

UJIAN AKHIR SEMESTER
LAPORAN PROTOTYPE INOVASI APLIKASI PENERJEMAH BAHASA
ISYARAT BISINDO DENGAN MACHINE LEARNING: SUARA TANGAN
SIP 118 Pemrograman Aplikasi Bergerak

NAMA ANGGOTA KELOMPOK (NIM):

- Runi Dwi Jiasta (202304560001)
 - Teresa Kaena Dharmayoto (202304560014)
 - Caroline Evarista Den Lau (202304560027)
 - Andrew Riza Rafael (202304560030)
-

DAFTAR ISI

I. TOPIK APLIKASI & PENJELASAN.....	2
II. TIGA FITUR UTAMA.....	2
2.1 Fitur Penerjemah.....	2
2.2 Halaman Bantuan.....	3
2.3 Halaman Notifikasi.....	5
III. PENERAPAN MATERI.....	6
3.1 Android Menu.....	6
3.2 Android Dialog.....	9
3.3 Android ListView.....	12
IV. PENERAPAN ANDROID DATABASE (CRUD).....	13
4.1 Arsitektur Database.....	13
4.2 <i>CREATE</i> (Menambahkan Data).....	14
4.3 <i>READ</i> (Mengambil Data).....	21
4.4 <i>UPDATE</i> (Memodifikasi Data).....	23
4.5 <i>DELETE</i> (Menghapus Data).....	28
4.6 Keamanan Data & Penanganan Error.....	31
V. KELEBIHAN DAN KEKURANGAN APLIKASI.....	32
5.1 Kelebihan.....	32
5.2 Kekurangan.....	32
VI. PEMBAGIAN TUGAS.....	33

I. TOPIK APLIKASI & PENJELASAN

Nama Aplikasi	: Suara Tangan
Topik Aplikasi	: Inovasi Aplikasi Penerjemah Bahasa Isyarat BISINDO dengan <i>Machine Learning</i>
Alasan	: Aplikasi Suara Tangan adalah aplikasi mobile yang menggunakan teknologi <i>Machine Learning</i> untuk menjadi jembatan komunikasi <i>real-time</i> antara teman tuli/sulit mendengar dengan teman dengar di Indonesia.
Pemilihan Topik	

Pemilihan topik ini didorong oleh:

- **Kebutuhan Mendesak:** Berdasarkan data yang bersumber dari Sistem Informasi Manajemen Penyandang Disabilitas Kementerian Kesehatan Republik Indonesia pada Maret 2022, **19.392** penyandang disabilitas tuna wicara-rungu di Indonesia yang menghadapi hambatan komunikasi signifikan, keterbatasan interaksi sosial, serta tantangan dalam pendidikan dan pekerjaan.
- **Inklusi Sosial:** Proyek ini bertujuan untuk menghilangkan hambatan bahasa, mengurangi stigma dan diskriminasi, serta menciptakan lingkungan masyarakat yang lebih adil dan setara bagi teman tuli untuk berpartisipasi penuh dalam berbagai aspek kehidupan.
- **Relevansi Teknologi:** Aplikasi ini dikembangkan untuk menggunakan dataset BISINDO (Bahasa Isyarat Indonesia) agar dapat menerjemahkan secara akurat dan *real-time* menjadi teks dan suara, sejalan dengan tema aksesibilitas dan komunikasi.

II. TIGA FITUR UTAMA

2.1 Fitur Penerjemah

Fitur inti dari aplikasi “Suara Tangan” adalah penerjemah (*Translate*) yang dirancang untuk menjembatani komunikasi *real-time* antara teman tuli atau sulit mendengar dengan teman dengar di Indonesia. Fitur ini berfokus pada penerjemahan komprehensif dua arah dengan memadukan teknologi *Machine Learning* untuk kemampuan *Gesture to Text* dan *Speech to Text*. Model *Machine Learning* yang digunakan dalam fitur ini dilatih menggunakan kombinasi dataset BISINDO (Bahasa Isyarat Indonesia) dan Bahasa Indonesia, sehingga mampu menginterpretasikan input dan menyajikan hasil akhirnya di layar dalam format teks yang ringkas dan mudah dipahami.

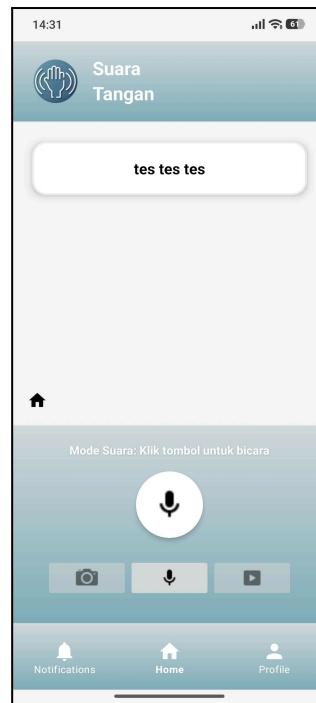
Secara fungsional, fitur penerjemah ini memberikan fleksibilitas tinggi kepada pengguna melalui tiga metode input utama, yaitu Kamera, Suara, dan Video. Mode Kamera memungkinkan pengguna menangkap dan menerjemahkan gerakan isyarat BISINDO secara waktu nyata menjadi teks. Sementara itu, mode Suara berfungsi untuk merekam ucapan lisan Bahasa Indonesia dan mengkonversinya menjadi teks agar dapat dibaca oleh teman tuli. Sebagai tambahan, tersedia mode Video yang memungkinkan pengguna mengunggah video isyarat untuk dianalisis dan diterjemahkan oleh sistem. Keberadaan ketiga mode ini bertujuan untuk menciptakan komunikasi dua arah yang lancar dan efektif serta memecah hambatan komunikasi dalam berbagai situasi.

Dari sisi desain antarmuka, fitur ini menampilkan estetika yang elegan dan kontemporer dengan dominasi palet warna biru muda dan putih. Desainnya

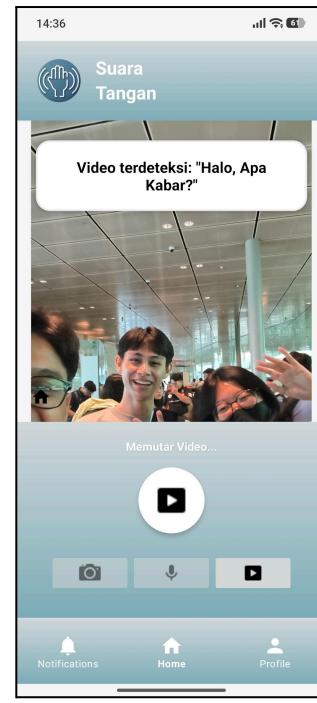
mengutamakan kemudahan penggunaan, di mana pengguna dapat berpindah fitur secara cepat antara terjemahan suara, kamera, dan video melalui bilah navigasi di bagian bawah layar. Fitur suara ditempatkan secara strategis di posisi tengah dengan ikon mikrofon besar yang dilengkapi petunjuk interaksi eksplisit, memastikan pengguna dapat mengakses dan memanfaatkan berbagai metode terjemahan dengan navigasi yang intuitif dan mudah dipahami.



Gambar 1. Penerjemah Mode Kamera



Gambar 2. Penerjemah Mode Suara



Gambar 3. Penerjemah Mode Video

2.2 Halaman Bantuan

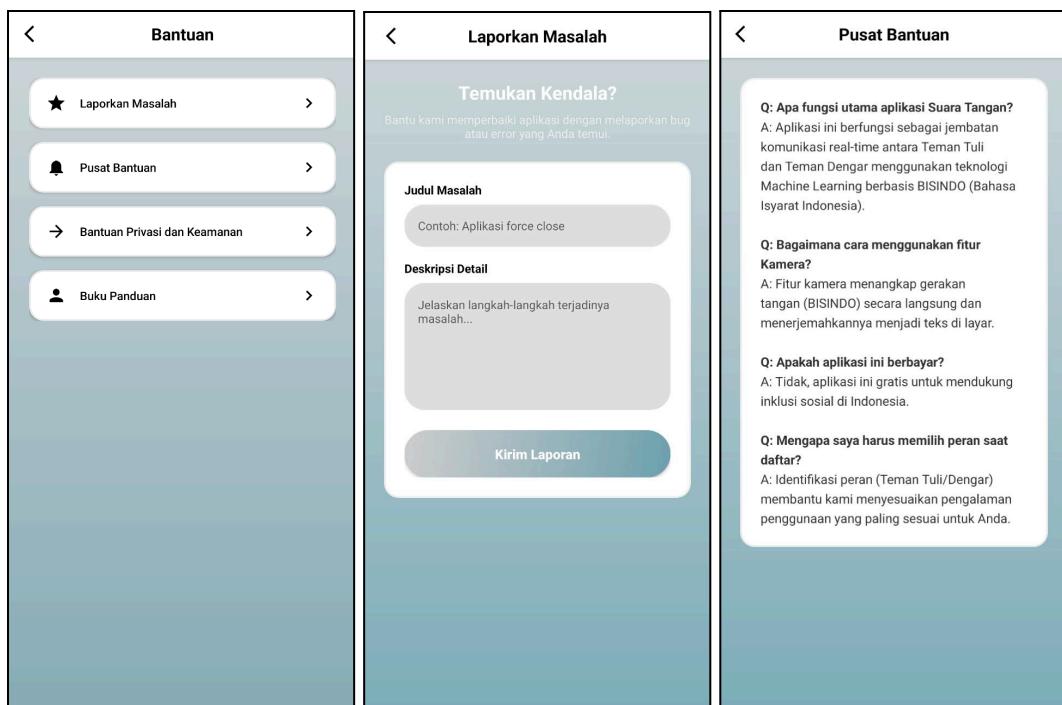
Fitur pendukung vital dalam aplikasi "Suara Tangan" adalah menu **Bantuan** yang dirancang sebagai pusat dukungan pengguna untuk memastikan kelancaran dan kenyamanan pengalaman penggunaan aplikasi. Fitur ini berfokus pada penyediaan *User Support* yang aktif dan responsif, bertindak sebagai hub sentral bagi pengguna untuk mendapatkan informasi yang dibutuhkan. Dengan adanya fitur ini, aplikasi memastikan bahwa setiap hambatan teknis maupun kebutuhan informasi pengguna dapat terakomodasi dengan baik guna mendukung pengembangan sistem yang berkelanjutan.

Secara fungsional, menu Bantuan menyediakan struktur dukungan yang komprehensif melalui empat opsi utama yang memiliki fungsi spesifik:

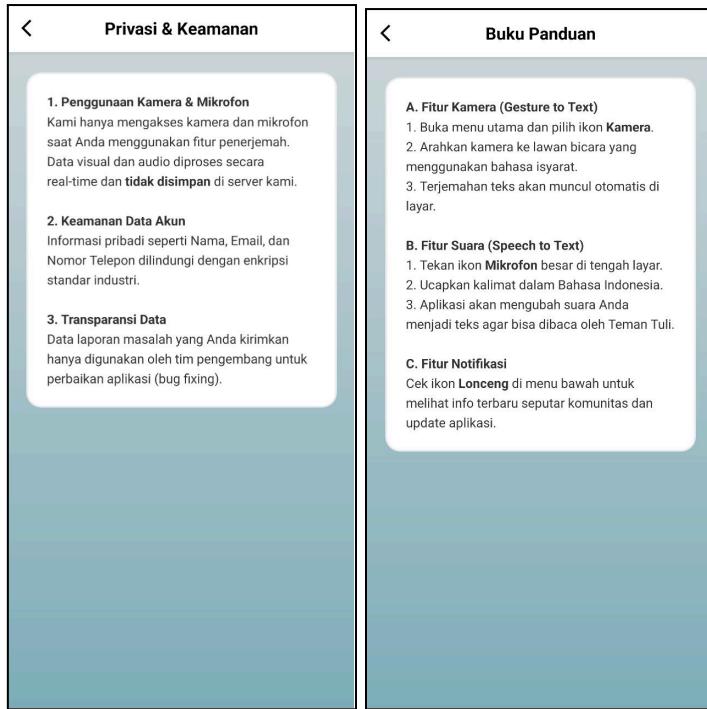
- Laporkan Masalah:** Fitur ini berfungsi sebagai saluran pengaduan dinamis di mana sistem secara otomatis mendeteksi identitas pengguna (`user_id`) untuk mempermudah pelacakan dan perbaikan bug oleh tim pengembang.

2. **Pusat Bantuan:** Bagian ini menyajikan informasi mendasar berupa FAQ untuk menjawab pertanyaan umum seputar peran pengguna dan mekanisme kerja aplikasi.
3. **Bantuan Privasi dan Keamanan:** Opsi ini memberikan transparansi krusial mengenai perlindungan data, menegaskan bahwa akses kamera dan mikrofon hanya terjadi secara real-time tanpa penyimpanan permanen, serta menjamin enkripsi data pribadi pengguna.
4. **Buku Panduan:** Fitur ini menyediakan instruksi teknis langkah demi langkah, memandu pengguna dalam mengoperasikan fitur utama seperti penerjemah isyarat berbasis kamera (*Gesture to Text*) dan fitur suara (*Speech to Text*).

Dari sisi desain antarmuka, fitur ini memisahkan fungsi Bantuan dalam tampilan yang *user friendly* dan terorganisir. Setiap kategori bantuan disajikan dengan ikon visual yang mudah dipahami, memudahkan pengguna dalam mengidentifikasi jenis bantuan yang mereka perlukan.



