

NIM	:	11S20018
Nama	:	Ruth Aulya Silalahi
Kelas	:	13IF1
Video Source Code	:	httpsyoutu.bejnrBpP_OcJA

A. Accessing the Features of Android Device**Sending an SMS Message**

Cari tahu kegunaan dari semua kode program pada line 38 sampai 52. Laporkan hasilnya. Kode di atas adalah sebuah fungsi untuk mengirim pesan teks (SMS) ke nomor telepon tertentu. Fungsi ini menerima dua parameter yaitu `userMessage` yang berisi pesan yang akan dikirim dan `userPhone` yang berisi nomor telepon penerima.

Pertama-tama, fungsi ini akan melakukan pengecekan apakah aplikasi memiliki izin untuk mengirim SMS dengan menggunakan fungsi `ContextCompat.checkSelfPermission()`. Jika izin belum diberikan, maka fungsi `ActivityCompat.requestPermissions()` akan dipanggil untuk meminta izin pada pengguna.

Jika izin sudah diberikan, fungsi akan membuat objek `SmsManager` untuk mengirim pesan SMS. Pada Android 12 (API level 31) atau yang lebih baru, `SmsManager` tidak dapat diakses secara langsung, sehingga untuk mengaksesnya, perlu menggunakan fungsi `getSystemService()`. Namun, pada versi Android yang lebih lama, `SmsManager.getDefault()` dapat langsung digunakan.

Terakhir, fungsi akan memanggil `smsManager.sendTextMessage()` untuk mengirim pesan teks ke nomor telepon yang ditentukan. Pesan akan dikirim menggunakan nilai parameter `userMessage`, sedangkan parameter `userPhone` digunakan untuk menentukan nomor telepon penerima.

PAM Praktikum 9 - 11S20018

Write your message

Enter phone number

SEND SMS


Sending an SMS Message

PAM Praktikum 9 - 11S20018

Hallo

089518861460

SEND SMS




Izinkan **PAM Praktikum 9 - 11S20018** mengirim dan melihat SMS?

IZINKAN

JANGAN IZINKAN

14:29 0,1KB/d 4G VoLTE

Pesan

 **1111**
Hallo

Baru saja 1

Sending E-mail

Cari tahu kegunaan dari semua kode program pada line 35 sampai 46. Laporkan hasilnya. Kode di atas adalah sebuah fungsi untuk mengirim email menggunakan aplikasi email yang terpasang pada perangkat Android. Fungsi ini menerima tiga parameter yaitu userAddress yang berisi alamat email penerima, userSubject yang berisi subjek email, dan userMessage yang berisi isi pesan email.

Sending E-mail

Fungsi ini pertama-tama akan membuat sebuah array emailAddress yang berisi alamat email penerima. Selanjutnya, fungsi akan membuat objek emailIntent dengan menggunakan Intent yang mengacu pada aksi ACTION_SENDTO. Objek emailIntent juga akan diatur untuk membuka aplikasi email dengan menggunakan Intent.ACTION_SENDTO dan Uri.parse("mailto:").

Selanjutnya, fungsi akan menambahkan beberapa data tambahan ke dalam emailIntent menggunakan metode putExtra(). Data tersebut adalah alamat email penerima (Intent.EXTRA_EMAIL), subjek email (Intent.EXTRA_SUBJECT), dan isi pesan email (Intent.EXTRA_TEXT) yang diambil dari nilai parameter.

Terakhir, fungsi akan memeriksa apakah ada aplikasi email yang terpasang pada perangkat dengan menggunakan metode resolveActivity() dan jika ada, fungsi akan membuka aplikasi email melalui metode startActivity(). Jika tidak ada aplikasi email yang terpasang, maka tidak akan ada tindakan yang diambil. Pada contoh kode ini, fungsi createChooser() digunakan untuk menampilkan dialog pemilihan aplikasi email yang akan digunakan oleh pengguna ketika membuka email.

