GUIDEBOOK

HydraTiva

Dosen Pembimbing

Aditya Wicaksono, S.Kom., M.Kom.
Gema Parasti Mindara, S.Si., M.Kom.
Endang Purnama Giri, S.Kom

Ketua Tim

Mochamad Tegar Santoso	J0403231113

Anggota

Muhammad Thariq Aziz	J0403231087
Riyadh Azhara	J0403231053
Dhiyaurrahman Hamizan Haikal Putra	J0403231043
Miftakhul Ushbah Nastaftian	J0403231093
Shofwan Imtiyaz	J0403231038

Sekolah Vokasi
IPB University
2024

Daftar Isi

03 **Apa itu Hydrativa?**
Desripsi dan tujuan proyek kami

04 **Tampilan Website**
Tampilan Front-End Website Hydrativa

09 **Source Code Website**
Source code fungsi-fungsi pada Website Hydrativa

14 **Tampilan Aplikasi**
Aplikasi Hydrativa

18 **Source Code Aplikasi**
Back-End aplikasi Hydrativa

37 **Alat Hydrativa**
Alat IoT penyiraman Hydrativa

Apa itu Hydrativa?

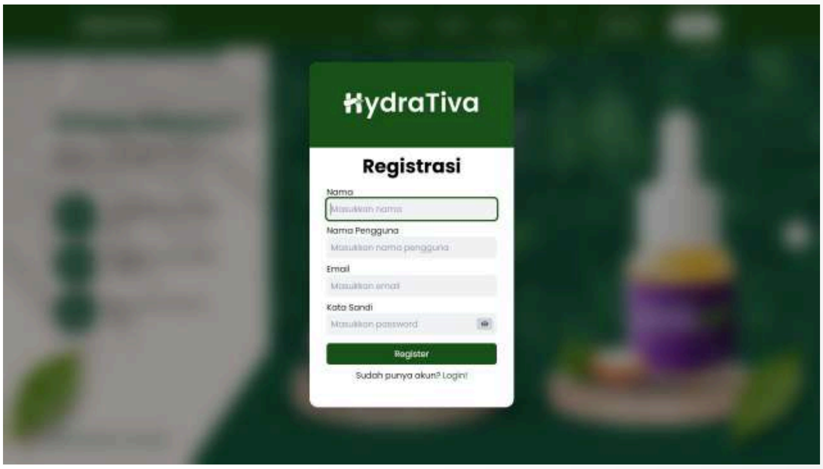
HydraTiva adalah sistem yang dirancang untuk meningkatkan produktivitas dan efisiensi di sektor perkebunan tanaman Stevia. Dengan menggabungkan teknologi IoT (Internet of Things) dan analisis data yang canggih, HydraTiva memungkinkan petani untuk memantau kondisi lahan secara real-time, mengatur penyiraman secara otomatis, dan membantu penyaluran hasil perkebunan stevia. Aplikasi ini memberikan informasi kadar tanah kebun stevia, histori penyiraman lahan stevia, dan rekomendasi tindakan berdasarkan data yang terkumpul dari sensor yang tersebar di seluruh lahan stevia.

Tujuan

Tujuan pembuatan Guidebook ini adalah guna memenuhi tugas Dokumentasi Bebas UAS Mata Kuliah Pemrograman Web, Pemrograman Mobile, dan Komputasi Awan Program Studi Teknologi Rekayasa Perangkat Lunak. Guidebook ini diharapkan dapat menjadi panduan dalam pembuatan proyek yang serupa.