Gambar 4-6. Halaman Bantuan, Laporkan Masalah, dan Pusat Bantuan



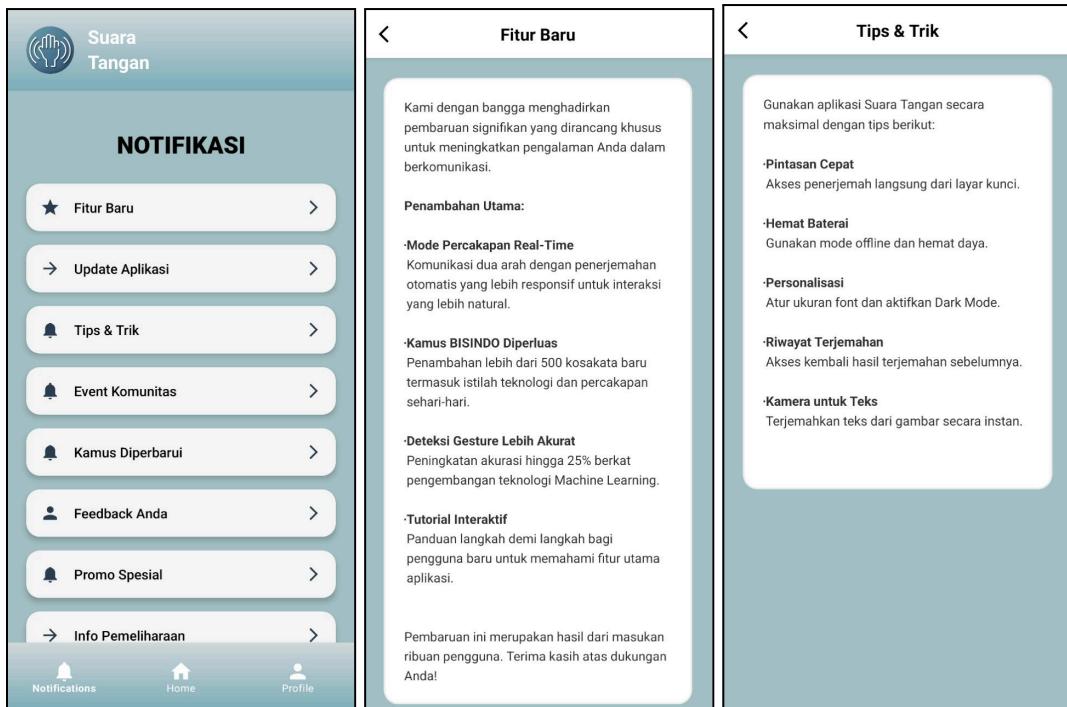
Gambar 7-8. Halaman Bantuan Privasi & Keamanan, dan Buku Panduan

2.3 Halaman Notifikasi

Halaman **Notifikasi** merupakan salah satu fitur dari aplikasi Suara Tangan yang berfungsi sebagai laman pemberitahuan pengguna tentang berbagai informasi penting secara tepat dan teratur. Dalam halaman ini, pengguna akan mendapat pemberitahuan mengenai pembaruan aplikasi, perubahan sistem, konten terbaru, serta informasi yang mendukung penggunaan aplikasi. Jenis pemberitahuan yang tersedia dalam fitur notifikasi ini mencakup informasi tentang Fitur Baru, *Update Aplikasi* yang mencangkup penambahan fasilitas, peningkatan kinerja, dan perbaikan masalah, Tips & Trik untuk membantu pengguna memahami dan menggunakan semua fitur secara optimal.

Selain itu, halaman ini juga memberi pemberitahuan tentang *Event Komunitas* untuk menginformasikan kegiatan atau acara yang relevan dengan pengguna, Lalu Kamus Diperbaharui (Bahasa) sebagai pemberitahuan adanya pembaruan kosakata ataupun bahasa yang baru ditambahkan dalam aplikasi, Feedback Anda digunakan sebagai saluran komunikasi antara pengguna dan pengembang, serta Promo Spesial yang berisi informasi mengenai penawaran khusus.

Halaman Notifikasi juga mencangkup Info Pemeliharaan untuk memberitahu pengguna mengenai jadwal atau proses perawatan sistem, serta Kuis Mingguan yang bertujuan untuk meningkatkan interaksi dan keterlibatan pengguna. Dengan adanya Fitur Notifikasi, pengguna dapat dengan mudah mendapatkan informasi terbaru tanpa harus mencari secara langsung, sehingga pengalaman menggunakan aplikasi menjadi lebih efektif, informatif, dan nyaman.



Gambar 9-11. Halaman Notifikasi dan Contoh Artikel Notifikasi

III. PENERAPAN MATERI

3.1 Android Menu

1. Bottom Navigation View (Menu Utama)

Bottom Navigation View adalah standar navigasi modern pada Android yang menampilkan menu di bagian bawah layar. Konsep ini memungkinkan pengguna berpindah antar fitur utama (Beranda, Notifikasi, Profil) dengan satu kali ketukan, memberikan akses cepat tanpa perlu memuat ulang *Activity* secara keseluruhan. Di dalam `MainActivity.java`, kami menggunakan `BottomNavigationView` untuk mengelola navigasi antara Beranda, Notifikasi, dan Profil. Kami menerapkan `setOnItemSelectedListener` untuk mendeteksi item mana yang diklik. Berdasarkan ID item tersebut, kami menggunakan `FragmentTransaction` untuk mengganti (*replace*) tampilan fragmen yang aktif.

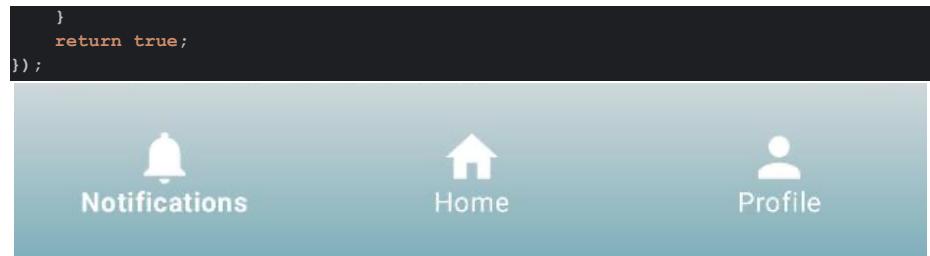
Kode Implementasi:

a. MainActivity.java

```
// Inisialisasi Bottom Navigation
BottomNavigationView bottomNav = findViewById(R.id.bottom_navigation);

// Listener untuk menangani perpindahan menu
bottomNav.setOnItemSelectedListener(item -> {
    Fragment selectedFragment = null;
    int itemId = item.getItemId();
    // Menentukan fragmen tujuan berdasarkan ID menu
    if (itemId == R.id.nav_notifications) {
        selectedFragment = new NotificationFragment();
    } else if (itemId == R.id.nav_home) {
        selectedFragment = new HomeFragment();
    } else if (itemId == R.id.nav_profile) {
        selectedFragment = new FragmentProfile();
    }

    // Mengganti tampilan fragmen
    if (selectedFragment != null) {
        loadFragment(selectedFragment);
    }
})
```



Gambar 12. Menu Utama

2. Dashboard Menu (Bantuan)

Android Menu biasanya identik dengan *Option Menu* (titik tiga di pojok atas). Namun, untuk fitur Bantuan, kami menerapkan konsep Menu Dashboard yang menggunakan tombol navigasi besar agar lebih mudah diakses.

Penerapan konsep Android Menu dengan gaya *Dashboard* pada halaman Bantuan menggunakan komponen `AppCompatActivity` yang berfungsi sebagai item navigasi. Tombol-tombol ini terhubung ke Activity melalui inisialisasi `findViewById()`, sehingga dapat dimanipulasi secara programatis. Penggunaan metode `setOnClickListener()` pada setiap tombol memungkinkan aplikasi mendeteksi *input* pengguna. Saat tombol ditekan, metode ini akan memicu pembuatan objek `Intent` dan memanggil `startActivity()`, yang secara otomatis memindahkan pengguna ke halaman fitur yang relevan (seperti formulir laporan atau detail panduan). Implementasi ini menunjukkan penggunaan menu navigasi visual yang intuitif, memprioritaskan kemudahan akses dibandingkan menu opsi standar.

Kode Implementasi:

a. BantuanActivity.java

```
// Inisialisasi tombol menu
Button btnLapor = findViewById(R.id.btn_lapor_masalah);
Button btnPusat = findViewById(R.id.btn_pusat_bantuan);
Button btnPrivasi = findViewById(R.id.btn_privasi);

// Navigasi ke Activity Form (Lapor Masalah)
btnLapor.setOnClickListener(v ->
    startActivity(new Intent(this, LaporanMasalahActivity.class))
);

// Navigasi ke Detail dengan membawa data konten (Pusat Bantuan)
    btnPusat.setOnClickListener(v -> openDetail("Pusat Bantuan", faqContent));

// Metode helper untuk membuka detail
private void openDetail(String title, String content) {
    Intent intent = new Intent(this, DetailBantuanActivity.class);
    intent.putExtra("TITLE", title);
    intent.putExtra("CONTENT", content);
    startActivity(intent);
}
```

b. activity_bantuan.xml (Layout Menu)

```
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/btn_lapor_masalah"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:text="Laporkan Masalah"
    android:drawableStart="@drawable/ic_new_feature"
    android:drawableEnd="@drawable/ic_arrow_right"
    android:paddingStart="20dp"
    android:paddingEnd="20dp"
    android:gravity="center_vertical|start" />
```

3. Dashboard Menu (Profil)

Halaman Profil juga mengadopsi konsep Menu *Dashboard* yang serupa dengan fitur Bantuan. Menu ini menyajikan daftar opsi pengaturan akun dalam bentuk tombol-tombol yang tersusun rapi, memberikan tampilan antarmuka yang bersih dan mudah dipahami (*user-friendly*).

Pada `FragmentProfile.java`, menu diimplementasikan sebagai sekumpulan tombol navigasi. Tombol-tombol ini dihubungkan ke tampilan melalui metode `findViewById()` di dalam `onCreateView`, menjadikannya elemen interaktif utama pada layar profil. Melalui metode `setOnClickListener()`, setiap tombol diberi instruksi spesifik untuk menangani transisi halaman. Ketika salah satu menu dipilih, sistem menggunakan `Intent` eksplisit untuk meluncurkan `Activity` baru tanpa menutup aplikasi, menjaga alur navigasi tetap lancar. Implementasi ini menunjukkan pengorganisasian fitur akun yang terstruktur dan konsisten, memudahkan pengguna mengelola data pribadi mereka.

Kode Implementasi:

a. FragmentProfile.java

```
// Inisialisasi tombol menu profil
Button btnInfoAkun = view.findViewById(R.id.btn_info_akun);
Button btnSandi = view.findViewById(R.id.btn_ubah_sandi);

// Navigasi ke halaman Informasi Akun
btnInfoAkun.setOnClickListener(v -> {
    Intent infoIntent = new Intent(getActivity(), InfoAkunActivity.class);
    // Meneruskan data intent dari activity induk jika ada
    if (getActivity().getIntent().getExtras() != null) {
        infoIntent.putExtras(getActivity().getIntent().getExtras());
    }
    startActivity(infoIntent);
});

// Navigasi ke halaman Ubah Kata Sandi
btnSandi.setOnClickListener(v ->
startActivity(new Intent(getActivity(), UbahSandiActivity.class))
);
```

b. fragment_profile.xml (Layout Menu)

```
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/btn_info_akun"
    style="@style/MenuButtonStyle"
    android:text="Informasi Akun"
    android:drawableStart="@drawable/ic_profile" />
```

4. Menu Navigasi (Fragment *Notification*)

Pada halaman Notifikasi, konsep Android menu diterapkan menggunakan sistem navigasi berbasis Fragment, di mana halaman Notifikasi bertindak sebagai salah satu menu utama yang dapat diakses oleh pengguna. Implementasi ini dilakukan melalui penggunaan `NotificationFragment` yang berfungsi sebagai ruang penyimpanan untuk menampilkan daftar notifikasi. Fragment ini dihubungkan dengan tampilan antarmuka melalui metode `onCreateView()`, yang menghubungkan Fragment dengan layout `activity_notification.xml` sebagai tampilan utama. Dengan pendekatan ini, setiap kali pengguna memilih

menu Notifikasi dari navigasi utama aplikasi, sistem akan menampilkan **NotificationFragment** tanpa perlu memuat ulang Activity utama.

Penggunaan Fragment sebagai elemen navigasi memungkinkan perpindahan antar menu yang lebih cepat, efisien, dan hemat sumber daya, sekaligus memberikan pengalaman pengguna yang lebih responsif serta sesuai dengan standar pengembangan aplikasi Android modern.

Kode Implementasi :

a. **NotificationFragment.java**

```
public class NotificationFragment extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
    ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.activity_notification, container,
    false);
    }
}
```

3.2 Android Dialog

1. Dialog Konfirmasi *Logout* (*Custom Alert Dialog*)

Custom Alert Dialog memungkinkan pengembang untuk menampilkan jendela *pop-up* dengan desain yang sepenuhnya disesuaikan dengan tema aplikasi, berbeda dengan *Alert Dialog* standar yang hanya berisi teks. Dialog ini dirancang untuk menangani interaksi pilihan ganda yang krusial, seperti memisahkan opsi “Keluar” (*Logout*) biasa dengan opsi “Hapus Akun” yang berisiko tinggi dalam satu tampilan yang rapi.

