style="list-style-type: none"> Asisten membuat media google form untuk pengumpulan post-test. Praktikan mengimplementasi "<i>recyclerview</i>". Praktikan membuat laporan yang menjelaskan hasil aktivitas praktikum pada pertemuan ke-lima.
6	Service	Minggu 6 Praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan ke-enam. Praktikan mengerjakan pre-test melalui elearning menu PRETEST 6
		Minggu 6 Praktikum Durasi : 70 menit	Praktikum: <ul style="list-style-type: none"> Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. Praktikan menuliskan kode program sederhana tentang pembuatan service di android yang terdapat pada buku petunjuk praktikum. Praktikan menjalankan hasil penulisan kode program.

			<ul style="list-style-type: none"> • Praktikan melakukan uji coba terhadap hasil penulisan kode program.
		Minggu 6 Praktikum Durasi : 6 hari	Post-test: <ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan post-test. • Praktikan membuat API tentang list film bioskop dan menyimpannya. • Praktikan membuat laporan yang menjelaskan hasil aktivitas praktikum pada pertemuan keenam.
7	Konsumsi REST API	Minggu 7 Praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> • Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan ketujuh. • Praktikan mengerjakan pre-test melalui elearning menu PRETEST 7
		Minggu 7 Praktikum Durasi : 70 menit	Praktikum: <ul style="list-style-type: none"> • Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. • Praktikan menuliskan kode program untuk keperluan konsumsi REST API • Praktikan menjalankan hasil penulisan kode program. • Praktikan melakukan uji coba terhadap hasil penulisan kode program.
		Minggu 7 Praktikum Durasi : 6 hari	Post-test: <ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan post-test. • Praktikan memodifikasi kode program untuk membuat list bioskop menggunakan REST API. • Praktikan membuat laporan yang menjelaskan hasil aktivitas praktikum pada pertemuan ketujuh.
8	Firebase Realtime Database	Minggu 8 Praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> • Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan kedelapan. • Praktikan mengerjakan pre-test melalui elearning menu PRETEST 8

		Minggu 8 Praktikum Durasi : 70 menit	Praktikum: <ul style="list-style-type: none"> Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. Praktikan menuliskan kode program untuk menerapkan firebase. Praktikan menjalankan hasil penulisan kode program. Praktikan melakukan uji coba terhadap hasil penulisan kode program.
		Minggu 8 Praktikum Durasi : 6 hari	Post-test: <ul style="list-style-type: none"> Asisten membuat media google form untuk pengumpulan post-test. Praktikan membuat CRUD kedalam realtime database kemudian melaporkan dengan menjelaskan struktur kode program laporan dan hasil praktikum pada pertemuan ke-delapan.
9	Firebase Storage	Minggu 9 Praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan ke-sembilan. Praktikan mengerjakan pre-test melalui elearning menu PRETEST 9
		Minggu 9 Praktikum Durasi : 70 menit	Praktikum: <ul style="list-style-type: none"> Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. Praktikan meng-<i>install</i> plugin Postman pada browser Google Chrome di perangkat komputernya. Praktikan membuat sebuah <i>local database</i> dengan nama "employee". Praktikan menuliskan kode program ke dalam database sesuai buku petunjuk praktikum. Praktikan menuliskan kode program koneksi database dan simpan dengan nama "employee.php" sesuai buku petunjuk praktikum.

			<ul style="list-style-type: none"> • Praktikan menjalankan hasil penulisan kode program. • Praktikan mengakses <i>local database</i> melalui plugin Postman sesuai buku petunjuk praktikum.
		Minggu 9 Praktikum Durasi : 6 hari	Post-test: <ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan post-test. • Praktikan membuat laporan yang menjelaskan struktur kode program laporan dan hasil praktikum pada pertemuan ke-sembilan.
10	Debugging	Minggu 10 Praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> • Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan ke-sepuluh. • Praktikan mengerjakan pre-test melalui elearning menu PRETEST 10
		Minggu 10 Praktikum Durasi : 70 menit	Praktikum: <ul style="list-style-type: none"> • Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i> pada setiap perangkat komputer praktikan. • Praktikan membuat file .apk dari hasil praktikum. • Praktikan menjalankan hasil penulisan kode program. • Praktikan melakukan uji coba terhadap hasil penulisan kode program.
		Minggu 10 Praktikum Durasi : 6 hari	<ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan post-test. • Praktikan membuat laporan yang menjelaskan struktur kode program dan hasil praktikum pada pertemuan ke-sepuluh.
11	Peta	Minggu 11 Praktikum Durasi : 20 menit	Pre-test: <ul style="list-style-type: none"> • Dosen/Asisten praktikum menjelaskan(diskusi) materi yang akan dipelajari pada pertemuan ke-sebelas. • Praktikan mengerjakan pre-test melalui elearning menu PRETEST 11
		Minggu 11 Praktikum Durasi : 70 menit	Praktikum: <ul style="list-style-type: none"> • Asisten praktikum memastikan alat(<i>software</i>) telah ter-<i>install</i>

			<p>pada setiap perangkat komputer praktikan.</p> <ul style="list-style-type: none"> • Praktikan menuliskan dua langkah praktikum yang terdapat pada buku petunjuk praktikum. • Praktikan menjalankan hasil penulisan kode program. • Praktikan melakukan uji coba terhadap hasil penulisan kode program.
		Minggu 11 Praktikum Durasi : 6 hari	<ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan post-test. • Praktikan membuat laporan yang menjelaskan struktur kode program dan hasil praktikum pada pertemuan ke-sebelas.
11	Responsi	Minggu 12 Praktikum Durasi : 24 jam	<ul style="list-style-type: none"> • Asisten membuat media google form untuk pengumpulan responsi. • Praktikan mengumpulkan tugas proyek + laporan.

PRAKTIKUM 1: STRUKTUR PROYEK ANDROID

Pertemuan ke : 1

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPMK-01	Mahasiswa memiliki kemampuan teoritis terkait konsep mobile programming, menerapkan konsep-teori dan model serta analisis dalam merancang dan membangun aplikasi

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menganalisa struktur proyek android
2. Menerapkan struktur proyek android ke dalam aplikasi yang dibuat

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-04	CPMK-01	Mengenal tentang struktur proyek android
CPL-03	CPMK-01	Menerapkan struktur proyek android ke dalam aplikasi yang dibuat

1.3. TEORI PENDUKUNG

A. Mengenal Android Studio

Android Studio adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android. Berbasis editor kode dan alat developer yang andal dari [IntelliJ IDEA](#), Android Studio menawarkan lebih banyak fitur yang mampu meningkatkan produktivitas Anda saat mem-build aplikasi Android, seperti:

- Sistem build berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android
- Edit Live untuk mengupdate composables di emulator dan perangkat fisik secara real time

- Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
- Framework dan alat pengujian yang lengkap
- Alat lint untuk merekam/memeriksa performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- Dukungan C++ dan NDK
- Dukungan bawaan untuk [Google Cloud Platform](#), yang memudahkan integrasi Google Cloud Messaging dan App Engine

B. Struktur project

Setiap project di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file resource. Jenis modul meliputi:

- Modul aplikasi Android
- Modul library
- Modul Google App Engine

Secara default, Android Studio menampilkan file project Anda dalam tampilan project Android, seperti yang ditunjukkan pada gambar 1. Tampilan ini disusun menurut modul untuk memberikan akses cepat ke file sumber utama project Anda. Semua file build terlihat di tingkat atas, di bagian **Gradle Scripts**. Setiap modul aplikasi berisi folder berikut:

- **manifest**: Berisi file AndroidManifest.xml.
- **java**: Berisi file kode sumber Kotlin dan Java, termasuk kode pengujian JUnit.
- **res**: Berisi semua resource non-kode, seperti string UI dan gambar bitmap.

Semua komponen yang kita gunakan untuk membuat aplikasi android akan diletakan pada folder-folder proyek android studio. Adapun struktur proyek yang ada pada android studio adalah :

1. Manifest

Klik folder Manifest maka akan terdapat AndroidManifest.xml. File ini digunakan untuk melakukan beberapa pengaturan seperti Nama aplikasi, Icon aplikasi, Theme style, User permission (jika membuat aplikasi yang membutuhkan akses hardware smartphone ataupun koneksi internet).

2. Java

Di folder ini terdapat file MainActivity.java yang digunakan untuk memberikan perintah agar aplikasi melakukan sesuatu menggunakan bahasa pemrograman java.

3. Res Drawable

Folder yang digunakan untuk memasukan gambar pendukung aplikasi yang kita buat baik berupa icon ataupun lainnya. Cara memasukan gambar ke folder ini cukup mudah yaitu:
o Klik kanan pada folder drawable, pilih show in explorer, kemudian copykan gambar yang sudah kita siapkan di komputer pada folder drawable.

4. Res Layout

Disini terdapat file activity_main.xml. File ini digunakan untuk pengaturan layout untuk interface utama pada aplikasi android yang kita buat. Saat pertama kali dibuka biasanya file ini memiliki kode-kode default.

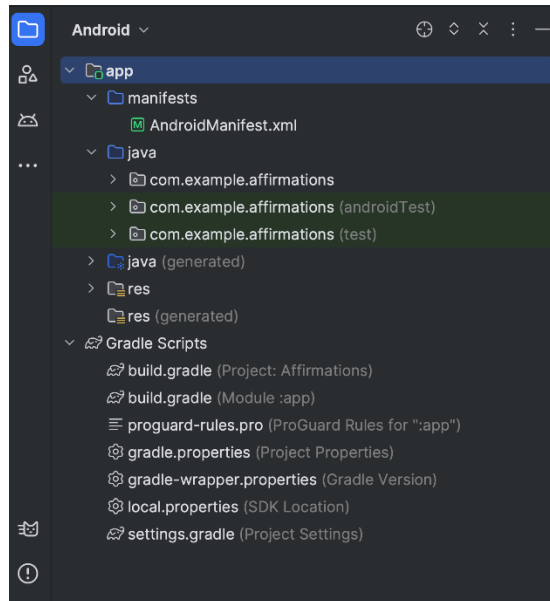
5. Mipmap

Folder ini digunakan untuk memasukan gambar berupa icon. Icon default aplikasi yang kita buat juga diambil dari folder ini.

6. Values

- **Colors.xml**. File ini berisi kode-kode untuk pengaturan warna. Warna status bar, teks, ataupun lainnya.
- **Strings.xml**. Digunakan untuk pengaturan teks-teks aplikasi yang kita buat.
- **Styles.xml**. Digunakan untuk memberikan nama warna setelah kode-kode warna dimasukan atau disetting pada color.xml. File ini juga bisa disebut theme dasar dari aplikasi android.

Untuk melihat struktur file project sebenarnya, pilih **Project**, bukan **Android**, dari menu **Project**.



Gambar 1. File project dalam tampilan project Android.

C. Sistem build Gradle

Android Studio menggunakan Gradle sebagai dasar sistem build, dengan lebih banyak kemampuan khusus Android yang disediakan oleh [plugin Android Gradle](#). Sistem build ini berjalan sebagai alat terintegrasi dari menu Android Studio dan terpisah dari command line. Anda dapat menggunakan fitur-fitur sistem build untuk:

- Menyesuaikan, mengonfigurasi, dan memperluas proses build.
- Membuat banyak APK untuk aplikasi Anda dengan berbagai fitur yang menggunakan project dan modul yang sama.
- Menggunakan kembali kode dan resource ke seluruh set sumber.
- Berkat fleksibilitas Gradle, Anda dapat mencapai semua ini tanpa mengubah file sumber inti aplikasi Anda.

File build Android Studio diberi nama `build.gradle.kts` jika Anda menggunakan [Kotlin](#) (direkomendasikan) atau `build.gradle` jika Anda menggunakan [Groovy](#). File tersebut adalah file teks biasa yang menggunakan sintaksis Kotlin atau Groovy untuk mengonfigurasi build dengan elemen yang disediakan oleh plugin Android Gradle. Setiap project memiliki satu file build level teratas untuk seluruh project dan file build level modul terpisah untuk setiap modul. Saat Anda mengimpor project yang ada, Android Studio akan otomatis menghasilkan file build yang diperlukan.

Varian build

Sistem build dapat membantu Anda membuat beberapa versi berbeda untuk aplikasi yang sama dari satu project. Hal ini berguna saat Anda menyediakan aplikasi dalam versi gratis dan berbayar atau jika Anda ingin mendistribusikan beberapa APK untuk berbagai konfigurasi perangkat di Google Play. Untuk mengetahui informasi selengkapnya tentang cara mengonfigurasi varian build, lihat [Mengonfigurasi varian build](#).

Dukungan multi-APK

Dukungan multi-APK memungkinkan Anda membuat beberapa APK sekaligus secara efisien berdasarkan kepadatan layar atau ABI. Misalnya, Anda dapat membuat APK aplikasi terpisah untuk kepadatan layar `hdpi` dan `mdpi`, sambil tetap mempertimbangkannya sebagai satu varian dan mengizinkannya berbagi setelan APK pengujian, `javac`, `dx`, dan ProGuard.

Untuk mengetahui informasi selengkapnya tentang dukungan multi-APK, baca [Membuat beberapa APK](#).

Penyingkatan resource

Penyingkatan resource di Android Studio otomatis menghapus resource yang tidak digunakan dari aplikasi terpaket dan dependensi library Anda. Misalnya, jika aplikasi Anda menggunakan [layanan Google Play](#) untuk mengakses fungsi Google Drive, dan saat ini Anda tidak menggunakan [Login dengan Google](#), penyingkatan resource dapat menghapus beragam aset drawable untuk tombol SignInButton.

Mengelola dependensi

Dependensi untuk project Anda ditetapkan berdasarkan nama dalam skrip build level modul. Gradle menemukan dependensi dan menyediakannya di build Anda. Anda dapat mendeklarasikan dependensi modul, dependensi biner jarak jauh, dan dependensi biner lokal dalam file build.gradle.kts Anda.

Android Studio mengonfigurasi project untuk menggunakan Maven Central Repository secara default. Konfigurasi ini disertakan dalam file build level atas untuk project.

Agar bisa lebih memahami lingkungan kerja sekaligus cara pembuatan proyek android, mulailah dengan sebuah proyek baru. Namun sebelum proses pembuatan proyek dilakukan, anda harus memiliki 3 buah komponen yang dapat anda unduh langsung dari situs resminya. Komponen tersebut adalah:

1. Java JDK

<http://www.oracle.com/technetwork/java/javase/downloads/index.html> dan pilih Java JDK terbaru.

2. Android Studio

Adalah sebuah IDE yang bisa digunakan untuk pengembangan aplikasi Android, dan dikembangkan oleh Google. **Android Studio** merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Saat ini Android Studio telah menggantikan Eclipse sebagai IDE resmi untuk pengembangan aplikasi android. <https://developer.android.com/studio/index.html>.

3. Android SDK

File Android SDK atau SDK android ini juga dapat di peroleh dari <http://developer.android.com/sdk/index.html>. Ketika mendownload, jangan lupa pilih platform atau sistem operasi yang sesuai dengan milik anda. Di dalam buku ini sistem operasi yang dipilih adalah windows.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio
3. MySQL

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Jelaskan komponen-komponen apa saja yang dibutuhkan ketika kita akan menginstall android studio.	50
2.	CPL-03	CPMK-01	Mengapa kita memerlukan Android Virtual Device (AVD) dalam melengkapi kebutuhan pengembangan aplikasi android?	50

1.6. LANGKAH PRAKTIKUM

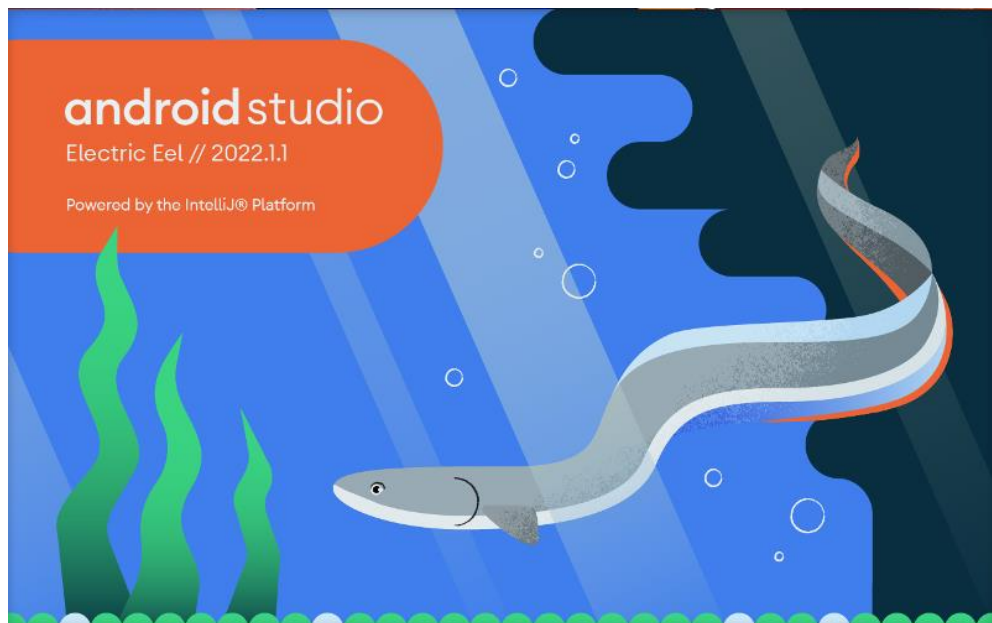
Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 -2	Hasil praktikum langkah 1 - 2	

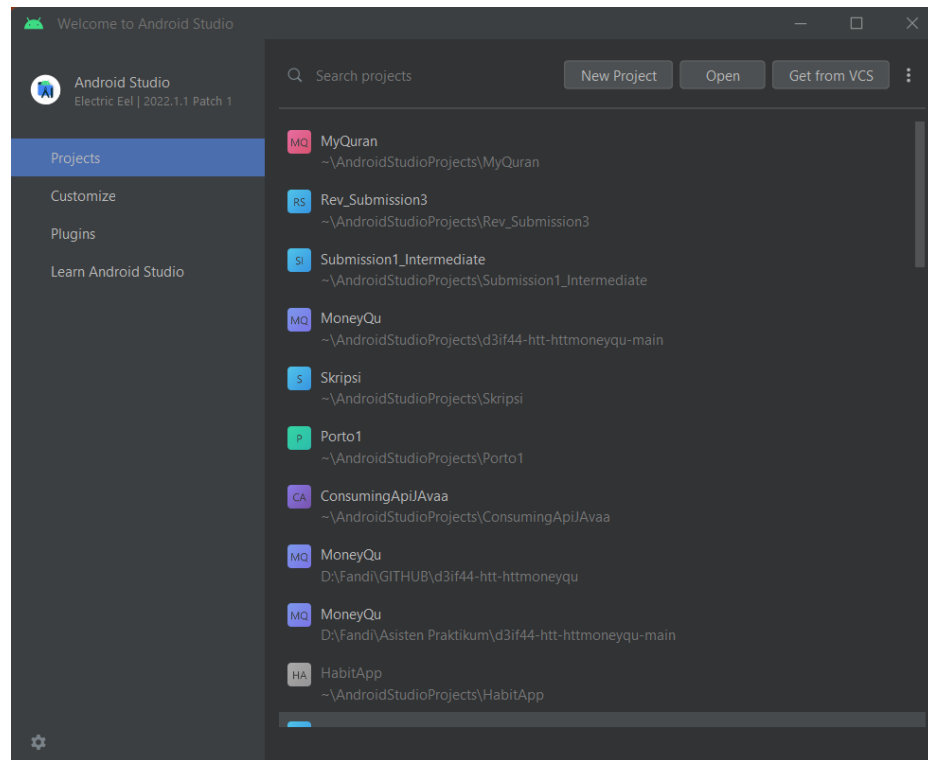
Langkah-Langkah Praktikum:

1. Membuat Project Android

- Jalankan aplikasi Android Studio dan tunggu Gambar 1.1 hingga selesai

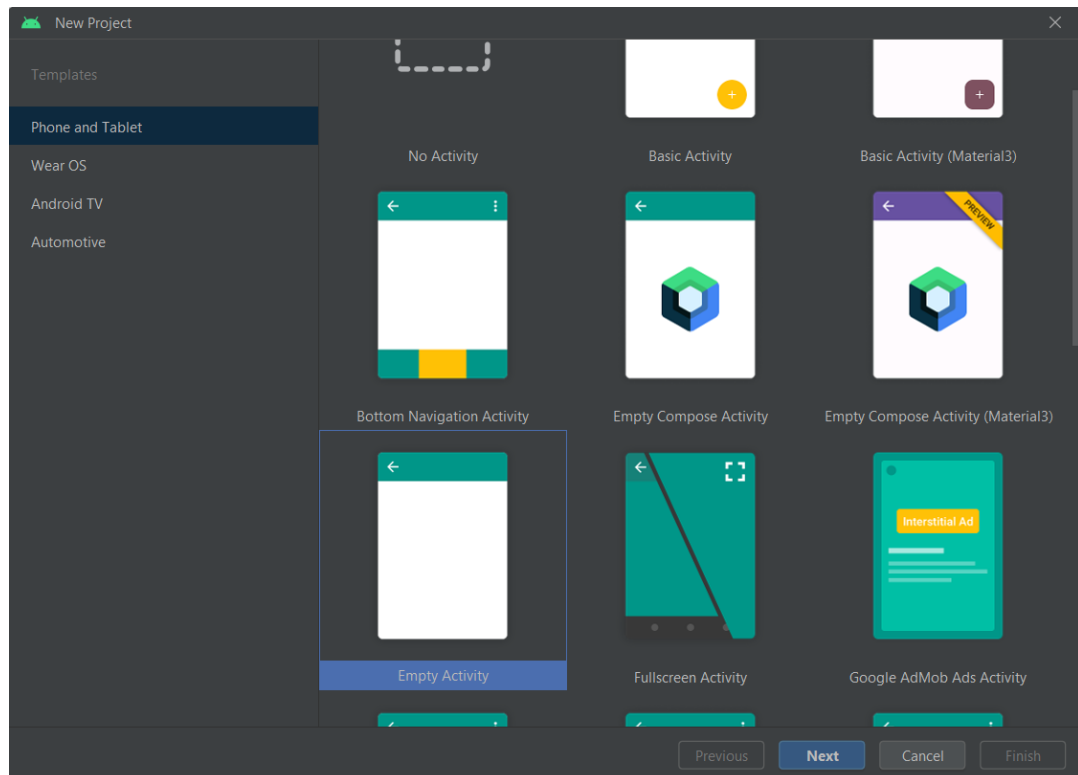


Gambar 1.1 Tampilan Masuk Android Studio



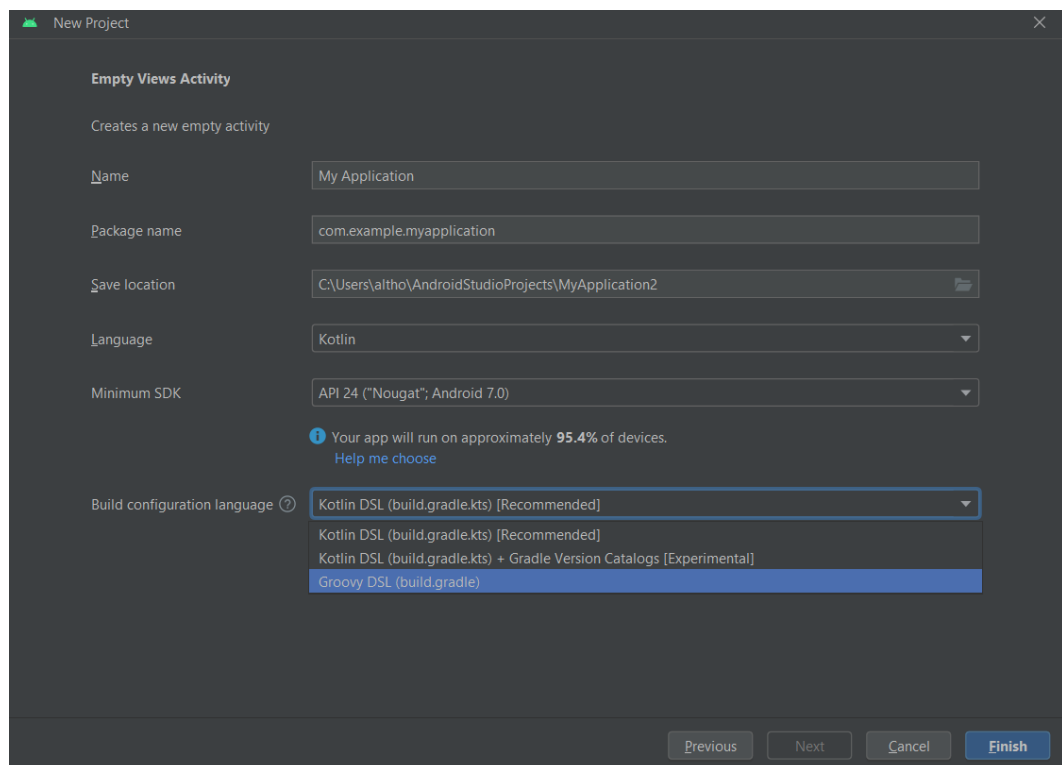
Gambar 1.2 Menu Awal

- b. Pada Gambar 1.2, pilih **New Project** untuk memulai pembuatan project android anda.
- c. Pada Gambar 1.3:
 - Pada bagian Templates, pilih **Phone and Tablet**
 - Anda dapat memilih sebuah template activity yang disediakan untuk mempermudah memulai aplikasi. Untuk saat ini, pilih **Empty Activity**.
 - **Next** jika sudah selesai.