Tampilan Website

1. Register



HydraTiva

Registrasi

Nama
Masukkan nama

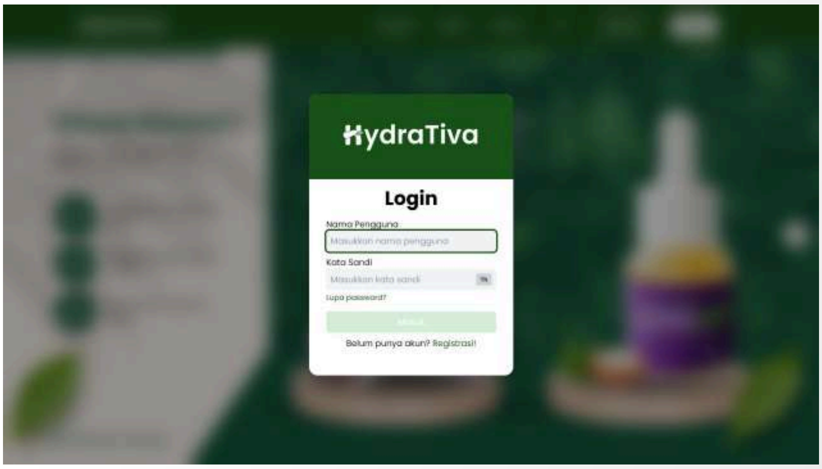
Nama Pengguna
Masukkan nama pengguna

Email
Masukkan email

Kata Sandi
Masukkan password

Sudah punya akun? Login!

2. Login



HydraTiva

Login

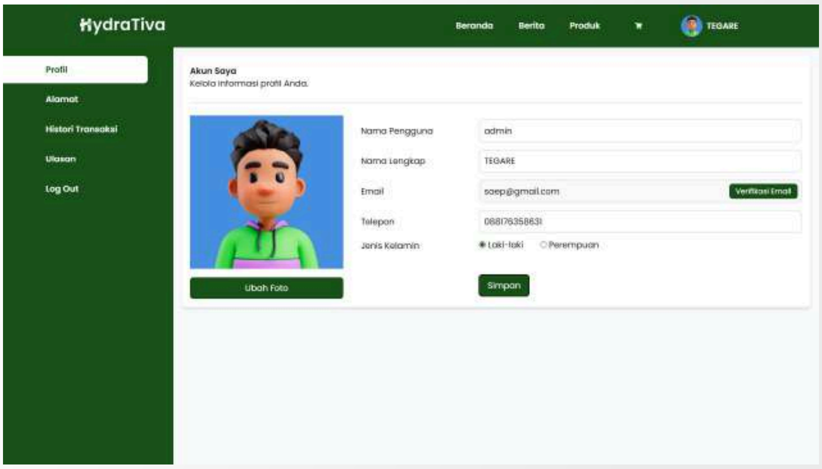
Nama Pengguna
Masukkan nama pengguna

Kata Sandi
Masukkan kata sandi

Lupa password?

Belum punya akun? Registrasi!

3. Profil



Akun Saya
Kelola informasi profil Anda.

