Tugas Besar 1 - Android

IF3210 Pengembangan Aplikasi pada Platform Khusus

# 

Aplikasi BondoMan



Deadline: **5 April 2024 17:00 WIB**

**Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

**2024**

# Latar Belakang



“*Alasan kenapa Bandung Bondowoso meminta para jin yang membuat aplikasi*”

Bandung Bondowoso merupakan seorang kepala proyek pembangunan seribu candi. Proyek tersebut dia dapatkan dari Roro yang tidak memahami segitiga manajemen proyek, yaitu *budget*, *deadline*, dan *quality*. Proyek ini memiliki *budget* yang sedikit, *deadline* yang hanya satu malam, dan Roro meminta kualitas candi yang bagus.

Bondowoso merasa pusing ketika mengerjakan proyek seribu candi dalam satu malam ini. Dalam pengerjaannya, terjadi banyak sekali transaksi pembelian bahan baku. Tentu saja dia harus berhati-hati dalam melakukan perhitungan biaya, sebab jika tidak seusai dengan RAB (Rancangan Anggaran Biaya) semula, Roro tidak ingin membayarnya.

Awalnya, Bondowoso menuliskan segala transaksi jual beli bahan baku di atas prasasti berbentuk batu. Namun, dengan skala proyek yang sangat besar ini, ia kewalahan dan meminta bantuan kalian, para jin yang saat ini berkuliah di Institut Jin semester 6, untuk membuat aplikasi manajemen keuangan. Ia percaya kalian bukanlah jin yang pandir sehingga dapat menyelesaikan aplikasi tersebut.

Saat ini, Bondowoso menginginkan aplikasi tersebut dapat berjalan di gawai yang ia miliki. Karena ia tidak suka buah apel, ia meminta aplikasi dibuat untuk sistem operasi Android. Bantulan Bondowoso memanajemen uangnya agar pembangunan candi berjalan lancar!

# Changelog

2 April 2024 - Pengubahan *deadline* tugas besar

# Tujuan

Tujuan dari tugas besar ini adalah

1. Mahasiswa memahami prinsip dasar dalam pengembangan aplikasi Android
2. Mahasiswa mampu bekerja sama dalam konteks pengembangan aplikasi Android
3. Mahasiswa mampu mempresentasikan aplikasi yang dibuat
4. Mahasiswa memahami dan mampu menjelaskan alasan dibalik pengembangan aplikasi

# Spesifikasi Aplikasi

Berikut adalah spesifikasi yang **harus diimplementasi:**

## Header dan Navbar

* Terdapat empat *navigation menu*, yaitu Menu Transaksi, Menu Scan, Menu Grafik, dan Menu Pengaturan
* Jika berada di menu tertentu, berikan *cue* pada icon menu tersebut, serta header-nya berubah menyesuaikan menu tersebut
* Referensi dapat dilihat pada [Lampiran](#_xsjxvxcks6as)

## Login - Halaman Login

* Pada awalnya, pengguna aplikasi harus *login* terlebih dahulu sebelum bisa menggunakan fitur yang ada di aplikasi
* Setelah *login*, *endpoint* akan memberikan token sebagai tanda *login* berhasil. Token akan digunakan untuk mengaksis *endpoint* lain
* Cara penyimpanan token dibebaskan, **pastikan** kalian menerapkan ***best practice*** dalam penyimpanan di Android
* *Login* dapat dilakukan dengan menembak *endpoint* **{{baseUrl}}/api/auth/login**
* *Request body* untuk *login* berbentuk JSON dengan format:

{

“email”: “{NIM}@std.stei.itb.ac.id”,

“password”: “password\_{NIM}”

}

## Logout - Halaman Pengaturan

* Pengguna dapat melakukan ***logout*** dari aplikasi
* *Logout* dapat dilakukan di halaman **Pengaturan**
* Setelah *logout*, pengguna **diarahkan** ke halaman **Login**