Gambar 1.3 Konfigurasi Android Studio 1

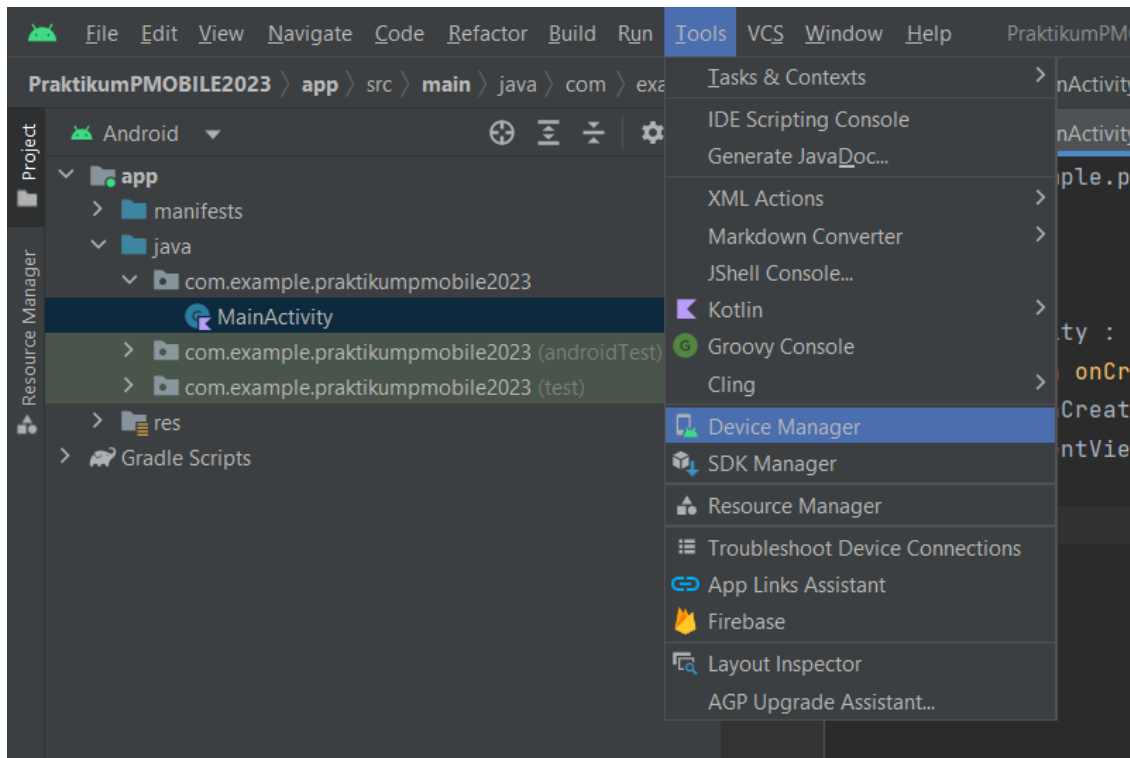
- d. Selanjutnya pada **Name** anda dapat menyesuaikan nama pada project yang akan dibuat. Kemudian pilih **Language** menjadi **Kotlin**. Pada **Minimum SDK** silahkan pilih sesuai keinginan seperti Gambar 1.4. kemudian pilih Build Configuration language **Groovy DSL** agar dapat sesuai dengan kode yang ada pada modul. **Finish** jika sudah selesai.



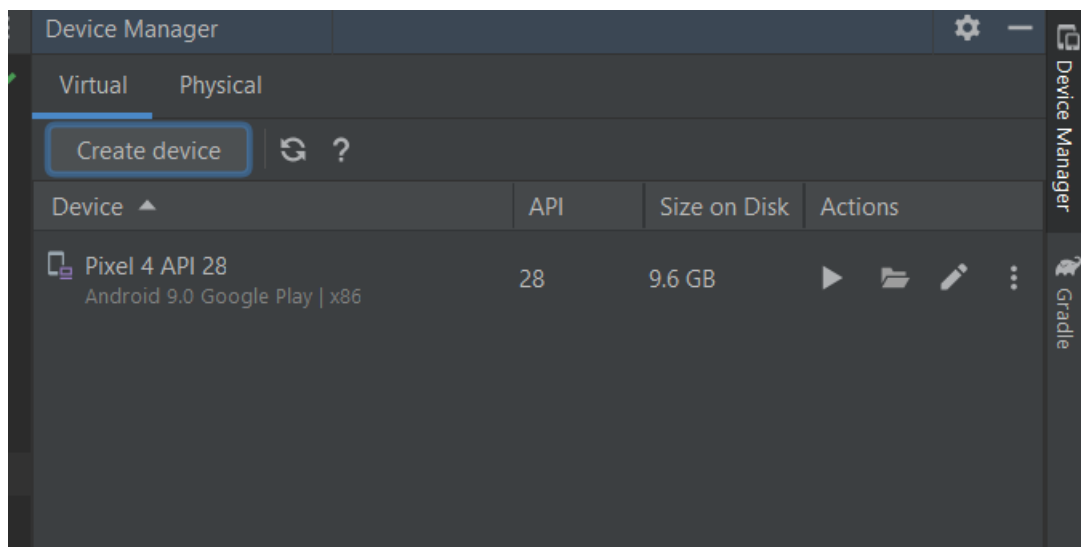
Gambar 1.4 Konfigurasi Android Studio 2

2. Membuat Android Virtual Device

- a. Pada Android Studio, di toolbar menu, pilih **Tools → AVD Manager**, sesuai Gambar 1.7

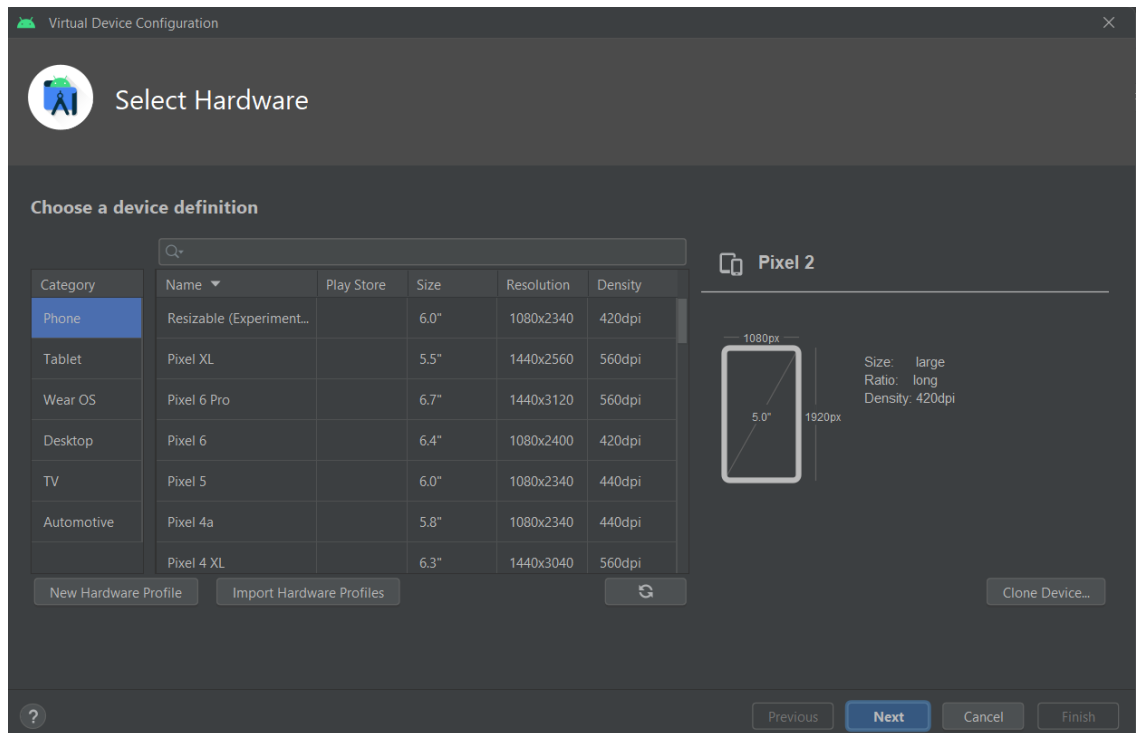


Gambar 1.5 Membuat AVD 1



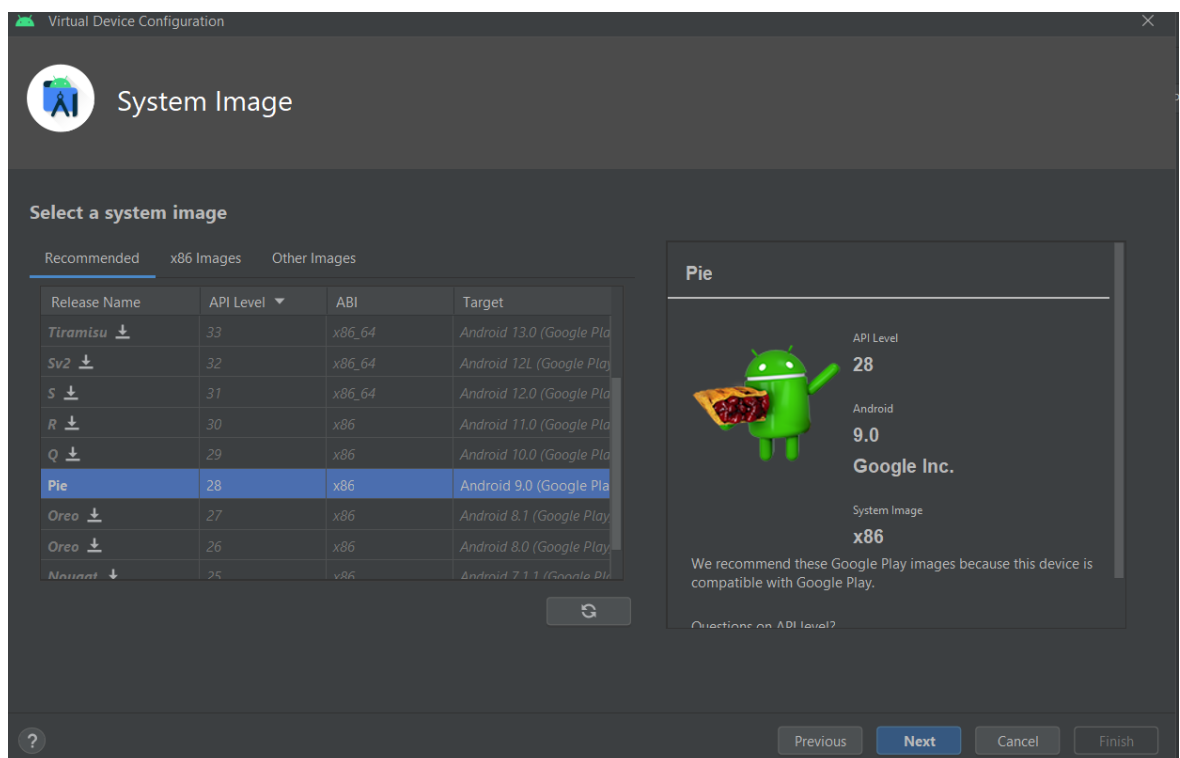
Gambar 1.6 Membuat AVD 2

- b. Pada kotak dialog, pilih **Create Device** sesuai Gambar 1.6



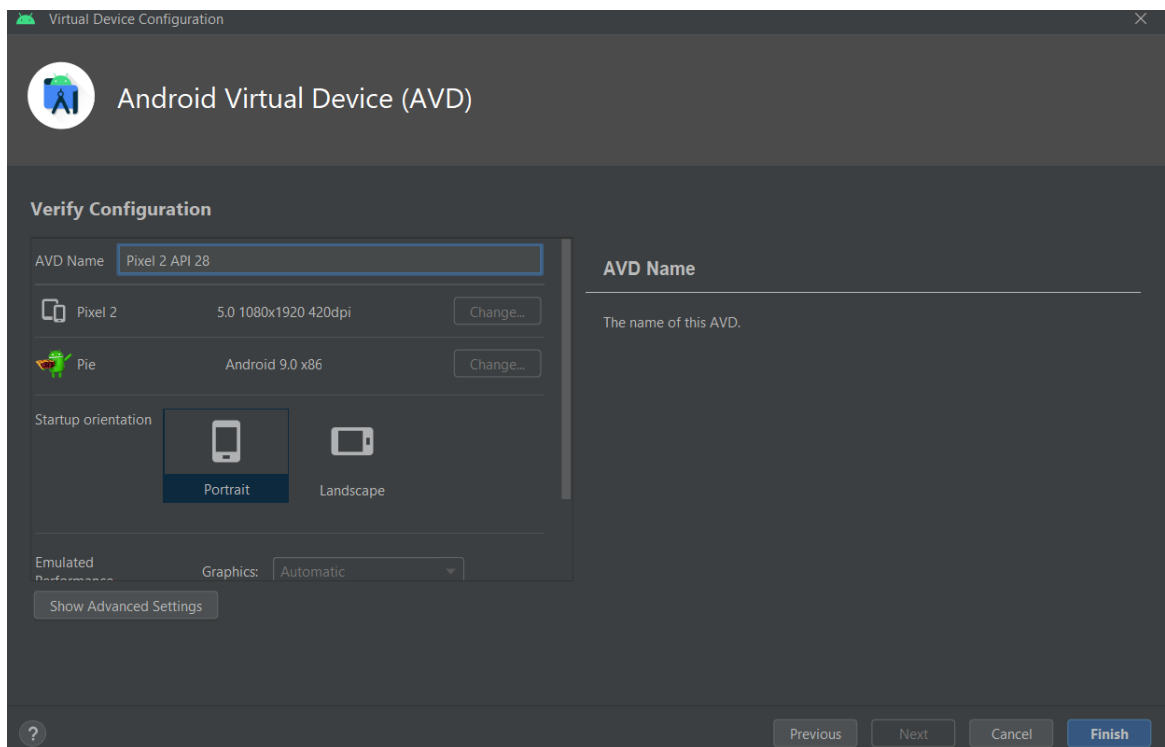
Gambar 1.7 Pilih Jenis Perangkat

- c. Pilih ukuran layar Virtual Device anda, **Next** terus hingga Gambar 1.7



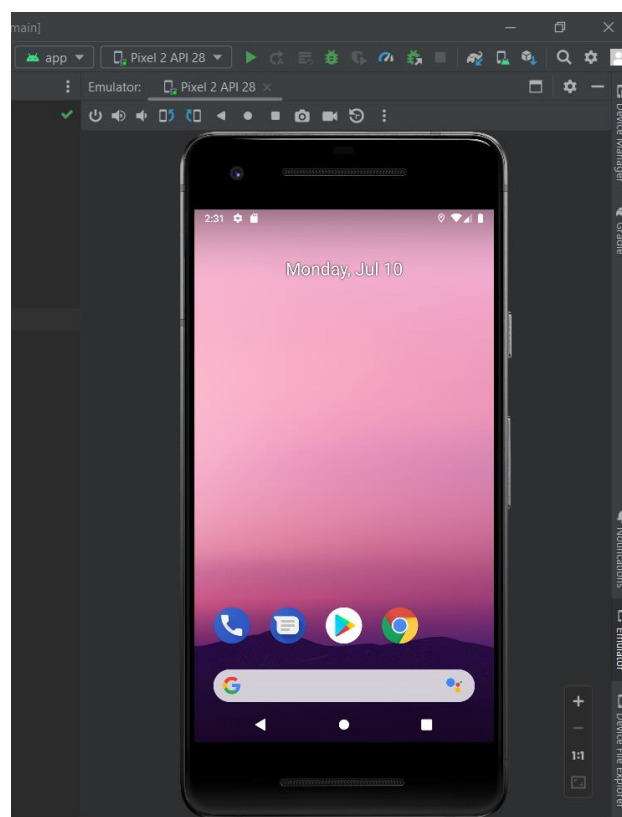
Gambar 1.8 Setting Versi Android AVD

- d. Pada Gambar 1.8 silahkan pilih jenis versi android yang diinginkan, **Next** jika sudah selesai.



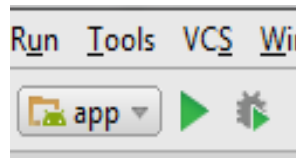
Gambar 1.9 Setting nama AVD

- e. Pada Gambar 1.9 silahkan sesuaikan nama AVD, jika sudah selesai klik **Finish**.



Gambar 1.10 AVD Siap Digunakan

- f. Selain menggunakan AVD, anda dapat menggunakan Smartphone anda dalam melakukan RUN aplikasi android yang telah anda buat nantinya.

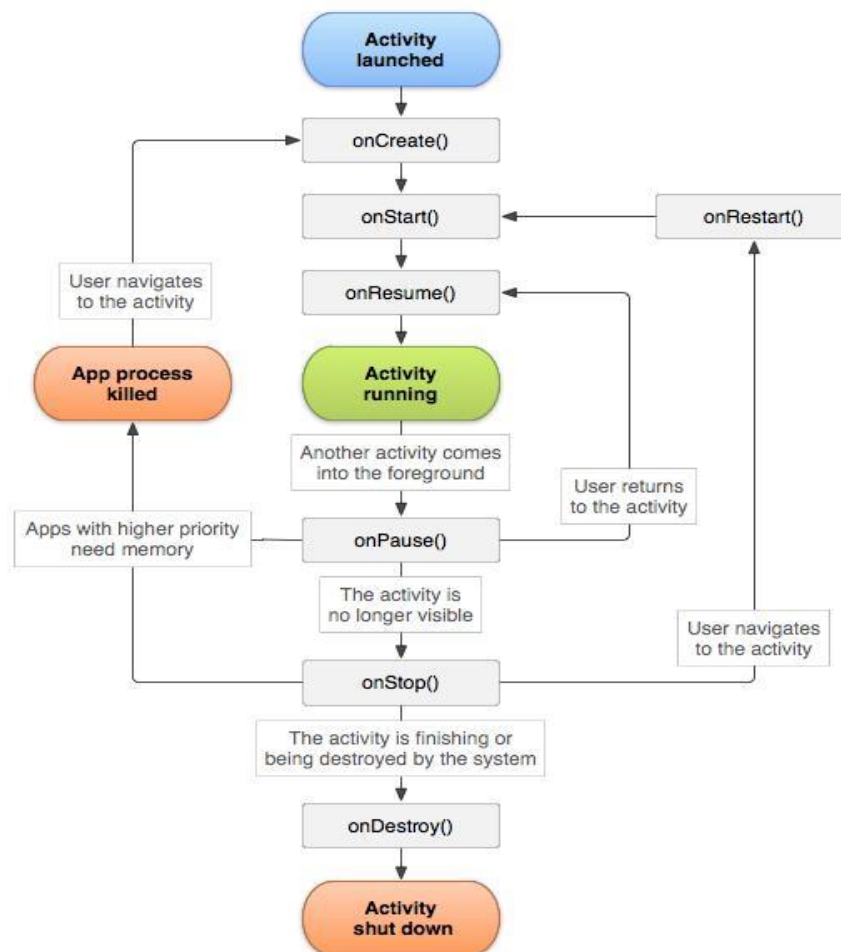


Gambar 1.11 Tombol Run App Menggunakan Smartphone

Klik play Button pada Gambar 1.11, lalu pilih device yang sudah anda koneksi kan ke laptop/PC menggunakan kabel USB.

3. Mengenal Activity

Activity adalah komponen aplikasi yang menyediakan tampilan layar untuk berinteraksi dengan pengguna. Setiap activity memiliki jendela tampilan di mana UI ditampilkan. Activity dapat dimulai, dihentikan, dilanjutkan, dan dihancurkan oleh sistem. Saat activity baru dimulai, activity sebelumnya diberhentikan dan dimasukkan ke dalam stack kembali. Activity dapat menerima callback siklus hidup yang memberikan kesempatan untuk melakukan tindakan tertentu sesuai dengan perubahan keadaan, seperti melepaskan dan mendapatkan kembali sumber daya saat dihentikan dan dilanjutkan. Berikut merupakan siklus dari Activity.



Gambar 1.12 Siklus Activity

- b. Untuk memahami siklus Activity lebih lanjut, silahkan sesuaikan **MainActivity.kt** seperti dibawah ini:

```

1 package com.example.praktikumpmobile2023
2

```



```

3 import android.content.ContentValues.TAG
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.util.Log
7
8 class MainActivity : AppCompatActivity() {
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12
13         Log.d(TAG, "Ini ada di onCreate")
14     }
15
16     override fun onResume() {
17         super.onResume()
18         Log.d(TAG, "Ini ada di onResume")
19     }
20
21     override fun onPause() {
22         super.onPause()
23         Log.d(TAG, "Ini ada di onPause")
24     }
25
26     override fun onStop() {
27         super.onStop()
28         Log.d(TAG, "Ini ada di onStop")
29     }
30
31     override fun onDestroy() {
32         super.onDestroy()
33         Log.d(TAG, "Ini ada di onDestroy")
34     }
35 }

```

c. Lakukan Run dengan shortcut **SHIFT+F10** atau klik icon run pada device manager

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Silahkan lakukan penginstalan di Laptop anda /menjalankan android studio yang ada di PC Lab	50
2.	CPL-03	CPMK-01	Silahkan dicoba berbagai fitur yang ada dalam android studio kemudian jelaskan fungsinya.	50

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		

2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 2: LAYOUT

Pertemuan ke : 2

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPMK-02	Mahasiswa mampu mengoperasikan bahasa pemrograman berorientasi obyek untuk memecahkan masalah dalam lingkup pemrograman mobile

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Memahami perbedaan jenis tampilan Linear dan Relative Layout
2. Mempraktekkan membuat Linear dan Relative Layout Android dengan Android Studio

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03	CPMK-02	Memahami perbedaan jenis tampilan Linear dan Relative Layout
CPL-04	CPMK-02	Mempraktekkan membuat Linear dan Relative Layout Android dengan Android Studio

1.3. TEORI PENDUKUNG

Layout yakni untuk mengatur tata letak suatu komponen pada pemrograman android, diperlukan sebuah file XML. File XML tersebut digunakan untuk mengatur antarmuka aplikasi android seperti komponen teks, tombol, dan sebagainya. Untuk meletakkan komponen-komponen tersebut diperlukan sebuah cara agar komponen yang dibuat sesuai dengan keinginan dan rapi. Cara yang digunakan untuk mengatur tata letak yaitu menggunakan layout. File XML untuk membuat antarmuka tersebut diletakkan pada **project_android -> res -> layout**.

Jenis Layout :

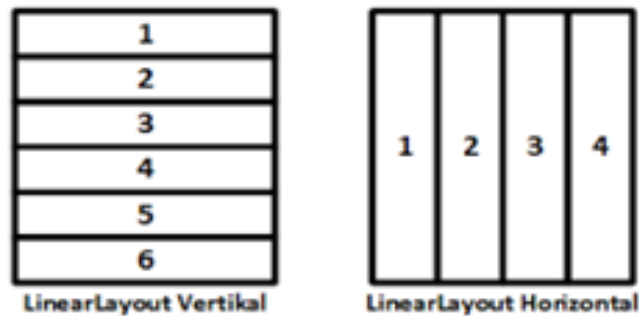
Ada beberapa jenis layout yang dapat digunakan pada android antara lain **Relative Layout**, **Linear Layout**, **Constraint Layout**, dan sebagainya.