Pada **FragmentProfile.java**, ketika pengguna menekan tombol “Keluar”, sebuah metode `showLogoutOptions()` dipanggil. Metode ini menggunakan **AlertDialog.Builder** untuk membuat dialog. Alih-alih menggunakan pesan teks standar, kami “menempelkan” (*inflate*) layout khusus (**dialog_custom_logout.xml**) ke dalam dialog tersebut menggunakan `builder.setView()`. Ini memungkinkan kami memiliki dua tombol aksi berbeda (“Log Out” dan “Hapus Akun”) dalam satu dialog.

Kode Implementasi:

a. **FragmentProfile.java**

```
private void showLogoutOptions() {
    // 1. Setup Builder dan Inflate Layout Custom
    AlertDialog.Builder builder = new AlertDialog.Builder(requireContext());
    View dialogView = LayoutInflater.from(requireContext()).inflate(R.layout.dialog_custom_logout,
    null);
    builder.setView(dialogView);
    AlertDialog dialog = builder.create();

    // 2. Membuat background dialog transparan (agar rounded corner terlihat)
    if (dialog.getWindow() != null)
        dialog.getWindow().setBackgroundDrawable(new
    ColorDrawable(Color.TRANSPARENT));
    }

    // 3. Inisialisasi komponen di dalam dialog
    Button btnLogout = dialogView.findViewById(R.id.btn_dialog_logout);
    Button btnDelete = dialogView.findViewById(R.id.btn_dialog_delete);
    TextView tvCancel = dialogView.findViewById(R.id.tv_dialog_cancel);

    // 4. Logika Tombol Logout
    btnLogout.setOnClickListener(v -> {

```

```

        dialog.dismiss(); // Tutup dialog
        performLogout(false); // Lakukan proses logout
    });

    // 5. Logika Tombol Hapus Akun (membuka dialog konfirmasi lanjutan)
    btnDelete.setOnClickListener(v -> {
        dialog.dismiss();
        showFinalDeleteConfirmation();
    });

    dialog.show();
}

```

b. dialog_custom_logout.xml

```

<LinearLayout ...
    android:background="@drawable/bg_white_dialog"
    android:orientation="vertical"
    android:padding="32dp">

    <TextView
        android:text="Konfirmasi Keluar"
        android:textSize="20sp"
        android:textStyle="bold" ... />

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btn_dialog_logout"
        android:text="Log Out" ... />

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btn_dialog_delete"
        android:background="@drawable/button_red_rounded"
        android:text="Hapus Akun Permanen" ... />
</LinearLayout>

```

2. Dialog Input Password (Validasi Input)

Dialog tidak hanya terbatas pada tombol pilihan, tetapi juga dapat dimanfaatkan untuk menerima *input* data (*User Input*). Dalam fitur keamanan, dialog *input* sangat efektif sebagai “jaring pengaman” terakhir. Dialog ini mewajibkan pengguna memasukkan kata sandi sebelum melakukan tindakan destruktif (menghapus akun) untuk memastikan bahwa tindakan tersebut benar-benar dilakukan oleh pemilik akun.

Pada `FragmentProfile.java`, kami menerapkan dialog yang memuat elemen `TextInputEditText` dari layout `dialog_password_input.xml`. Logika validasi dilakukan langsung di dalam dialog: ketika tombol konfirmasi ditekan, aplikasi mengambil teks dari `EditText` tersebut dan mengirimkannya ke fungsi penghapusan akun.

Kode Implementasi:

a. FragmentProfile.java

```

// Method untuk menampilkan Dialog Input Password
private void showPasswordInputDialog(int userId, SharedPreferences prefs) {
    AlertDialog.Builder builder = new AlertDialog.Builder(requireContext());
    // Inflate layout yang memuat EditText
    View dialogView = LayoutInflater.from(requireContext()).inflate(R.layout.dialog_password_input,
        null);
    builder.setView(dialogView);

    // Mengambil referensi input dari dalam dialog
    TextInputEditText etPassword = dialogView.findViewById(R.id.et_password_confirm);
    Button btnConfirm = dialogView.findViewById(R.id.btn_confirm_password);

    btnConfirm.setOnClickListener(v -> {
        String password = etPassword.getText().toString().trim();
        // Validasi password dan eksekusi API hapus akun
    });
}

```

```

        if (!password.isEmpty()) {
            dialog.dismiss();
            performDeleteAccount(userId, password, prefs);
        }
    });

    dialog.show();
}

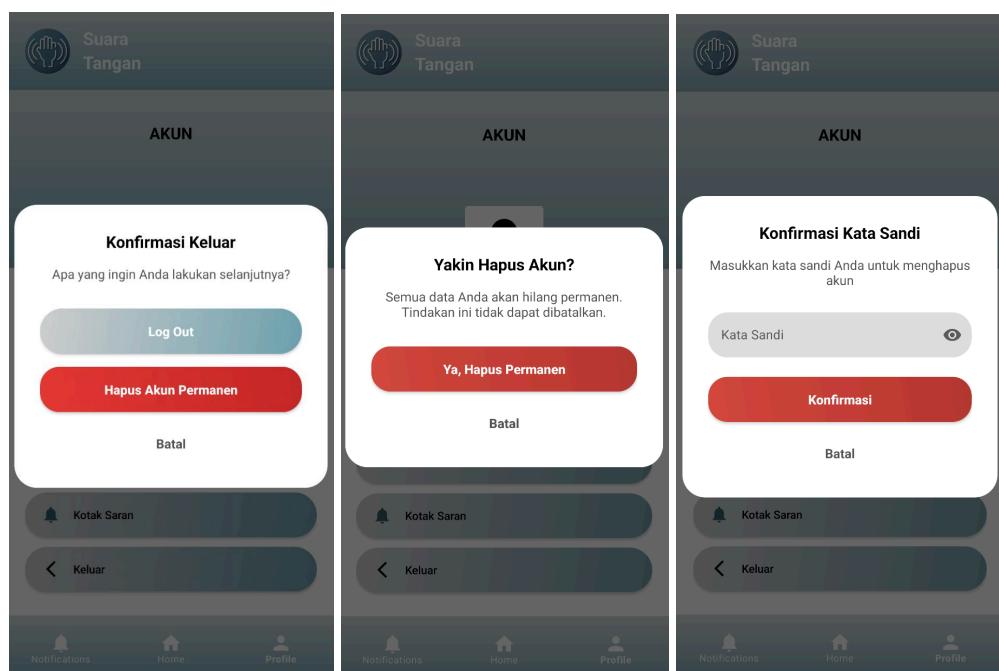
```

b. dialog_password_input.xml

```

<LinearLayout ... >
    <com.google.android.material.textfield.TextInputLayout ...>
        <com.google.android.material.textfield.TextInputEditText
android:id="@+id/et_password_confirm"
android:inputType="textPassword" />
    </com.google.android.material.textfield.TextInputLayout>
</LinearLayout>

```



Gambar 13-15. Dialog Konfirmasi *Logout*, Menghapus Akun, dan Verifikasi Kata Sandi untuk Menghapus Akun

3. Dialog Detail Notification

Pada halaman Notifikasi, penggunaan konsep Android Dialog diterapkan melalui mekanisme tampilan halaman detail notifikasi yang muncul sebagai respons terhadap interaksi pengguna terhadap item notifikasi. Implementasi ini terlihat pada `NotificationAdapter`, di mana setiap item pada daftar notifikasi dilengkapi dengan `event listener` menggunakan `setOnClickListener()`. Ketika pengguna menekan salah satu notifikasi, aplikasi akan membuka `NotificationDetailActivity` dengan membawa data dan isi notifikasi melalui objek `Intent`. Halaman detail ini dirancang untuk menjadi media penyampaian informasi yang lebih informatif namun dengan tampilan layar penuh (*full-screen dialog*) agar konten yang ditampilkan lebih lengkap dan nyaman dibaca.

Di dalam `NotificationDetailActivity`, data yang diterima dari `Intent` ditampilkan pada komponen `TextView`, sementara tombol kembali

diimplementasikan menggunakan `ImageView` dengan `listener finish()` untuk menutup halaman detail. Pendekatan ini memungkinkan penyajian informasi notifikasi secara interaktif dan terstruktur, sekaligus memberikan pengalaman pengguna yang lebih baik dibandingkan dialog konvensional yang terbatas.

Kode Implementasi:

a. NotificationAdapter.java

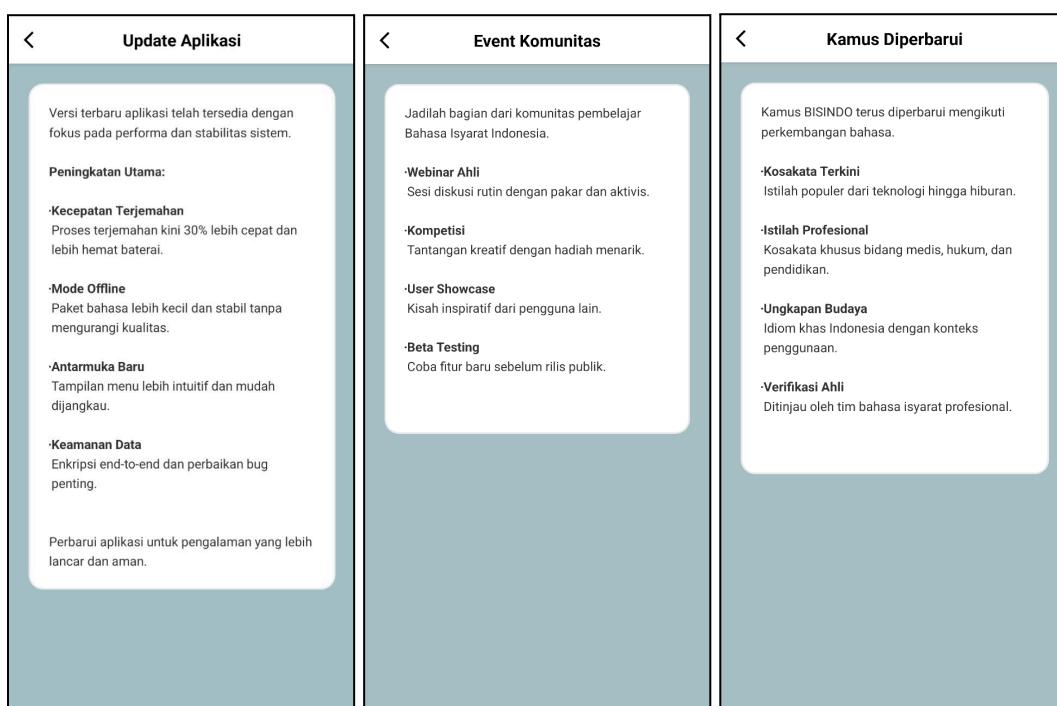
```
itemView.setOnClickListener(v -> {
    Context context = v.getContext();
    String title = notification.getTitle();
    String content = getNotificationContent(title);

    Intent intent = new Intent(context, NotificationDetailActivity.class);
    intent.putExtra("NOTIFICATION_TITLE", title);
    intent.putExtra("NOTIFICATION_CONTENT", content);
    context.startActivity(intent);
});
```

b. NotificationDetailActivity.java

```
String title = getIntent().getStringExtra("NOTIFICATION_TITLE");
String content = getIntent().getStringExtra("NOTIFICATION_CONTENT");

if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.N) {
    tvDetailContent.setText(Html.fromHtml(content, Html.FROM_HTML_MODE_LEGACY));
} else {
    tvDetailContent.setText(Html.fromHtml(content));
}
```



Gambar 16-18. Contoh Detail Notifikasi

3.3 Android ListView

1. Daftar Notifikasi menggunakan `RecyclerView`

Penerapan konsep `ListView` pada fitur Notifikasi diimplementasikan menggunakan `RecyclerView`, yang merupakan pengembangan dari `ListView` dengan performa dan fleksibilitas yang lebih optimal. Implementasi ini dapat

dilihat pada `NotificationFragment`, di mana `RecyclerView` digunakan untuk menampilkan daftar notifikasi dalam bentuk list vertikal. `RecyclerView` diatur menggunakan `LinearLayoutManager` agar setiap item notifikasi tersusun secara berurutan dari atas ke bawah dan dapat di gulir kebawah maupun keatas (*scroll*).

Data notifikasi disimpan dalam sebuah objek List, yang berisi judul notifikasi dan ikon pendukung, serta menangani interaksi pengguna pada setiap item. Penggunaan `ViewHolder` pada adapter berfungsi untuk meningkatkan efisiensi pemrosesan tampilan dengan mengurangi pemanggilan ulang komponen UI, sehingga aplikasi tetap ringan dan responsif meskipun menampilkan banyak notifikasi. Dengan implementasi ini, fitur Notifikasi mampu menyajikan informasi secara terstruktur, dinamis, dan sesuai dengan pengembangan Android modern.