## Melakukan Penambahan, Pengubahan, dan Penghapusan Transaksi

* Pengguna dapat melakukan manipulasi data transaksi, berupa penambahan, pengubahan, dan penghapusan transaksi
* Dalam penambahan transaksi, seminimal-minimalnya, pengguna perlu memasukkan data atribut, berupa: judul transaksi, kategori (Pemasukan, Pengeluaran), nominal transaksi, dan lokasi transaksi. Sistem perlu secara otomatis merekam tanggal transaksi berupa tanggal dimasukkannya transaksi dan lokasi transaksi sebagai lokasi pengguna saat ini apabila pengguna mengizinkan pengaksesan lokasi
* Dalam pengubahan transaksi, sistem secara otomatis mengisi data atribut sesuai transaksi yang sudah ada (termasuk *hidden* atribut sebelumnya, contohnya: tanggal) dan pengguna dapat mengubah data tersebut kecuali atribut yang dispesifikkan. Setidaknya, atribut yang dapat diubah adalah judul, nominal, dan lokasi; serta atribut yang tidak dapat diubah adalah kategori dan tanggal.
* Dalam penghapusan transaksi, data transaksi yang terhapus perlu langsung hilang dari list transaksi yang ditampilkan

## Melihat Daftar Transaksi yang Sudah Dilakukan

* Daftar transaksi disimpan dalam Room menggunakan Repository Pattern agar data dapat diakses secara *offline*
* Halaman ini akan menampilkan seluruh daftar transaksi yang sudah dilakukan
* Setiap transaksi akan menampilkan kategori transaksi, harga, nama transaksi, lokasi, dan tanggal transaksi
* Pengguna dapat menekan transaksi untuk membuka Google Maps sesuai dengan lokasi transaksi
* Pengguna dapat menambahkan transaksi baru dengan adanya tombol “+”

## Melakukan *Scan* Nota - Halaman *Scan* Nota

* Pengguna dapat menuju halaman untuk melakukan *scan nota*
* Proses *scan nota* adalah mengirimkan gambar dari sisi *client* menuju *server*
* Proses pembacaan pada sisi *server* adalah *dummy* data
* Pengguna dapat mengirimkan gambar baik melalui kamera secara langsung maupun dengan mengunggah foto dari galeri
* Setelah melakukan *scan nota,* tampilkan hasilnya pada layar. Tanyakan kepada pengguna apakah ingin mengulangi proses pembacaan nota atau tidak. Jika tidak, simpan pada daftar transaksi.
* Pengiriman gambar dapat dilakukan dengan mengakses *endpoint* **{{baseUrl}}/api/bill/upload** dengan body berbentuk **form-data** dengan **key “file”.**

## Melihat Graf Rangkuman Transaksi - Halaman Graf

* Pengguna dapat melihat graf yang berisi rangkuman transaksi yang sudah dilakukan
* Tampilan graf yang digunakan dibebaskan
* Halaman ini harus bisa menangani *layout portrait* dan *landscape*
* Dalam membuat graf, kakas yang digunakan dibebaskan asalkan masih menggunakan Kotlin

## Menyimpan Daftar Transaksi dalam Format .xlsx, .xls - Halaman Pengaturan

* Pengguna aplikasi dapat menyimpan seluruh daftar transaksi yang telah dilakukan ke dalam sebuah berkas *spreadsheet* (xlsx atau xls) yang dapat diunduh dan dibuka di luar aplikasi
* Terdapat sebuah tombol untuk menyiapkan berkas yang berisi daftar transaksi. Pengguna dapat memilih antara dua format berkas yang didukung yaitu .xlsx atau .xls sebelum proses ekspor berkas dimulai.
* Berkas yang diekspor akan berisi kolom Tanggal, Kategori Transaksi (Pemasukan atau Pengeluaran), Nominal Transaksi, Nama Transaksi, dan Lokasi. Berkas dapat disimpan pada penyimpanan internal perangkat

## Intent GMail - Halaman Pengaturan

* Pengguna dapat mengirimkan daftar transaksi via aplikasi GMail dengan tujuan email yaitu email yang digunakan untuk login.
* Daftar transaksi dikirim dalam berkas *spreadsheet* (xlsx atau xls) yang akan dikirim sebagai *attachment* GMail.
* *Subject* dan *text message* perlu ditambahkan juga sebagai *intent*. Isi dari *subject* maupun *text* dibebaskan.