12. Relative Layout

Relative Layout merupakan sebuah layout dimana posisi dari sebuah komponen dapat diatur terhadap komponen lainnya. Misalnya tombol “OK” posisinya berada dibawah “EditText”, kemudian tombol “cancel” posisinya berada di sebelah kanan tombol “OK” dan dibawah “EditText”. Intinya, saling berkaitan antara posisi satu tombol dengan yang lain.

13. Linear Layout

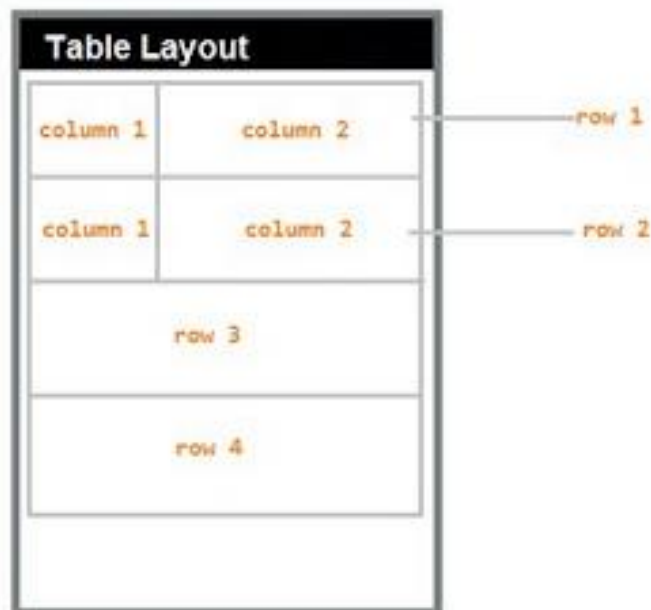
Linear Layout merupakan layout yang memberikan susunan tata letak komponen secara garis lurus baik vertikal maupun horizontal. Gambar 2 menunjukkan model orientasi LinearLayout.



Gambar 2.1 Linear Layout

14. Table Layout

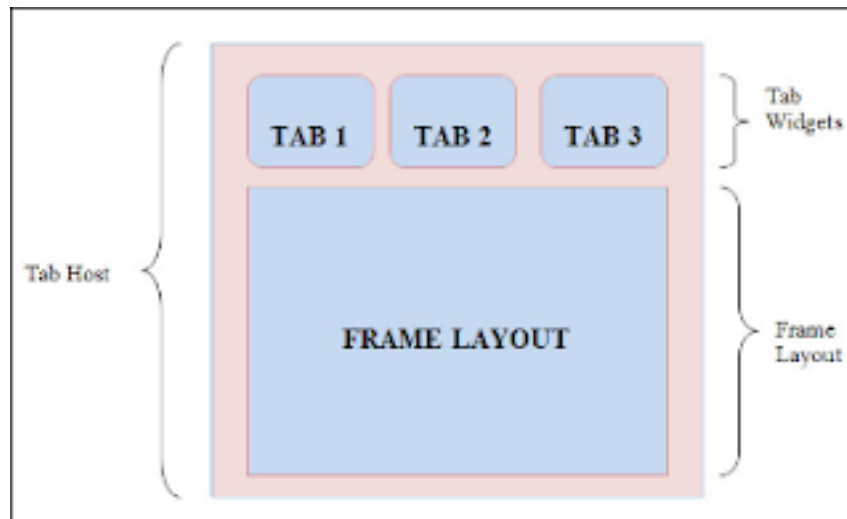
Table Layout merupakan layout yang memberikan susunan tata letak komponen berdasarkan susunan tabel (baris dan kolom) mirip dengan struktur table pada HTML.



Gambar 2.2 Table Layout

15. Frame Layout

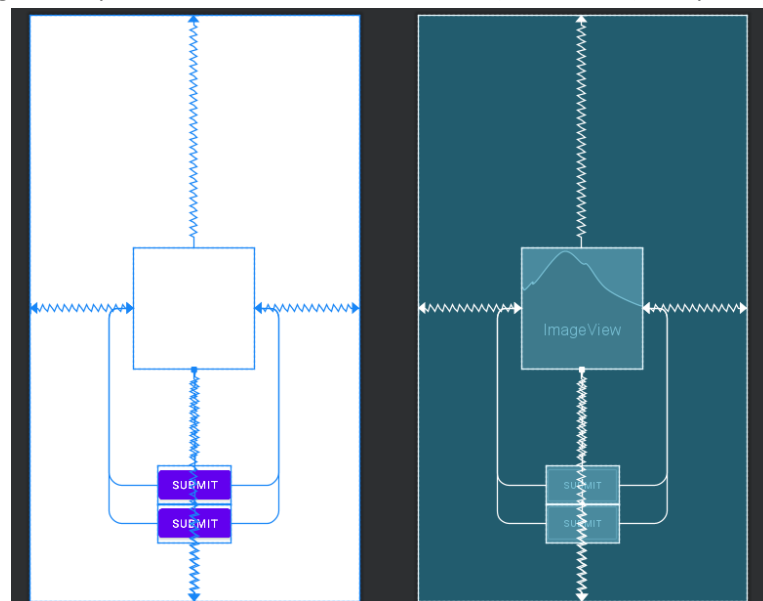
Frame Layout merupakan layout yang memberikan susunan tata letak komponen yang bertumpuk.



Gambar 2.3 Frame Layout

16. Constraint Layout

Constraint Layout merupakan salah satu komponen ViewGroup yang dapat kita gunakan untuk menyusun tampilan aplikasi yang kompleks tanpa adanya nested layout. ConstraintLayout tersedia dengan dukungan kompatibilitas mulai dari Android 2.3 (API Level 9) sampai dengan yang terbaru.



Gambar 2.4 Constraint Layout

17. Scroll View

Dalam pengembangan aplikasi android biasanya sering terkendala dengan ukuran tampilan layar. Konten yang ingin dimasukkan cukup banyak sehingga layar tidak dapat menampilkannya dalam satu layar. ScrollView digunakan untuk mengatasi masalah tersebut. Pasti sudah tidak asing lagi dengan Scroll View ini?

18. Kombinasi Layout

Beberapa layout yang telah dijelaskan sebelumnya juga dapat dikombinasikan. Kombinasi layout tersebut bertujuan agar antarmuka yang dibangun menjadi lebih bagus dan sesuai dengan keinginan. Kombinasi layout dapat dipilih sesuai dengan kebutuhan antarmuka.

Bagian ini menjadi dasar untuk mengembangkan User Interface sebuah aplikasi. Jangan pedulikan scriptnya dahulu, karena kelihatannya banyak dan susah, namun sebenarnya sangat sederhana dan mudah dimengerti. Semakin banyak Anda berlatih coding, semakin paham juga maksud programnya.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Sebut dan Jelaskan jenis layout yang dapat digunakan untuk membuat aplikasi android?	50
2.	CPL-04	CPMK-02	Jelaskan ketika kondisi bagaimana kita bisa menerapkan penggunaan relative layout?	50

1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-04	CPMK-02	Selesaikan percobaan 1 - 3	Hasil praktikum langkah 1 - 3	

Langkah-Langkah Praktikum:

1. Percobaan 1 (Linear Layout)

- a. Jalankan Android Studio, **Buatlah sebuah project android.**
- b. Pada Java Explorer, pilih **res/layout/activity_main.xml**. sesuaikan menjadi seperti berikut:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/activity_main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context="com.example.praktikumpmobile2023.MainActivity">
9   <!-- sesuaikan tools:context dengan package project anda -->
10
11
12   <LinearLayout
13     android:layout_width="match_parent"
14     android:layout_height="match_parent"
15     android:layout_weight="2"
16     android:orientation="horizontal">
17
18     <TextView
```

```

19     android:layout_width="wrap_content"
20     android:layout_height="match_parent"
21     android:layout_weight="1"
22     android:background="#aa0000"
23     android:gravity="center_horizontal"
24     android:text="Merah" />
25
26     <TextView
27         android:layout_width="wrap_content"
28         android:layout_height="match_parent"
29         android:layout_weight="1"
30         android:background="#03fa40"
31         android:gravity="center_horizontal"
32         android:text="Hijau" />
33
34 </LinearLayout>
35 </LinearLayout>

```

c. Pastikan pada MainActivity.kt seperti berikut:

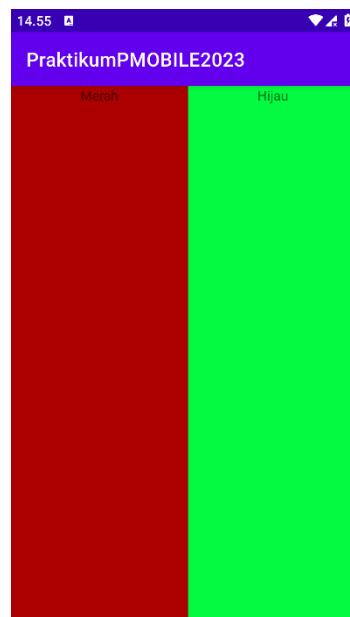
```

1 package com.example.praktikumpmobile2023
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10    }
11 }

```

d. Lakukan Run dengan shortcut **SHIFT+F10** atau klik Run Button

e. Pastikan tampilan program pada AVD/Smartphone anda seperti gambar 2.5



Gambar 2.5 Percobaan Linear Layout

2. Percobaan 2 (Relative Layout)

- Buatlah satu file baru xml dengan klik kanan para folder **layout**, lalu pilih New->Xml->Layout Xml File. Beri nama relative_layout lalu **Finish**.
- Isikan kotak dialog new seperti berikut
- Pada **relative_layout.xml**. sesuaikan menjadi seperti berikut:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <TextView
7         android:id="@+id/textview1"
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:layout_alignParentLeft="true"
11        android:layout_alignParentTop="true"
12        android:paddingTop="10dp"
13        android:text="Type here:" />
14
15    <EditText
16        android:id="@+id/entry"
17        android:layout_width="match_parent"
18        android:layout_height="wrap_content"
19        android:layout_below="@id/textview1"
20        android:hint="Input disini"
21        android:inputType="text" />
22
23    <Button
24        android:id="@+id/ok"
25        android:layout_width="wrap_content"
26        android:layout_height="wrap_content"
27        android:layout_below="@id/entry"
28        android:layout_alignParentRight="true"
29        android:layout_marginLeft="10dp"
30        android:text="OK" />
31
32    <Button
33        android:layout_width="wrap_content"
34        android:layout_height="wrap_content"
35        android:layout_alignTop="@id/ok"
36        android:layout_toLeftOf="@id/ok"
37        android:text="Cancel" />
38 </RelativeLayout>
39

```

- Pastikan pada MainActivity.kt seperti berikut:

```

1 package com.example.praktikummobile2023
2
3 import androidx.appcompat.app.AppCompatActivity

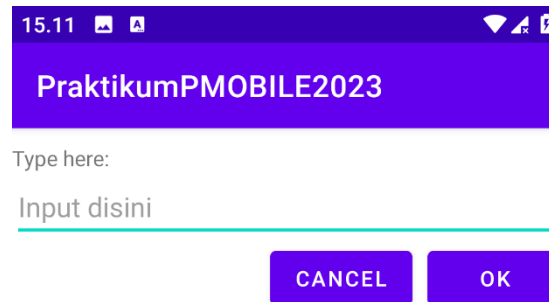
```

```

4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.relative_layout)
10    }
11 }

```

- e. Lakukan Run dengan shortcut **SHIFT+F10**.
- f. Pastikan tampilan AVD anda seperti Gambar 2.6



Gambar 2.6 Percobaan Relative Layout

3. Percobaan 3 (Constraint Layout)

- a. Buatlah satu file baru xml dengan klik kanan para folder **layout**, lalu pilih New->Xml->Layout Xml File. Beri nama **constraint_layout** lalu **Finish**.
- b. Isikan kotak dialog new seperti berikut
- c. Pada **constraint_layout.xml**. sesuaikan menjadi seperti berikut:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <ImageView
9         android:id="@+id/imageView"
10        android:layout_width="150dp"
11        android:layout_height="150dp"
12        app:layout_constraintBottom_toBottomOf="parent"
13        app:layout_constraintEnd_toEndOf="parent"
14        app:layout_constraintHorizontal_bias="0.498"
15        app:layout_constraintStart_toStartOf="parent"
16        app:layout_constraintTop_toTopOf="parent"
17        android:background="@color/black"
18        android:src="@drawable/ic_launcher_foreground"/>
19
20    <Button
21        android:id="@+id/button"

```

```

22     android:layout_width="wrap_content"
23     android:layout_height="wrap_content"
24     android:text="Submit"
25     app:layout_constraintBottom_toBottomOf="parent"
26     app:layout_constraintEnd_toEndOf="@+id/imageView"
27     app:layout_constraintStart_toStartOf="@+id/imageView"
28     app:layout_constraintTop_toBottomOf="@+id/imageView" />
29
30 <Button
31     android:layout_width="wrap_content"
32     android:layout_height="wrap_content"
33     android:text="Submit"
34     app:layout_constraintEnd_toEndOf="@+id/imageView"
35     app:layout_constraintStart_toStartOf="@+id/imageView"
36     app:layout_constraintTop_toBottomOf="@+id/button" />
37 </androidx.constraintlayout.widget.ConstraintLayout>

```

d. Ubahlah file MainActivity.java menjadi seperti ini:

```

1 package com.example.praktikummobile2023
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.constraint_layout)
10    }
11 }

```

e. Lakukan Run dengan shortcut **SHIFT+F10**.

f. Pastikan tampilan AVD anda seperti Gambar 2.7



Gambar 2.7 Percobaan Constraint Layout

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Menerapkan/membuat desain layout untuk aplikasi Movie Catalog. Mulai dari menu awal sampai kepada konten didalam masing2 menu, dengan menggunakan jenis-jenis layout yang telah dipelajari.	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-04	CPMK-02	20%		
2.	Praktik	CPL-04	CPMK-02	30%		
3.	Post-Test	CPL-04	CPMK-02	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 3: PENGENALAN WIDGET FORM

Pertemuan ke : 3

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPMK-02	Mahasiswa mampu mengoperasikan bahasa pemrograman berorientasi obyek untuk memecahkan masalah dalam lingkup pemrograman mobile

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Membuat sebuah tampilan Form Widget Basic Views seperti Textview, Radiobutton, RadioGroup, Check Box, Edittext, Button, ImageButton
2. Mempraktekkan tampilan Form Widget Basic Views dalam aplikasi yang dibuat

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-04	CPMK-02	Membuat sebuah tampilan Form Widget Basic Views seperti Textview, Radiobutton, RadioGroup, Check Box, Edittext, Button, ImageButton
CPL-04	CPMK-02	Mempraktekkan tampilan Form Widget Basic Views dalam aplikasi yang dibuat

1.3. TEORI PENDUKUNG

Basic Views merupakan alat dasar yang digunakan untuk menampilkan sesuatu di dalam android sesuai dengan kebutuhannya, umumnya seperti TextView, EditText, Button, ImageButton, ImageView, Checkbox, ToggleButton, RadioButton, dan RadioGroup.

a. Atribut

Untuk menentukan bagaimana objek View akan ditampilkan di layar maka kita perlu menambahkan atribut pada objek View. Kita dapat menambahkan atribut pada tag xml. Contoh : Mengatur atribut panjang dan tinggi suatu View dapat kita tentukan dengan menuliskan kode di bawah :

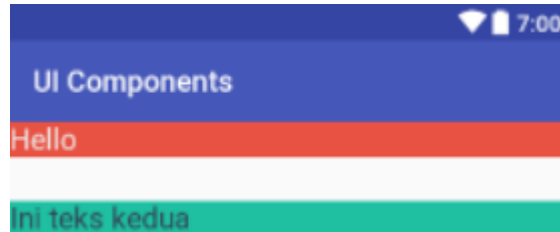
```
android:layout_width="match_parent"
android:layout_height="wrap_content"
```

b. Identifier

Kita menggunakan atribut “id” pada tag xml untuk memberikan identifier pada objek View kita. Id tersebut akan kita gunakan untuk mereferensikan objek tersebut. Gunakan atribut `id="@+id/namald”` untuk menambahkan id pada objek View.

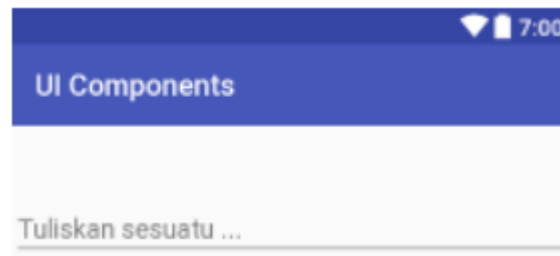
Adapun penjelasan dari masing-masing basic view tersebut:

a. **TextView** digunakan untuk menampilkan teks di layar.



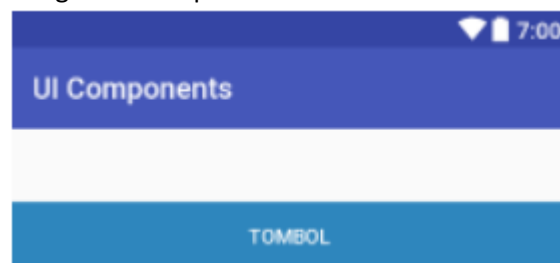
Gambar 3.1 Tampilan TextView

b. **EditText** merupakan subclass dari Views TextView, bedanya memungkinkan pengguna untuk mengedit isi teks.



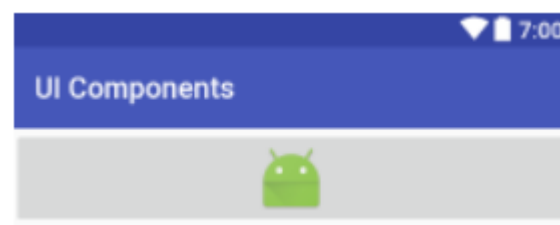
Gambar 3.2 Tampilan EditText

c. **Button** mewakili sebuah widget tombol push.



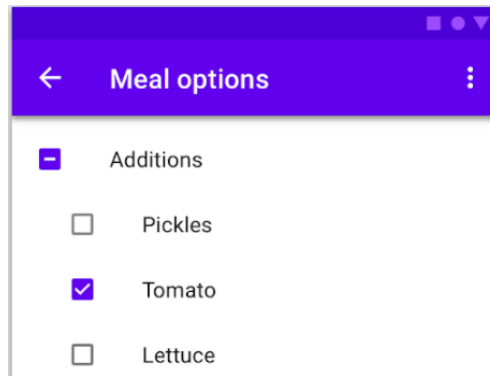
Gambar 3.3 Tampilan button

d. **ImageButton** mirip dengan tampilan Button, perbedaannya dapat menampilkan gambar sebagai tombol



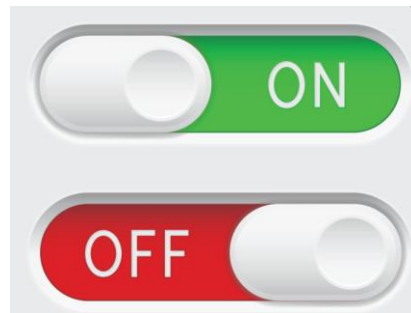
Gambar 3.4 Tampilan ImageButton

e. **CheckBox** berfungsi sebagai tipe khusus dari tombol yang memiliki dua keadaan yaitu checked atau unchecked



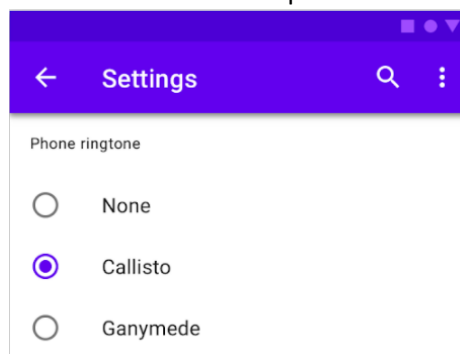
Gambar 3.5 Tampilan Checkbox

- f. **ToggleButton** berfungsi untuk menampilkan keadaan checked atau unchecked menggunakan lampu indikator



Gambar 3.6 Tampilan ToggleButton

- g. **RadioButton** memiliki dua keadaan yaitu checked atau unchecked. Sekali sebuah RadioButton diberi checked, maka tidak dapat di unchecked kembali. Sebuah RadioGroup digunakan untuk mengelompokkan satu atau lebih views RadioButton, dengan demikian mengizinkan hanya satu RadioButton yang akan di checked dalam RadioGroup



Gambar 3.7 Tampilan RadioButton

- c. **View Binding**
View binding memungkinkan pengikatan langsung antara file layout XML dan kode Java/Kotlin tanpa perlu menggunakan **findViewById()**. Langkah-langkah penggunaannya meliputi mengaktifkan view binding di proyek Android Studio.
- d. **Intent**
Intent adalah mekanisme yang digunakan dalam platform Android untuk berkomunikasi antara komponen aplikasi, seperti Activity, Service, BroadcastReceiver, dan ContentProvider. Intent digunakan untuk memulai komponen baru, memulai layanan latar belakang, mengirim dan menerima pesan antar komponen, dan melakukan tugas lain yang melibatkan komunikasi antar komponen dalam aplikasi Android.
Intent dapat memiliki dua jenis :

a) **Explicit Intent**

berfungsi untuk mengaktifkan komponen-komponen dalam satu aplikasi yang sama. Misalnya seperti : Berpindah Activity.

Explicit intents mendefinisikan suatu komponen yang harus dipanggil oleh system android dengan menggunakan java class sebagai identifier

b) **Implicit Intent**

berfungsi untuk memanggil fungsi activity yang sudah ada di fungsi internal android seperti Dial Number, Open Browser dan lainnya.

Implicit intents meminta sistem untuk melakukan service tanpa memberitahu sistem yangmana suatu kelas Java harus melakukan layanan ini. Dalam membangun sebuah implicit intents Anda harus menentukan tindakan yang harus dilakukan dan opsional suatu URI yang harus digunakan untuk tindakan ini. Misalnya Anda bisa mengatakan sistem yang ingin Anda lihat (tindakan) adalah sebuah halaman web (URI). Dengan memulai intent untuk ini data sistem akan mencoba untuk menemukan sebuah aplikasi yang terdaftar untuk acara ini, misalnya browser.

Hasil Intent:

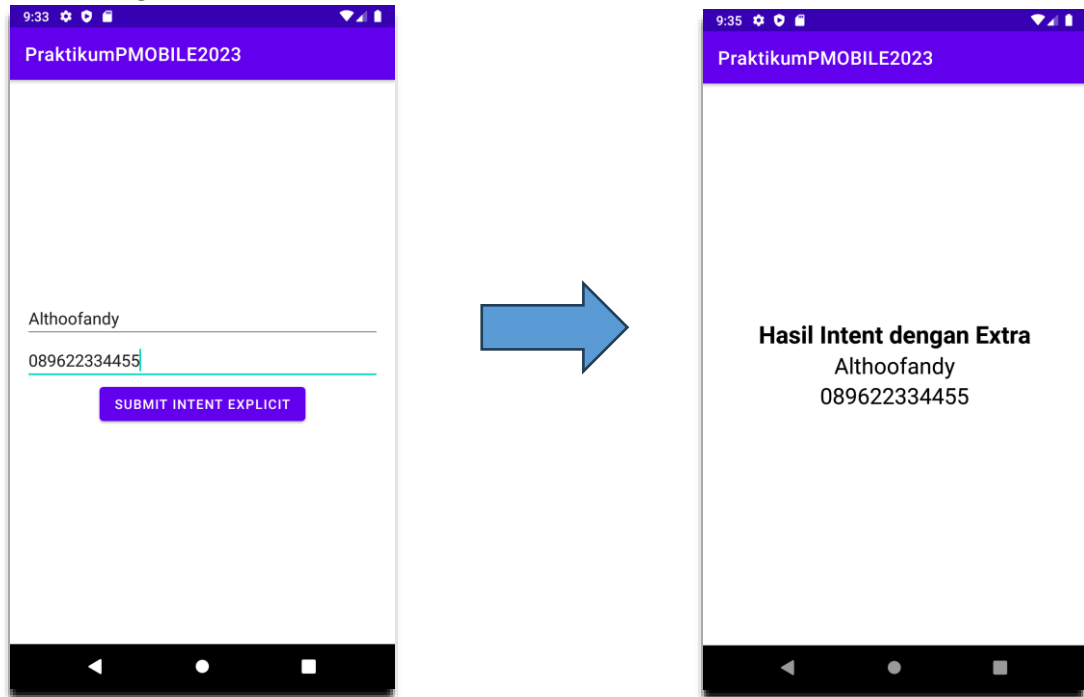
a) **Explicit Intent**



b) **Implicit Intent**



c) Mengirim data melalui Intent



Baris berikut menunjukkan bagaimana membuat sebuah explicit intent dan mengirimkan suatu variable ke system android. Jika ada class yang merepresentasikan sebuah activity Intents maka harus di start oleh android system.