Nama Pengguna: admin

Nama Lengkap: TEGARE

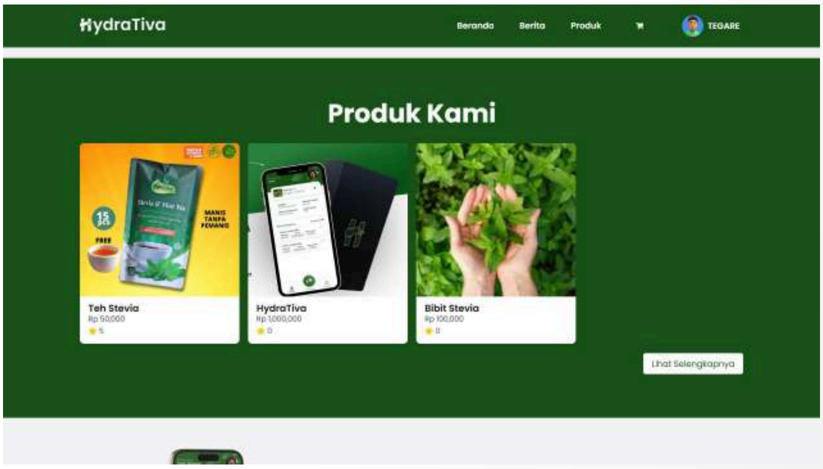
Email: saep@gmail.com

Telepon: 088176358631

Jenis Kelamin: ● Laki-laki ○ Perempuan

Tampilan Website

4. Produk



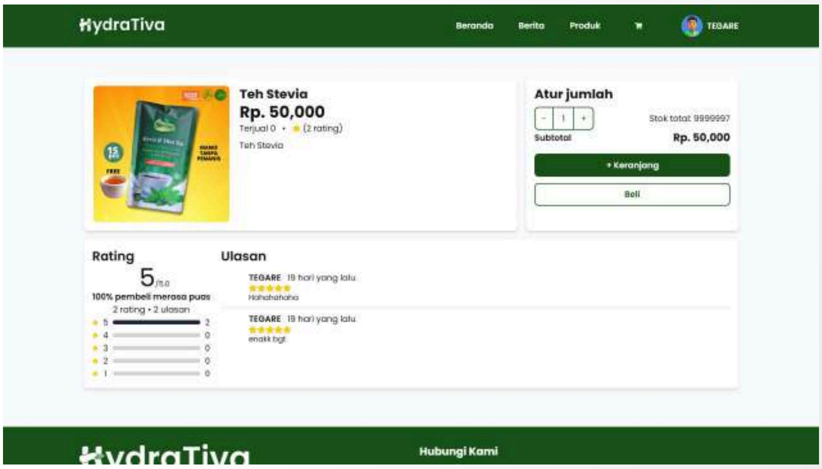
Produk Kami

Teh Stevia
Rp 50,000
★ 5

HydraTiva
Rp 1,000,000
★ 0

Bibit Stevia
Rp 100,000
★ 0

5. Detail Produk



Teh Stevia
Rp. 50,000
Terjual 0 • ★ (2 rating)
Teh Stevia

Atur jumlah
Stok total: 9999997
Subtotal Rp. 50,000

Rating
5/5.0
100% pembeli merasa puas
2 rating • 2 ulasan

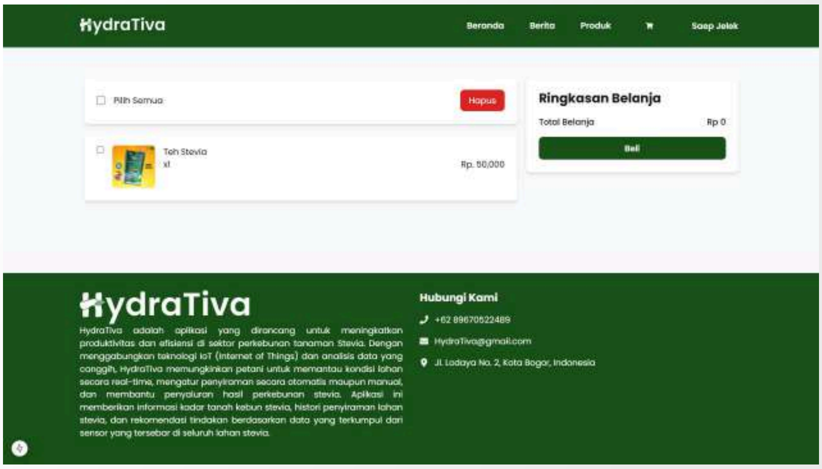
★ 5	2
★ 4	0
★ 3	0
★ 2	0
★ 1	0

Ulasan

TEGARE 19 hari yang lalu
★★★★★
Hahahahaha

TEGARE 19 hari yang lalu
★★★★★
enakk bgt

6. Keranjang



☐ Pilih Semua

☐ Teh Stevia
x1
Rp. 50,000

Ringkasan Belanja
Total Belanja Rp 0

HydraTiva

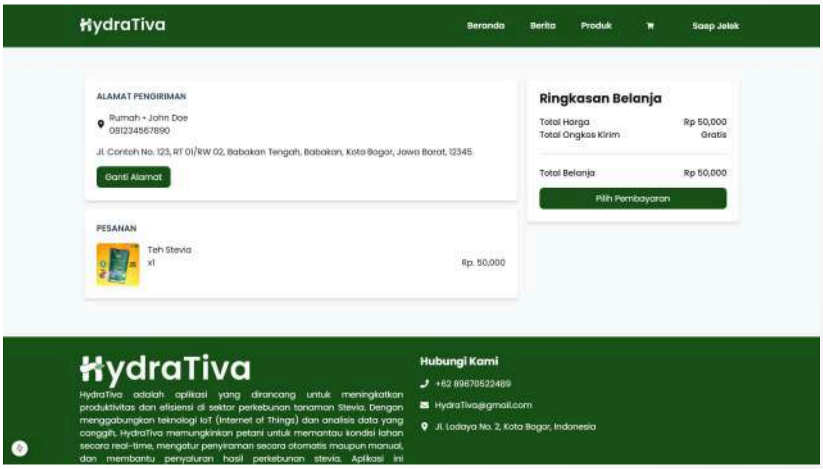
HydraTiva adalah aplikasi yang dirancang untuk meningkatkan produktivitas dan efisiensi di sektor perkebunan tanaman Stevia. Dengan menggabungkan teknologi IoT (Internet of Things) dan analisis data yang canggih, HydraTiva memungkinkan petani untuk memantau kondisi lahan secara real-time, mengatur penyiraman secara otomatis maupun manual, dan membantu penyaluran hasil perkebunan stevia. Aplikasi ini memberikan informasi kadar tanah kebun stevia, histori penyiraman lahan stevia, dan rekomendasi tindakan berdasarkan data yang terkumpul dari sensor yang tersebar di seluruh lahan stevia.