## *Background Service* - Mengecek *expiry* JWT

* Aplikasi memiliki *background service* yang mengecek apakah JWT sudah *expired*.
* Apabila sudah *expired*, silahkan berkreasi dalam mendefinisikan perilaku dari *service* (misal, melakukan *logout* sendiri atau melakukan *re-login*).
* Masa berlaku JWT adalah 5 menit.
* Pengecekan JWT dapat dilakukan dengan menembak *endpoint* **{{baseURL}}/api/auth/token** dengan header **“Authorization: Bearer {token}”**

## *Network Sensing* - Deteksi Sinyal

* Aplikasi dapat mengecek apakah kondisi hp saat ini tersambung dengan internet, baik data ataupun wifi atau tidak.
* Jika hp terputus dari koneksi internet, keluarkan sebuah *pop up* sederhana yang menunjukkan bahwa saat ini hp tidak terhubung dengan internet.
* Walaupun tanpa internet, aplikasi dapat dibuka, akan tetapi aplikasi tidak bisa melakukan permintaan dan penarikan data dari server.
* Jika ada halaman yang memerlukan data dari server saat koneksi internet terputus, keluarkan halaman sederhana yang menunjukkan bahwa internet sedang tidak ada sebagai penggantinya (jangan hanya halaman kosong saja).

## *Broadcast Receiver* - *Randomize* Transaksi dari Pengaturan

* *Fragment* penambahan transaksi memiliki *broadcast receiver* untuk menerima *intent* dari *broadcast* yang dikirim oleh tombol “Randomize transaksi” di Halaman Pengaturan.
* Terdapat minimal satu *field* yang terisi ketika *fragment* menerima *intent*.

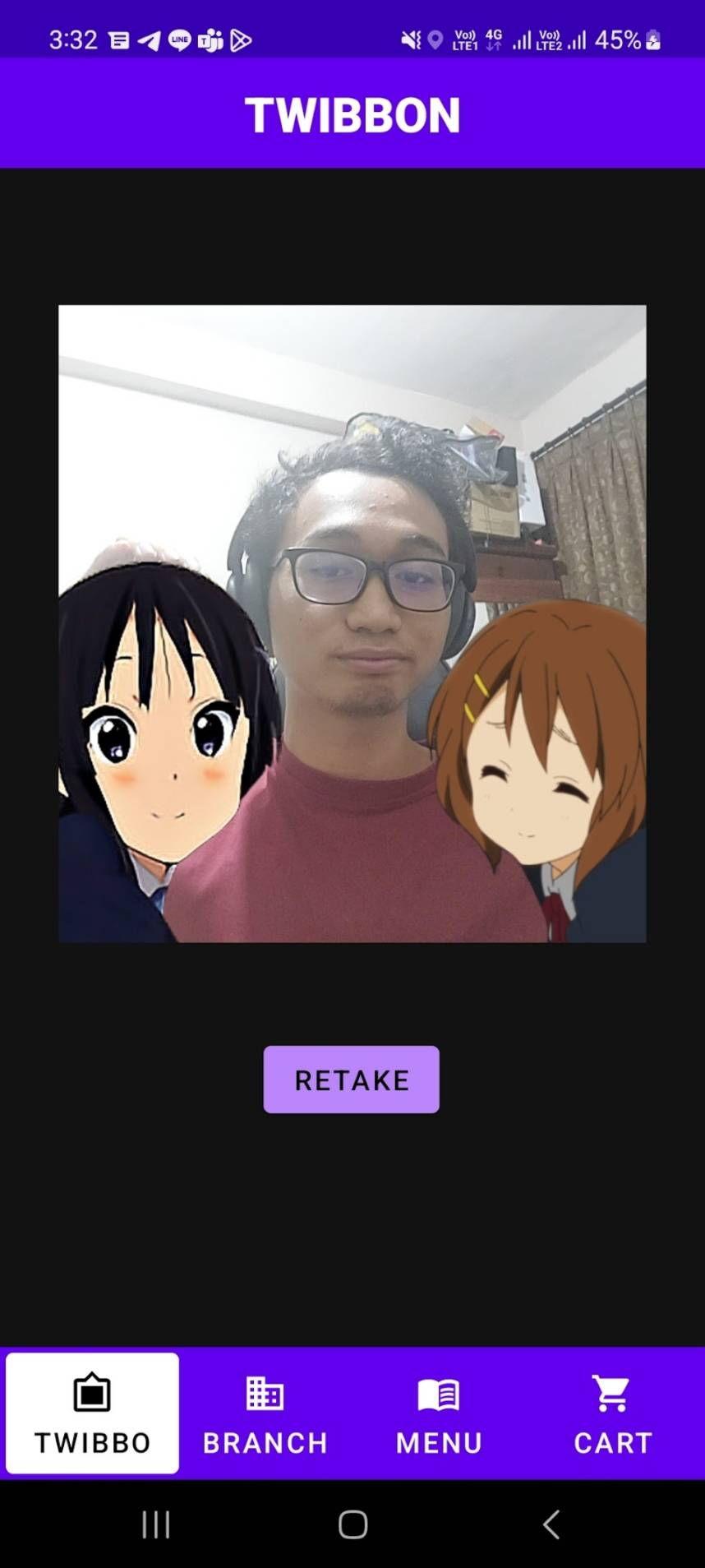
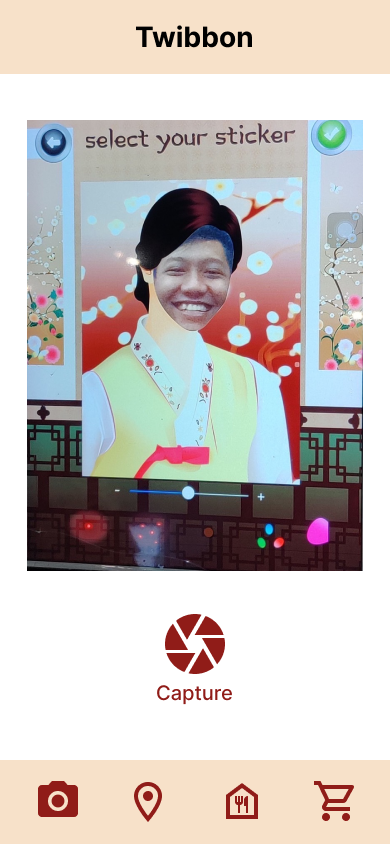
# Spesifikasi Bonus