```
val intent = Intent(this, ActivityTujuan::class.java)
    .putExtra("KEY_VALUE1", "Value1")
    .putExtra("KEY_VALUE2", "Value2")
    startActivity(intent)
```

Komponen yang menerima Intents dapat menggunakan metode `getIntent().getExtras()` untuk mendapatkan variable data yang dikirim.

```
val value1 = intent.getStringExtra("KEY_VALUE1")
val value2 = intent.getStringExtra("KEY_VALUE2")
```

Sedangkan implicit intents dikhususkan untuk sebuah aksi yang harus dilaksanakan dan terdapat data opsional dari aksi tersebut.

```
val intent = Intent(Intent.ACTION_VIEW).apply {
    data = Uri.parse("https://www.google.com")
}
startActivity(intent)
```

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Jelaskan pengertian widget pada android.	
2.	CPL-04	CPMK-02	Jelaskan kegunaan view binding dan intent.	

1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-04	CPMK-02	Selesaikan langkah praktikum 1 - 6	Hasil praktikum langkah 1 - 6	

Langkah-Langkah Praktikum:

1. Jalankan Android Studio, **File > New > New Project**
2. Isikan kotak dialog new seperti berikut

Project Name	My Second Android
Company Domain	praktikum.com
Build Target	API 15/Android 4.0.3
Activity Selected	Empty Activity
Activity Name	MainActivity
Layout Name	activity_main

3. Pada Package Explorer, pilih **gradle scripts/build_gradle(module:app)**. Kemudian tambahkan pada bagian **android** kode berikut.

```
buildFeatures {
    viewBinding = true
}
```

4. Pada Package Explorer, pilih **app/res/layout/activity_main.xml**. sesuaikan menjadi seperti berikut:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/activity_main"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   android:orientation="vertical"
9   tools:context="com.example.praktikumpmobile2023.MainActivity">
10
```

```

11 <TextView
12     android:id="@+id/heading"
13     android:layout_width="match_parent"
14     android:layout_height="wrap_content"
15     android:text="Silahkan Isi Nama Anda" />
16
17 <EditText
18     android:id="@+id/isinama"
19     android:layout_width="match_parent"
20     android:layout_height="wrap_content" />
21
22 <RadioGroup
23     android:id="@+id/radiogrup"
24     android:layout_width="match_parent"
25     android:layout_height="wrap_content">
26
27     <RadioButton
28         android:id="@+id/rdb1"
29         android:layout_width="match_parent"
30         android:layout_height="wrap_content"
31         android:checked="true"
32         android:text="Laki-laki" />
33
34     <RadioButton
35         android:id="@+id/rdb2"
36         android:layout_width="match_parent"
37         android:layout_height="wrap_content"
38         android:checked="false"
39         android:text="Perempuan" />
40 </RadioGroup>
41
42 <CheckBox
43     android:id="@+id/checkAgree"
44     android:layout_width="match_parent"
45     android:layout_height="wrap_content"
46     android:text="Semua Yang Diisikan Benar" />
47
48 <ImageButton
49     android:layout_width="match_parent"
50     android:layout_height="wrap_content"
51     android:id="@+id/btnSubmit"
52     android:src="@mipmap/ic_launcher"/>
53 </LinearLayout>

```

5. Buatlah sebuah Activity baru dengan nama HasilActivity.kt, kemudian ubah layout **activity_hasil** pada direktori **app/res/layout** menjadi sebagai berikut:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6

```

```

7 <TextView
8     android:id="@+id/tampilNama"
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    android:text="Tampil Nama" />
12
13 <TextView
14     android:id="@+id/tampilJenisKelamin"
15     android:layout_width="match_parent"
16     android:layout_height="wrap_content"
17     android:text="Tampil Jenis Kelamin" />
18
19 <TextView
20     android:id="@+id/tampilSetuju"
21     android:layout_width="match_parent"
22     android:layout_height="wrap_content"
23     android:text="Tampil Setuju" />
24
25 </LinearLayout>

```

6. Ubah file MainActivity.java pada direktori **src** menjadi seperti berikut:

```

1 import android.content.Intent
2 import androidx.appcompat.app.AppCompatActivity
3 import android.os.Bundle
4 import com.example.praktikumpmobile2023.HasilActivity.Companion.DATA_KELAMIN
5 import com.example.praktikumpmobile2023.HasilActivity.Companion.DATA_NAMA
6 import com.example.praktikumpmobile2023.HasilActivity.Companion.DATA_SETUJU
7 import com.example.praktikumpmobile2023.databinding.ActivityMainBinding
8
9 class MainActivity : AppCompatActivity() {
10     private lateinit var binding: ActivityMainBinding
11     private var jenisKelamin = ""
12     private var tampilSetuju = true
13
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         binding = ActivityMainBinding.inflate(layoutInflater)
17         setContentView(binding.root)
18
19         binding.apply {
20             btnSubmit.setOnClickListener {
21                 val nama = isinama.text.toString()
22                 if (rdb1.isChecked) {
23                     jenisKelamin = "Laki-laki"
24                 }
25                 if (rdb2.isChecked) {
26                     jenisKelamin = "Perempuan"
27                 }
28                 tampilSetuju = checkAgree.isChecked()
29                 val intent = Intent(this@MainActivity, HasilActivity::class.java)
30                 .putExtra(DATA_NAMA, nama)

```

```

31         .putExtra(DATA_KELAMIN,jenisKelamin)
32         .putExtra(DATA_SETUJU,tampilSetuju)
33         startActivity(intent)
34     }
35 }
36 }
37 }

```

7. Ubah file HasilActivity.java pada direktori **src** menjadi seperti berikut

```

1
2 import android.graphics.Color
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import com.example.praktikummobile2023.databinding.ActivityHasilBinding
6
7 class HasilActivity : AppCompatActivity() {
8     private lateinit var binding: ActivityHasilBinding
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         binding = ActivityHasilBinding.inflate(layoutInflater)
12         setContentView(binding.root)
13
14         val nama = intent.getStringExtra(DATA_NAMA)
15         val jenisKelamin = intent.getStringExtra(DATA_KELAMIN)
16         val resultSetuju = intent.getBooleanExtra(DATA_SETUJU,false)
17
18         binding.apply {
19             tampilNama.text = nama
20             tampilJenisKelamin.text = jenisKelamin
21
22             if(jenisKelamin=="Laki-laki"){
23                 tampilJenisKelamin.setBackgroundColor(Color.GRAY)
24             }
25             if(jenisKelamin=="Perempuan"){
26                 tampilJenisKelamin.setBackgroundColor(Color.MAGENTA)
27             }
28
29             if(resultSetuju==true){
30                 tampilSetuju.setText("Data yang diisikan benar")
31                 tampilSetuju.setBackgroundColor(Color.GREEN)
32             }else{
33                 tampilSetuju.setText("Ada yang salah!")
34                 tampilSetuju.setBackgroundColor(Color.RED)
35             }
36         }
37     }
38     companion object{
39         const val DATA_NAMA = "data_nama"
40         const val DATA_KELAMIN = "data_kelamin"
41         const val DATA_SETUJU = "data_setuju"
42     }
43 }
44

```

45

8. Lakukan Run dengan shortcut **SHIFT+F10**

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Dari desain layout yang telah dibuat pada praktikum kedua, silahkan ditambahkan dengan widget yang sesuai dengan kebutuhannya	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 4: ROOM DATABASE

Pertemuan ke : 4

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPMK-02	Mahasiswa mampu mengoperasikan bahasa pemrograman berorientasi obyek untuk memecahkan masalah dalam lingkup pemrograman mobile

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Memahami mekanisme/siklus Activity

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-04	CPMK-02	Mempraktekkan berbagai jenis aktifitas dalam siklus Activity
--------	---------	--

1.3. TEORI PENDUKUNG

Room Database adalah library dari Android Jetpack yang menyediakan cara lebih mudah untuk membuat, mengelola, dan berinteraksi dengan database SQLite pada aplikasi Android. Room menyediakan lapisan abstraksi di atas SQLite untuk menyederhanakan penggunaan database dalam proyek Android.

Room Database adalah salah satu cara untuk menyimpan data secara lokal di aplikasi Android. Dalam konteks ini, penyimpanan lokal berarti data disimpan di perangkat pengguna, biasanya di dalam database SQLite, sehingga dapat diakses dan digunakan ketika aplikasi dijalankan, bahkan tanpa koneksi internet.

Pengertian dan fitur utama dari Room Database:

1. **Basis Data SQLite:** Room menggunakan SQLite sebagai basis data lokal untuk menyimpan data secara persisten dalam aplikasi Android.
2. **Lapisan Abstraksi:** Room menyediakan lapisan abstraksi yang memungkinkan Anda berinteraksi dengan database menggunakan konsep objek daripada menggunakan kueri SQL mentah. Anda

dapat mengakses database menggunakan metode yang didefinisikan di dalam DAO (Data Access Object) tanpa harus menulis kueri SQL secara langsung.

3. **Entity:** Entity adalah representasi kelas Java yang mewakili tabel dalam database SQLite. Setiap entitas (Entity) biasanya memiliki atribut yang mewakili kolom dalam tabel.
4. **DAO (Data Access Object):** DAO adalah antarmuka yang berisi deklarasi metode untuk mengakses data dari database. Metode-metode ini dapat digunakan untuk mengambil, menyisipkan, memperbarui, atau menghapus entitas dari database.
5. **Database:** Database adalah entitas yang mewakili basis data SQLite dalam aplikasi Anda. Ini biasanya merupakan kelas abstract yang mengextend RoomDatabase. Database menyediakan metode untuk mendapatkan instance DAO dan menginisialisasi database.
6. **Migrasi:** Room menyediakan mekanisme migrasi yang memungkinkan Anda memperbarui skema database tanpa kehilangan data yang ada.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Jelaskan pengertian room database	
2.	CPL-04	CPMK-02	Dimanakah data akan disimpan ketika menggunakan room database?	

1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-04	CPMK-02	Selesaikan langkah praktikum pada percobaan 1	Hasil praktikum Percobaan 1	
2.	CPL-04	CPMK-02	Selesaikan langkah praktikum pada percobaan 2	Hasil praktikum Percobaan 2	

Langkah-Langkah Praktikum:

1. Percobaan 1

- a. Jalankan Android Studio, **Buatlah project baru**
- b. Isikan kotak dialog new seperti berikut

Project Name	My Fourth Android
Company Domain	praktikum.com
Build Target	API 15/Android 4.0.3
Activity Selected	Empty Activity

Activity Name MainActivity

Layout Name activity_main

- c. Buka file build.gradle(module:app) pada direktori **Gradle Script**. Modifikasi kode plugins pada baris pertama menjadi seperti ini.

```
plugins {
    id "com.android.application"
    id "org.jetbrains.kotlin.android"
    id "kotlin-kapt"
    id "kotlin-parcelize"
}
```

- d. Kemudian pada bagian **dependencies**, tambahkan library berikut.

```
implementation "androidx.room:room-runtime:2.5.2"
kapt "androidx.room:room-compiler:2.5.2"
```

- e. Setelah berhasil, silahkan klik **sync now**.
f. Ubah file MainActivity.java pada direktori **src** menjadi seperti berikut:

```
import android.R
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ArrayAdapter
import androidx.lifecycle.LiveData
import androidx.lifecycle.Observer
import com.example.praktikumpmobile2023.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private lateinit var dbBarang: DatabaseBarang
    private lateinit var barangDao: BarangDao
    private lateinit var appExecutors: AppExecutor

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        appExecutors = AppExecutor()
        dbBarang = DatabaseBarang.getDatabase(applicationContext)
        barangDao = dbBarang.barangDao()

        binding.apply {
            fabAdd.setOnClickListener {
                appExecutors.diskIO.execute {
```

```

        val barangTitles = listOf("Meja", "Semen", "Triplek", "Pasir")
        val jenisBarang = listOf("Perabotan", "Material", "Material", "Material")
        val hargaBarang = listOf(50000, 48000, 15000, 68000)
        for(i in 1..4){
            val newBarang = Barang(i, barangTitles[i-1], jenisBarang[i-1], hargaBarang[i-1])
            barangDao.insert(newBarang)
        }
    }
}

val barangList: LiveData<List<Barang>> = barangDao.getAllBarang()
barangList.observe(this@MainActivity, Observer { list ->
    val namaBarangList = list.map { it.nama }
    lvRoomDb.adapter = ArrayAdapter(this@MainActivity,
    android.R.layout.simple_list_item_1, namaBarangList)
    lvRoomDb.setOnItemClickListener { _, _, position, _ ->
        val selectedBarang = list[position]

        // Dapatkan ID atau data lain yang perlu dikirim ke halaman detail
        val detailIntent = Intent(this@MainActivity, DetailActivity::class.java)
        detailIntent.putExtra("barang_id", selectedBarang.id) // Contoh: Kirim ID
barang
        startActivity(detailIntent)
    }
})
}
}
}

```

g. Kemudian ubahlah **activity_main** sebagai berikut

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <ListView
        android:id="@+id/lvRoomDb"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fabAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"

```

```

        android:layout_alignParentEnd="true"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:src="@drawable/baseline_add_24"
        app:fabSize="normal"
    />

```

```
</RelativeLayout>
```

h. Kemudian buat lah data kelas **Barang**

```

import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity
data class Barang(
    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "id")
    var id: Int = 0,

    @ColumnInfo(name = "nama")
    var nama: String? = null,

    @ColumnInfo(name = "jenis")
    var jenis: String? = null,

    @ColumnInfo(name = "harga")
    var harga: Int? = null,
)

```

i. Buat kelas **BarangDao**

```

import androidx.lifecycle.LiveData
import androidx.room.*

@Dao
interface BarangDao {
    @Insert(onConflict = OnConflictStrategy.IGNORE)
    fun insert(barang: Barang)

    @Update
    fun update(barang: Barang)

    @Delete
    fun delete(barang: Barang)

    @Query("SELECT * from barang ORDER BY id ASC")
    fun getAllBarang(): LiveData<List<Barang>>

    @Query("SELECT * FROM barang WHERE id = :barangId")
    fun getBarangById(barangId: Int): Barang

```

```
}
```

j. Buatlah kelas **DatabaseBarang**

```
import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase

@Database(entities = [Barang::class], version = 1, exportSchema = false)
abstract class DatabaseBarang : RoomDatabase() {

    abstract fun barangDao(): BarangDao

    companion object {
        @Volatile
        private var INSTANCE: DatabaseBarang? = null

        fun getDatabase(context: Context): DatabaseBarang {
            return INSTANCE ?: synchronized(this) {
                val instance = Room.databaseBuilder(
                    context.applicationContext,
                    DatabaseBarang::class.java,
                    "db_barang"
                ).build()
                INSTANCE = instance
                instance
            }
        }
    }
}
```

k. Kemudian buatlah kelas **AppExecutor**

```
import android.os.Handler
import android.os.Looper
import java.util.concurrent.Executor
import java.util.concurrent.Executors

class AppExecutor {
    //kelas ini berfungsi untuk menjalankan database lokal diluar main thread
    //tanpa kelas ini maka proses CRUD tidak dapat dilakukan/error
    val diskIO: Executor = Executors.newSingleThreadExecutor()
    val networkIO: Executor = Executors.newFixedThreadPool(3)
    val mainThread: Executor = MainThreadExecutor()

    private class MainThreadExecutor : Executor {
        private val mainThreadHandler = Handler(Looper.getMainLooper())

        override fun execute(command: Runnable) {
            mainThreadHandler.post(command)
        }
    }
}
```

```
}
```

- l. Kemudian buatlah Activity baru bernama **DetailActivity**
- m. Modifikasi layout activity_detail seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp"
    tools:context=".DetailActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Data Barang"
        android:gravity="center"
        android:textSize="24sp"
        android:textColor="@color/black"
        android:textStyle="bold"
    />

    <EditText
        android:id="@+id/etNama"
        android:layout_width="match_parent"
        android:hint="Nama Barang"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/etJenis"
        android:layout_width="match_parent"
        android:hint="Jenis Barang"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/etharga"
        android:layout_width="match_parent"
        android:hint="Harga Barang"
        android:layout_height="wrap_content"/>

    <Button
        android:id="@+id/btnUpdate"
        android:layout_marginTop="16dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Update data"/>

    <Button
```



```

        android:id="@+id/btnDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hapus data"/>
</LinearLayout>

```

n. Selanjutnya ubah **DetailActivity** sebagai berikut.

```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.example.praktikummobile2023.databinding.ActivityDetailBinding

class DetailActivity : AppCompatActivity() {
    private lateinit var binding: ActivityDetailBinding
    private lateinit var appExecutors: AppExecutor
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityDetailBinding.inflate(layoutInflater)
        setContentView(binding.root)
        appExecutors = AppExecutor()
        val barangId = intent.getIntExtra("barang_id", -1)

        if (barangId != -1) {
            appExecutors.diskIO.execute {
                val dao = DatabaseBarang.getDatabase(this@DetailActivity).barangDao()
                val selectedBarang = dao.getBarangById(barangId)

                binding.apply {
                    etNama.setText(selectedBarang.nama)
                    etJenis.setText(selectedBarang.jenis)
                    etHarga.setText(selectedBarang.harga.toString())

                    btnUpdate.setOnClickListener {
                        val updatedBarang = selectedBarang.copy(
                            nama = etNama.text.toString(),
                            jenis = etJenis.text.toString(),
                            harga = etHarga.text.toString().toInt()
                        )
                        appExecutors.diskIO.execute {
                            dao.update(updatedBarang)
                            // Lakukan tindakan update lainnya jika diperlukan
                        }
                    }

                    btnDelete.setOnClickListener {
                        appExecutors.diskIO.execute {
                            dao.delete(selectedBarang)
                            // Lakukan tindakan delete lainnya jika diperlukan
                            finish() // Kembali ke MainActivity setelah menghapus
                        }
                    }
                }
            }
        }
    }
}

```

```
}
}
```

- o. Lakukan Run dengan shortcut **SHIFT+F10**.
- p. Setelah berhasil melakukan langkah praktikum, **harap simpan file project dengan benar**. Karena pada materi selanjutnya akan menggunakan project ini juga sebagai langkah praktikum.

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Jelaskan perbedaan keadaan antara ketika menekan tombol back dan tombol home screen.	
2.	CPL-04	CPMK-02	Dari pengujian anda jelaskan dengan bahasa anda sendiri apakah yang dimaksud dengan keadaan onCreate(), onResume(), onPause(), onStop(), onDestroy()	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 5: RECYCLERVIEW

Pertemuan ke : 5

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPMK-02	Mahasiswa mampu mengoperasikan bahasa pemrograman berorientasi obyek untuk memecahkan masalah dalam lingkup pemrograman mobile

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Membuat sebuah tampilan menu di Android

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-04	CPMK-02	Menerapkan pemrograman menu Android dengan Android Studio
--------	---------	---

1.3. TEORI PENDUKUNG

RecyclerView adalah komponen antarmuka pengguna (UI) yang disediakan oleh Android untuk menampilkan daftar item dengan cara yang lebih fleksibel dan efisien daripada ListView. Ini adalah bagian dari Android Jetpack, yang merupakan kumpulan alat untuk membantu Anda membangun aplikasi Android yang lebih baik dan lebih mudah.

Berikut adalah beberapa poin penting tentang RecyclerView:

- a. **View Recycling:** Salah satu fitur paling penting dari RecyclerView adalah kemampuannya untuk mendaur ulang tampilan (View Recycling). Ini berarti saat Anda menggulir melalui daftar, tampilan item yang tidak terlihat dihapus dari layar dan digunakan kembali untuk menampilkan item-item baru yang muncul.
- b. **LayoutManager:** RecyclerView memiliki konsep LayoutManager yang memungkinkan Anda untuk mengatur cara tampilan item ditampilkan di dalam RecyclerView. Anda dapat menggunakan LinearLayoutManager untuk membuat daftar linear vertikal atau horizontal, GridLayoutManager untuk tata letak kotak, atau StaggeredGridLayoutManager untuk tampilan kotak dengan ukuran berbeda.

- c. **Adapter:** Seperti ListView, RecyclerView juga menggunakan konsep adapter untuk menghubungkan data dengan tampilan. Anda perlu membuat kelas Adapter yang mewarisi RecyclerView.Adapter. Adapter ini bertanggung jawab untuk mengembalikan tampilan (ViewHolder) yang sesuai dan mengisi data ke dalam tampilan tersebut.
- d. **ViewHolder:** ViewHolder adalah komponen yang menyimpan referensi ke tampilan individu dalam daftar. Ini membantu dalam menghindari pemanggilan findViewById berulang-ulang, yang dapat merugikan kinerja aplikasi. Setiap item dalam RecyclerView memiliki ViewHolder yang berkorespondensi dengan tampilan item itu sendiri.
- e. **Listener dan Interaksi:** Anda dapat menambahkan pendengar klik atau interaksi lainnya pada tampilan item dalam ViewHolder. Ini memudahkan implementasi tindakan yang sesuai ketika pengguna berinteraksi dengan item dalam daftar.
- f. **Fleksibilitas dan Kustomisasi:** Dibandingkan dengan ListView, RecyclerView jauh lebih fleksibel dan memungkinkan Anda untuk mengatur tampilan dan perilaku item dengan lebih terperinci. Dalam keseluruhan, RecyclerView merupakan pilihan yang lebih disarankan saat Anda ingin menampilkan daftar item dalam aplikasi Android Anda, karena kemampuannya untuk mendaur ulang tampilan dan kinerja yang lebih baik.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Jelaskan perbedaan RecyclerView dengan ListView	
2.	CPL-04	CPMK-02	Apa yang harus disiapkan ketika ingin menggunakan RecyclerView?	

1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-04	CPMK-02	Selesaikan langkah praktikum 1 - 5	Hasil praktikum langkah 1 -5	

Langkah-Langkah Praktikum:

1. Jalankan Android Studio, bukalah project praktikum sebelumnya.
2. Ubah file activity_main.xml pada direktori res/menu menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">
```

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvRoomDb"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fabAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:src="@drawable/baseline_add_24"
    app:fabSize="normal"
/>

</RelativeLayout>

```

3. Buatlah sebuah layout bernama **item_barang.xml**. dan modifikasi xml tersebut

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardCornerRadius="20dp"
    android:layout_marginStart="12dp"
    android:layout_marginEnd="12dp"
    android:layout_marginTop="8dp"
    android:backgroundTint="@color/cardview_light_background"
    android:layout_height="80dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/tvNama"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Nama barang"
            android:textColor="@color/black"
            android:textStyle="bold"
            android:layout_marginStart="16dp"
            android:layout_marginEnd="16dp"
            android:textSize="18sp"/>

        <TextView
            android:id="@+id/tvJenis"
            android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:text="Jenis barang"
        android:textColor="@color/black"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"

        android:textSize="16sp"/>

    </LinearLayout>

</androidx.cardview.widget.CardView>

```

4. Setelah membuat layout **item_barang.xml**, buatlah kelas yang bernama **BarangAdapter**. kemudian modifikasi sebagai berikut.

```

import android.content.Intent
import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example.praktikumpmobile2023.databinding.ItemBarangBinding

class BarangAdapter(private val barangList: List<Barang>) :
    RecyclerView.Adapter<BarangAdapter.BarangViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): BarangViewHolder {
        val binding = ItemBarangBinding.inflate(LayoutInflater.from(parent.context), parent, false)
        return BarangViewHolder(binding)
    }

    override fun onBindViewHolder(holder: BarangViewHolder, position: Int) {
        val namaBarang = barangList[position]
        holder.bind(namaBarang)
    }

    override fun getItemCount(): Int {
        return barangList.size
    }

    class BarangViewHolder(private val binding: ItemBarangBinding) :
        RecyclerView.ViewHolder(binding.root) {
        fun bind(namaBarang: Barang) {
            binding.apply {
                tvNama.text = namaBarang.nama.toString()
                tvJenis.text = namaBarang.jenis.toString()

                itemView.setOnClickListener {
                    val detailIntent = Intent(it.context, DetailActivity::class.java)
                    detailIntent.putExtra("barang_id", namaBarang.id)
                    it.context.startActivity(detailIntent)
                }
            }
        }
    }
}

```

```
}
}
```

5. Setelah membuat **BarangAdapter**, selanjutnya modifikasi kelas **MainActivity.kt**.

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.lifecycle.LiveData
import androidx.lifecycle.Observer
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.praktikumpmobile2023.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private lateinit var dbBarang: DatabaseBarang
    private lateinit var barangDao: BarangDao
    private lateinit var appExecutors: AppExecutor