Hubungi Kami

☎ +62 89670522489
HydraTiva@gmail.com
Jl. Ladaya No. 2, Kota Bogor, Indonesia

Tampilan Website

7. Checkout



HydraTiva

Beranda Berita Produk Saep Jelek

ALAMAT PENGIRIMAN

Rumah • John Doe
081234567890

Jl. Contoh No. 123, RT 01/RW 02, Babakan Tengah, Babakan, Kota Bogor, Jawa Barat, 12345

Ganti Alamat

PESANAN

Teh Stevia
x1
Rp. 50,000

Ringkasan Belanja

Total Harga	Rp 50,000
Total Ongkos Kirim	Gratis
Total Belanja	Rp 50,000

Pilih Pembayaran

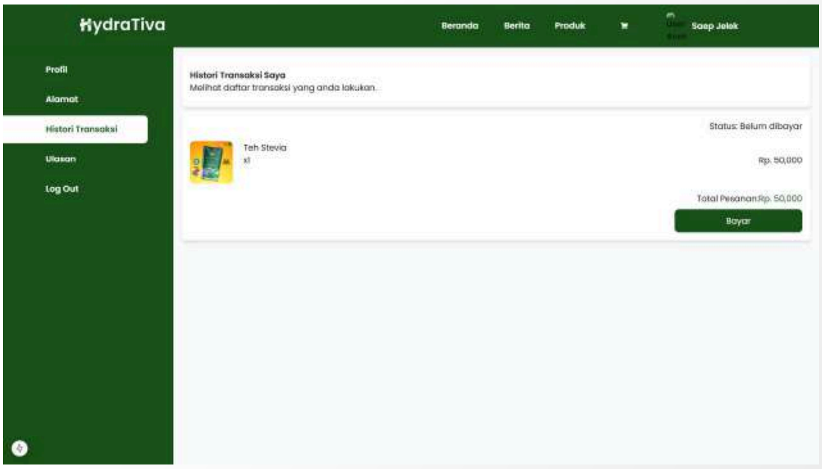
HydraTiva

HydraTiva adalah aplikasi yang dirancang untuk meningkatkan produktivitas dan efisiensi di sektor perkebunan tanaman Stevia. Dengan menggabungkan teknologi IoT (Internet of Things) dan analisis data yang canggih, HydraTiva memungkinkan petani untuk memantau kondisi lahan secara real-time, mengatur penyiraman secara otomatis maupun manual, dan membantu penyaluran hasil perkebunan stevia. Aplikasi ini

Hubungi Kami

+62 89670522489
HydraTiva@gmail.com
Jl. Lodaya No. 2, Kota Bogor, Indonesia

8. Histori Transaksi



HydraTiva

Beranda Berita Produk Saep Jelek

Profil
Alamat
Histori Transaksi
Ulasan
Log Out

Histori Transaksi Saya
Melihat daftar transaksi yang anda lakukan.