PAM Praktikum 9 - 11S20018

ifs20018@gmail.com

Perkenalan Diri

Halo, Saya Ruth Silalahi

SEND E-MAIL

Sending E-mail

PAM Praktikum 9 - 11S20018

ifs20018@gmail.com

Perkenalan Diri

Halo, Saya Ruth Silalahi

SEND E-MAIL

Choose an app

PRIBADI

KERJA



PAM
Praktikum 9



Gmail



Tutanota

← Tulis



Dari ruthauliyas@gmail.com



Kepada  ifs20018@gmail.com



Perkenalan Diri

Halo, Saya Ruth Silalahi

Making a Call

Cari tahu kegunaan dari semua kode program pada line 33 sampai 42. Laporkan hasilnya.

Kode di atas adalah sebuah fungsi untuk memulai panggilan telepon pada perangkat Android. Fungsi ini menerima satu parameter yaitu `userNumber` yang berisi nomor telepon yang akan dipanggil.

Fungsi ini akan melakukan pengecekan apakah aplikasi memiliki izin untuk melakukan panggilan telepon dengan menggunakan fungsi `ContextCompat.checkSelfPermission()`. Jika izin belum diberikan, maka fungsi `ActivityCompat.requestPermissions()` akan dipanggil untuk meminta izin pada pengguna dengan kode permintaan 100.

Jika izin sudah diberikan, fungsi akan membuat objek intent dengan menggunakan Intent yang mengacu pada aksi `ACTION_DIAL`. Objek intent juga akan diatur untuk membuka aplikasi panggilan dengan menggunakan `Uri.parse("tel:${userNumber}")` sebagai data URI. Nilai parameter `userNumber` akan dimasukkan ke dalam URI tersebut dengan menggunakan placeholder `${ }`.

Terakhir, fungsi akan membuka aplikasi panggilan melalui metode `startActivity()`. Jika aplikasi panggilan tidak tersedia di perangkat atau terjadi kesalahan lainnya, maka tidak akan ada tindakan yang diambil.

PAM Praktikum 9 - 11S20018

089518861460

CALL

Izinkan **PAM Praktikum 9 - 11S20018** melakukan dan mengelola panggilan telepon?

IZINKAN

JANGAN IZINKAN



089518861460

1

2

3

Converting Speech to Text

Cari tahu kegunaan dari semua kode program pada line 29 sampai 39 dan line 46 sampai line 51. Laporkan hasilnya.

Line 29 sampai 39

Kode di atas adalah contoh penggunaan `ActivityResultLauncher` pada Kotlin untuk menangani hasil dari aktivitas yang dimulai dengan menggunakan `startActivityForResult()`. `ActivityResultLauncher` adalah sebuah interface yang diperkenalkan pada Android Jetpack versi 1.3.0 untuk menggantikan fungsi `onActivityResult()` yang telah digunakan sejak lama. Interface ini digunakan untuk memperoleh hasil dari aktivitas yang diluncurkan dengan menggunakan metode `startActivityForResult()` pada suatu komponen, seperti `Activity` atau `Fragment`.

Pada kode di atas, `registerForActivityResult()` digunakan untuk mendaftarkan sebuah instance dari `ActivityResultLauncher` yang baru dengan menggunakan `ActivityResultContracts.StartActivityForResult()` sebagai kontrak. Kontrak tersebut menetapkan bahwa aktivitas yang diluncurkan akan menghasilkan data yang akan dikembalikan sebagai objek `ActivityResult`. Setelah instance `ActivityResultLauncher` berhasil didaftarkan, objek tersebut dapat digunakan untuk meluncurkan aktivitas dan menangani hasilnya dengan menggunakan metode `launch()` dan sebuah fungsi lambda sebagai `ActivityResultCallback`.

Pada kode di atas, setelah aktivitas yang diluncurkan selesai, hasilnya akan diperoleh melalui objek `result`. Objek `result` kemudian akan digunakan untuk memeriksa kode hasil dan objek data dengan menggunakan `resultCode` dan `data`. Jika hasilnya berhasil (`resultCode` sama dengan `RESULT_OK`) dan objek data tidak null, maka data suara yang dihasilkan akan diekstrak dari hasil aktivitas menggunakan `getStringArrayListExtra()` dan ditampilkan pada `tvResult`.