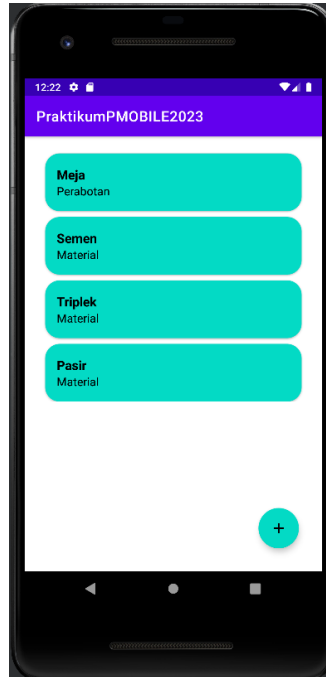
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        appExecutors = AppExecutor()
        dbBarang = DatabaseBarang.getDatabase(applicationContext)
        barangDao = dbBarang.barangDao()

        binding.apply {
            fabAdd.setOnClickListener {
                appExecutors.diskIO.execute {
                    val barangTitles = listOf("Meja", "Semen", "Triplek", "Pasir")
                    val jenisBarang = listOf("Perabotan", "Material", "Material", "Material")
                    val hargaBarang = listOf(50000, 48000, 15000, 68000)
                    for(i in 1..4){
                        val newBarang = Barang(i, barangTitles[i-1], jenisBarang[i-1], hargaBarang[i-1])
                        barangDao.insert(newBarang)
                    }
                }
            }
        }
        val barangList: LiveData<List<Barang>> = barangDao.getAllBarang()
        barangList.observe(this@MainActivity, Observer { list ->
            val layoutManager = LinearLayoutManager(this@MainActivity)
            rvRoomDb.layoutManager = layoutManager

            val adapter = BarangAdapter(list)
            rvRoomDb.adapter = adapter
        })
    }
}
```


6. Lakukan Run dengan shortcut **SHIFT+F10**.
7. Maka tampilan **RecyclerView** akan menjadi:



Gambar 5.1 Tampilan RecyclerView

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	implementasikanlah recyclerview ke dalam aplikasi yang anda buat.	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-04	CPMK-02	20%		
2.	Praktik	CPL-04	CPMK-02	30%		
3.	Post-Test	CPL-04	CPMK-02	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 6: SERVICE

Pertemuan ke : 6

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPMK-02	Mahasiswa mampu mengoperasikan bahasa pemrograman berorientasi obyek untuk memecahkan masalah dalam lingkup pemrograman mobile

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Mengetahui service pada aplikasi android
2. Menerapkan pembuatan service pada aplikasi android

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-04	CPMK-02	Mengenal tentang service pada aplikasi android
CPL-04	CPMK-02	Menerapkan pembuatan service pada aplikasi android

1.3. TEORI PENDUKUNG

Secara umum **service** adalah Komponen Aplikasi android yang bekerja dibelakang layar, tanpa user interface dan interaksi dengan user. Contohnya adalah ketika kita menggunakan Gmail yang tanpa kita buka Aplikasinya bisa menotifikasi kalau ada email baru masuk. Service bisa juga dikatakan sebuah komponen aplikasi yang bisa melakukan operasi yang berjalan lama di latar belakang dan tidak menyediakan antarmuka pengguna. Komponen aplikasi lain bisa memulai layanan dan komponen aplikasi tersebut akan terus berjalan di latar belakang walaupun pengguna beralih ke aplikasi lain. Selain itu, komponen bisa mengikat ke layanan untuk berinteraksi dengannya dan bahkan melakukan komunikasi antarproses (IPC). Misalnya, layanan mungkin menangani transaksi jaringan, memutar musik, melakukan file I/O, atau berinteraksi dengan penyedia materi dari latar belakang. Inilah tiga tipe layanan yang berbeda:

Foreground

Layanan latar depan melakukan beberapa operasi yang terlihat oleh pengguna. Misalnya, aplikasi audio akan menggunakan layanan latar depan untuk memutar track audio. Layanan latar depan harus menampilkan Notifikasi. Layanan latar depan terus berjalan bahkan saat pengguna tidak berinteraksi dengan aplikasi.

Background

Layanan latar belakang melakukan operasi yang tidak terlihat secara langsung oleh pengguna. Misalnya, jika aplikasi menggunakan layanan untuk memadatkan penyimpanannya, aplikasi tersebut biasanya akan menjadi layanan latar belakang.

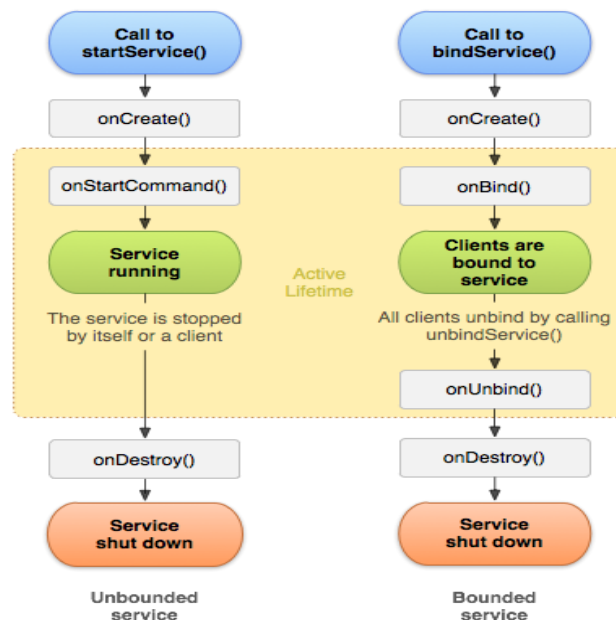
Bound

Sebuah layanan akan *bound* bila komponen aplikasi mengikatkan kepadanya dengan memanggil `bindService()`. Layanan terikat menawarkan antarmuka klien-server yang memungkinkan komponen berinteraksi dengan layanan tersebut, mengirim permintaan, mendapatkan hasil dan bahkan melakukannya pada sejumlah proses dengan komunikasi antarproses (IPC). Layanan terikat hanya berjalan selama komponen aplikasi terikat padanya. Beberapa komponen bisa diikat ke layanan sekaligus, namun bila semuanya telah dilepas, layanan akan dimusnahkan.

Walaupun dokumentasi ini umumnya membahas layanan yang dimulai dan terikat secara terpisah, layanan Anda bisa bekerja dengan dua cara—layanan tersebut bisa dimulai (untuk berjalan terus-menerus) dan juga memungkinkan pengikatan. Ini hanyalah masalah apakah Anda mengimplementasikan beberapa metode `callback: onStartCommand()` untuk memungkinkan komponen memulainya dan `onBind()` untuk memungkinkan pengikatan.

Apakah service sudah dimulai, bound, atau keduanya, semua komponen aplikasi bisa menggunakan service (bahkan dari aplikasi terpisah), demikian pula semua komponen bisa menggunakan suatu aktivitas—dengan memulainya dengan `Intent`. Akan tetapi, Anda bisa mendeklarasikan layanan sebagai *privat* pada file manifes, dan memblokir akses dari aplikasi lain. Hal ini dibahas selengkapnya di bagian tentang Mendeklarasikan layanan dalam manifes.

Seperti activity, service memiliki metode callback daur hidup yang bisa Anda implementasikan untuk memantau perubahan keadaan layanan dan melakukan pekerjaan pada waktu yang sesuai. Service berikut memperagakan setiap metode daur hidup:



Gambar 1. Daer hidup layanan. Diagram di sebelah kiri menampilkan daur hidup bila layanan dibuat dengan `startService()` dan diagram di sebelah kanan menampilkan daur hidup bila layanan dibuat dengan `bindService()`.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Jelaskan fungsi penggunaan intent dalam aplikasi android.	
2.	CPL-04	CPMK-02	Jelaskan perbedaan penggunaan implicit dan explicit intents.	

1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-04	CPMK-02	Selesaikan langkah praktikum 1 - 8	Hasil praktikum langkah 1 - 8	

Langkah-Langkah Praktikum:

1. Buat sebuah database baru pada phpmyadmin dengan nama employee.
2. Pilih database employee, dan pilih tab sql/query, lalu tuliskan kode berikut:

```
CREATE TABLE IF NOT EXISTS tbl_employee (
  employee_id int(4) NOT NULL AUTO_INCREMENT,
  employee_name varchar(60) NOT NULL,
  jabatan varchar(30) NOT NULL,
  tgl_kerja date NOT NULL,
  gaji int(10) NOT NULL,
  PRIMARY KEY (employee_id)
)
ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

INSERT INTO tbl_employee (employee_id, employee_name, jabatan,
tgl_kerja, gaji) VALUES
(1, 'Reza', 'VP', '2013-08-01', 60000),
(2, 'Adi', 'Executive', '2014-10-09', 20000),
(3, 'Isgi', 'Manager', '2013-08-20', 40000),
(4, 'Alfin', 'Executive', '2013-06-01', 25000),
(5, 'Onie', 'Trainee', '2014-10-01', 10000);
```

3. Buka editor text yang biasa digunakan , buat file baru dengan nama: **employee.php**, simpan ke direktori **C:/xampp/htdocs/**
4. Tuliskan kode berikut pada employee.php:

```
<?php
//koneksi ke database employee
$connection =
mysqli_connect("localhost","root","041013","employee") or
die("Error " . mysqli_error($connection));
```

```
//query ke table employee
$sql = "select * from tbl_employee";
$result = mysqli_query($connection, $sql) or die("Error in
Selecting " . mysqli_error($connection));

//pembuatan array
$emparray = array();
while($row =mysqli_fetch_assoc($result))
{
    $emparray[] = $row;
}

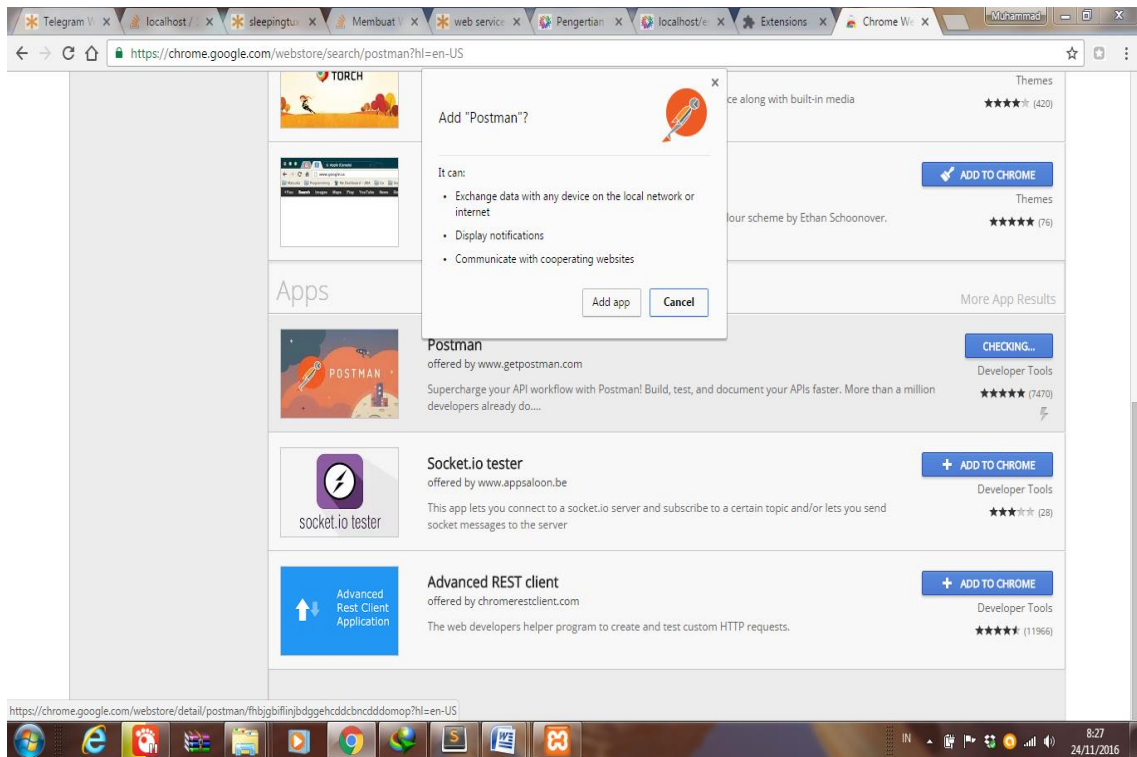
//pembuatan json
echo json_encode($emparray);

//tutup koneksi
mysqli_close($connection);
?>
```

5. Akses melalui browser pada localhost/employee.php, maka hasilnya akan seperti berikut:

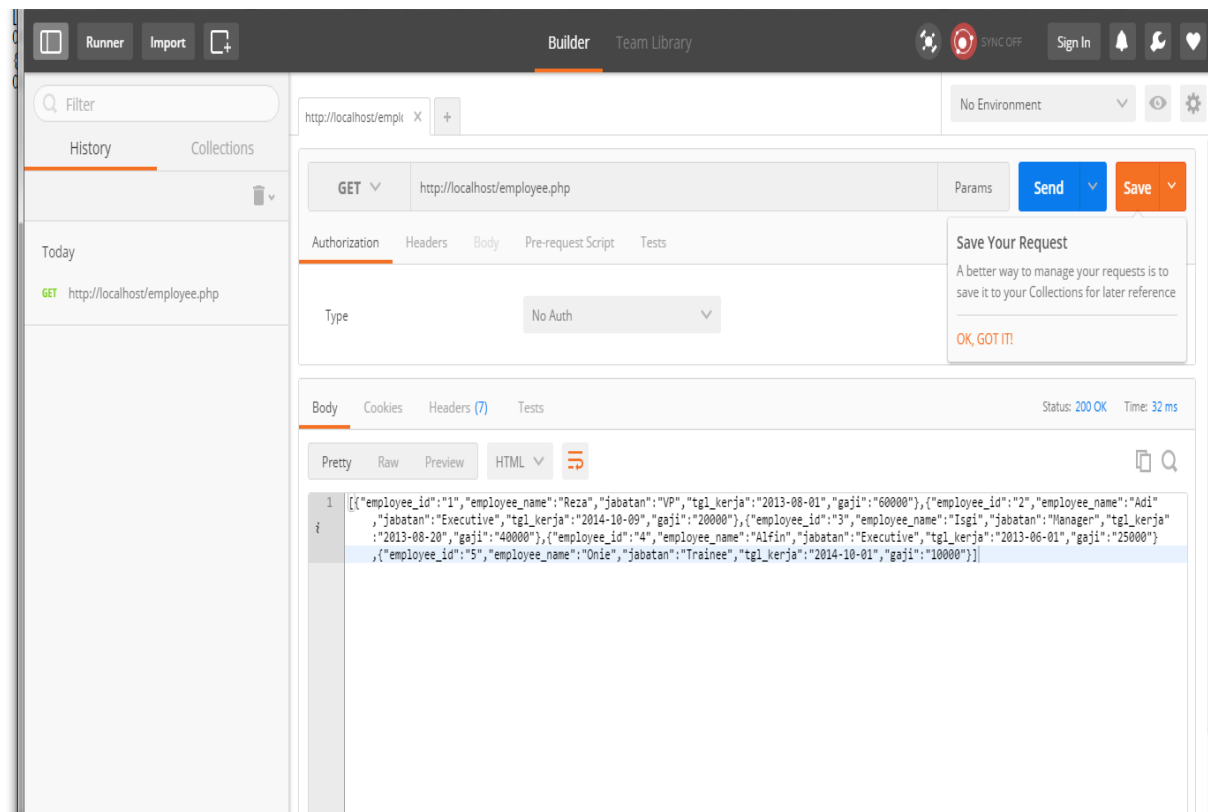
```
[{"employee_id":"1","employee_name":"Reza","jabatan":"VP","tgl_kerja":"20
13-08-
01","gaji":"60000"}, {"employee_id":"2","employee_name":"Adi","jabatan":"E
xecutive","tgl_kerja":"2014-10-
09","gaji":"20000"}, {"employee_id":"3","employee_name":"Isgi","jabatan":"
Manager","tgl_kerja":"2013-08-
20","gaji":"40000"}, {"employee_id":"4","employee_name":"Alfin","jabatan":
"Executive","tgl_kerja":"2013-06-
01","gaji":"25000"}, {"employee_id":"5","employee_name":"Onie","jabatan":
"Trainee","tgl_kerja":"2014-10-01","gaji":"10000"}]
```

6. Add Postman melalui Google Chrome anda:



Gambar 6.1 Menambahkan Postman Pada Google Chrome

7. Akses localhost/employee.php melalui postman:



Gambar 6.2 Akses Localhost

- Agar lebih baik, API dapat langsung dihosting melalui hosting gratis seperti 000webhost dan penyedia lainnya.

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-04	CPMK-02	Silahkan buat sebuah API yang menampilkan list bioskop yang ada di jogja. Harap API disimpan karena akan dipakai pada pertemuan berikutnya!	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 7: KONSUMSI REST API

Pertemuan ke : 7

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-03	Mahasiswa mampu membuat program untuk akses basis data, mampu mengkompilasi dan menjalankan sebuah aplikasi

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Mengetahui konsumsi rest API dalam android
2. Menerapkan konsumsi rest API android ke dalam aplikasi yang dibuat

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-07	CPMK-03	Mengenal tentang konsumsi rest API dalam android
CPL-07	CPMK-03	Menerapkan konsumsi rest API android ke dalam aplikasi yang dibuat

1.3. TEORI PENDUKUNG

Apa itu API?

API adalah singkatan dari Application Programming Interface yaitu sebuah software yang memungkinkan para developer untuk mengintegrasikan dan mengizinkan dua aplikasi yang berbeda secara bersamaan untuk saling terhubung satu sama lain.

Tujuan penggunaan dari API adalah untuk saling berbagi data antar aplikasi yang berbeda tersebut, Tujuan penggunaan API lainnya yaitu untuk mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah function yang terpisah sehingga para developer tidak perlu lagi membuat fitur yang serupa.

Istilah “API” sebetulnya tidak ada hubungannya dengan hal-hal yang berkaitan dengan web, karena istilah tersebut sudah ada sebelum web. Hal ini semacam dikooptasi yang berarti “pemanggilan web

service”. Tapi secara tradisional, Pengertian API bukan seperti itu. Tapi lebih berkaitan dengan fungsi-fungsi yang disediakan oleh Sistem Operasi.

Apa itu REST API?

Representational State Transfer disingkat REST. Ini adalah API yang mengikuti seperangkat aturan untuk aplikasi dan layanan untuk berkomunikasi satu sama lain.

REST API bekerja dengan mengajukan permintaan untuk sumber daya dan mengembalikan semua informasi yang relevan tentang sumber daya, diterjemahkan ke dalam format yang dapat dengan mudah ditafsirkan oleh klien (format ini ditentukan oleh API yang menerima permintaan). Klien juga dapat memodifikasi item di server dan bahkan menambahkan item baru ke server melalui REST API.

Apa itu RESTful API?

RESTful API adalah Application Programming Interface (API) yang mematuhi batasan arsitektur REST dan berinteraksi dengan layanan web RESTful. RESTful API juga mengikuti prinsip REST API. RESTful API lebih skalabel dan memiliki masa pakai yang lebih lama. RESTful API menggunakan permintaan HTTP untuk mengakses dan menggunakan data. Ada empat permintaan HTTP dasar yang dapat dibuat klien adalah:

GET — Untuk mengambil sumber daya.

POST — Untuk membuat sumber daya baru.

PUT — Untuk mengedit atau memperbarui sumber daya yang ada.

DELETE — Untuk menghapus sumber daya.

OPTION — Untuk mendapatkan operasi yang disupport oleh resource pada rest server EndPoint REST API dan RESTful API.

Saat pengguna membuat respons, API mengirim informasi ‘request’ dari aplikasi web atau server web, ia akan menerima ‘response’. Lokasi tempat API mengirim request atau tempat sumber daya berada adalah endpoint. Dengan REST API, endpoint adalah salah satu ujung saluran komunikasi.

Kenapa Kamu Harus Menggunakan API

Kenapa harus memilih menggunakan Web API? Jika dilihat dari penjelasan diatas pasti sudah paham tentang apa saja kelebihan dan manfaat yang bisa kamu dapatkan saat menggunakan Web API dalam pengembangan aplikasi.

Berikut alasan mengapa memilih Web API :

- a. Web API bersifat Open Source
- b. Saat membutuhkan Web Service dan tidak perlu SOAP, maka ASP.NET Web API adalah solusi dan pilihan terbaik.
- c. Web API dapat digunakan untuk membangun Layanan HTTP sederhana, non-SOAP di atas WCF message pipeline.
- d. Konfigurasi yang mudah dan tidak seperti pada layanan WCF REST.
- e. Pembuatan Service dengan API Web cukup sederhana, Berbeda dengan Layanan WCF REST, pembuatan service menjadi cukup sulit.
- f. Berdasarkan HTTP sehingga mudah untuk didefinisikan, mengekspos dan mengkonsumsi dengan cara RESTful.
- g. Berdasarkan arsitektur RESTful yang ringan dan bagus untuk perangkat yang memiliki bandwidth terbatas seperti ponsel pintar.

Perbedaan antara REST API dan RESTful API.

Meskipun arsitektur REST dan RESTful API serupa, keduanya berbeda dengan sedikit perbedaan. REST API mengikuti semua aturan Arsitektur REST. Ini memiliki sistem lapisan client-server, stateless, cacheable, dengan antarmuka yang seragam, sedangkan aplikasi web RESTful memiliki semua fitur arsitektur REST dengan fitur tambahan yang unik.

Di REST Protokolnya kuat, dan mewarisi banyak langkah keamanan, yang merupakan lapisan arsitektur bawaan. Dalam RESTful API Ini adalah multi-layer dan memiliki protokol transport yang membuat sistem kurang aman jika dibandingkan dengan REST.

Saat menggunakan layanan web REST API, kerja URL didasarkan pada permintaan dan respons. Cara kerja RESTful sepenuhnya didasarkan pada aplikasi REST.

Ini sangat mudah beradaptasi dan ramah pengguna untuk semua perusahaan bisnis dan IT yang menggunakan REST API, RESTful API ini terlalu fleksibel jika dibandingkan dengan layanan web RESTLESS.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Jelaskan tujuan penggunaan API dalam aplikasi android.	
2.	CPL-07	CPMK-03	RESTful API menggunakan permintaan HTTP untuk mengakses dan menggunakan data. Jelaskan empat permintaan HTTP dasar yang digunakan oleh klien.	

1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-07	CPMK-03	Selesaikan langkah praktikum 1 - 15	Hasil praktikum langkah 1 - 15	

Langkah-Langkah Praktikum:

1. Jalankan Android Studio, Buatlah Project Baru.
2. Isikan kotak dialog new seperti berikut

Project Name	KaryawanApp
Company Domain	praktikum.com
Build Target	API 28
Activity Selected	Empty Activity
Activity Name	MainActivity
Layout Name	activity_main

3. Pada praktikum kali ini kita akan menggunakan API yang sudah dibuat pada langkah praktikum pertemuan sebelumnya, yaitu `employee.php`. (silahkan pakai API <https://praktikum10consumingtest.000webhostapp.com/index.php>, jika tidak memiliki API yang telah dihosting).
4. Pertemuan kali ini akan menerapkan beberapa materi praktikum yang sudah dipelajari sebelumnya, seperti **recyclerview**, **intent** dan **service**.

5. Pertama tama silahkan tambahkan library Fast Android Networking pada **Gradle Scripts/build.gradle(module:app)** dengan library.

```
implementation "com.amitshekhra.android:android-networking:1.0.0"
```

6. Kemudian jangan lupa juga untuk mengaktifkan **viewbinding** serta **Permission internet** dan sync now

Berikut Permission Internet didalam Manifest

```
<uses-permission android:name="android.permission.INTERNET"/>
```

7. Silahkan buat data class bernama **ModelKaryawan**, dan modifikasi seperti berikut.