Kode Implementasi:

a. `NotificationFragment.java`

```
RecyclerView rvNotifications = view.findViewById(R.id.rv_notifications);
rvNotifications.setLayoutManager(new
LinearLayoutManager(getApplicationContext()));

List<Notification> notificationList = new ArrayList<>();

notificationList.add(new Notification("Fitur Baru",
R.drawable.ic_new_feature));
notificationList.add(new Notification("Update Aplikasi",
R.drawable.ic_system_update));
notificationList.add(new Notification("Tips & Trik", R.drawable.ic_notif));
notificationList.add(new Notification("Event Komunitas",
R.drawable.ic_notifications));
notificationList.add(new Notification("Kamus Diperbarui",
R.drawable.ic_notif));
notificationList.add(new Notification("Feedback Anda",
R.drawable.ic_profile));
notificationList.add(new Notification("Promo Spesial",
R.drawable.ic_notif));
notificationList.add(new Notification("Info Pemeliharaan",
R.drawable.ic_system_update));
notificationList.add(new Notification("Kuis Mingguan",
R.drawable.ic_home));

NotificationAdapter adapter = new NotificationAdapter(notificationList);
rvNotifications.setAdapter(adapter);
}
```

IV. PENERAPAN ANDROID *DATABASE* (CRUD)

4.1 Arsitektur *Database*

Aplikasi Suara Tangan mengimplementasikan arsitektur *Client-Server* dengan Android sebagai *client* dan PHP MySQL sebagai *backend server*. Data pengguna, laporan masalah, dan saran disimpan secara terpusat di *database* MySQL yang diakses melalui API RESTful. Komunikasi antara Android dan server menggunakan *library* Volley untuk HTTP *requests* dengan format *response* JSON. Berikut adalah struktur *database* yang digunakan:

1. Tabel `users`: Menyimpan data akun (`id`, `nama_lengkap`, `username`, `email`, `telepon`, `password`, `peran`)

id	nama_lengkap	username	email	telepon	password	peran
8	Teresa Kaena Dharmanyoto	kaee	teresa@mail.com	081282825601	\$2y\$10\$TW72YWWS5Eq1IRi1FxxBKYe6kdS/SssqW0NFSHUkhlE...	Teman Tuli
9	Runi Dwi Jiasta	rungi	rungi@mail.com	082222222222	\$2y\$10\$xMTW4TGJievwGE.p5YpUZen/ZspuIQ.vhECKk4eL0k...	Teman Dengar
10	Andrew Riza Rafael	andrew	andrew@mail.com	08111111111	\$2y\$10\$2gI6QhSaWnWUQs0TW/QMo.QuoVpnwyAv2yyuHnE2ULk...	Teman Tuli
11	Caroline Evarista Den Lau	evaa	evaa@mail.com	083333333333	\$2y\$10\$FVQ34ZnlmP073qO83.DOKCMlgeomsfFwcrHj3z4Ml...	Teman Dengar

Gambar 19. Tampilan Tabel `users` dalam PHP MySQL

2. Tabel `laporan_masalah`: Menyimpan laporan *bug* (`id`, `user_id`, `judul`, `deskripsi`, `tanggal`)

id	user_id	judul	deskripsi	tanggal	status
2	11	Bug dalam Header Notifikasi	Header notifikasi berwarna hijau. Tidak selaras dengan...	2026-01-08 00:40:51	pending
3	8	Terjemahan Kurang Akurat	Hasil penerjemahan tidak sesuai dengan arti bahasa...	2026-01-08 00:42:44	pending
4	9	Aplikasi Terlalu Bagus	:)	2026-01-08 00:44:03	pending

Gambar 20. Tampilan Tabel `laporan_masalah` dalam PHP MySQL

3. Tabel `kotak_saran`: Menyimpan *feedback User* (`id`, `user_id`, `saran`, `tanggal`)

id	user_id	saran	tanggal
1	8	Bismillah PAB dapet A++ :D	2026-01-08 00:51:54
2	11	Puji Tuhan kelar	2026-01-08 00:55:39
3	9	Tetap semangat, yaa!	2026-01-08 00:58:44

Gambar 21. Tampilan Tabel `kotak_saran` dalam PHP MySQL

Berikut adalah rangkaian teknologi yang digunakan:

1. Android Volley: HTTP *client* untuk *network requests*
2. PHP PDO: *Database access layer* dengan *prepared statements*
3. JSON: Format pertukaran data antara *client-server*
4. `SharedPreferences`: Penyimpanan lokal untuk *session management*

4.2 **CREATE** (Menambahkan Data)

A. Registrasi Pengguna Baru

Proses registrasi dimulai ketika *User* mengisi formulir pendaftaran dengan data lengkap (nama, `username`, email, telepon, `password`, dan peran). Sistem melakukan validasi *input* di sisi Android terlebih dahulu untuk memastikan format data benar sebelum dikirim ke *server*. Setelah validasi lolos, data dikirim via POST *request* menggunakan Volley ke *endpoint* register.php. *Server* kemudian memverifikasi data, meng-hash `password` dengan `bcrypt`, dan menyimpan ke *database*.

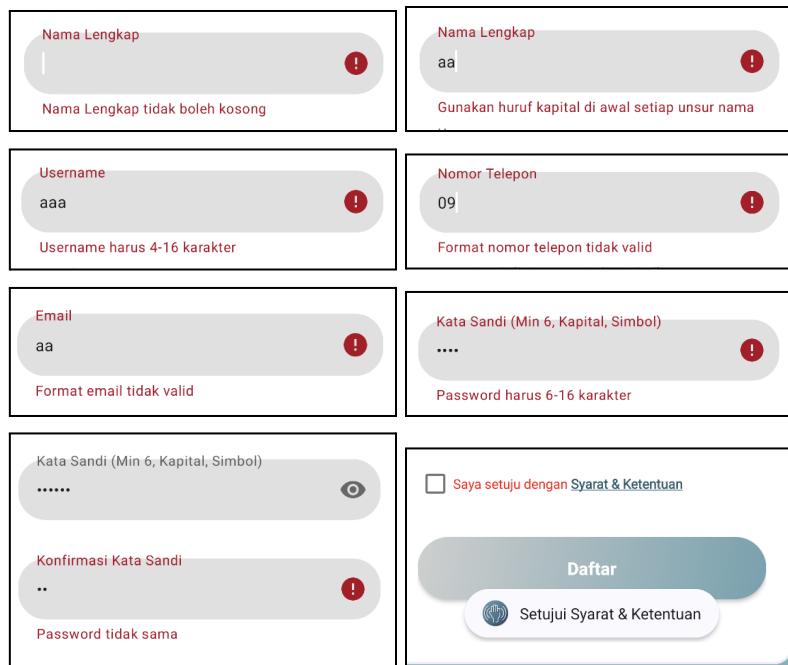
Kode Validasi Kolom “Nama Lengkap” dalam RegisterFragment.java:

```
if (nama.isEmpty()) {
    tilNama.setError("Nama Lengkap tidak boleh kosong");
    return;
}
boolean isKapitalValid = true;
String[] words = nama.split(" "));
```

```

for (String word : words) {
    if (word.length() > 0) {
        if (!Character.isUpperCase(word.charAt(0))) {
            isKapitalValid = false;
            break;
        }
    }
}

```



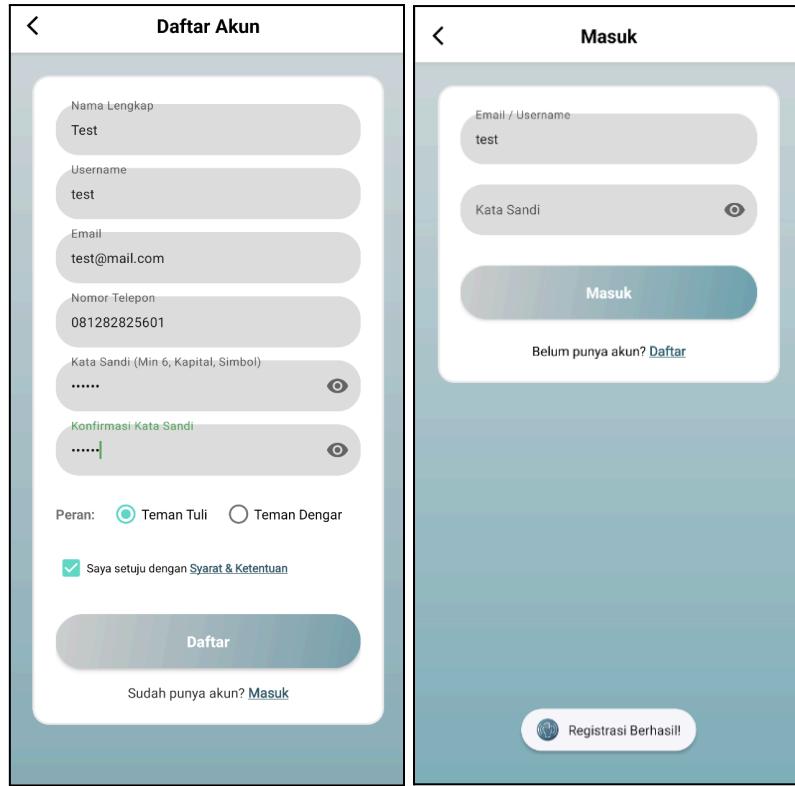
Gambar 22. *Toast* dan Peringatan Jika Kolom Form Tidak Terisi dengan Benar

Snippet Kode Network Request dalam RegisterFragment.java:

```

StringRequest stringRequest = new StringRequest(Request.Method.POST,
ApiConfig.REGISTER_URL,
response -> { JSONObject jsonObject = new JSONObject(response);
if ("success".equals(jsonObject.getString("status"))) {
    Toast.makeText(getApplicationContext(), "Registrasi Berhasil!",
    LENGTH_SHORT).show(); navigateToLogin(bundle);
}
},
error -> { /* handle error */ }
) {
    @Override
    protected Map<String, String> getParams() {
        Map<String, String> params = new HashMap<>();
        params.put("nama_lengkap", nama);
        params.put("username", username);
        params.put("email", email);
        params.put("telepon", telepon);
        params.put("password", password);
        params.put("peran", peran);
        return params;
    }
};

```



Gambar 23. Contoh Form Registrasi yang Terisi dengan Benar
 Gambar 24. *Toast* Registrasi Berhasil dan *Redirect* Otomatis ke Halaman *Login*

Kode *Insert Database* dalam register.php ketika Pendaftaran *User* Berhasil:

```
$hashed_pass = password_hash($pass, PASSWORD_BCRYPT);

try {
    $sql = "INSERT INTO users (nama_lengkap, username, email, telepon,
password, peran)
VALUES (:nama, :user, :email, :telp, :pass, :peran)";
$stmt = $pdo->prepare($sql);

$stmt->execute([
    ':nama' => $nama,
    ':user' => $user,
    ':email' => $email,
    ':telp' => $telp,
    ':pass' => $hashed_pass,
    ':peran' => $peran
]);

echo json_encode(["status" => "success", "message" => "User
registered successfully!"]);
}
```

id	nama_lengkap	username	email	telepon	password	peran
8	Teresa Kaena Dharmanyoto	kaee	teresa@mail.com	081282825601	\$2y\$10\$TW72YWW5Eq1IR1FxxBKYe6kdS/SssqW0NFSHUKhIE2...	Teman Tuli
9	Runi Dwi Jiasta	runi	runi@mail.com	082222222222	\$2y\$10\$xMiW4TGJiewvGE p5YpUZen/ZspuIQ.vhECKk4eLC0k...	Teman Dengar
10	Andrew Riza Rafael	andrew	andrew@mail.com	081111111111	\$2y\$10\$2gj6QhSaWnWUQs0TW/QMo.QuoVpnwyAv2yyuHnE2ULk...	Teman Tuli
11	Caroline Evarista Den Lau	evaa	evaa@mail.com	083333333333	\$2y\$10\$FVQ34ZNrmImP073qO83.DOKCMJgeomsffwcrHj3z4Ml...	Teman Dengar
14	Test	test	test@mail.com	081282825601	\$2y\$10\$4GTCio6Trp8UyjvHBMjb.vqbu3ad8vPSFYs2CWo8Lb...	Teman Tuli

Gambar 25. Data *User* Baru Tersimpan di Tabel **users** (ID: 14)

B. Submit Laporan Masalah

Fitur laporan masalah memungkinkan *User* melaporkan *bug* atau kendala dalam aplikasi. *User* mengisi judul dan deskripsi masalah, kemudian sistem mengambil `user_id` dari `SharedPreferences` untuk mengidentifikasi pelapor. Data dikirim ke *server* dan disimpan di tabel `laporan_masalah` dengan relasi *foreign key* ke tabel `users`.