Berikut adalah spesifikasi bonus yang ***optional* untuk diimplementasi:**

## Twibbon

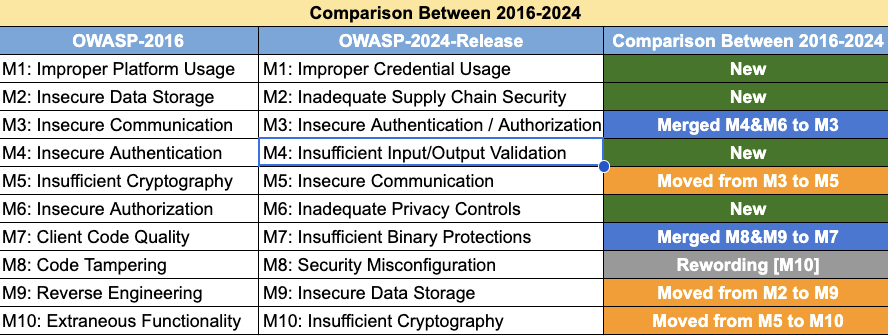
Twibbon merupakan sebuah gambar yang memiliki gambar lain sebagai *overlay*. Apa? Kalian bingung apa urusan twibbon dalam aplikasi mengenai manajemen uang? Sama, penulis juga kebingungan. Namun ini permintaan dari *stakeholder*, Bandung Bondowoso. Penulis hanya mengikuti keinginannya. Tetapi, jika tidak diimplementasikan-pun, seharusnya dia tidak marah 😶.

Fitur twibbon ini diimplementasikan dalam halaman yang berbeda. Dalam halaman twibbon ini, aplikasi akan membuka kamera yang sudah dipasang oleh twibbon. Banyak opsi gambar twibbon setidaknya 1. Jika lebih dari 1, silahkan berikan opsi untuk mengganti twibbon. Untuk mengambil foto twibbon, berikan tombol *capture*. Hasil pengambilan foto tidak perlu disimpan dalam perangkat, cukup ditampilkan saja. Setelah itu, pengguna bisa mengambil foto twibbon lagi. Gambar twibbon yang digunakan dibebaskan.



## OWASP

Bondowoso memiliki intel yang menyebutkan bahwa Roro akan bertindak curang pada proses pembangunan candi. Dia takut jika suatu saat Roro akan meng-*hack* aplikasi miliknya sehingga anggaran pembangunan candinya membengkak melebihi RAB (Rencana Anggaran Biaya) yang sudah dia buat. Oleh karena itu, dia ingin agar aplikasi yang dibuat oleh para jin memenuhi standar keamanan berdasarkan OWASP 2024.



OWASP Top 10 Mobile adalah daftar sepuluh kerentanan keamanan yang paling umum ditemukan pada aplikasi mobile. Kerentanan ini dikeluarkan oleh Open Web Application Security Project (OWASP), sebuah organisasi nirlaba yang berfokus pada keamanan perangkat lunak. Daftar ini bertujuan untuk meningkatkan kesadaran akan ancaman keamanan yang ada dalam pengembangan aplikasi mobile, sehingga pengembang dapat mengambil langkah-langkah untuk mengurangi risiko keamanan.