```
data class ModelKaryawan(
    val employee_id: String,
    val employee_name: String,
    val jabatan: String,
    val tgl_kerja: String,
    val gaji: String
)
```

8. Kemudian silahkan modifikasi kelas **MainActivity.kt** menjadi sebagai berikut.

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.recyclerview.widget.LinearLayoutManager
import com.androidnetworking.AndroidNetworking
import com.androidnetworking.common.Priority
import com.androidnetworking.interfaces.JSONArrayRequestListener
import com.example.karyawanapp.databinding.ActivityMainBinding

import org.json.JSONArray

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        AndroidNetworking.initialize(applicationContext)
        val layoutManager = LinearLayoutManager(this)
        binding.rvEmployee.layoutManager = layoutManager

        AndroidNetworking.get("https://praktikum10consumingtest.000webhostapp.com/index.php")
            .setPriority(Priority.MEDIUM)
            .build()
            .getAsJSONArray(object : JSONArrayRequestListener {
                override fun onResponse(response: JSONArray) {
                    // Tangani respons JSONArray di sini
                    val employees = mutableListOf<ModelKaryawan>()
```

```

    for (i in 0 until response.length()) {
        val employee = response.getJSONObject(i)
        val employeeId = employee.getString("employee_id")
        val employeeName = employee.getString("employee_name")
        val jabatan = employee.getString("jabatan")
        val tglKerja = employee.getString("tgl_kerja")
        val gaji = employee.getString("gaji")

        employees.add(ModelKaryawan(employeeId, employeeName, jabatan, tglKerja, gaji))
    }
    val adapter = Adapter(employees)
    binding.rvEmployee.adapter = adapter
}

override fun onError(anError: com.androidnetworking.error.ANError) {
    // Tangani kesalahan di sini
}
})
}
}

```

9. Langkah selanjutnya adalah modifikasi **activity_main.xml**.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvEmployee"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>

```

10. Setelah itu silahkan buat layout baru dengan nama **item_employee**, dan modifikasi seperti berikut.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardCornerRadius="20dp"
    android:layout_marginStart="12dp"
    android:layout_marginEnd="12dp"
    android:layout_marginTop="8dp"
    android:backgroundTint="@color/cardview_light_background"
    android:layout_height="80dp">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tvNama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nama Karyawan"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:textSize="18sp"/>
    <TextView
        android:id="@+id/tvJabatan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Jabatan"
        android:textColor="@color/black"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:textSize="16sp"/>
</LinearLayout>
</androidx.cardview.widget.CardView>

```

11. Setelah itu buatlah kelas baru bernama **Adapter**, dan modifikasi seperti berikut.

```

import android.content.Intent
import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example.karyawanapp.DetailActivity.Companion.DATA_GAJI
import com.example.karyawanapp.DetailActivity.Companion.DATA_JABATAN
import com.example.karyawanapp.DetailActivity.Companion.DATA_NAMA
import com.example.karyawanapp.DetailActivity.Companion.DATA_TGL_MASUK
import com.example.karyawanapp.databinding.ItemEmployeeBinding

class Adapter(private val data: List<ModelKaryawan>) :
    RecyclerView.Adapter<Adapter.EmployeeViewHolder>() {

    class EmployeeViewHolder(private val binding: ItemEmployeeBinding) :
        RecyclerView.ViewHolder(binding.root) {
        fun bind(employee: ModelKaryawan) {
            binding.tvNama.text = employee.employee_name
            binding.tvJabatan.text = employee.jabatan

            binding.root.setOnClickListener {
                val detailIntent = Intent(it.context, DetailActivity::class.java)
                detailIntent.putExtra(DATA_NAMA, employee.employee_name)
            }
        }
    }
}

```



```

        detailIntent.putExtra(DATA_JABATAN, employee.jabatan)
        detailIntent.putExtra(DATA_TGL_MASUK, employee.tgl_kerja)
        detailIntent.putExtra(DATA_GAJI, employee.gaji)
        it.context.startActivity(detailIntent)
    }
}
}

override fun onCreateView(parent: ViewGroup, viewType: Int): EmployeeViewHolder {
    val binding = ItemEmployeeBinding.inflate(
        LayoutInflater.from(parent.context),
        parent,
        false
    )
    return EmployeeViewHolder(binding)
}

override fun onBindViewHolder(holder: EmployeeViewHolder, position: Int) {
    val data = data[position]
    holder.bind(data)
}

override fun getItemCount(): Int = data.size
}

```

12. Silahkan membuat Activity baru dengan nama **DetailActivity.kt**, kemudian ubah kode nya menjadi sebagai berikut.

```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.example.karyawanapp.databinding.ActivityDetailBinding

class DetailActivity : AppCompatActivity() {
    private lateinit var binding: ActivityDetailBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityDetailBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val namaKry = intent.getStringExtra(DATA_NAMA)
        val jabatanKry = intent.getStringExtra(DATA_JABATAN)
        val tglMasuk = intent.getStringExtra(DATA_TGL_MASUK)
        val gajiKry = intent.getStringExtra(DATA_GAJI)

        binding.apply {
            tvNamaKaryawan.text = namaKry
            tvJabatanKaryawan.text = jabatanKry
            tvTanggalKerja.text = tglMasuk
            tvGaji.text = gajiKry
        }
    }
}

```

```

companion object{
    const val DATA_NAMA = "data_nama"
    const val DATA_JABATAN = "data_jabatan"
    const val DATA_TGL_MASUK = "data_tgl_masuk"
    const val DATA_GAJI = "data_gaji"
}
}

```

13. Dan langkah terakhir adalah modifikasi layout **activity_detail** menjadi seperti.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:background="@color/teal_200"
    android:layout_height="match_parent"
    android:gravity="center"
    android:padding="16dp"
    tools:context=".DetailActivity">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        app:cardCornerRadius="15dp"
        android:layout_height="wrap_content">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textSize="18sp"
                android:layout_marginTop="16dp"
                android:gravity="center"
                android:text="Detail Karyawan"
                android:textStyle="bold"
                android:textColor="@color/black"
                android:layout_marginBottom="16dp"/>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_weight="2"
                android:orientation="horizontal"
                android:layout_marginBottom="12dp">

                <TextView
                    android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:gravity="center"
        android:text="Nama Karyawan : "
        android:layout_weight="1"
        android:textColor="@color/black" />

<TextView
    android:id="@+id/tvNamaKaryawan"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="14sp"
    android:gravity="center"
    android:layout_weight="1"
    android:hint="Nama"
    android:textColor="@color/black" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:orientation="horizontal"
    android:layout_marginBottom="12dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:gravity="center"
        android:text="Jabatan : "
        android:layout_weight="1"
        android:textColor="@color/black" />

    <TextView
        android:id="@+id/tvJabatanKaryawan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:gravity="center"
        android:layout_weight="1"
        android:hint="Jabatan"
        android:textColor="@color/black" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:orientation="horizontal"
    android:layout_marginBottom="12dp">

```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="14sp"
    android:gravity="center"
    android:text="Tanggal Kerja :"
    android:layout_weight="1"
    android:textColor="@color/black" />

<TextView
    android:id="@+id/tvTanggalKerja"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="14sp"
    android:gravity="center"
    android:layout_weight="1"
    android:hint="Tanggal Kerja"
    android:textColor="@color/black" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:orientation="horizontal"
    android:layout_marginBottom="12dp">

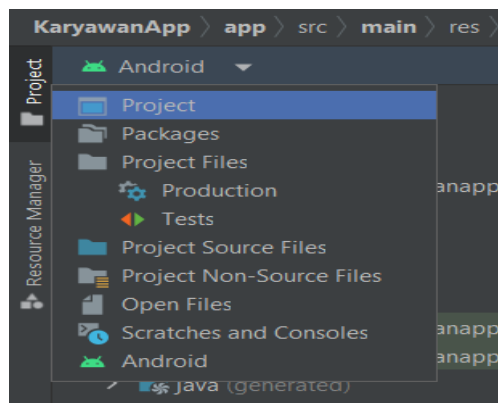
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:gravity="center"
        android:text="Gaji :"
        android:layout_weight="1"
        android:textColor="@color/black" />

    <TextView
        android:id="@+id/tvGaji"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:gravity="center"
        android:layout_weight="1"
        android:hint="Gaji"
        android:textColor="@color/black" />

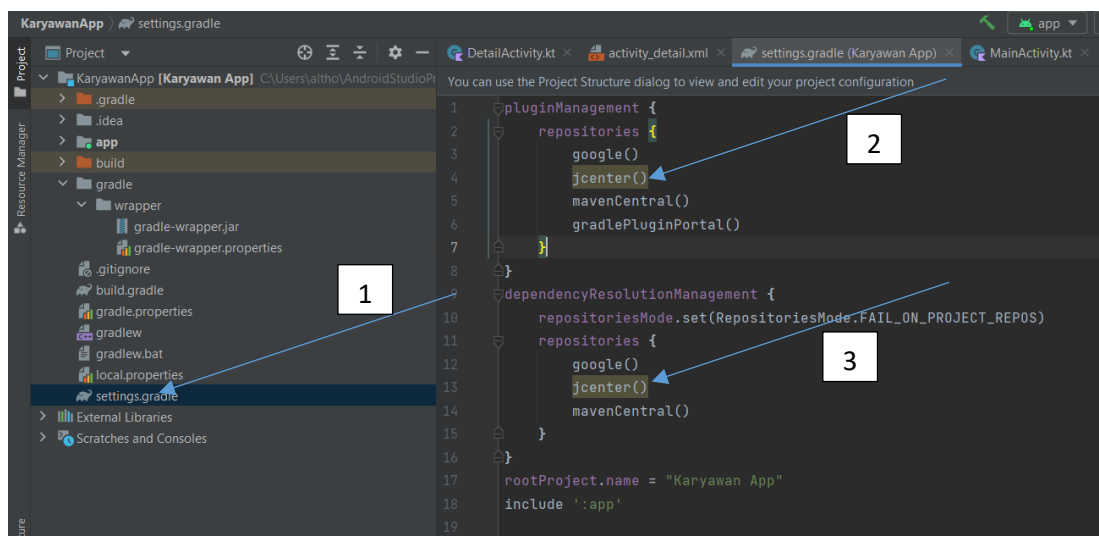
</LinearLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>

```

14. Jika pada **MainActivity.kt** masih terdapat error seperti library tidak terbaca, maka bisa melakukan penambahan repository **jcenter()** pada **settings.gradle** yang berada pada projects.



Gambar 7.1 Mengubah dari Android menjadi Project



Gambar 7.2 Penambahan repo jcenter()

15. Jika sudah silahkan melakukan sync now dan run aplikasi.

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Ulangi langkah praktikum dengan menggunakan API Bioskop yang ada di jogja yang telah dibuat pada posttest sebelumnya.	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-07	CPMK-03	20%		
2.	Praktik	CPL-07	CPMK-03	30%		

3.	Post-Test	CPL-07	CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama :	Asisten:	Tanggal:
NIM :	Paraf Asisten:	Nilai:

--

PRAKTIKUM 8: FIREBASE REALTIME DATABASE

Pertemuan ke : 8

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-03	Mahasiswa mampu membuat program untuk akses basis data, mampu mengkompilasi dan menjalankan sebuah aplikasi

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menganalisa penggunaan firebase dalam android
2. Menerapkan pembuatan database firebase ke dalam aplikasi yang dibuat

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-07	CPMK-03	Memilih database yang tepat digunakan dalam aplikasi yang dibuat
CPL-07	CPMK-03	Menerapkan pembuatan database firebase ke dalam aplikasi yang dibuat

1.3. TEORI PENDUKUNG

Mengenal Apa itu Firebase?

Firebase adalah suatu layanan dari Google untuk memberikan kemudahan bahkan mempermudah para developer aplikasi dalam mengembangkan aplikasinya. Firebase alias BaaS (*Backend as a Service*) merupakan solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan developer.

Dengan menggunakan Firebase, apps developer bisa fokus dalam mengembangkan aplikasi tanpa memberikan *effort* yang besar untuk urusan *backend*.

Singkat cerita mengenai sejarah dari Firebase didirikan pertama kali pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk Firebase yang pertama kali adalah Realtime Database. Realtime Database digunakan developer untuk menyimpan data dan *synchronize* ke banyak *user*. Kemudian ia berkembang sebagai layanan pengembang aplikasi. Pada bulan Oktober 2014, perusahaan tersebut diakuisisi oleh Google.

Mengenai segi layanan, dulu Firebase memberikan *service trial* (percobaan), namun saat ini kamu bisa memanfaatkan dan menggunakan layanan Firebase secara *free* (gratis). Tentu saja dengan adanya batasan-batasan tertentu.

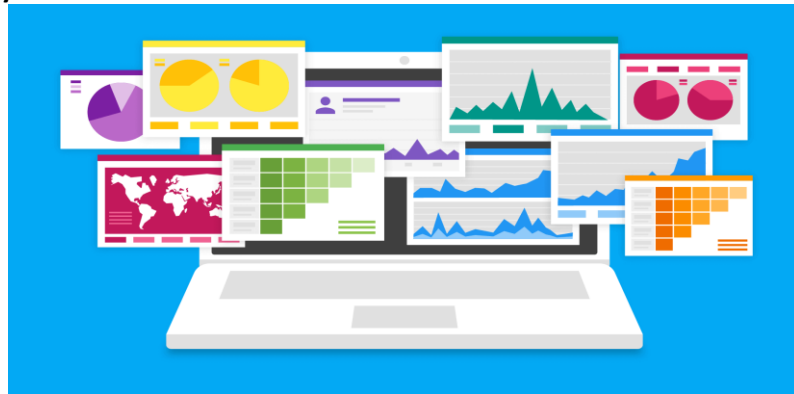
Layanan-layanan yang tersedia dari Firebase ada 2 pilihan, di antaranya:

- **SPARK** : kita bisa menggunakan layanan secara gratis.
- **BLAZE** : kita akan dikenakan biaya sesuai dengan pemakaian layanan.

Mengetahui Jenis atau Fitur Firebase

Selanjutnya kita akan membahas jenis atau fitur dari Firebase itu sendiri, berikut detailnya:

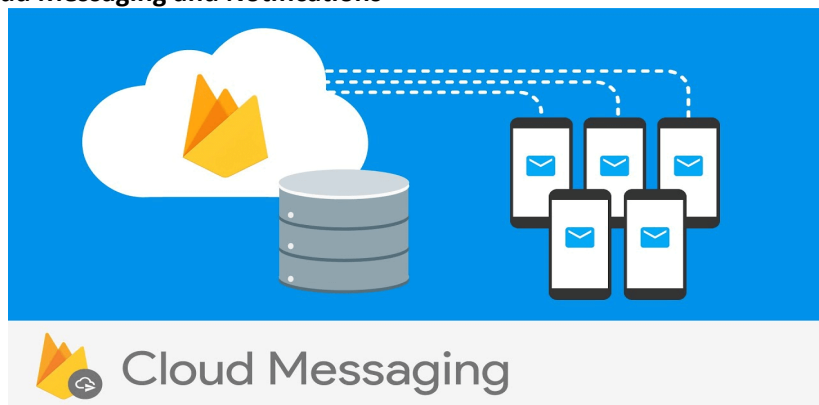
#1 Firebase Analytics



Fitur *Analytics* adalah salah satu fitur pada Firebase yang digunakan sebagai koleksi data dan *reporting* untuk aplikasi Android maupun iOS. Koleksi data pun bervariasi. Sebagai contoh, kamu dapat membuat suatu laporan atau *report* untuk pengguna aplikasi di negara Indonesia saja, atau mungkin negara lain seperti Singapura. Kamu juga bisa melihat bagian mana saja dari aplikasi yang paling sering digunakan oleh *user*.

Fitur ini mempunyai kelebihan yang memungkinkan kita untuk bisa membuat segmentasi *user* berdasarkan *user attribute*. *User attribute* adalah suatu parameter yang bisa kita gunakan sebagai *filter* yang bertujuan untuk *reporting* dan notifikasi. Contohnya pada aplikasi *online shop*. Dengan *user attribute*, kamu bisa tahu jumlah *user* yang membeli *handphone* merk 'O' atau bahkan bisa mencari tahu jam berapa transaksi yang dilakukan *user* sering terjadi.

#2 Firebase Cloud Messaging and Notifications

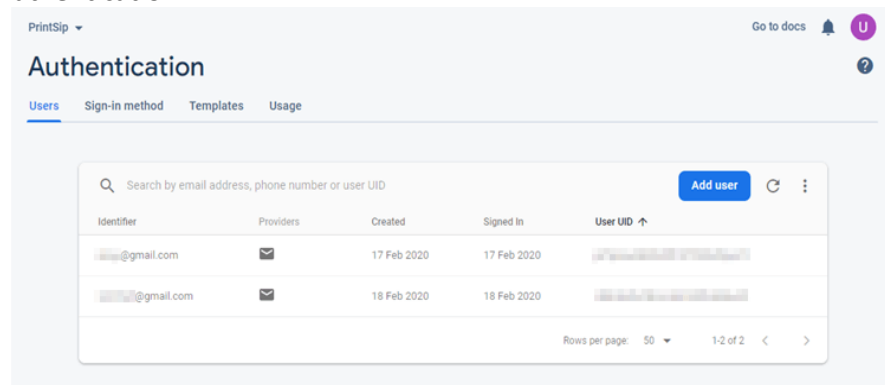


FCM (*Firebase Cloud Messaging*) yaitu menyediakan koneksi yang handal dan tentunya hemat baterai antar server maupun antar *device*. Sehingga kamu dapat mengirim dan menerima pesan serta notifikasi di Android, iOS, dan web tanpa perlu biaya.

Untuk menargetkan pesan lanjutan, kamu bisa targetkan pesan dengan mudah menggunakan *segment* yang telah ditentukan sebelumnya yakni menggunakan demografi dan *behavior*/perilaku. Anda dapat menargetkan pesan ke perangkat yang telah berlangganan pada topik tertentu. Selain itu, Anda bisa juga menargetkan hanya ke satu perangkat untuk mendapatkan informasi data yang terperinci. Biasanya ini dilakukan untuk proses pengujian.

Pesan notifikasi ini terintegrasi sepenuhnya dengan Google Analytics for Firebase, sehingga kamu memiliki akses pada interaksi dan tracking konversi secara detail. Nah, Anda dapat memantau suatu efektivitas dari satu dashboard tanpa perlu *coding* atau membuat program sendiri.

#3 Firebase Authentication



Firebase Authentication adalah salah satu layanan *back-end*, fitur Android dan iOS, SDK yang mudah digunakan, dan tampilan *interfaces* yang siap pakai untuk mengautentikasi pengguna ke aplikasi yang kamu buat. Firebase Authentication mendukung autentikasi menggunakan nomor telepon, sandi, penyedia identitas gabungan populer seperti seperti Google, Facebook, dan sebagainya.

Firebase Authentication terintegrasi dengan fitur layanan Firebase lainnya. Sistem ini memanfaatkan berbagai jenis standar industri, seperti OAuth 2.0 dan OpenID Connect, yang memudahkan integrasi dengan *backend* khusus buatanmu.

Kamu juga dapat memudahkan pengguna untuk *login* ke aplikasi dengan menggunakan fitur Firebase UI (tampilan *interfaces*), sebagai alternatif *full drop-in authentication*.

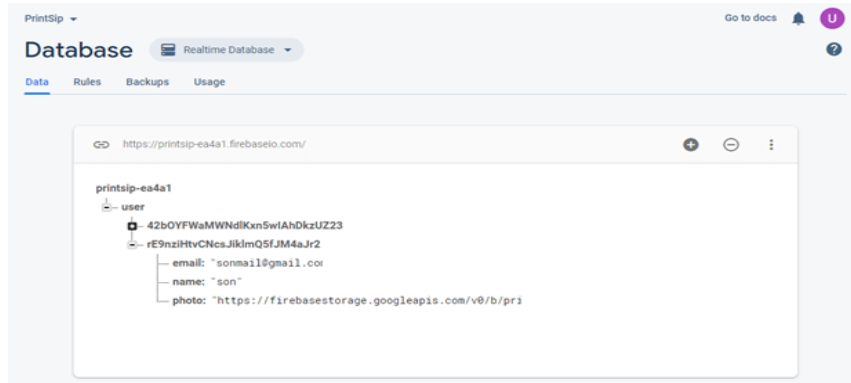
#4 Firebase Cloud Firestore



Cloud Firestore adalah *database* yang bersifat fleksibel dan terukur untuk pengembangan perangkat seperti seluler, web, dan server di Firebase dan *Google Cloud Platform*. Seperti halnya Firebase Realtime Database, Cloud Firestore membuat datamu tetap terkoneksi di aplikasi *user* melalui *listener realtime* dan menawarkan layanan secara *offline* untuk aplikasi seluler dan web. Dengan begitu, kamu dapat membuat aplikasi yang *powerfull*, responsif, dan mampu bekerja tanpa bergantung pada latensi koneksi internet.

Cloud Firestore merupakan database NoSQL yang dihosting di *cloud* dan dapat diakses melalui SDK *real* oleh aplikasi iOS, Android dan web.

#5 Firebase Realtime Database



Firebase Realtime Database adalah *database* yang di-host melalui cloud. Data disimpan dan dieksekusi dalam bentuk *JSON* dan disinkronkan secara *realtime* ke setiap *user* yang terkoneksi. Hal ini berfungsi memudahkan kamu dalam mengelola suatu *database* dengan skala yang cukup besar. Ketika kamu membuat aplikasi lintas-platform/*multiplatform* menggunakan SDK Android, iOS, dan juga JS (*JavaScript*), semua pengguna akan berbagi sebuah instance *Realtime Database* dan menerima *update*-an data secara serentak dan otomatis.

Kemampuan lain dari Firebase Realtime Database adalah tetap responsif bahkan saat *offline* karena SDK Firebase Realtime Database menyimpan data langsung ke *disk device* atau memori lokal. Setelah perangkat terhubung kembali dengan internet, perangkat pengguna (*user*) akan menerima setiap perubahan yang terjadi.

#6 Firebase Hosting



Selanjutnya ada Firebase Hosting, suatu layanan *hosting* konten web. Hanya dengan satu instruksi, kamu dapat mengimplementasikan aplikasi web serta menyajikan konten statis maupun dinamis ke CDN (jaringan penayangan konten) global dengan cepat.

Kegunaan dari Firebase Hosting itu sendiri yaitu mampu menayangkan konten melalui koneksi yang begitu aman, mengirimkan konten secara cepat, dan mendukung semua jenis konten untuk di *hosting*, mulai dari file HTML dan CSS hingga API atau layanan mikro Express.js.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio

1.5. PRE-TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Jelaskan kelebihan penggunaan firebase dalam membangun aplikasi android.	

2.	CPL-07	CPMK-03	Jelaskan sintaks SQL dalam membuat sebuah table dalam database. (dari kasus yang anda buat)	
----	--------	---------	---	--

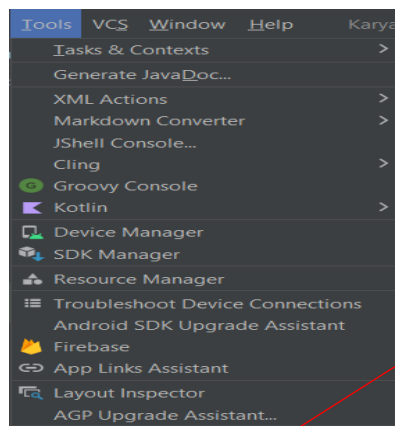
1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-07	CPMK-03	Selesaikan langkah praktikum 1 -12	Hasil praktikum langkah 1 -12	

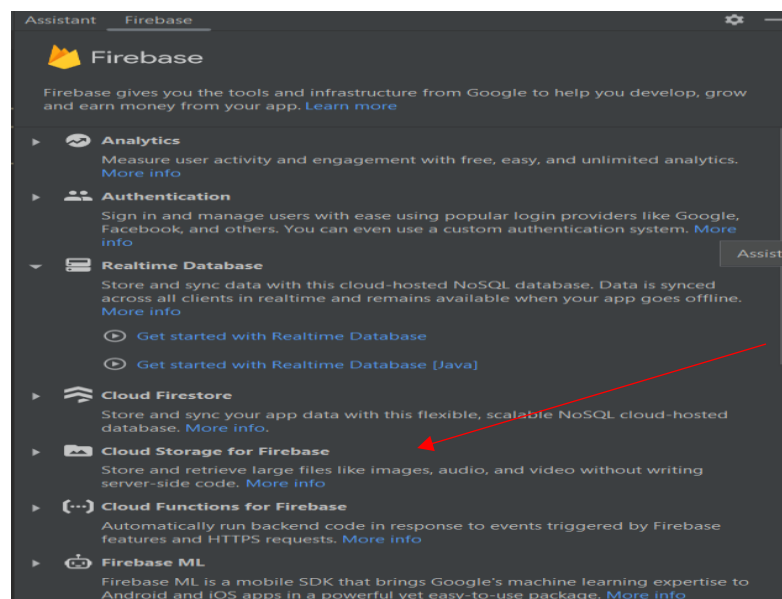
Langkah-Langkah Praktikum:

1. Jalankan Android studio, buat project baru
2. Silahkan buka project langkah praktikum pertemuan sebelumnya.
3. Kemudian silahkan koneksikan Firebase dengan project saat ini dengan cara **tools->Firebase**.