Kode Implementasi:

a. LaporanMasalahActivity.java memanggil `user_id`:

```
private void submitLaporan(String judul, String deskripsi) {
    SharedPreferences prefs = getSharedPreferences("GoScorePrefs",
        Context.MODE_PRIVATE);
    int userId = prefs.getInt("USER_ID", -1);

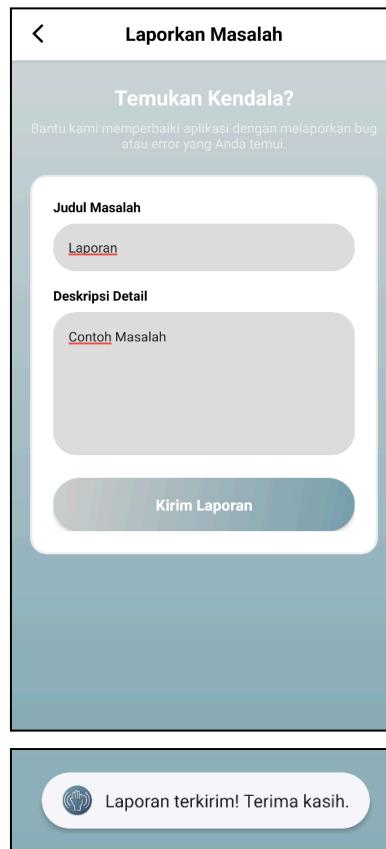
    if (userId == -1) {
        Toast.makeText(this, "Error: User not found",
            Toast.LENGTH_SHORT).show();
        return;
    }
    performSubmitLaporan(userId, judul, deskripsi);
}
```

b. LaporanMasalahActivity.java POST Request:

```
StringRequest stringRequest = new StringRequest(Request.Method.POST,
    ApiConfig.SUBMIT_LAPORAN_URL,
    response -> {
        try {
            JSONObject jsonObject = new JSONObject(response);
            String status = jsonObject.getString("status");
            String message = jsonObject.optString("message", "");

            if ("success".equals(status)) {
                Toast.makeText(this, "Laporan terkirim! Terima
                    kasih.", Toast.LENGTH_LONG).show();
                finish();
            } else {
                Toast.makeText(this, message,
                    Toast.LENGTH_SHORT).show();
            }
        }
    }

    @Override
    protected Map<String, String> getParams() {
        Map<String, String> params = new HashMap<>();
        params.put("user_id", String.valueOf(userId));
        params.put("judul", judul);
        params.put("deskripsi", deskripsi);
        return params;
    }
}
```



Gambar 26. Contoh Pengisian Laporan Masalah
dan *Toast* Ketika Berhasil Terkirim

Kode untuk *Insert* Laporan dalam *Database* (*submit_laporan.php*):

```
$sql = "INSERT INTO laporan_masalah (user_id, judul, deskripsi)
VALUES (:user_id, :judul, :deskripsi)";
$stmt = $pdo->prepare($sql);
$stmt->execute([
    ':user_id' => $user_id,
    ':judul' => $judul,
    ':deskripsi' => $deskripsi
]);

echo json_encode([
    "status" => "success",
    "message" => "Laporan berhasil dikirim!",
    "laporan_id" => $pdo->lastInsertId()
]);
```

id	user_id	judul	deskripsi	tanggal	status
2	11	Bug dalam Header Notifikasi	Header notifikasi berwarna hijau. Tidak selaras de...	2026-01-08 00:40:51	pending
3	8	Terjemahan Kurang Akurat	Hasil penerjemahan tidak sesuai dengan arti bahasa...	2026-01-08 00:42:44	pending
4	9	Aplikasi Terlalu Bagus	:)	2026-01-08 00:44:03	pending
5	14	Laporan	Contoh Masalah	2026-01-08 01:58:12	pending

Gambar 27. Laporan Baru Tersimpan dalam Tabel *laporan_masalah* (ID: 14)

C. Submit Saran dalam Kotak Saran

Fitur kotak saran memungkinkan *User* memberikan *feedback* atau masukan untuk pengembangan aplikasi. Berbeda dengan laporan masalah yang bersifat *bug report*, kotak saran lebih fokus pada ide perbaikan atau fitur baru yang diinginkan *User*. Proses dimulai dengan *User* mengetik saran di `EditText`, kemudian sistem mengambil `user_id` dari `SharedPreferences` untuk identifikasi pengirim. Data dikirim ke *server* via POST request dan disimpan di tabel `kotak_saran` dengan *timestamp* otomatis.

Kode Implementasi:

a. KotakSaranActivity.java memanggil user_id:

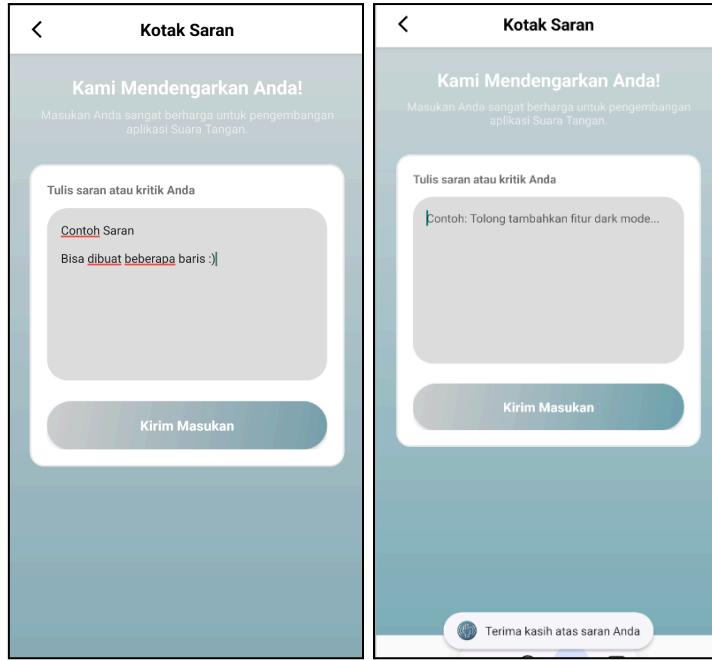
```
private void submitLaporan(String judul, String deskripsi) {
    SharedPreferences prefs = getSharedPreferences("GoScorePrefs",
        Context.MODE_PRIVATE);
    int userId = prefs.getInt("USER_ID", -1);

    if (userId == -1) {
        Toast.makeText(this, "Error: User not found",
            Toast.LENGTH_SHORT).show();
        return;
    }
    performSubmitLaporan(userId, judul, deskripsi);
}
```

b. KotakSaranActivity.java POST Request:

```
RequestQueue queue = Volley.newRequestQueue(this);
StringRequest stringRequest = new StringRequest(Request.Method.POST,
    ApiConfig.SUBMIT_SARAN_URL,
    response -> {
        try {
            JSONObject jsonObject = new JSONObject(response);
            String status = jsonObject.getString("status");
            String message = jsonObject.optString("message", "");

            if ("success".equals(status)) {
                Toast.makeText(this, "Terima kasih atas saran Anda",
                    Toast.LENGTH_SHORT).show();
                etSaran.setText("");
            } else {
                Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
            }
        } catch (JSONException e) {
            Toast.makeText(this, "Error submitting suggestion",
                Toast.LENGTH_SHORT).show();
        }
    },
    error -> Toast.makeText(this, "Network error: " + error.getMessage(),
        Toast.LENGTH_SHORT).show()
) {
    @Override
    protected Map<String, String> getParams() {
        Map<String, String> params = new HashMap<>();
        params.put("user_id", String.valueOf(userId));
        params.put("saran", saran);
        return params;
    }
};
```



Gambar 28-29. Contoh Pengisian Kotak Saran
dan *Toast* Ketika Berhasil Terkirim

Kode untuk *Insert* Saran dalam Database (*submit_saran.php*):

```
$sql = "INSERT INTO kotak_saran (user_id, saran) VALUES (:user_id,
:saran)";
$stmt = $pdo->prepare($sql);
$stmt->execute([
    ':user_id' => $user_id,
    ':saran' => $saran
]);

echo json_encode([
    "status" => "success",
    "message" => "Terima kasih! Masukan Anda telah kami terima.",
    "saran_id" => $pdo->lastInsertId()
]);
```

id	user_id	saran	tanggal
1	8	Bismillah PAB dapet A++ :D	2026-01-08 00:51:54
2	11	Puji Tuhan kelar	2026-01-08 00:55:39
3	9	Tetap semangat, yaa!	2026-01-08 00:58:44
Contoh Saran			
4	14	Bisa dibuat beberapa baris :)	2026-01-11 21:33:04

Gambar 30. Saran Baru Tersimpan dalam Tabel `kotak_saran` (ID: 14)

4.3 READ (Mengambil Data)

A. Verifikasi Kredensial dalam Fitur Login

Proses *login* memungkinkan *User* masuk dengan *username* ATAU email. Sistem mendeteksi otomatis jenis *input* berdasarkan keberadaan karakter @. Data *credential* dan *password* dikirim ke *server*, kemudian *server* melakukan *query* ke *database* untuk mencari *User* yang sesuai. Jika ditemukan, *password* di-*verify* menggunakan `password_verify()` terhadap *hash* yang tersimpan. Jika cocok, *server* mengirim data *User* lengkap yang kemudian disimpan di `SharedPreferences` untuk *session management*.

Kode Implementasi:

a. LoginFragment.java (Fleksibilitas Kredensial):

```
String credential = etCredential.getText().toString().trim();
String password = etPassword.getText().toString().trim();

boolean isEmail = credential.contains("@");
if (isEmail) {
    if (!Patterns.EMAIL_ADDRESS.matcher(credential).matches()) {
        tilCredential.setError("Format email tidak valid");
        return;
    }
}
```

b. LoginFragment.java (Menyimpan SharedPreferences):

```
if ("success".equals(status)) {
    SharedPreferences prefs = requireActivity().getSharedPreferences("GoScorePrefs",
    Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = prefs.edit();

    editor.putString("LOGGED_IN_CREDENTIAL", credential);

    Intent intent = new Intent(getActivity(), MainActivity.class);
}
```

c. login.php

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE email = :cred OR
username = :cred");
$stmt->execute([':cred' => $credential]);
$user = $stmt->fetch();

if ($user && password_verify($pass, $user['password'])) {
    unset($user['password']);
    echo json_encode([
        "status" => "success",
        "message" => "Login successful",
        "user" => $user
    ]);
} else {
    echo json_encode(["status" => "error", "message" => "Invalid
username or password"]);
}
```

B. Menarik Data Informasi Akun User

Proses *login* memungkinkan *User* masuk dengan *username* ATAU email. Sistem mendeteksi otomatis jenis *input* berdasarkan keberadaan karakter @. Data *credential* dan *password* dikirim ke *server*, kemudian *server* melakukan

query ke *database* untuk mencari *User* yang sesuai. Jika ditemukan, *password* di-*verify* menggunakan `password_verify()` terhadap *hash* yang tersimpan. Jika cocok, *server* mengirim data *User* lengkap yang kemudian disimpan di `SharedPreferences` untuk *session management*.

Kode Implementasi:

a. InfoAkunActivity.java:

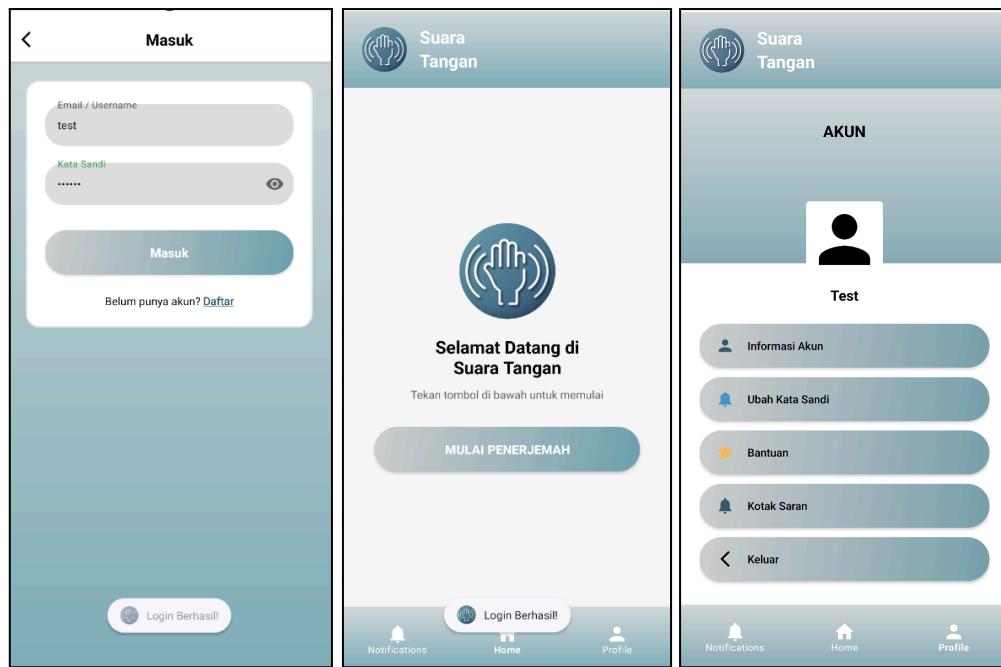
```
private void loadUserData() {
    RequestQueue queue = Volley.newRequestQueue(this);
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
ApiConfig.GET_USER_URL,
    response -> {
        try {
            JSONObject jsonObject = new JSONObject(response);
            String status = jsonObject.getString("status");

            if ("success".equals(status)) {
                JSONObject user = jsonObject.getJSONObject("user");
                tvNama.setText(namaLengkap);
                tvEmail.setText(email);
                tvTelepon.setText(telepon);
                tvPeran.setText(peran);
            } else {
                Toast.makeText(this, "Failed to load user data",
Toast.LENGTH_SHORT).show();
            }
        } {
            @Override
            protected Map<String, String> getParams() {
                Map<String, String> params = new HashMap<>();
                params.put("user_id", String.valueOf(userId));
                return params;
            }
        };
    });
}
```

b. get_user.php:

```
$stmt = $pdo->prepare("SELECT id, nama_lengkap, username, email,
telepon, peran FROM users WHERE id = :id");
$stmt->execute([':id' => $user_id]);
$user = $stmt->fetch();

if ($user) {
    echo json_encode([
        "status" => "success",
        "user" => $user
    ]);
} else {
    echo json_encode([
        "status" => "error",
        "message" => "User not found"
    ]);
}
```



Gambar 30-32. Login Berhasil hingga Menampilkan Informasi Akun yang Tepat

4.4 UPDATE (Memodifikasi Data)

A. Edit Informasi Akun

Fitur *Edit Profile* memungkinkan *User* mengubah nama lengkap, nomor telepon, dan peran mereka. Halaman ini memiliki dua mode: *display mode (read-only)* dan *edit mode (editable)*. Saat *User* menekan tombol “Edit”, *layout* berganti ke *edit mode* dengan `TextInputLayout` yang dapat dimodifikasi. Setelah *User* mengubah data dan menekan “Simpan”, sistem melakukan validasi, mengirim data ke *server* untuk *update database*, kemudian menyimpan perubahan ke `SharedPreferences` dan kembali ke *display mode*.

Kode Implementasi:

a. InfoAkunActivity.java (*Toggle Edit Mode*):

```
private void setEditMode(boolean enabled) {
    isEditMode = enabled;
    if (enabled) {
        layoutDisplayMode.setVisibility(View.GONE);
        layoutEditMode.setVisibility(View.VISIBLE);
        btnEdit.setText("Simpan");

        etNama.setText(tvNama.getText().toString());
        etTelepon.setText(tvTelepon.getText().toString());
        tvEmailEdit.setText(tvEmail.getText().toString());

        String peran = tvPeran.getText().toString().trim();
        if (peran.equals("Teman Dengar")) {
            ((RadioButton) findViewById(R.id.rb_teman_dengar)).setChecked(true);
        } else if (peran.equals("Teman Tuli")) {
            ((RadioButton) findViewById(R.id.rb_teman_tuli)).setChecked(true);
        }
    } else {
        layoutDisplayMode.setVisibility(View.VISIBLE);
        layoutEditMode.setVisibility(View.GONE);
        btnEdit.setText("Edit");
    }
}
```

```

        layoutDisplayMode.setVisibility(View.VISIBLE);
        layoutEditMode.setVisibility(View.GONE);
        btnEdit.setText("Edit");
    }
}

```

b. InfoAkunActivity.java (*Update Request*):

```

StringRequest stringRequest = new StringRequest(Request.Method.POST,
ApiConfig.UPDATE_PROFILE_URL,
    response -> {
        android.util.Log.d("InfoAkunActivity", "Server response: " +
response);

        try {
            JSONObject jsonObject = new JSONObject(response);
            String status = jsonObject.getString("status");
            String message = jsonObject.optString("message", "");

            if ("success".equals(status)) {
                Toast.makeText(this, "Profil berhasil diperbarui!",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

c. update_profile.php:

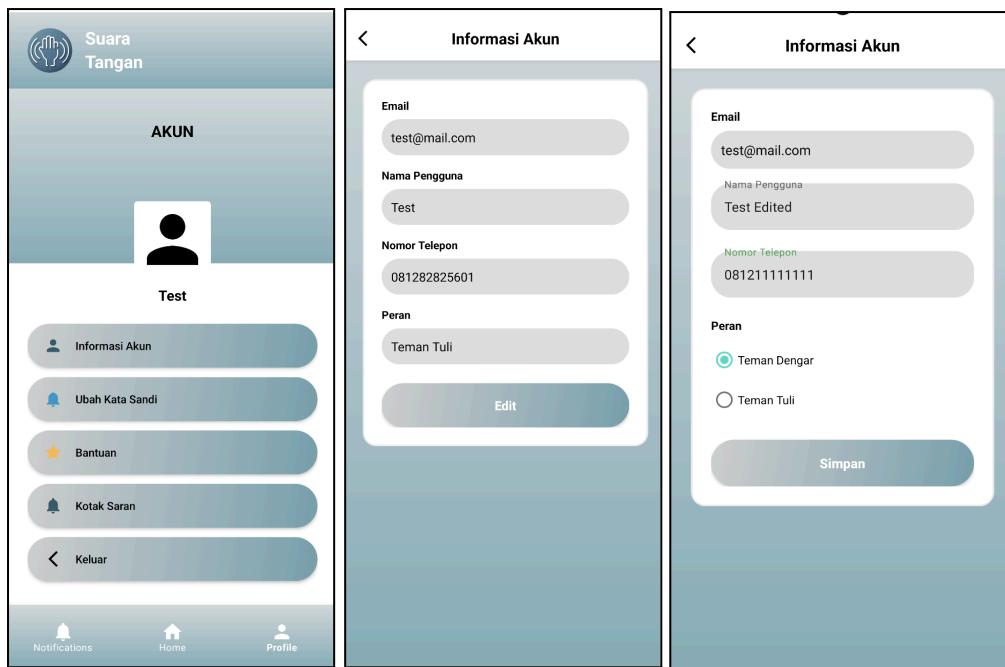
```

$updates = [];
$params = [':id' => $user_id];
if (!empty($nama)) {
    $updates[] = "nama_lengkap = :nama";
    $params[':nama'] = $nama;
}
if (!empty($telp)) {
    $updates[] = "telepon = :telp";
    $params[':telp'] = $telp;
}
if (!empty($peran)) {
    $updates[] = "peran = :peran";
    $params[':peran'] = $peran;
}
if (empty($updates)) {
    echo json_encode(["status" => "error", "message" => "No fields
to update"]);
    exit;
}

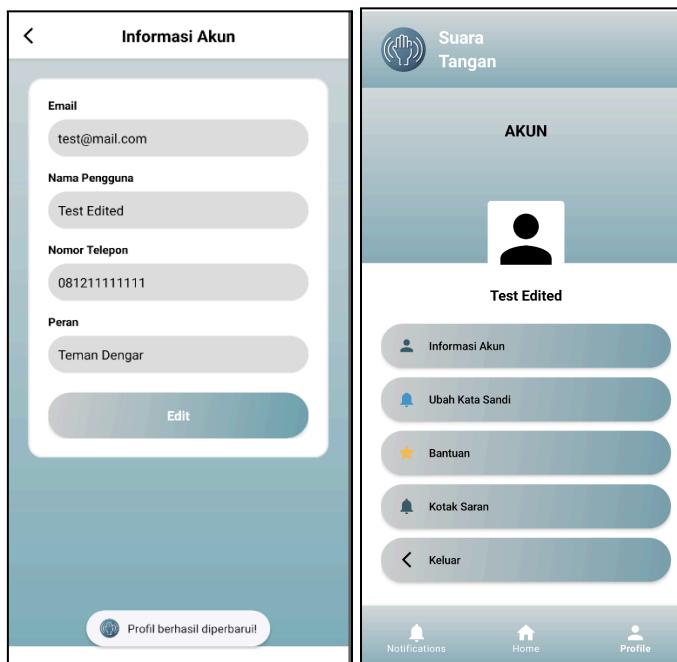
$sql = "UPDATE users SET " . implode(", ", $updates) . " WHERE id
= :id";
$stmt = $pdo->prepare($sql);
$stmt->execute($params);

if ($stmt->rowCount() > 0) {
    $stmt = $pdo->prepare("SELECT id, nama_lengkap, username,
email, telepon, peran FROM users WHERE id = :id");
    $stmt->execute([':id' => $user_id]);
    $user = $stmt->fetch();
}

```



Gambar 33-35. Tampilan Informasi Akun Sebelum dan Selagi Diubah



Gambar 36-37. Tampilan Informasi Akun Setelah Diubah

id	nama_lengkap	username	email	telepon	password	peran
8	Teresa Kaena Dharmanyoto	kaeee	teresa@mail.com	081282825601	\$2y\$10\$TW72YWWS5Eq1IRl1FxxBKYe6kdS/SssqW0NFSHUkHIE2...	Teman Tuli
9	Runi Dwi Jiasta	rungi	rungi@mail.com	082222222222	\$2y\$10\$xMtW4TGJievwGE.p5YpUZen/ZspuJQ.vhECKk4eLC0k...	Teman Dengar
10	Andrew Riza Rafael	andrew	andrew@mail.com	081111111111	\$2y\$10\$2gj6QhSaWnWUQs0TW/QMo.QuoVpnwyAv2yyuHnE2ULk...	Teman Tuli
11	Caroline Evarista Den Lau	evaa	evaa@mail.com	083333333333	\$2y\$10\$FVQ34ZNrmImP073qO83.DOKCMJgeomsfFwcrHj3z4Ml...	Teman Dengar
14	Test	test	test@mail.com	081282825601	\$2y\$10\$4GTcio6Tp8UyjvHBIMjb.vqbu3ad8vPSFYs2CWo8Lb...	Teman Tuli

Gambar 38. Isi Tabel **users** Sebelum Data Diubah

id	nama_lengkap	username	email	telepon	password	peran
8	Teresa Kaena Dharmanyoto	kaee	teresa@mail.com	081282825601	\$2y\$10\$TW72YWW5Eq1IRI1FxxBKYe6kds/SssqWONFSHUKhIE2...	Teman Tuli
9	Runi Dwi Jiasta	runi	runi@mail.com	082222222222	\$2y\$10\$xMlW4TGJievwGE.p5YpUZen/ZspuJQ.vhECKk4eLC0k...	Teman Dengar
10	Andrew Riza Rafael	andrew	andrew@mail.com	081111111111	\$2y\$10\$2gl6QhSaWnWUQsoTW/QMo.QuoVpnwyAv2yyuHnE2ULk...	Teman Tuli
11	Caroline Evarista Den Lau	evaa	evaa@mail.com	083333333333	\$2y\$10\$FVQ34ZNrmImP073qOB3.DOKCMJgeoms!FwcrHj3z4MI...	Teman Dengar
14	Test Edited	test	test@mail.com	081211111111	\$2y\$10\$4GTcio6Tpr8UyjvHBIMjb.vqbu3ad8vPSFYs2CWo8Lb...	Teman Dengar

Gambar 39. Isi Tabel **users** Sesudah Data Diubah

B. Perbarui Kata Sandi

Fitur ubah *password* memerlukan tiga *input*: *password* lama (untuk verifikasi), *password* baru, dan konfirmasi *password* baru. Sistem melakukan validasi ketat di *client-side* (minimal 6 karakter, harus ada huruf besar dan simbol) sebelum mengirim ke *server*. Di *server*, *password* lama diverifikasi terlebih dahulu dengan **password_verify()**. Jika cocok, *password* baru di-*hash* dan di-*update* dalam *database*. Proses ini memastikan keamanan tinggi karena *password* lama harus diketahui untuk melakukan perubahan.

Kode Implementasi:

a. UbahSandiActivity.java (Validasi):

```
private void validateAndChangePassword() {
    tilLama.setError(null);
    tilBaru.setError(null);
    tilKonfirmasi.setError(null);

    String sandiLama = etLama.getText().toString().trim();
    String sandiBaru = etBaru.getText().toString().trim();
    String konfirmasi = etKonfirmasi.getText().toString().trim();

    if (sandiLama.isEmpty()) {
        tilLama.setError("Kata sandi lama tidak boleh kosong");
        return;
    }
    if (sandiBaru.isEmpty()) {
        tilBaru.setError("Kata sandi baru tidak boleh kosong");
        return;
    }
    if (sandiBaru.length() < 6) {
        tilBaru.setError("Password minimal 6 karakter");
        return;
    }
    boolean hasCapital = sandiBaru.matches("[A-Z].*");
    boolean hasSymbol = sandiBaru.matches("[^a-zA-Z0-9].*");

    if (!hasCapital || !hasSymbol) {
        tilBaru.setError("Password harus memiliki minimal 1 huruf besar dan 1 simbol");
        return;
    }
    if (konfirmasi.isEmpty()) {
        tilKonfirmasi.setError("Konfirmasi password tidak boleh kosong");
        return;
    }
    if (!sandiBaru.equals(konfirmasi)) {
        tilKonfirmasi.setError("Password tidak sama");
        return;
    }
    if (sandiLama.equals(sandiBaru)) {
```

```

        tilBaru.setError("Kata sandi baru harus berbeda dengan kata sandi lama");
        return;
    }
    performPasswordUpdate(sandiLama, sandiBaru, konfirmasi);
}