Line 46 sampai 51

Kode di atas adalah sebuah fungsi yang digunakan untuk mengkonversi suara menjadi teks dengan menggunakan fitur speech recognition bawaan dari Android. Pada fungsi `convertSpeech()`, pertama-tama sebuah objek `Intent` dibuat dengan aksi `ACTION_RECOGNIZE_SPEECH`. Aksi ini mengindikasikan bahwa aktivitas yang diluncurkan akan melakukan pengenalan suara untuk menghasilkan teks.

Kemudian, dua nilai tambahan (extra) ditambahkan ke dalam objek `Intent`. Nilai pertama adalah `RecognizerIntent.EXTRA_LANGUAGE_MODEL`, yang menetapkan bahwa

Converting Speech to Text

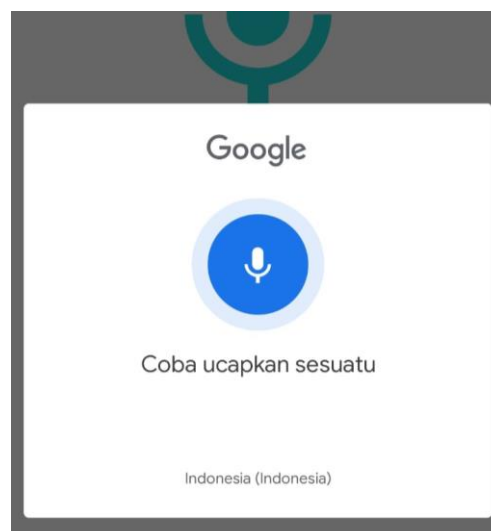
model bahasa yang digunakan adalah `LANGUAGE_MODEL_FREE_FORM`. Model bahasa ini digunakan untuk mengenali teks yang diucapkan secara bebas tanpa batasan atau aturan khusus. Nilai tambahan kedua adalah `RecognizerIntent.EXTRA_LANGUAGE`, yang menentukan bahasa yang digunakan untuk mengenali suara. Pada kode di atas, bahasa yang digunakan diambil dari nilai default sistem dengan menggunakan `Locale.getDefault()`.

Terakhir, objek `Intent` yang telah disiapkan akan diluncurkan dengan menggunakan `activityResultLauncher.launch(intent)`. Tujuan dari peluncuran ini adalah untuk menampilkan dialog pengenalan suara dan mengizinkan pengguna untuk mengucapkan kata-kata yang akan diubah menjadi teks. Setelah pengguna selesai berbicara, hasil dari pengenalan suara akan dikembalikan ke pemanggil melalui `ActivityResultLauncher` yang sudah didaftarkan sebelumnya, sehingga teks yang dihasilkan dapat ditampilkan pada tampilan aplikasi.

PAM Praktikum 9 - 11S20018



Please tap to microphone to speak



Converting Speech to Text

PAM Praktikum 9 - 11S20018



ruth

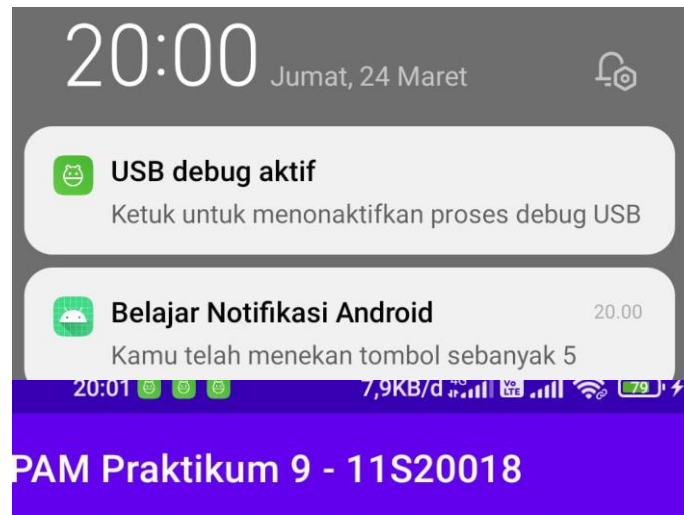
B. Notification**Local Notification**

Cari tahu kegunaan dari semua kode program pada line 48 sampai 70. Laporkan hasilnya.