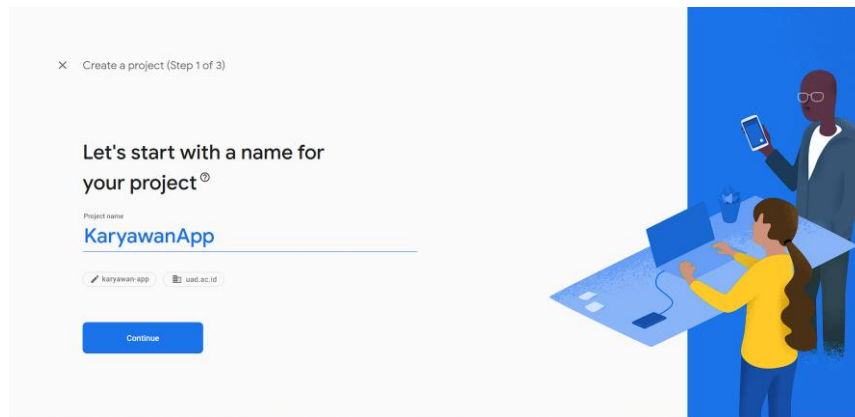


Gambar 8.1 Firebase tools

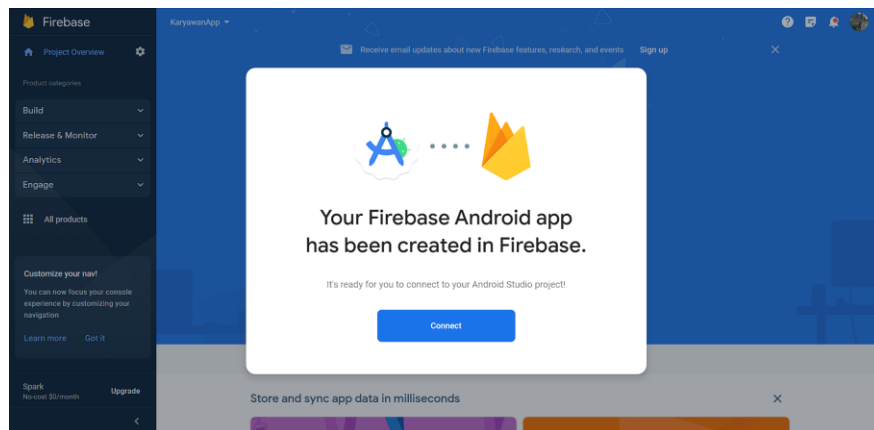
4. Setelah muncul menu fitur firebase, silahkan pilih **Realtime Database** dan pilih **Get started with Realtime Database**.



5. Kemudian silahkan koneksikan Realtime Database dengan firebase dan kemudian klik **Add Dependency**.

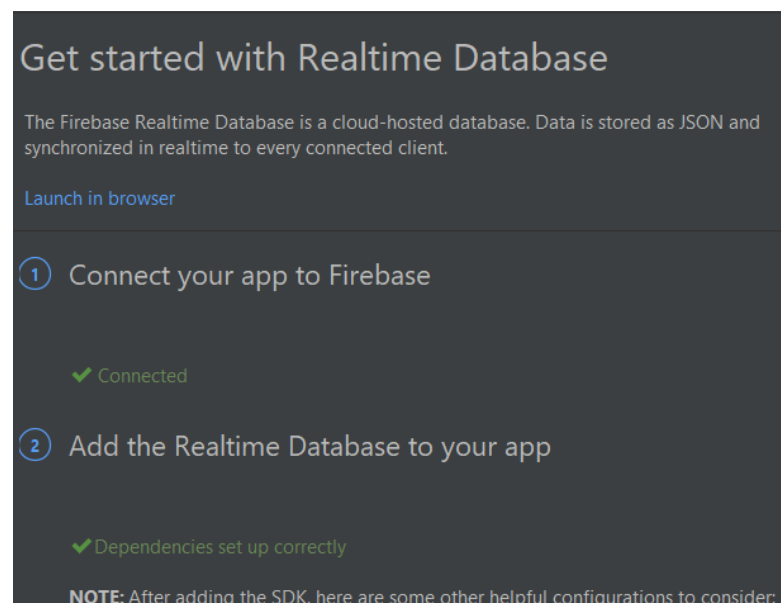


Gambar 8.2 Membuat project baru pada Firebase Console



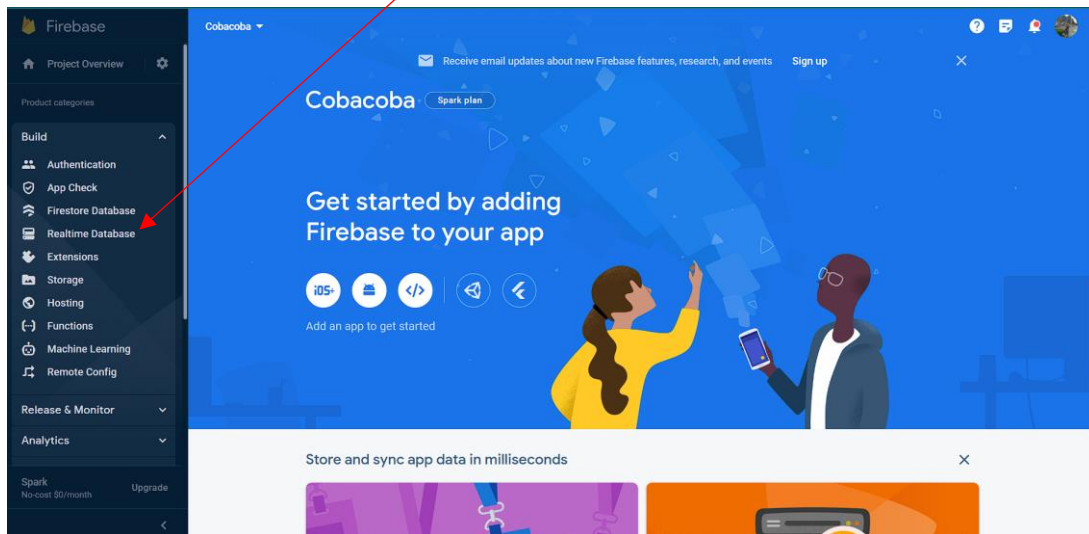
Gambar 8.3 Pop up untuk koneksi prproject

6. Jika sudah berhasil terkoneksi, maka tampilannya akan menjadi seperti.



Gambar 8.4 Gambar berhasil koneksi Firebase

7. Setelah koneksi firebase kedalam project, langkah selanjutnya adalah mengaktifkan realtime database.



Gambar 8.5 Aktifkan Realtime Database

8. Jika realtime database sudah aktif, kemudian lakukan modifikasi kode pada **activity_main.xml** menjadi seperti ini.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/rvEmployee"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

        <com.google.android.material.floatingactionbutton.FloatingActionButton
            android:id="@+id/fabAddEmployee"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="end|bottom"
            android:layout_margin="16dp"
            android:backgroundTint="@color/purple_200"
            android:src="@drawable/baseline_movie_24" />
    </FrameLayout>
</LinearLayout>
```

9. Setelah itu silahkan membuat activity baru bernama **PekerjaanActivity.kt** beserta dengan layout nya. Kemudian modifikasi **activity_pekerjaan.xml** menjadi.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp"
    android:orientation="vertical"
    tools:context=".PekerjaanActivity">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/rvPekerjaan"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

        <com.google.android.material.floatingactionbutton.FloatingActionButton
            android:id="@+id/fabAddPekerjaan"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="end|bottom"
            android:layout_margin="16dp"
            android:backgroundTint="@color/purple_200"
            android:src="@drawable/baseline_add_24" />
    </FrameLayout>
</LinearLayout>
```

10. Buat sebuah data class bernama **ModelPekerjaan** dan modifikasi kode nya:

```
data class ModelPekerjaan(
    val nama: String? = null,
    val shift: String? = null,
)
```

11. Buatlah semua layout xml dengan nama **item_pekerjaan.xml** dan ubah kode nya menjadi:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardCornerRadius="20dp"
    android:layout_marginStart="12dp"
    android:layout_marginEnd="12dp"
    android:layout_marginTop="8dp"
    android:backgroundTint="@color/teal_200"
    android:layout_height="80dp">
```



```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tvNama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nama Pekerjaan"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:textSize="18sp"/>

    <TextView
        android:id="@+id/tvShift"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Shift Pekerjaan"
        android:textColor="@color/black"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:textSize="14sp"/>

</LinearLayout>
</androidx.cardview.widget.CardView>

```

12. Kemudian buat adapter dengan nama **AdapterPekerjaan** dan modifikasi kode nya:

```

class AdapterPekerjaan(private val data: List<ModelPekerjaan>) :
    RecyclerView.Adapter<AdapterPekerjaan.PekerjaanViewHolder>() {

    class PekerjaanViewHolder(private val binding: ItemPekerjaanBinding) :
        RecyclerView.ViewHolder(binding.root) {
        fun bind(employee: ModelPekerjaan) {
            binding.tvNama.text = employee.nama
            binding.tvShift.text = employee.shift
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PekerjaanViewHolder {
        val binding = ItemPekerjaanBinding.inflate(
            LayoutInflater.from(parent.context),
            parent,
            false
        )
        return PekerjaanViewHolder(binding)
    }
}

```



```

override fun onBindViewHolder(holder: PekerjaanViewHolder, position: Int) {
    val data = data[position]
    holder.bind(data)

}
override fun getItemCount(): Int = data.size
}

```

13. Kemudian buatlah layout xml baru lagi dengan nama **upload_dialog.xml**, dan modifikasi menjadi.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <EditText
        android:id="@+id/editTextJobName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Nama Pekerjaan" />

    <EditText
        android:id="@+id/editTextShift"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Shift" />

</LinearLayout>

```

14. Kemudian ubah kode pada kelas **PekerjaanActivity** menjadi seperti:

```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.LayoutInflater
import android.widget.EditText
import android.widget.Toast
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.karyawanapp.databinding.ActivityPekerjaanBinding
import com.google.android.material.dialog.MaterialAlertDialogBuilder
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.ValueEventListener

class PekerjaanActivity : AppCompatActivity() {
    private lateinit var binding: ActivityPekerjaanBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityPekerjaanBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}

```

```

binding.rvPekerjaan.layoutManager = LinearLayoutManager(this)
binding.rvPekerjaan.setHasFixedSize(true)
showData()
tambahData()
}

private fun showData() {
    val dataRef = FirebaseDatabase.getInstance().getReference("Pekerjaan")
    dataRef.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            val pekerjaanList = mutableListOf<ModelPekerjaan>()
            val adapter = AdapterPekerjaan(pekerjaanList)
            for (dataSnapshot in snapshot.children) {
                val pekerjaanKey = dataSnapshot.getValue(ModelPekerjaan::class.java)
                pekerjaanKey?.let {
                    pekerjaanList.add(it)
                    Log.d("cek data firebase", it.toString())
                }
            }

            if (pekerjaanList.isNotEmpty()) {
                binding.rvPekerjaan.adapter = adapter
            } else {
                Toast.makeText(this@PekerjaanActivity, "Data not found",
                Toast.LENGTH_LONG).show()
            }
        }

        override fun onCancelled(snapshot: DatabaseError) {
            // Handle onCancelled
        }
    })
}

private fun tambahData(){
    val database: FirebaseDatabase = FirebaseDatabase.getInstance()
    val pekerjaanRef: DatabaseReference = database.getReference("Pekerjaan")
    binding.fabAddPekerjaan.setOnClickListener {
        val dialogView = LayoutInflater.from(this).inflate(R.layout.upload_dialog, null)

        MaterialAlertDialogBuilder(this)
            .setTitle("Tambah Pekerjaan")
            .setView(dialogView)
            .setPositiveButton("Tambah") { dialog, _ ->
                val jobName =
                dialogView.findViewById<EditText>(R.id.editTextJobName).text.toString()
                val shift = dialogView.findViewById<EditText>(R.id.editTextShift).text.toString()
                val pekerjaanData = HashMap<String, Any>()
                pekerjaanData["nama"] = jobName
                pekerjaanData["shift"] = shift

                val newPekerjaanRef = pekerjaanRef.push()
                newPekerjaanRef.setValue(pekerjaanData)
            }
    }
}

```

```

        dialog.dismiss()

    }

    .setNegativeButton("Batal") { dialog, _ ->
        dialog.dismiss()
    }

    .show()
}
}
}

```

15. Ubahlah Class MainActivity menjadi seperti berikut ini:

```

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        AndroidNetworking.initialize(applicationContext)
        val layoutManager = LinearLayoutManager(this)
        binding.rvEmployee.layoutManager = layoutManager

        AndroidNetworking.get("https://praktikum10consumingtest.000webhostapp.com/index.php")
            .setPriority(Priority.MEDIUM)
            .build()
            .getAsJSONArray(object : JSONArrayRequestListener {
                override fun onResponse(response: JSONArray) {
                    // Tangani respons JSONArray di sini
                    val employees = mutableListOf<ModelKaryawan>()

                    for (i in 0 until response.length()) {
                        val employee = response.getJSONObject(i)
                        val employeeId = employee.getString("employee_id")
                        val employeeName = employee.getString("employee_name")
                        val jabatan = employee.getString("jabatan")
                        val tglKerja = employee.getString("tgl_kerja")
                        val gaji = employee.getString("gaji")
                        Log.d("cekdata", employeeName)
                        employees.add(ModelKaryawan(employeeId, employeeName, jabatan, tglKerja, gaji))
                    }
                    val adapter = Adapter(employees)
                    binding.rvEmployee.adapter = adapter
                }

                override fun onError(anError: com.androidnetworking.error.ANError) {
                }
            })

        binding.fabAddEmployee.setOnClickListener {
            val intent = Intent(this, PekerjaanActivity::class.java)

```

```

        startActivity(intent)
    }
}

```

16. Lakukan running.

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Buatlah read dan create data kedalam realtime database firebase. Data yang diinputkan adalah nama film yang akan tayang dibioskop, jam tayang, dan harga tiket.	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-07	CPMK-03	20%		
2.	Praktik	CPL-07	CPMK-03	30%		
3.	Post-Test	CPL-07	CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 9: FIREBASE STORAGE

Pertemuan ke : 9

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-03	Mahasiswa mampu membuat program untuk akses basis data, mampu mengkompilasi dan menjalankan sebuah aplikasi

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Membuat web service dari database
2. Menerapkan pembuatan web service dengan postman ke dalam aplikasi yang dibuat

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-07	CPMK-03	Membuat web service dari database
CPL-07	CPMK-03	Membuat web service dengan postman ke dalam aplikasi yang dibuat

1.3. TEORI PENDUKUNG

Cloud Storage for Firebase dibuat untuk developer aplikasi yang perlu menyimpan dan menayangkan konten buatan pengguna, seperti foto atau video. Cloud Storage for Firebase adalah layanan penyimpanan objek yang andal, sederhana, dan hemat biaya yang dibuat untuk skala Google. Firebase SDK untuk Cloud Storage menambahkan keamanan Google pada upload dan download file untuk aplikasi Firebase Anda, bagaimanapun kualitas jaringannya.

Kemampuan utama

Operasi yang stabil	Firebase SDK untuk Cloud Storage melakukan upload dan download, bagaimanapun kualitas jaringannya. Upload dan download bersifat stabil, artinya proses ini akan dilanjutkan dari posisi berhenti, sehingga menghemat waktu dan bandwidth pengguna
---------------------	---

Keamanan yang kuat	Firebase SDK untuk Cloud Storage terintegrasi dengan Firebase Authentication untuk menyediakan autentikasi yang mudah dan intuitif bagi developer. Anda dapat menggunakan model keamanan deklaratif kami untuk mengizinkan akses berdasarkan nama file, ukuran, jenis konten, dan metadata lainnya.
Skalabilitas tinggi	Cloud Storage dibuat untuk skala exabyte ketika aplikasi Anda menjadi viral. Berkembang dengan mudah dari prototipe ke tahap produksi menggunakan infrastruktur yang sama yang mendukung Spotify dan Google Foto

Bagaimana cara kerjanya?

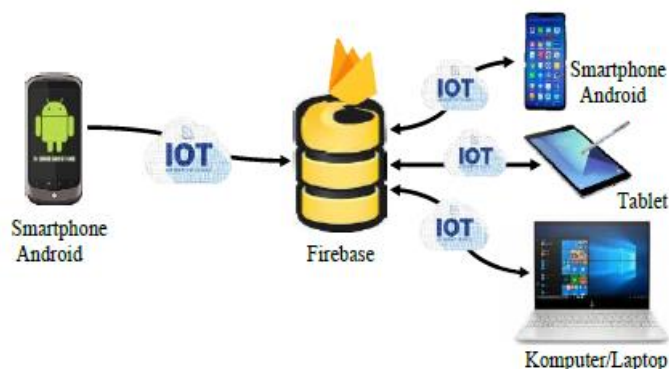
Developer menggunakan Firebase SDK untuk Cloud Storage untuk mengupload dan mendownload file langsung dari klien. Jika koneksi jaringan buruk, klien bisa mencoba operasi ini lagi dari posisi terakhir saat terhenti, sehingga menghemat waktu dan bandwidth pengguna.

Cloud Storage for Firebase menyimpan file Anda di bucket [Google Cloud Storage](#), sehingga membuatnya mudah diakses melalui Firebase dan Google Cloud. Dengan begitu, Anda memiliki fleksibilitas untuk mengupload dan mendownload file dari klien seluler melalui Firebase SDK untuk Cloud Storage. Selain itu, Anda dapat melakukan pemrosesan sisi server seperti pemfilteran gambar atau transcoding video menggunakan [Google Cloud Storage API](#). Cloud Storage otomatis melakukan penskalaan, sehingga Anda tidak perlu melakukan migrasi ke penyedia lain. Pelajari semua manfaat [integrasi kami dengan Google Cloud](#) lebih lanjut.

Firebase SDK untuk Cloud Storage terintegrasi sempurna dengan [Firebase Authentication](#) untuk mengidentifikasi pengguna, dan kami menyediakan [bahasa keamanan deklaratif](#) yang dapat Anda gunakan untuk menyetel kontrol akses pada tiap file atau kumpulan file, sehingga Anda bisa menyetelnya sebagai publik atau pribadi sesuai keinginan.

Alur implementasi

1	Mengintegrasikan Firebase SDK untuk Cloud Storage.	Sertakan klien dengan cepat melalui Gradle, CocoaPods, atau script include.
2	Membuat Referensi	Referensi jalur file, seperti "images/mountains.png", untuk mengupload, mendownload, atau menghapusnya.
3	Mengupload atau Mendownload	Upload atau download ke jenis native dalam memori atau disk.
4	Mengamankan File	Gunakan Aturan Keamanan Firebase untuk Cloud Storage untuk mengamankan file Anda.
5	(Opsional) Membuat dan membagikan URL Download	Gunakan Firebase Admin SDK untuk membuat URL yang dapat dibagikan agar pengguna dapat mendownload objek.



Gambar 2. Arsitektur Sistem Database Firebase

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio
3. MySQL

1.5. PRE-TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Jelaskan jenis-jenis yang ada pada Firebase Database	
2.	CPL-07	CPMK-03	Jelaskan cara kerja Firebase Database	

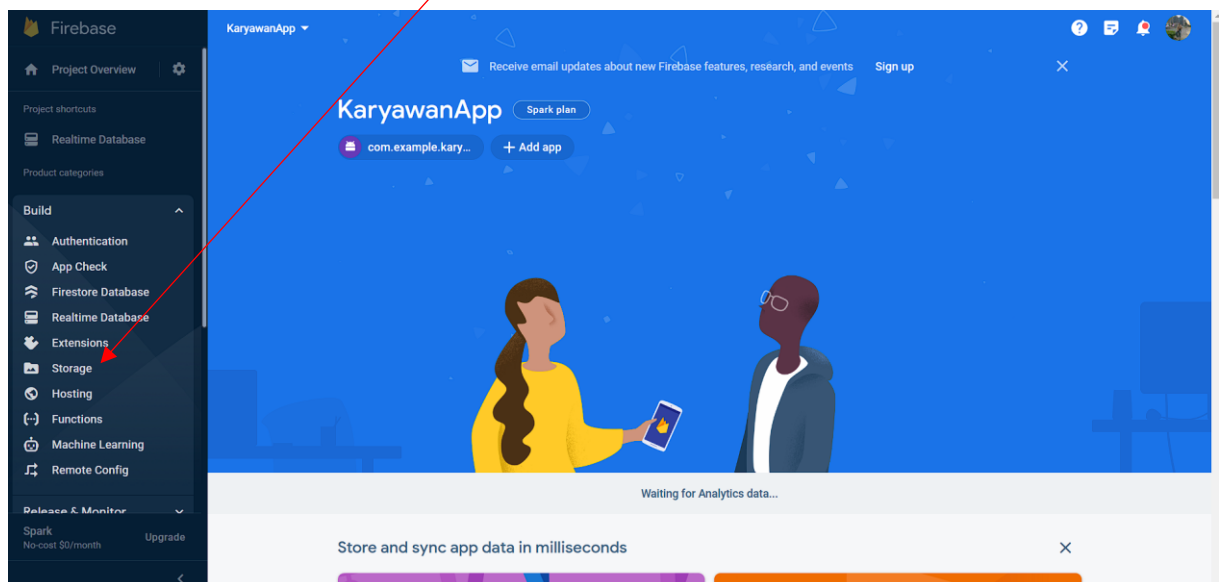
1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-07	CPMK-03	Selesaikan langkah praktikum 1 -7	Hasil praktikum langkah 1 -7	

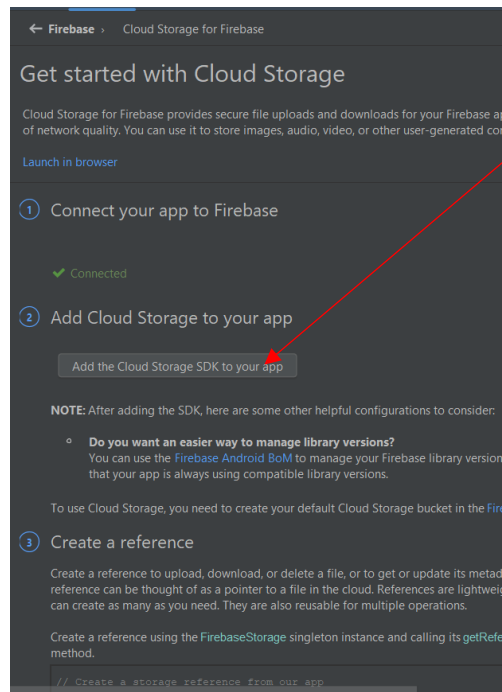
Langkah-Langkah Praktikum:

1. Buka project langkah praktikum pada praktikum sebelumnya.
2. Kemudian buka firebase console dan tambahkan fitur **storage**.



Gambar 9.1 aktifkan fitur storage

3. Jika fitur **storage** sudah aktif, langkah selanjutnya adalah menambahkan dependency kedalam gradle dengan cara buka **tools->firebase->cloud storage**.