Dari 10 kerentanan pada tahun 2024, analisislah keamanan aplikasi kalian berdasarkan 3 kerentanan berikut:

1. M4: Insufficient Input/Output Validation
2. M8: Security Misconfiguration
3. M9: Insecure Data Storage

Setelah melakukan analisis, cobalah untuk melakukan perbaikan pada aplikasi kalian jika ada. Dokumentasikan analisis dan perbaikan pada aplikasi kalian di README. Dokumentasi diperkenankan dalam bentuk teks, gambar, dan/atau video. **Gunakan asumsi bahwa *server* tidak aman dan tidak ada *hacker* yang langsung mengakses API *server* sehingga segala validasi dilakukan melalui *client-side*.**

## **Accessibility Testing**

Karena sering bekerja pada malam hari, Bondowoso mengalami rabun senja. Oleh karena itu, dia tidak dapat melihat aplikasi pada gawainya secara jelas sehingga dia meminta kalian, para jin, untuk membuat aplikasi yang memiliki aksesibilitas yang bagus.

Accessibility Scanner adalah kakas yang dikembangkan oleh Google dan digunakan untuk memindai antarmuka pengguna aplikasi untuk memberikan rekomendasi terkait cara meningkatkan aksesibilitas aplikasi. Accessibility Scanner memungkinkan siapa saja, tidak hanya developer, untuk dapat dengan cepat dan mudah mengidentifikasi berbagai peningkatan aksesibilitas umum; misalnya, memperbesar target sentuh kecil, meningkatkan kontras untuk teks dan gambar, serta menyediakan deskripsi konten untuk elemen grafis yang tidak berlabel.

Dengan meningkatkan aksesibilitas aplikasi, pengembang dapat menjangkau audiens yang lebih luas dan memberikan pengalaman yang lebih inklusif, terutama bagi pengguna penyandang disabilitas. Peningkatan aksesibilitas sering kali mengarah pada peningkatan kepuasan pengguna, rating aplikasi, dan retensi pengguna.

**Kerjakan bonus ini setelah aplikasi kalian selesai dikembangkan.** Tampilkan pada README halaman *suggestion* sebelum aplikasi kalian diperbaiki seperti gambar dibawah ini.

## 

Setelah itu, lakukan perbaikan berdasarkan *suggestion* yang diberikan hingga tidak ada lagi *suggestion* yang ada. Tampilkan pada README bukti tangkapan layarnya. Selain itu, dokumentasikan juga perbaikan apa saja yang sudah kalian lakukan.

# Batasan Aplikasi

Berikut adalah batasan implementasi dari aplikasi:

1. Referensi daftar halaman dapat dilihat melalui [Figma](https://www.figma.com/file/AT53CebszSHtiEnDPGaQpQ/PBD?type=design&node-id=0%3A1&mode=design&t=DjHiCfrrEKhgiAts-1). Informasi yang ditampilkan minimal sesuai dengan yang ada di Figma, tetapi boleh ditambah dan diubah
2. Dalam aplikasi ini, pengguna diharuskan untuk melakukan komunikasi dengan Google Maps untuk membuka lokasi transaksi dan GMail untuk mengirimkan *cash flow* menggunakan Intent. Penggunaan Intent pada fitur lain dibebaskan
3. Penggunaan *library* untuk implementasi graf dan pembuatan *file* xlsx dan xls dibebaskan, tetapi harus dalam bahasa Kotlin
4. Implementasi *broadcast receiver* dibebaskan baik secara *local* atau tidak. Pastikan kalian memiliki alasan dibalik implementasi kalian.

# Backend

*Backend* untuk aplikasi dapat diakses melalui pranala berikut <https://pbd-backend-2024.vercel.app/>.