Kode di atas adalah sebuah fungsi yang digunakan untuk menampilkan notifikasi pada aplikasi Android. Berikut adalah penjelasan dari masing-masing baris kode:

Pertama-tama, dibuat sebuah objek NotificationCompat.Builder dengan menggunakan this@MainActivity dan CHANNEL_ID sebagai parameter. CHANNEL_ID adalah sebuah string yang digunakan untuk menentukan channel notifikasi pada versi Android yang lebih baru. Kemudian, dilakukan pengecekan apakah perangkat yang digunakan telah menggunakan versi Android yang lebih baru dari Oreo (versi 8.0). Jika ya, maka dilakukan pembuatan NotificationChannel dan ditetapkan sebagai IMPORTANCE_DEFAULT. Selain itu, juga dilakukan pembuatan objek NotificationManager dan digunakan untuk membuat channel notifikasi dengan manager.createNotificationChannel(channel).

Setelah itu, dilakukan pengaturan pada notifikasi seperti judul, deskripsi, dan ikon yang akan ditampilkan pada notifikasi. Judul dan deskripsi notifikasi diambil dari sebuah string yang telah ditetapkan sebelumnya. Sedangkan untuk ikon notifikasi, digunakan R.drawable.small_icon. Objek NotificationManagerCompat digunakan untuk menampilkan notifikasi dengan memanggil notificationManagerCompat.notify(1, builder.build()).

Local Notification**PAM Praktikum 9 - 11S20018****0****PAM Praktikum 9 - 11S20018****5****Repeated Notification**

Cari tahu kegunaan dari setiap kode program dari fungsi **”showNotification()”** pada line 60 sampai 106. Laporkan hasilnya.

Fungsi `showNotification()` digunakan untuk menampilkan time picker ketika tombol "Set Notification" di klik. Setelah waktu diatur, fungsi akan membuat intent untuk memanggil kelas `NotificationReceiver`, lalu membuat `pendingIntent` dengan kode permintaan 100 yang akan digunakan untuk membatalkan alarm nanti. Selanjutnya, fungsi akan membuat objek `AlarmManager`, dan memanggil metode `setInexactRepeating()` pada objek tersebut untuk mengatur alarm dengan interval waktu yang diatur oleh pengguna, dan menetapkan `pendingIntent` yang dibuat sebelumnya sebagai tindakan yang akan diambil ketika alarm berbunyi. Terakhir, fungsi menampilkan pesan "The alarm has been set" dengan Toast.