4. Kemudian tambahkan dependency image loader yaitu glide.

```
implementation "com.github.bumptech.glide:glide:4.13.0"
```

5. langkah selanjutnya adalah buka layout **item_pekerjaan.xml** dan modifikasi kode nya menjadi.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardCornerRadius="20dp"
    android:layout_marginStart="12dp"
    android:layout_marginEnd="12dp"
    android:layout_marginTop="8dp"
    android:backgroundTint="@color/teal_200"
    android:layout_height="80dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        android:gravity="center">

        <androidx.cardview.widget.CardView
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:layout_marginLeft="16dp"
            app:cardCornerRadius="15dp">

            <ImageView
                android:id="@+id/ivPhotos"
                android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        android:scaleType="centerCrop"/>
</androidx.cardview.widget.CardView>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tvNama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nama Pekerjaan"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:textSize="18sp"/>

    <TextView
        android:id="@+id/tvShift"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Shift Pekerjaan"
        android:textColor="@color/black"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:textSize="14sp"/>

</LinearLayout>

</LinearLayout>
</androidx.cardview.widget.CardView>

```

6. Kemudian modifikasi kode yang ada didalam **ModelPekerjaan**:

```

data class ModelPekerjaan(
    val nama: String? = null,
    val shift: String? = null,
    val photoUrl: String? = null,
)

```

7. Pada layout **upload_dialog.xml** tambahkan button baru, sehingga kode nya menjadi.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <EditText
        android:id="@+id/editTextJobName"
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:hint="Nama Pekerjaan" />

<EditText
    android:id="@+id/editTextShift"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Shift" />
<Button
    android:id="@+id/btnUpload"
    android:text="Upload gambar"
    android:layout_width="wrap_content"
    android:layout_gravity="center"
    android:layout_height="wrap_content"/>
</LinearLayout>

```

8. Modifikasi adapter AdapterPekerjaan menjadi.

```

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import com.example.karyawanapp.databinding.ItemPekerjaanBinding

class AdapterPekerjaan(private val data: List<ModelPekerjaan>) :
    RecyclerView.Adapter<AdapterPekerjaan.PekerjaanViewHolder>() {

    class PekerjaanViewHolder(private val binding: ItemPekerjaanBinding) :
        RecyclerView.ViewHolder(binding.root) {
        fun bind(employee: ModelPekerjaan) {
            binding.tvNama.text = employee.nama
            binding.tvShift.text = employee.shift
            Glide.with(binding.root.context)
                .load(employee.photoUrl)
                .centerCrop()
                .into(binding.ivPhotos)
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PekerjaanViewHolder {
        val binding = ItemPekerjaanBinding.inflate(
            LayoutInflater.from(parent.context),
            parent,
            false
        )
        return PekerjaanViewHolder(binding)
    }

    override fun onBindViewHolder(holder: PekerjaanViewHolder, position: Int) {
        val data = data[position]
        holder.bind(data)
    }

    override fun getItemCount(): Int = data.size

```

}

9. Kemudian modifikasi kelas **PekerjaanActivity.kt** menjadi seperti.

```
import android.app.ProgressDialog
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.provider.OpenableColumns
import android.util.Log
import android.view.LayoutInflater
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.activity.result.contract.ActivityResultContracts
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.karyawanapp.databinding.ActivityPekerjaanBinding
import com.google.android.material.dialog.MaterialAlertDialogBuilder
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.ValueEventListener
import com.google.firebase.storage.FirebaseStorage
import com.google.firebase.storage.StorageReference

class PekerjaanActivity : AppCompatActivity() {
    private lateinit var binding: ActivityPekerjaanBinding
    private var url2: String=""
    private var name: String=""
    private lateinit var storage: StorageReference
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityPekerjaanBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.rvPekerjaan.layoutManager = LinearLayoutManager(this)
        binding.rvPekerjaan.setHasFixedSize(true)
        showData()
        tambahData()
    }
    private fun showData() {
        val dataRef = FirebaseDatabase.getInstance().getReference("Pekerjaan")
        dataRef.addValueEventListener(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                val pekerjaanList = mutableListOf<ModelPekerjaan>()
                val adapter = AdapterPekerjaan(pekerjaanList)
                for (dataSnapshot in snapshot.children) {
                    val pekerjaanKey = dataSnapshot.getValue(ModelPekerjaan::class.java)
                    pekerjaanKey?.let {
                        pekerjaanList.add(it)
                        Log.d("cek data firebase", it.toString())
                    }
                }
            }
        })
        if (pekerjaanList.isNotEmpty()) {
```

```

        binding.rvPekerjaan.adapter = adapter
    } else {
        Toast.makeText(this@PekerjaanActivity, "Data not found",
Toast.LENGTH_LONG).show()
    }
}

override fun onCancelled(snapshot: DatabaseError) {
    // Handle onCancelled
}

})
}

private fun tambahData(){
    val database: FirebaseDatabase = FirebaseDatabase.getInstance()
    val pekerjaanRef: DatabaseReference = database.getReference("Pekerjaan")
    binding.fabAddPekerjaan.setOnClickListener {
        val dialogView = LayoutInflater.from(this).inflate(R.layout.upload_dialog, null)

        AlertDialog.Builder(this)
            .setTitle("Tambah Pekerjaan")
            .setView(dialogView)
            .setPositiveButton("Tambah") { dialog, _ ->
                val jobName =
dialogView.findViewById<EditText>(R.id.editTextJobName).text.toString()
                val shift = dialogView.findViewById<EditText>(R.id.editTextShift).text.toString()
                val pekerjaanData = HashMap<String, Any>()
                pekerjaanData["nama"] = jobName
                pekerjaanData["shift"] = shift
                pekerjaanData["photoUrl"] = url2
                name = jobName

                val newPekerjaanRef = pekerjaanRef.push()
                newPekerjaanRef.setValue(pekerjaanData)
                dialog.dismiss()

            }
            .setNegativeButton("Batal") { dialog, _ ->
                dialog.dismiss()
            }
            .show()

        dialogView.findViewById<Button>(R.id.btnUpload).setOnClickListener {
            choosePicture()
        }
    }
}

private val getMultipleContentsPicture =
registerForActivityResult(ActivityResultContracts.GetContent()) { uri: Uri? ->
    if (uri != null){
        uploadPic(uri)
    }
}
}

```

```

private fun choosePicture() {
    getMultipleContentsPicture.launch("image/*")
}
private fun getFileName(uri: Uri): String {
    val cursor = contentResolver.query(uri, null, null, null, null)
    cursor.use {
        if (it != null && it.moveToFirst()) {
            val nameIndex = it.getColumnIndex(OpenableColumns.DISPLAY_NAME)
            if (nameIndex != -1) {
                return it.getString(nameIndex)
            }
        }
    }
}

return System.currentTimeMillis().toString()
}
private fun uploadPic(uris: Uri) {
    val selectedFile = mutableListOf<String>()
    val progressDialog = ProgressDialog(this)
    progressDialog.setMessage("Uploading Picture...")
    progressDialog.setCancelable(false)

    progressDialog.show()
    storage = FirebaseStorage.getInstance().getReference("User")
    val fileName = getFileName(uris)
    selectedFile.add(fileName)
    val contentResolver = this.contentResolver
    val nama = name
    // Upload semua file ke Firebase Storage
    val picRef = storage.child("image/${nama}/${uris.lastPathSegment}")

    try {
        val inputStream = contentResolver.openInputStream(uris)
        picRef.putStream(inputStream!!)
        .addOnSuccessListener { taskSnapshot ->
            // Upload berhasil
            progressDialog.dismiss()
            picRef.downloadUrl.addOnSuccessListener { uri ->
                val downloadUrl = uri.toString()
                url2 = downloadUrl

            }.addOnFailureListener {
            }
            Log.d("TAG", "File uploaded to Firebase Storage")
        }
        .addOnFailureListener { exception ->
            Log.e("TAG", "Upload failed", exception)
        }
    } catch (e: Exception) {
        Log.e("TAG", "Failed to open input stream", e)
    }
}
}

```

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Buatlah database eksternal dalam aplikasi yang anda buat, kemudian implementasikan postman untuk mengakses database anda.	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-07	CPMK-03	20%		
2.	Praktik	CPL-07	CPMK-03	30%		
3.	Post-Test	CPL-07	CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 10: DEBUGGING

Pertemuan ke : 10

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-03	Mahasiswa mampu membuat program untuk akses basis data, mampu mengkompilasi dan menjalankan sebuah aplikasi

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Memahami proses debugging aplikasi
2. Melakukan debugging aplikasi yang dibuat

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-07	CPMK-03	Memahami proses debugging aplikasi yang dibuat
CPL-07	CPMK-03	Menerapkan debugging ke dalam aplikasi yang dibuat

1.3. TEORI PENDUKUNG

Apa Itu Debugging?

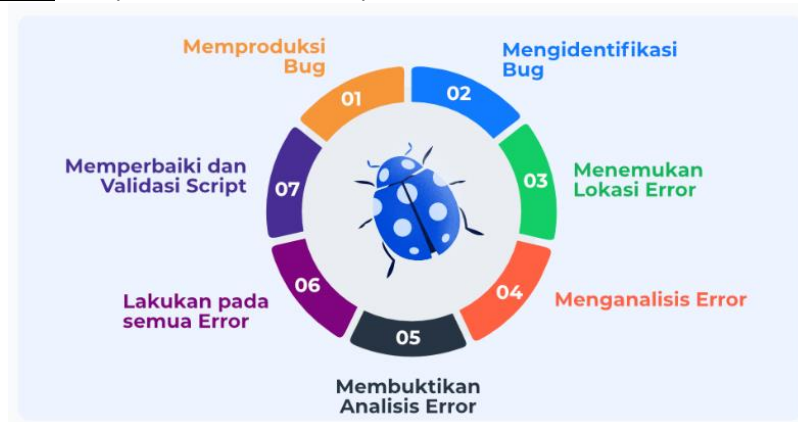
Debugging adalah proses mengidentifikasi dan menghapus bug atau error di dalam kode. Karena sistem pengkodean suatu program itu rumit dan kompleks, maka satu saja kesalahan kode dapat berpengaruh pada keseluruhan program. BUG memang bisa menyebabkan aplikasi atau software mengalami error seperti gagal login, gagal input data, fitur tidak berjalan, blue screen, dll.

Melakukan debug adalah salah satu hal wajib saat membangun sebuah aplikasi/software. Selain tujuan debugging adalah untuk mengatasi bug, proses ini memiliki beberapa manfaat lainnya, seperti:

- a. Error dapat terdeteksi lebih awal;
- b. Proses perbaikan dapat dilakukan sedini mungkin;
- c. Menghindari kesalahan desain program lebih lanjut;
- d. Dapat memberikan informasi struktur data;
- e. Membantu developer mengurangi informasi yang tidak berguna;

- f. Mengurangi resiko hacker yang menyusup melalui celah bug atau error;
- g. Developer dapat menghindari proses testing yang rumit sehingga menghemat waktu dan energi saat coding.

Biasanya, debugging akan dilakukan oleh para debugger. Namun, keahlian ini juga harus dimiliki programmer ataupun full stack developer.



Gambar 3. Tahapan Kerja Debugging

Sebagai langkah awal, harus mengetahui tahapan/cara kerja debugging sebagai berikut:

1. Mereproduksi Bug

Proses debugging pertama adalah mendokumentasikan semua issue bug yang ada dengan benar.

Ada beberapa tujuan mengapa tahap ini sangat penting, yaitu:

- a. Menentukan skala prioritas penyelesaian bug;
- b. Bug dapat diselesaikan dengan tim yang tepat;
- c. Memetakan mana bug yang belum diselesaikan, siap di uji coba, dan lulus quality control;
- d. Uji coba di berbagai perangkat dan browser untuk melihat apakah ada bug lainnya.

Ketika kita sudah mereproduksi bug artinya Anda sudah mengantongi bug apa saja yang harus dituntaskan. Nah, Anda dapat menunjuk tim-tim terkait untuk melakukan tahap berikutnya.

2. Mengidentifikasi Error

Proses debugging tidak dapat dilakukan tanpa proses identifikasi error. Kita bisa mulai mengumpulkan laporan kesalahan pada program dan kemungkinan penyebabnya. Sebagai contoh, ketika perintah pada perangkat lunak gagal dieksekusi, maka kemungkinan ada syntax bug pada skrip software tersebut. Daftar error ini biasanya didapatkan saat melakukan uji coba program, baik pengujian internal atau oleh klien. Pada tahap debug ini, pastikan benar-benar mengidentifikasi program secara menyeluruh. Dengan begitu, proses debugging bisa lebih efisien karena tidak ada lagi bug yang tersisa.

3. Menemukan Lokasi Bug

Ketika sudah tahu adanya error, tapi masih bingung di bagian mana kendala terjadi. Itulah kenapa penting untuk melanjutkan prosesnya dengan menemukan letak error di code tersebut. Misalnya, di baris kode mana ada kesalahan yang menyebabkan sistem tidak dapat berjalan. Kendala pada perangkat lunak bisa terjadi di bagian navigasi, tombol perintah, dan berbagai menu lainnya. Itu mengapa identifikasi bug di awal pengembangan perangkat lunak penting dilaksanakan.

4. Menganalisis Error

Ketika sudah tahu ada bug dan lokasinya pada program. Maka, dilanjutkan dengan analisis error agar dapat melakukan perbaikan yang tepat. Teknik bottom up dalam proses debug akan membantu mengetahui bug-bug lain yang terkait. Tak hanya itu, namun juga dapat meminimalisir resiko error tambahan pada saat perbaikan.

5. Membuktikan Analisis Error

Sudah benarkah hasil analisis error pada langkah debug sebelumnya? Sebelum memperbaikinya, kita perlu membuktikan bahwa tidak ada error lain yang bisa saja muncul ketika melakukan perbaikan bug. Salah satu caranya, dengan **menulis tes otomatis** pada lokasi bug dengan bantuan framework test. Setelah uji skrip tidak ditemukan error lain muncul, maka bisa lanjut ke tahap berikutnya.

6. Lakukan Debugging pada Semua Error

Bug yang di temukan bisa saja lebih dari satu. Pun demikian, semua bug tersebut harus mampu diatasi dengan tuntas.

Kita dapat mengumpulkan semua unit test kode yang ingin diperbaiki. Kemudian, jalankan pengujian pada setiap kode yang ingin diubah. Ingat, semuanya harus mampu berjalan dengan baik di perangkat lunak atau program tersebut.

7. Memperbaiki dan Validasi Skrip

Sudah menyiapkan kode perbaikan untuk debugging? Masukkan semua kode yang sudah di perbaiki dan sudah lolos uji coba ke dalam framework. Pun begitu, selalu lakukan pengecekan berulang. Pastikan kembali semua skrip sudah sesuai dan pastikan program berjalan sesuai yang diharapkan. Bagaimana jika ada bug lagi? Jangan ragu untuk mengulang proses debugging dari awal.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio
3. MySQL

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Jelaskan apa debugging pada aplikasi mobile	
2.	CPL-07	CPMK-03	Jelaskan tahapan debugging android.	

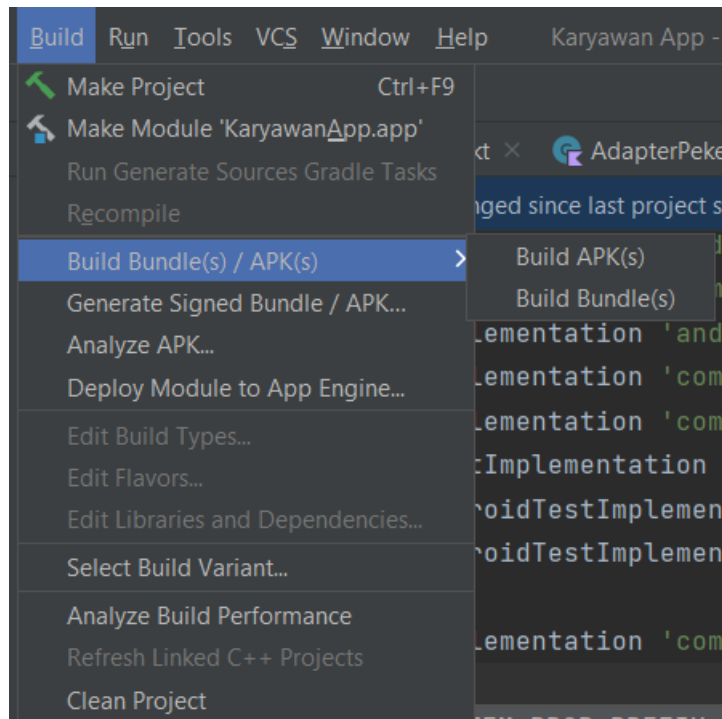
1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-07	CPMK-03	Selesaikan langkah praktikum 1 -6	Hasil praktikum langkah 1 -6	

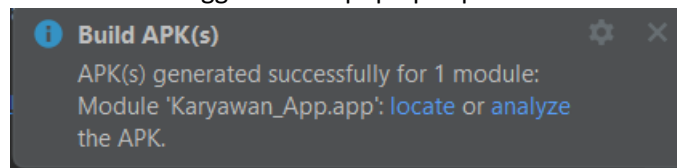
Langkah-Langkah Praktikum:

1. Buka project yang ingin kalian debug.
2. Jika sudah membuka project yang ingin di debug, silahkan pilih menu **Build -> Build Bundle/APK-> Build APK(s)**



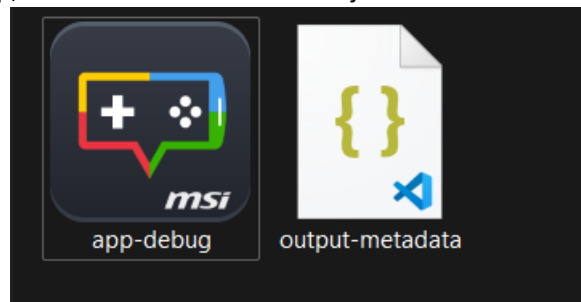
Gambar 10.1 Tools membuat debug

3. tunggu hingga proses selesai hingga muncul pop up seperti ini.



Gambar 10.2 popup setelah berhasil build file APK

4. setelah muncul popup, bisa klik locate untuk menuju ke lokasi file APK.



Gambar 10.3 Bentuk file APK hasil debugging.

5. Setelah menemukan file APK hasil debugging, maka kalian bisa sebar luaskan Aplikasi yang sudah dibuat dengan banyak cara, mulai dari googledrive dan lainnya.

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Dari aplikasi yang telah anda buat dipraktikum sebelumnya silahkan lakukan debugging dan berikan hasil pencermatan anda.	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-07	CPMK-03	20%		
2.	Praktik	CPL-07	CPMK-03	30%		
3.	Post-Test	CPL-07	CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 11: PETA

Pertemuan ke : 11

Total Alokasi Waktu : 90 menit

- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 30 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-03	Mahasiswa mampu membuat program untuk akses basis data, mampu mengkompilasi dan menjalankan sebuah aplikasi

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Membuat API Key untuk keperluan peta
2. Menerapkan peta dan marker ke dalam aplikasi yang dibuat

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-07	CPMK-03	Membuat API Key untuk keperluan peta
CPL-07	CPMK-03	Menerapkan pembuatan peta dan marker ke dalam aplikasi yang dibuat

1.3. TEORI PENDUKUNG

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Android studio
3. MySQL

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Mengapa kita memerlukan API Key saat akan menggunakan peta dalam aplikasi android?	
2.	CPL-07	CPMK-03		

1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-07	CPMK-03	Selesaikan langkah praktikum A nomor 1-26	Hasil praktikum langkah 1 -26	
2.	CPL-07	CPMK-03	Selesaikan langkah praktikum B nomor 1-10	Hasil praktikum langkah 1 -10	

Langkah-Langkah Praktikum:

A. Membuat API KEY di google

Berikut cara untuk mendapatkan API Key google maps untuk android :

1. Kunjungi situs **Google Cloud Platform Console** melalui link <https://console.cloud.google.com/google/maps-apis/overview>
2. Dibagian header kiri atas bisa pilih projek yang pernah dibuat atau buat projek baru
3. Kemudian klik pada bagian Menu dan pilih API dan LAYANAN, lalu pilih kredensial
4. Pada halaman kredensial klik buat kredensial, pilih Kunci Api
5. Setelah muncul Api nya kemudian klik batasi kunci
6. Pada halaman batasi kunci nama kunci secara default adalah Kunci API 1
7. Pada judul pembatasan aplikasi, pilih aplikasi android
8. Pada judul batasi pengguna aplikasi android anda, pilih tambahkan item
9. Setelah itu muncul nama package projek android dan SHA-1
10. Langkah selanjutnya untuk mengisi form tersebut kita harus membuat projek android
11. Buat projek baru pada android studio
12. Jika sudah build maka ada nama package atau paket di MainActivity.java bagian paling atas kemudian
13. Isikan pada kolom nama paket
14. Kemudian isi sidik jari sertifikat SHA-1
15. Pada kana atas android studio projek anda, ada tampilan atau text gradle, kemudian klik
16. Setelah itu klik pada bagian App, android, lalu signingReport, klik dua kali,
17. Tunggu sampai run selesai dan akan muncul SHA-1
18. Kemudian kopi kan ke form input tadi, klik selesai
19. Pada judul pembatasan API
20. Kemudian simpan.
21. Lalu klik Library pada menu sebelah kiri
22. Pilih Api Map SDK for Android
23. Klik aktifkan
24. Kemduian akan dibawa kehalaman berjudul google maps
25. Klik kredensial
26. Disitu terdapat kunci api dengan api key google maps yang nantinya akan digunakan dalam projek android.

Itulah cara mendapatkan Api key untuk key google maps andorid selanjutnya kita akan mencoba untuk menampilkan maps pada aplikasi android yang akan kita buat.

B. Membuat peta

1. Dari proyek yang sudah kita punya, pertama yang kita lakukan yaitu install library google maps di build.gradle(Module:app), ketikan seperti berikut pada module app tersebut:

```
1 implementation 'com.google.android.gms:play-services-maps:16.1.0'
```

2. Pada AndroidManifest.xml tambahkan permission:

```
1 <uses-permission android:name="android.permission.INTERNET"/>
```

```
2 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

3. Masih di AndroidManifest.xml, tambahkan kode berikut pada bagian <application>

```
1 <meta-data
2     android:name="com.google.android.gms.version"
3     android:value="@integer/google_play_services_version" />
4 <!-- The API key for Google Maps-based APIs. -->
5 <meta-data
6     android:name="com.google.android.geo.API_KEY"
7     android:value="@string/google_maps_key" />
8 <uses-library
9     android:name="org.apache.http.legacy"
10    android:required="false" />
11
```

4. Pada strings.xml tambahkan api key anda:

```
1 <string name="google_maps_key">KUNCI API KEY ANDA</string>
```

5. Lalu pada gradle.properties tambahkan juga:

```
1 GOOGLE_MAPS_API_KEY=KUNCI API KEY ANDA
```

6. activity_main.xml juga ditambahkan fragment maps:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <fragment
10         android:id="@+id/map"
11         android:name="com.google.android.gms.maps.SupportMapFragment"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"
14         tools:context="com.example.mapwithmarker.MapsMarkerActivity" />
15
16 </RelativeLayout>
```

7. Selanjutnya di MainActivity.java pada public class MainActivity extends AppCompatActivity tambahkan implements OnMapReadyCallback, jadinya akan seperti berikut:

```
1 public class MainActivity extends AppCompatActivity implements OnMapReadyCallback{
2
3     @Override
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(R.layout.activity_main);
7     }
8 }
```

```
8 }

```

8. Habis itu buat implements metode OnMapReady, jadi seperti berikut:

```
1 public class MainActivity extends AppCompatActivity implements OnMapReadyCallback{
2
3     @Override
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(R.layout.activity_main);
7     }
8
9     @Override
10    public void onMapReady(GoogleMap googleMap) {
11
12    }
13 }
```

9. Kemudian secara keseluruhan penambahan pemanggil id fragment yang ada pada activity_main.xml dan isi dari OnMapReady tersebut, seperti berikut:

```
1 package com.belajar.maps;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import com.google.android.gms.maps.CameraUpdateFactory;
6 import com.google.android.gms.maps.GoogleMap;
7 import com.google.android.gms.maps.OnMapReadyCallback;
8 import com.google.android.gms.maps.SupportMapFragment;
9 import com.google.android.gms.maps.model.LatLng;
10 import com.google.android.gms.maps.model.MarkerOptions;
11
12 public class MainActivity extends AppCompatActivity implements OnMapReadyCallback{
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
20             .findFragmentById(R.id.map);
21         mapFragment.getMapAsync(this);
22     }
23
24     @Override
25     public void onMapReady(GoogleMap googleMap) {
26         // Add a marker in Yogyakarta,Indonesia,
27         // and move the map's camera to the same location.
28         LatLng Yogyakarta = new LatLng(-7.8332349,110.3809325);
29         googleMap.addMarker(new MarkerOptions().position(Yogyakarta)
30             .title("Lokasi Kampus 4 UAD"));
31         googleMap.moveCamera(CameraUpdateFactory.newLatLng(Yogyakarta));
32     }
33
34 }
```

10. Setelah itu tinggal dijalankan pada android.

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Implementasikanlah peta & marker lokasi pada aplikasi yang anda buat	

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-07	CPMK-03	20%		
2.	Praktik	CPL-07	CPMK-03	30%		
3.	Post-Test	CPL-07	CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

DAFTAR PUSTAKA

1. Professional Android 4 Development (Reto Meier) : Chapter 4 Building User Interfaces
2. Beginning Android Application Development (Wei Meng Lee) : Chapter 5 Designing Your User Interface with Views
3. <https://developer.android.com/studio/index.html>
4. <https://google-developer-training.github.io/android-developer-fundamentals-course-concepts>



**LABORATORIUM
S1 INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN**



2023