```

b. UbahSandiActivity.java (*Update Request*):

```

private void performPasswordUpdate(String currentPassword, String newPassword, String confirmPassword) {
    RequestQueue queue = Volley.newRequestQueue(this);
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
ApiConfig.UPDATE_PASSWORD_URL,
response -> {
    try {
        JSONObject jsonObject = new JSONObject(response);
        String status = jsonObject.getString("status");
        String message = jsonObject.optString("message", "");

        if ("success".equals(status)) {
            Toast.makeText(this, "Kata sandi berhasil diperbarui!",
Toast.LENGTH_LONG).show();

            etLama.setText("");
            etBaru.setText("");
            etKonfirmasi.setText("");
            finish();
        } else {
            // Show error message
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```

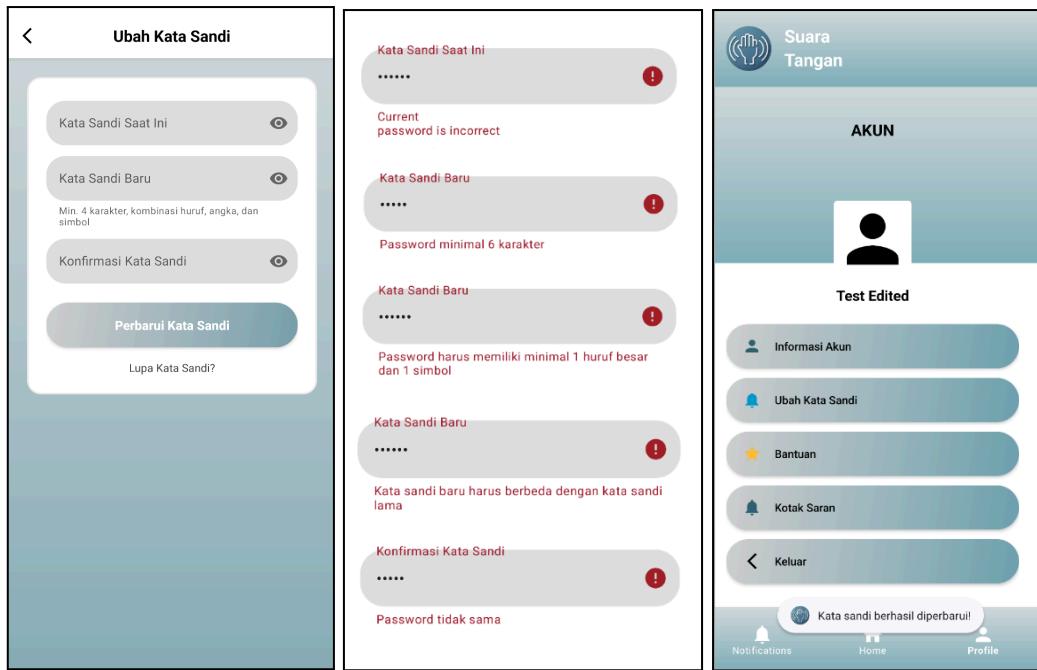
c. update_password.php:

```

$stmt = $pdo->prepare("SELECT password FROM users WHERE id =
:id");
$stmt->execute([':id' => $user_id]);
$user = $stmt->fetch();

if (!password_verify($current_password, $user['password'])) {
    echo json_encode(["status" => "error", "message" => "Current
password is incorrect"]);
    exit;
}
$hashed_pass = password_hash($new_password, PASSWORD_BCRYPT);
$stmt = $pdo->prepare("UPDATE users SET password = :pass WHERE id
= :id");
$stmt->execute([
    ':pass' => $hashed_pass,
    ':id' => $user_id
]);
echo json_encode([
    "status" => "success",
    "message" => "Password updated successfully!"
]);

```



Gambar 40-42. Tampilan Halaman Mengubah Sandi, Peringatan Jika *Password* Tidak Diisi dengan Benar, dan *Toast* serta *Redirect* Ketika Berhasil

15 Test test test@mail.com 081211111111 \$2y\$10\$kQkkSfDjFWAb2L6zvBtO10sONIP2hZT:jujrEi6F... Teman Dengar

Gambar 43. *Hash Password* Sebelum Diubah dalam Tabel `users`

15 Test test test@mail.com 081211111111 \$2y\$10\$gN1F7pyU.CRfmSlI74el7eF9YNym8A/lphAkJg9tUvo... Teman Dengar

Gambar 44. *Hash Password* Sesudah Diubah dalam Tabel `users`

4.5 ***DELETE*** (Menghapus Data)

A. Menghapus Akun Permanen

Fitur *Delete Account* menggunakan sistem *multi-step confirmation* untuk mencegah penghapusan tidak disengaja. Proses ini melibatkan tiga dialog berturut-turut: (1) Dialog pilihan “*Log Out*” atau “Hapus Akun Permanen”, (2) Dialog konfirmasi akhir dengan peringatan data akan hilang permanen, dan (3) Dialog *input password* untuk verifikasi identitas. Ketiga dialog ini dapat dilihat pada Gambar 13-15. Setelah *password* diverifikasi di *server*, akun dihapus dari *database* beserta semua data terkait (CASCADE DELETE), `SharedPreferences` dibersihkan, dan *User* diarahkan ke halaman registrasi.

a. `FragmentProfile.java` (*Multi-Step Dialog*):

```
private void showLogoutOptions() {
    if (getActivity() == null) return;

    AlertDialog.Builder builder = new AlertDialog.Builder(requireContext());
    View dialogView =
    LayoutInflater.from(requireContext()).inflate(R.layout.dialog_custom_logout, null);
    builder.setView(dialogView);
    AlertDialog dialog = builder.create();
```

```

        if (dialog.getWindow() != null) {
            dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
        }
        Button btnLogout = dialogView.findViewById(R.id.btn_dialog_logout);
        Button btnDelete = dialogView.findViewById(R.id.btn_dialog_delete);
        TextView tvCancel = dialogView.findViewById(R.id.tv_dialog_cancel);
        btnLogout.setOnClickListener(v -> {
            dialog.dismiss();
            performLogout(false);
        });
        btnDelete.setOnClickListener(v -> {
            dialog.dismiss();
            showFinalDeleteConfirmation();
        });
        tvCancel.setOnClickListener(v -> dialog.dismiss());
        dialog.show();
    }

    // DIALOG 2
    private void showFinalDeleteConfirmation() {
        // Menampilkan peringatan "Data akan hilang permanen" // Jika user confirm, lanjut
        ke showPasswordInputDialog()
    }
    // DIALOG 3
    private void showPasswordInputDialog(int userId, SharedPreferences prefs) {
        // User input password untuk verifikasi // Jika password benar, execute
        performDeleteAccount()
    }

```

b. FragmentProfile.java (*Delete Request*):

```

private void performDeleteAccount(int userId, String password,
SharedPreferences prefs) {
    if (getActivity() == null) return;

    com.android.volley.RequestQueue queue =
com.android.volley.toolbox.Volley.newRequestQueue(getApplicationContext());
    com.android.volley.toolbox.StringRequest stringRequest = new
com.android.volley.toolbox.StringRequest(
        com.android.volley.Request.Method.POST,
        ApiConfig.DELETE_ACCOUNT_URL,
        response -> {
            try {
                org.json.JSONObject jsonObject = new
org.json.JSONObject(response);
                String status = jsonObject.getString("status");
                String message = jsonObject.optString("message",
"");

                if ("success".equals(status)) {
                    // Clear SharedPreferences
                    SharedPreferences.Editor editor = prefs.edit();
                    editor.clear();
                    editor.apply();

                    Toast.makeText(getApplicationContext(), "Akun berhasil
dihapus", Toast.LENGTH_SHORT).show();

```

```

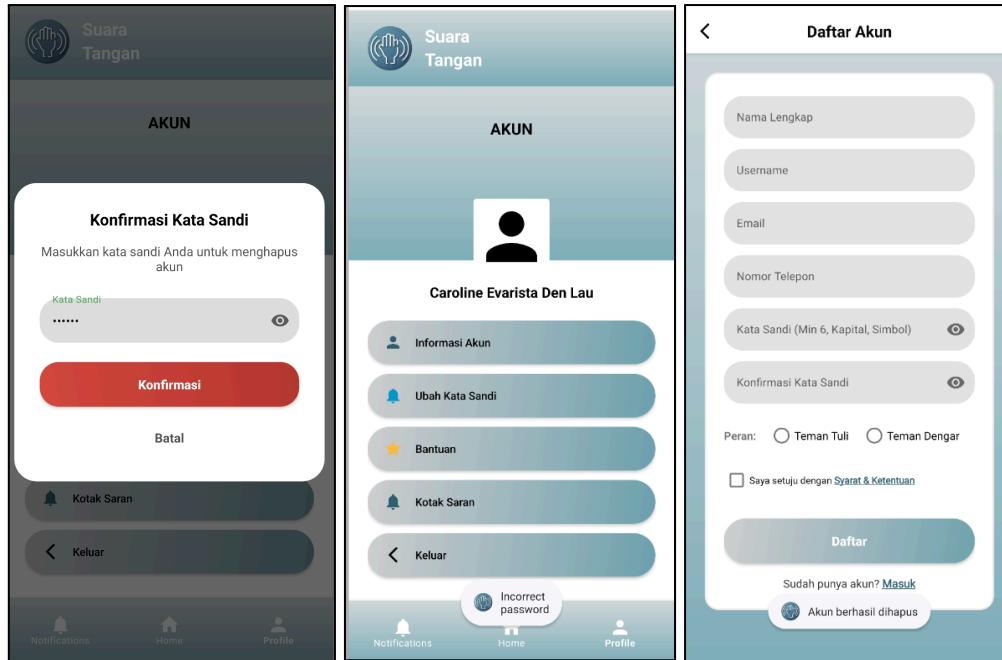
        // Go to sign up page
        Intent intent = new Intent(getActivity(),
AuthActivity.class);
        intent.putExtra("SHOW_SIGN_UP", true);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
        getActivity().finish();
    } else {
        Toast.makeText(getApplicationContext(), message,
Toast.LENGTH_SHORT).show();
    }
} catch (org.json.JSONException e) {
    Toast.makeText(getApplicationContext(), "Error deleting
account", Toast.LENGTH_SHORT).show();
}
},
error -> Toast.makeText(getApplicationContext(), "Network error: " +
error.getMessage(), Toast.LENGTH_SHORT).show()
)
{
@Override
protected java.util.Map<String, String> getParams() {
    java.util.Map<String, String> params = new
java.util.HashMap<>();
    params.put("user_id", String.valueOf(userID));
    params.put("password", password);
    return params;
}
};
}
;
```

c. delete_account.php:

```

$stmt = $pdo->prepare("SELECT password FROM users WHERE id =
:id");
$stmt->execute([':id' => $user_id]);
$user = $stmt->fetch();
if (!$user) {
    echo json_encode(["status" => "error", "message" => "User not
found"]);
    exit;
}
if (!password_verify($password, $user['password'])) {
    echo json_encode(["status" => "error", "message" => "Incorrect
password"]);
    exit;
}
$stmt = $pdo->prepare("DELETE FROM users WHERE id = :id");
$stmt->execute([':id' => $user_id]);