# Pengerjaan

Tugas besar 1 - Android **dikerjakan secara berkelompok.**

Sheets kelompok dapat diakses melalui **Files di Teams di *channel* Tubes 1.**

* Kelompok dibebaskan kepada mahasiswa dan boleh lintas kelas. Harap isi segera Sheets Daftar Kelompok Android pada Teams.
* Pengisian anggota kelompok **wajib** diisi pada slot yang sudah disediakan. **Tidak diperkenankan membuat slot baru**.
* Jumlah anggota kelompok adalah **3 - 4 orang**, perhatikan slot yang tersedia.
* Mahasiswa pertukaran sudah dimasukkan pada kelompok dengan nomor 2 hingga 13. **Harap tidak menukar atau menghapus nim yang sudah dimasukkan asisten.**
* **Utamakan mengisi kelompok dari nomor terkecil.**
* Tugas besar 1 **wajib** dikerjakan menggunakan ***Kotlin-Android Native***. **Bukan JAVA** dan **bukan *framework*** lain seperti Flutter, React Native, dll.
* Pastikan aplikasi bisa berjalan. Jika tidak berjalan sama sekali, asisten tidak bisa menilai apapun.
* **Disediakan *backend***dari asisten dan **wajib menggunakan *backend*** yang telah disediakan.
* Setiap aplikasi dibuat pada *repository* Gitlab Informatika dengan format nama sebagai berikut  
  **IF3210-2024-Android-XXX** dengan **XXX** = penomoran kelompok (3 huruf unik numerik sesuai pada Sheets Kelompok).
* Buat *release* pada gitlab untuk mengumpulkan, format *release* bebas, yang dipakai untuk menilai adalah release paling terakhir. Menu *release* ada di ***deployments***, tidak perlu menyertakan APK yang telah di *build*, cukup ***release*** ***tag*** yang ingin dikumpulkan. Pastikan isi dari ***tag*** tidak kosong dan berisikan kode yang kalian ingin kumpulkan!

Pada setiap *repository* ditambahkan sebuah README yang setidaknya berisi:

1. Deskripsi aplikasi.
2. *Library* yang digunakan.
3. *Screenshot* aplikasi (dimasukkan dalam folder *screenshot*).
4. Pembagian kerja anggota kelompok.
5. Jumlah jam persiapan dan pengerjaan untuk masing-masing anggota.

# Kriteria Penilaian

Berikut merupakan kriteria penilaian dari tugas besar ini:

1. Memilih teknik dan teknologi yang tepat untuk memenuhi spesifikasi.
2. Pemahaman terhadap platform Android.
3. Kreativitas & Keindahan aplikasi.
4. *User Experience* Asisten pada aplikasi android yang dibuat saat dilakukan *testing* pada saat Demo.  
   Contoh yang **mengurangi** *user experience*: *random crash*, Halaman yang sulit untuk digapai (harus ubah-ubah kode), *random behavior* (data kadang muncul, kadang tidak), dsb.
5. Kinerja dan *best practice* dari aplikasi yang dibangun.
6. Kerja sama dan manajemen kelompok.

# Question and Answer

Jika ada pertanyaan, silakan tulis pada ***channel* Tubes 1** di MSTeams kelas PBD 2023/2024.

*Kenapa tidak spreadsheet? Agar Tim Dosen dapat melihat keaktifan kalian dan lebih mudah memantau.*

# Link

1. [Email intent for Android](https://developer.android.com/guide/components/intents-common#Email)
2. [Google Maps Intent documentation](https://developers.google.com/maps/documentation/urls/android-intents#kotlin)
3. [How to read and write Excel file in Android](https://www.baeldung.com/kotlin/excel-read-write)
4. [SQLite Database Using Anko Kotlin | by Hanif Abdullah | SANDEC | Medium](https://medium.com/sandec/sqlite-database-using-anko-kotlin-40bcd3cc774c)
5. [Implementasi RecyclerView + CardView di Kotlin | by Anto D.](https://erthru.medium.com/implementasi-recyclerview-cardview-di-kotlin-268b8d896075)
6. [BottomNavigationView with Fragments](https://www.youtube.com/watch?v=OV25x3a55pk)
7. [Simple GET request using Retrofit in Android | Engineering Education (EngEd) Program | Section](https://www.section.io/engineering-education/making-api-requests-using-retrofit-android/)
8. [OWASP 10 Mobile](https://owasp.org/www-project-mobile-top-10/)
9. [Android - Broadcast Overview](https://developer.android.com/develop/background-work/background-tasks/broadcasts)
10. [Tutorial Broadcast Receiver](https://lobothijau.medium.com/broadcast-receivers-untuk-pemula-65a14ed03d88)

# 

# Lampiran

| Splash Screen | Halaman Login | Halaman Transaksi |
| --- | --- | --- |
| Halaman Penambahan Transaksi | Halaman Scan | Halaman Graf - Portrait |
| Halaman Graf - Landscape | | Halaman Pengaturan |