Repeated Notification

PAM Praktikum 9 - 11S20018

UBAH WAKTU

PAM Praktikum 9 - 11S20018

UBAH WAKTU

BELAJAR MEMBUAT REPEATED NOTIFICATION

09 : 01 AM
Jam Menit

BATAL OKE

PAM Praktikum 9 - 11S20018

UBAH WAKTU

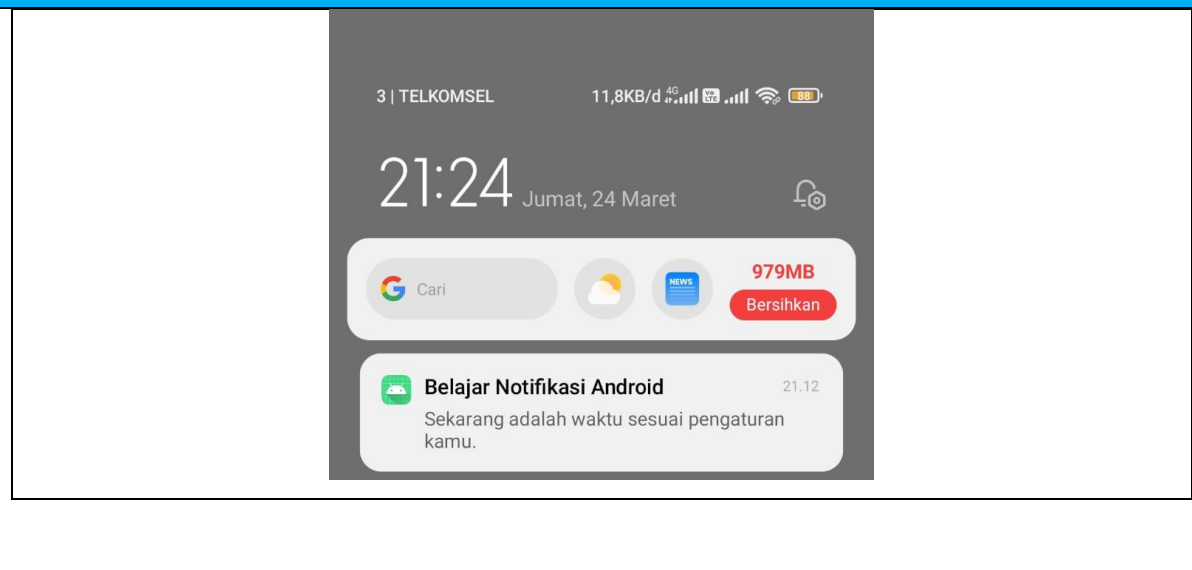
BELAJAR MEMBUAT REPEATED NOTIFICATION

09 : 15 AM
PM

00 05 10 15 20 25 30 35 40 45 50 55

BATAL OKE

The alarm has been set

Repeated Notification**Notification Procedures**

Cari tahu perbedaan kode program di atas dengan kode program pada tahap ke 6, serta laporkan pada line berapa kode program dimodifikasi. Laporkan hasilnya.

Kedua kode merupakan implementasi notifikasi di aplikasi Android menggunakan NotificationCompat. Namun, terdapat perbedaan antara keduanya. Kode ini memiliki tambahan fitur yang tidak ada pada kode step5, yaitu:

- Tambahkan aksi "Show Toast" pada notifikasi, yang akan menampilkan pesan "Ini adalah pesan notifikasi." ketika di-klik.
- Tambahkan tombol "Dismiss" pada notifikasi untuk membatalkan notifikasi.
- Gunakan gambar (bitmap) untuk notifikasi, dengan menambahkan large icon dan menggunakan gaya notifikasi BigPictureStyle untuk menampilkan gambar dalam notifikasi.
- Kode step5 hanya menampilkan notifikasi dasar dengan judul "Belajar Notifikasi Android" dan isi teks "Kamu telah menekan tombol sebanyak \${counter}" dan tidak memiliki fitur aksi dan tombol seperti pada kode pertama.