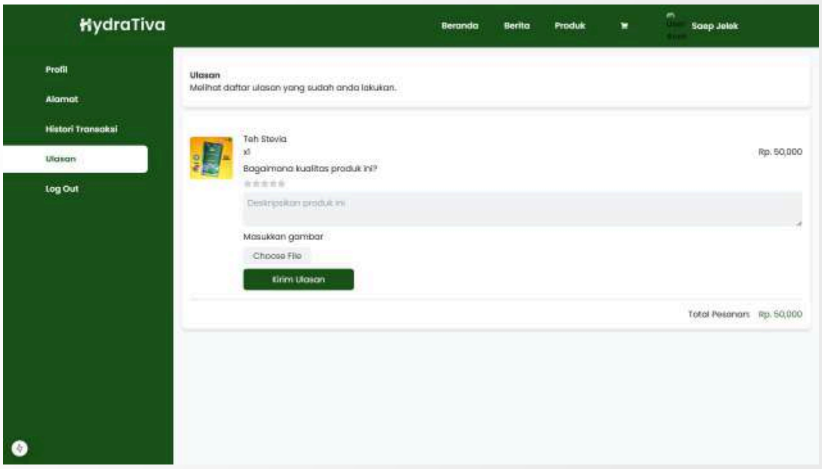
Status: Belum dibayar

Teh Stevia
x1
Rp. 50,000

Total PesananRp. 50,000

Bayar

9. Ulasan



HydraTiva

Beranda Berita Produk Saep Jelek

Profil
Alamat
Histori Transaksi
Ulasan
Log Out

Ulasan
Melihat daftar ulasan yang sudah anda lakukan.

Teh Stevia
x1
Rp. 50,000

Bagaimana kualitas produk ini?

☆☆☆☆☆

Deskripsikan produk ini

Masukkan gambar

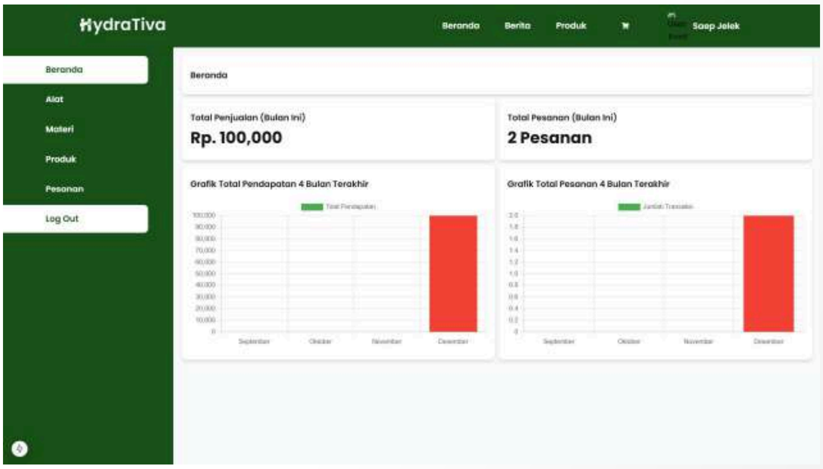
Choose File

Kirim Ulasan

Total Pesanan: Rp. 50,000

Tampilan Website

10. Beranda Admin



Beranda

Total Penjualan (Bulan Ini)

Rp. 100,000

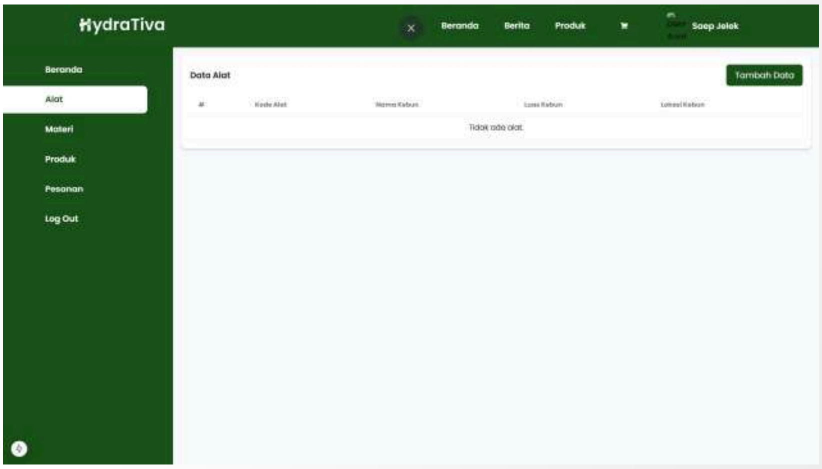
Total Pesanan (Bulan Ini)