echo json_encode([
    "status" => "success",
    "message" => "Account deleted successfully"
]);
```



Gambar 45. Dialog Konfirmasi Penghapusan Akun dengan *Input Password*, *Toast* jika *Password* salah, serta *Redirect* Ketika Berhasil

id	nama_lengkap	username	email	telepon	password	peran
8	Teresa Kaena Dharmanyoto	kaee	teresa@mail.com	081282825601	\$2y\$10\$TW72YWW5Eq1IRl1FxxBKYe6kdS/SssqW0NFSHUkhlE2...	Teman Tuli
9	Runi Dwi Jiasta	rungi	rungi@mail.com	082222222222	\$2y\$10\$xMtW4TGJievwGE,p5YpUZen/ZspuJQ.vhECKk4eLC0k...	Teman Dengar
10	Andrew Riza Rafael	andrew	andrew@mail.com	081111111111	\$2y\$10\$2g6QhSaWnWUQs0TW/QMo.QuoVpnwyAv2yyuHnE2ULk...	Teman Tuli
15	Test	test	test@mail.com	081211111111	\$2y\$10\$gN1F7pyU.CRfmsIi74el7eF9YNym8A/lphAkJg9tUvo...	Teman Dengar

Gambar 46. Tabel `users` Sebelum Menghapus Akun (ID:15)

id	nama_lengkap	username	email	telepon	password	peran
8	Teresa Kaena Dharmanyoto	kaee	teresa@mail.com	081282825601	\$2y\$10\$TW72YWW5Eq1IRl1FxxBKYe6kdS/SssqW0NFSHUkhlE2...	Teman Tuli
9	Runi Dwi Jiasta	rungi	rungi@mail.com	082222222222	\$2y\$10\$xMtW4TGJievwGE,p5YpUZen/ZspuJQ.vhECKk4eLC0k...	Teman Dengar
10	Andrew Riza Rafael	andrew	andrew@mail.com	081111111111	\$2y\$10\$2g6QhSaWnWUQs0TW/QMo.QuoVpnwyAv2yyuHnE2ULk...	Teman Tuli

Gambar 47. Tabel `users` Sesudah Menghapus Akun (ID:15 terhapus)

4.6 Keamanan Data & Penanganan Error

Aplikasi mengimplementasikan beberapa layer keamanan:

1. *Password Hashing*: Menggunakan `PASSWORD_BCRYPT` di PHP untuk *hash password* sebelum disimpan
2. *Prepared Statements*: Mencegah SQL Injection dengan PDO *prepared statements*
3. *Input Validation*: Validasi di *client-side* (Android) dan *server-side* (PHP)
4. *Session Management*: `user_id` disimpan di `SharedPreferences` dengan mode private

Setiap network request dilengkapi dengan error listener untuk handle:

1. `TimeoutError`: Koneksi *timeout*
2. `NoConnectionError`: Tidak ada *internet*
3. `ServerError`: HTTP status 500
4. `AuthFailureError`: HTTP status 401 (*unauthorized*)

V. KELEBIHAN DAN KEKURANGAN APLIKASI

5.1 Kelebihan

Secara keseluruhan, aplikasi ini menunjukkan penerapan arsitektur Android yang efisien dan berfokus pada keamanan serta kenyamanan pengguna.

- **Efisiensi Kode (*Code Reusability*):** Aplikasi menerapkan prinsip penggunaan ulang kode yang sangat baik pada fitur Bantuan. Satu *Activity* (`DetailBantuanActivity`) digunakan secara dinamis untuk menampilkan berbagai jenis konten (FAQ, Privasi, Panduan) yang dikirim melalui *Intent*, sehingga menghemat penulisan kode berulang.
- **Keamanan Tindakan Destruktif:** Fitur penghapusan akun dilindungi dengan mekanisme keamanan berlapis. Pengguna diwajibkan memverifikasi identitas dengan memasukkan kata sandi sebelum akun dapat dihapus, mencegah ketidaksengajaan atau akses tidak sah.
- **Desain Antarmuka Cerdas (*Smart UI*):** Pada fitur Profil, penggunaan satu *Activity* untuk dua mode tampilan (Lihat & Edit) dengan memanipulasi visibilitas (`View.VISIBLE/GONE`) memberikan pengalaman transisi yang sangat mulus dan responsif bagi pengguna.
- **Otomatisasi Pengisian Data:** Fitur pelaporan masalah dirancang untuk meminimalkan beban *input* pengguna. Sistem secara otomatis mengambil `USER_ID` dari sesi lokal, memastikan laporan terhubung ke akun yang benar tanpa perlu *input* manual.
- **Validasi Input Sisi Klien:** Sebelum mengirim data ke *server*, aplikasi melakukan pengecekan format yang ketat (seperti kombinasi *password* dan format nomor telepon). Ini memberikan umpan balik instan kepada pengguna dan mengurangi beban pemrosesan *server*.

5.2 Kekurangan

Terdapat beberapa aspek teknis dan fungsional yang masih dapat ditingkatkan untuk pengembangan selanjutnya.

- **Konten Bersifat Statis (*Hardcoded*):** Informasi pada menu Bantuan (FAQ, Panduan) ditulis langsung di dalam kode Java. Hal ini membuat pembaruan informasi menjadi tidak fleksibel karena memerlukan pembaruan aplikasi secara keseluruhan, bukan cukup melalui perubahan *database*.
- **Fitur Belum Sepenuhnya Lengkap:** Pada fitur keamanan, tombol “Lupa Kata Sandi” saat ini masih berupa *placeholder* (hanya menampilkan pesan “akan segera hadir”) dan belum memiliki fungsi pemulihan akun yang berjalan.
- **Tidak Adanya Fitur Pencarian:** Menu Bantuan menggunakan daftar statis tanpa fitur pencarian (*search bar*). Hal ini dapat menyulitkan pengguna untuk menemukan topik spesifik dengan cepat jika jumlah konten bantuan bertambah banyak di kemudian hari.
- **Manajemen Sesi Terbatas:** Proses *logout* saat ini hanya membersihkan data sesi di penyimpanan lokal (`SharedPreferences`). Aplikasi belum

mengirimkan permintaan ke *server* untuk membatalkan token sesi, yang merupakan praktik keamanan yang lebih ideal.

VI. PEMBAGIAN TUGAS

Nama	Tugas
Runi Dwi Jiasta (202304560001)	<p>Penyempurnaan Otentikasi (<i>Login & Register</i>):</p> <ul style="list-style-type: none"> Menyempurnakan antarmuka pengguna (<i>User Interface</i>) pada LoginFragment dan RegisterFragment agar tata letak terlihat rapi, responsif, dan konsisten dengan tema aplikasi. Melengkapi logika validasi input sisi klien (<i>client-side validation</i>) pada RegisterFragment secara menyeluruh, yang meliputi: <ul style="list-style-type: none"> Pengecekan format nama lengkap (harus diawali huruf kapital). Validasi keamanan kata sandi (wajib kombinasi huruf, angka, dan simbol). Validasi format email dan panjang digit nomor telepon. <p>Fitur Pendukung Otentikasi:</p> <ul style="list-style-type: none"> Membuat dan mengimplementasikan TermsActivity (Syarat & Ketentuan) dari awal. Menghubungkan halaman Syarat & Ketentuan ke dalam form registrasi menggunakan teks interaktif (<i>ClickableSpan</i>). <p>Modul Profil Pengguna:</p> <ul style="list-style-type: none"> Membangun FragmentProfile sebagai pusat kontrol akun dengan navigasi menu bergaya <i>dashboard</i>. Merancang antarmuka cerdas (<i>Smart UI</i>) pada InfoAkunActivity, yang memungkinkan satu halaman menangani dua mode tampilan (<i>Display</i> dan <i>Edit</i>) tanpa berpindah <i>Activity</i>. Mengimplementasikan fitur keamanan akun pada UbahSandiActivity dengan memastikan sandi baru berbeda dari sandi lama. Membangun mekanisme Logout dan Hapus Akun yang aman dengan menerapkan: <ul style="list-style-type: none"> <i>Custom Alert Dialog</i> untuk opsi tindakan keluar. Dialog verifikasi kata sandi sebagai langkah keamanan terakhir sebelum penghapusan akun permanen. <p>Modul Pusat Bantuan:</p> <ul style="list-style-type: none"> Membangun BantuanActivity sebagai hub pusat informasi dengan navigasi bergaya <i>Dashboard Menu</i>, menggunakan AppCompatButton berikon untuk aksesibilitas yang jelas. Mengimplementasikan strategi <i>Code Reusability</i> (penggunaan ulang kode) yang efisien dengan membuat DetailBantuanActivity. Activity tunggal ini mampu merender berbagai jenis konten (FAQ, Kebijakan Privasi, dan Buku Panduan) secara dinamis dengan menerima data HTML melalui Intent, sehingga menghemat penggunaan memori dan file.

	<ul style="list-style-type: none"> Menyusun konten edukasi pengguna (FAQ & Panduan) dalam format HTML <i>hardcoded</i> di dalam BantuanActivity untuk memastikan tampilan teks yang rapi dan terstruktur. Membuat antarmuka untuk fitur umpan balik pengguna melalui LaporanMasalahActivity dan KotakSaranActivity. <p>Dokumentasi Proyek:</p> <ul style="list-style-type: none"> Menyusun dan merapikan seluruh laporan proyek secara sistematis, mulai dari Bagian I - VI, memastikan format dan bahasa sesuai dengan standar penulisan yang berlaku.
Teresa Kaena Dharmanyoto (202304560014)	<p>Arsitektur & Setup Awal Aplikasi:</p> <ul style="list-style-type: none"> Membangun struktur navigasi awal menggunakan BottomNavigationView dan Fragment pada MainActivity. Merancang SplashActivity dengan <i>auto-redirect</i> dan membuat komponen <i>header</i> universal yang konsisten. <p>Modul Pusat Bantuan:</p> <ul style="list-style-type: none"> Membuat antarmuka fitur umpan balik melalui LaporanMasalahActivity dan KotakSaranActivity. <p>Penyempurnaan UI Notifikasi:</p> <ul style="list-style-type: none"> Memperbaiki tampilan NotificationDetailActivity dengan penambahan <i>header</i> yang sesuai, <i>formatting</i> HTML, dan CardView yang modern. <p>Penyempurnaan Home Fragment:</p> <ul style="list-style-type: none"> Memperbaiki antarmuka penerjemah dengan <i>overlay screen</i> dan tombol <i>gradient</i> yang konsisten. Memperbaiki UI mode-mode penerjemahan dasar agar sesuai dengan desain keseluruhan aplikasi. <p>Database Setup & Integration (CRUD):</p> <ul style="list-style-type: none"> Membangun infrastruktur <i>backend</i> (XAMPP, MySQL, PHP) dengan keamanan <i>Password Hashing</i> (BCRYPT) dan <i>Prepared Statements</i>. Membuat file PHP untuk semua operasi <i>database</i>: register.php, login.php, get_user.php, update_user.php, change_password.php, submit_feedback.php. Mengintegrasikan operasi CRUD (Create, Read, Update, Delete) dari Android ke <i>server</i> menggunakan <i>library</i> Volley. Membuat ApiConfig terpusat dan sistem penanganan error (<i>Error Handling</i>) yang komprehensif. Mengimplementasikan fitur Edit Profile (InformasiAkunActivity) yang dinamis dengan format <i>layout</i> dan tombol yang terintegrasi dengan <i>database</i>. Mengimplementasikan fitur Change Password yang dinamis dan terintegrasi dengan <i>database</i>. Mengimplementasikan fitur Kotak Saran yang dinamis dan terintegrasi dengan <i>database</i>. <p>Dokumentasi Proyek:</p> <ul style="list-style-type: none"> Menyusun laporan proyek secara sistematis (Bagian I - VI) lengkap dengan <i>code snippets</i>, bukti screenshot, dan <i>formatting</i> yang sesuai.

	<p>Kontribusi Tambahan:</p> <ul style="list-style-type: none"> Implementasi manajemen <i>Intent & Navigation</i> yang efisien serta penerapan prinsip <i>Clean Code</i> dan <i>Material Design</i> di seluruh aplikasi.
Caroline Evarista Den Lau (202304560027)	<p>Perancangan & Optimalisasi Modul Notifikasi:</p> <ul style="list-style-type: none"> Merancang & membangun struktur antarmuka awal untuk daftar notifikasi. Merapikan detail visual pada laman notifikasi, termasuk tata letak (<i>layout</i>), skema warna dan pemilihan font yang disesuaikan dengan lama lainnya. Menyusun dan mengintegrasikan konten data untuk setiap entri notifikasi agar informasi tersampaikan dengan jelas. Melakukan kurasi dan penataan ulang isi lama notifikasi untuk memastikan tampilan tersusun rapi dan terstruktur. Menambahkan logika interaksi pada laman notifikasi, mencangkup fungsi membuka, menutup detail notifikasi serta optimasi fitur <i>scrolling</i> keatas maupun kebawah sehingga tampilan lebih responsif. <p>Komponen Android (Teknis)</p> <ul style="list-style-type: none"> Mengimplementasikan Adapter & <i>RecyclerView</i> untuk tampilan daftar notifikasi yang lebih dinamis dan efisien. Menambahkan asset ikon gambar/simbol pada setiap item notifikasi untuk memperkuat konteks informasi. <p>Integrasi & Fungsi (Logic)</p> <ul style="list-style-type: none"> Mengintegrasikan <i>Intent</i> untuk memberikan pengarahan pengguna dari laman utama notifikasi menuju halaman detail notifikasi secara <i>seamless</i>. <p>Dokumentasi Proyek</p> <ul style="list-style-type: none"> Menyusun laporan proyek secara sistematis untuk bagian Notifikasi lengkap dengan bukti ss tampilan, code <i>snippets</i> dan menggunakan formatting yang sesuai.
Andrew Riza Rafael (202304560030)	<p>Konfigurasi Sistem & Perizinan (System Configuration)</p> <ul style="list-style-type: none"> Menambahkan library CameraX (<code>camera-core</code>, <code>camera-camera2</code>, <code>camera-lifecycle</code>, <code>camera-view</code>) untuk menangani fitur kamera modern. Menyiapkan <i>dependencies TensorFlow Lite</i> (persiapan untuk integrasi model AI di pengembangan berikutnya). Menambahkan izin wajib: <code>CAMERA</code>, <code>RECORD_AUDIO</code>, dan <code>INTERNET</code>. Menambahkan blok <code><queries></code> untuk <code>android.speech.RecognitionService</code>. Ini adalah kunci agar fitur <i>Speech-to-Text</i> bisa berjalan di Android 11+ tanpa diblokir oleh sistem keamanan HP <p>Desain Antarmuka Pengguna (User Interface / XML)</p> <ul style="list-style-type: none"> Menempatkan <code>PreviewView</code> (untuk kamera) dan <code>VideoView</code> (untuk pemutar video) di lapisan paling belakang agar memenuhi layar. Membuat <code>RelativeLayout</code> khusus berisi logo, teks

- sambutan, dan tombol "**MULAI PENERJEMAH**".
- Membuat alur Sesi Aktif (*Active Session Controls*) yang berisi papan hasil terjemahan (*CardView*), teks instruksi, tombol utama besar, dan 3 tombol navigasi mode (Kamera, Mic, Video).
- Logika Pemrograman (*Java Logic - HomeFragment*)**
- Manajemen Sesi (*Session Management*)
 - **Fungsi `startSession()`:** Dipanggil saat tombol "Mulai" ditekan. Berfungsi menyembunyikan layar sambutan, menampilkan kontrol fitur, dan baru kemudian menyalakan kamera/mic.
 - **Fungsi `stopSession()`:** Dipanggil saat tombol "Logo Home" ditekan. Berfungsi mematikan total kamera (`unbindAll`), mematikan mic, menyetop video, dan mereset tampilan kembali ke layar sambutan.
 - Logika Ganti Mode (*Mode Switching*)
 - Saat masuk **Mode Video/Mic**: Kamera fisik dimatikan total (bukan cuma disembunyikan) untuk menghemat baterai dan memori.
 - Saat masuk **Mode Kamera**: Kamera dinyalakan kembali.
 - Implementasi Fitur Kamera (*Gesture Detection*)
 - Menggunakan **CameraX ImageAnalysis**.
 - Karena saat ini aplikasi Suara Tangan belum menerapkan model AI, untuk itu saat ini masih menggunakan **Logika Dummy (Simulasi)** dimana sistem mendeteksi waktu (`System.currentTimeMillis`), dan setiap 1,5 detik menampilkan kata acak dari daftar kosakata ("Halo", "Nama", "Saya", dll) seolah-olah sedang mendeteksi gerakan tangan.
 - Implementasi Fitur Suara (*Speech-to-Text*)
 - Menggunakan **Google Voice Popup** (`startActivityForResult`).
 - Implementasi Fitur Video
 - Menggunakan **ActivityResultLauncher** untuk membuka galeri HP.
 - Menambahkan simulasi teks "Menganalisis video..." yang berubah menjadi hasil terjemahan setelah beberapa detik.