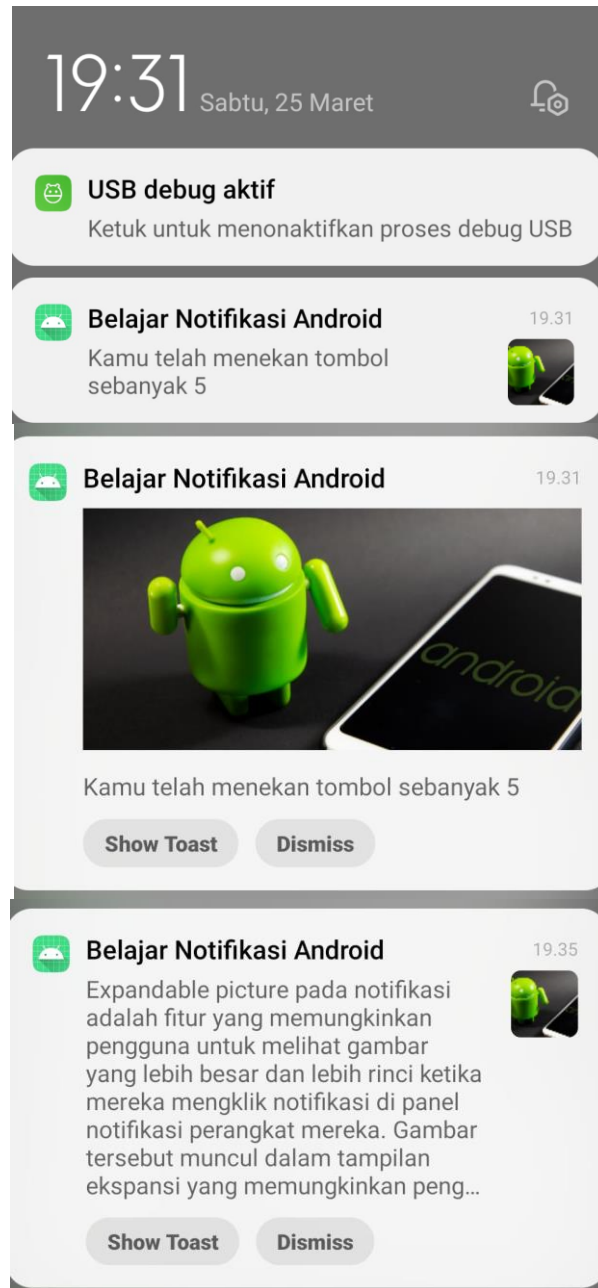
PAM Praktikum 9 - 11S20018

0

Notification Procedures

PAM Praktikum 9 - 11S20018

5



C. Architectura Room Database**Note APP**

Silahkan melakukan analisis terkait cara kerja dari aplikasi "NoteApp". Laporkan hasilnya.

Pada tahap pertama, activity utama ini mengatur tampilan RecyclerView untuk menampilkan catatan-catatan yang telah disimpan. Kemudian, activity memperoleh instance dari NoteViewModel dan memanggil fungsi observe pada instance tersebut, sehingga activity akan selalu memperbarui tampilan catatan ketika ada perubahan pada data catatan.

Ketika pengguna menambahkan catatan baru, activity akan menampilkan NoteAddActivity untuk menambahkan catatan baru. Activity utama mendaftarkan dua ActivityResultLauncher, yaitu addActivityResultLauncher dan editActivityResultLauncher, untuk menangani permintaan dari NoteAddActivity. Ketika pengguna menambahkan catatan baru, addActivityResultLauncher akan memperoleh data dari NoteAddActivity dan kemudian menciptakan objek Note baru dan memanggil fungsi insert dari instance NoteViewModel untuk menambahkan catatan ke database.

Ketika pengguna menghapus catatan, activity akan menampilkan dialog konfirmasi untuk menghapus semua catatan atau hanya catatan yang dipilih dengan men-swipe ke kiri atau ke kanan pada catatan. Jika pengguna memilih untuk menghapus semua catatan, activity akan memanggil fungsi deleteAllNotes dari NoteViewModel untuk menghapus semua catatan dari database. Jika pengguna menghapus catatan secara individual, ItemTouchHelper akan menangani swipe dan memanggil fungsi delete pada NoteViewModel untuk menghapus catatan dari database.

Kode di atas juga mendefinisikan dua ActivityResultLauncher untuk meng-handle pembaruan catatan dan menambahkan catatan baru. Ketika pengguna memilih untuk meng-update catatan, editActivityResultLauncher akan memperoleh data yang diperbarui dari NoteAddActivity dan kemudian memperbarui catatan yang bersesuaian dengan memanggil fungsi update pada NoteViewModel.

Note APP

NoteApp_11S20018



Title 1

Description 1

Title 2

Description 2

Title 3

Description 3

Praktikum PAM ke-9

Belajar membuat aplikasi note

PAM Praktikum 9 - 11S20018

ifs20018@gmail.com

Perkenalan Diri

Halo, Saya Ruth Silalahi

SEND E-MAIL

Note APP

NoteApp_11S20018



Title 1

Description 1

Title 2

Description 2

Title 3

Description 3

Praktikum PAM ke-9

Belajar membuat aplikasi note

NoteApp_11S20018



Title 1

Description 1

Title 2

Description 2

Praktikum PAM ke-9

Belajar membuat aplikasi note

NoteApp_11S20018