2 Pesanan

Grafik Total Pendapatan 4 Bulan Terakhir

Grafik Total Pesanan 4 Bulan Terakhir

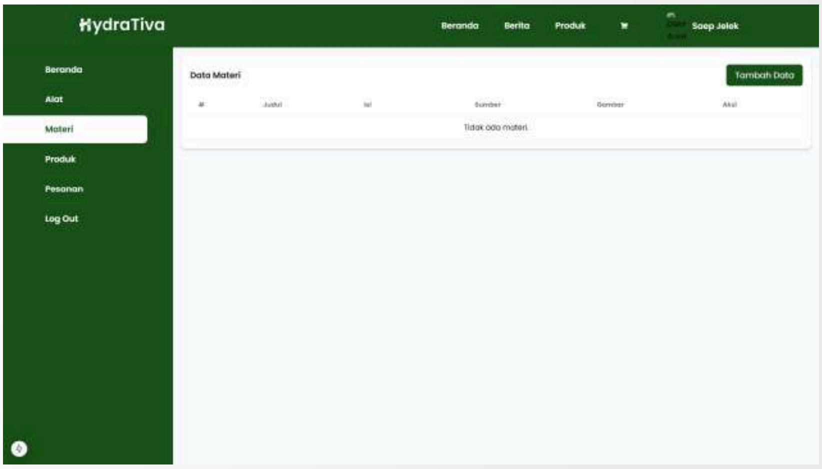
8. Alat



Data Alat

#	Kode Alat	Nama Kebun	Luas Kebun	Lokasi Kebun
Tidak ada alat.				

9. Materi

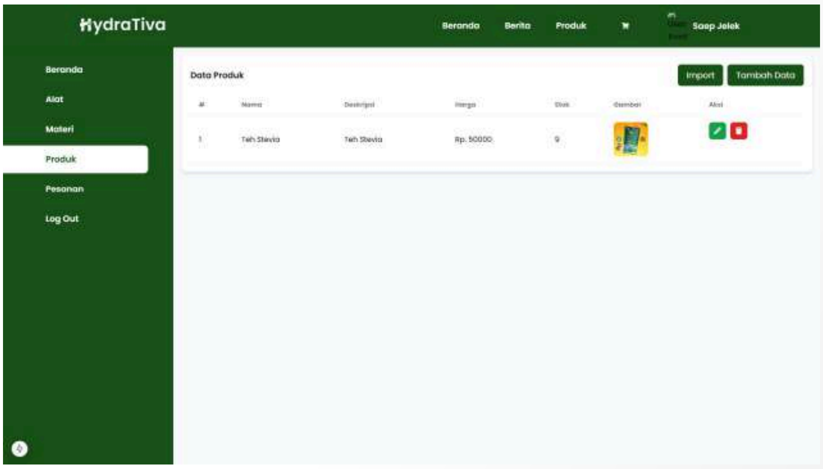


Data Materi

#	Judul	Isi	Sumber	Gambar	Aksi
Tidak ada materi.					

Tampilan Website

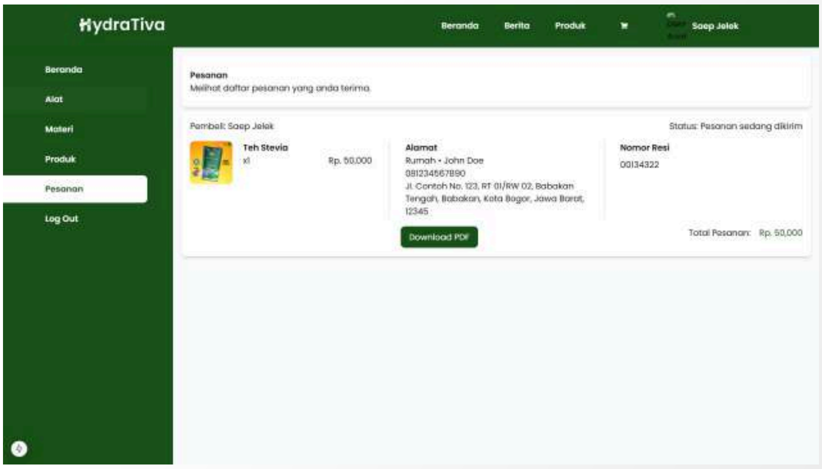
10. Produk



Data Produk

#	Nama	Deskripsi	Harga	Stok	Gambar	Aksi
1	Teh Stevia	Teh Stevia	Rp. 50000	9		

11. Pesanan



Pesanan
Melihat daftar pesanan yang anda terima

Pembeli: Saep Jelek

Status: Pesanan sedang dikirim

Teh Stevia
x1
Rp. 50,000

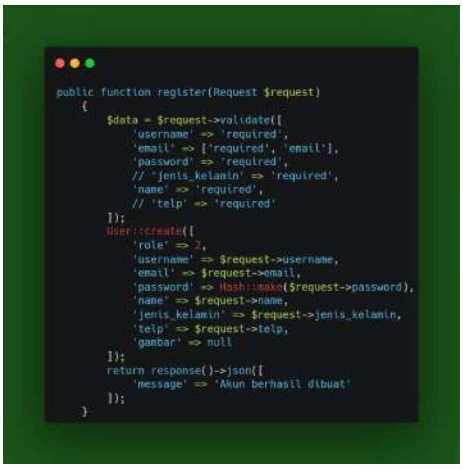
Alamat
Rumah • John Doe
081234567890
Jl. Contoh No. 123, RT 01/RW 02, Babakan Tengah, Babakan, Kota Bogor, Jawa Barat, 12345

Nomor Resi
00134322

Total Pesanan: Rp. 50,000

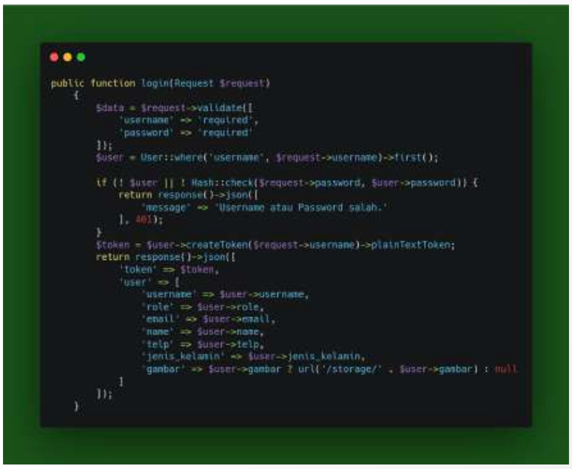
Source Code Website

1. Register



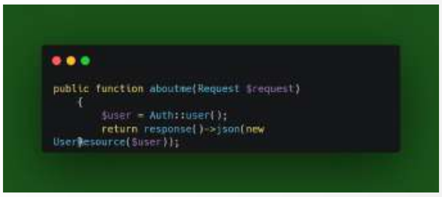
```
public function register(Request $request)
    {
        $data = $request->validate([
            'username' => 'required',
            'email' => ['required', 'email'],
            'password' => 'required',
            // 'jenis_kelamin' => 'required',
            'name' => 'required',
            // 'telp' => 'required'
        ]);
        User::create([
            'role' => 2,
            'username' => $request->username,
            'email' => $request->email,
            'password' => Hash::make($request->password),
            'name' => $request->name,
            'jenis_kelamin' => $request->jenis_kelamin,
            'telp' => $request->telp,
            'gambar' => null
        ]);
        return response()->json([
            'message' => 'Akun berhasil dibuat'
        ]);
    }
```

2. Login



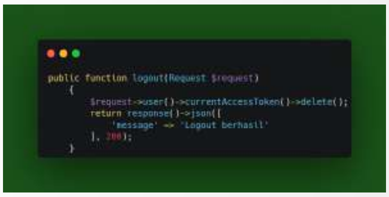
```
public function login(Request $request)
    {
        $data = $request->validate([
            'username' => 'required',
            'password' => 'required'
        ]);
        $user = User::where('username', $request->username)->first();

        if (! $user || ! Hash::check($request->password, $user->password)) {
            return response()->json([
                'message' => 'Username atau Password salah.'
            ], 401);
        }
        $token = $user->createToken($request->username)->plainTextToken;
        return response()->json([
            'token' => $token,
            'user' => [
                'username' => $user->username,
                'role' => $user->role,
                'email' => $user->email,
                'name' => $user->name,
                'telp' => $user->telp,
                'jenis_kelamin' => $user->jenis_kelamin,
                'gambar' => $user->gambar ? url('/storage/' . $user->gambar) : null
            ]
        ]);
    }
```

3. Profil



```
public function aboutme(Request $request)
    {
        $user = Auth::user();
        return response()->json(new
UserResource($user));
```

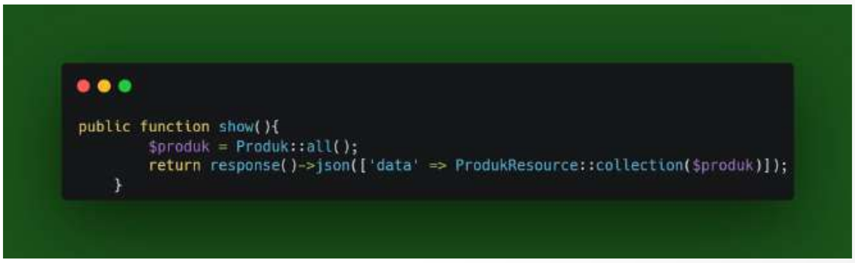
4. Logout



```
public function logout(Request $request)
    {
        $request->user()->currentAccessToken()->delete();
        return response()->json([
            'message' => 'Logout berhasil'
        ], 200);
    }
```

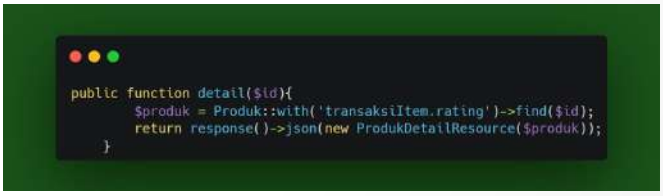
Source Code Website

5. Produk



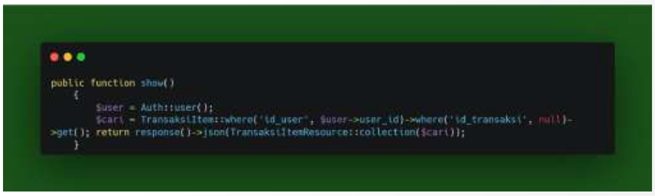
```
public function show(){
        $produk = Produk::all();
        return response()->json(['data' => ProdukResource::collection($produk)]);
    }
```

6. Detail Produk



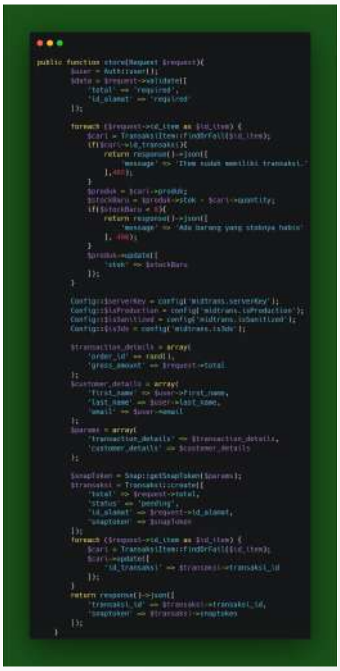
```
public function detail($id){
        $produk = Produk::with('transaksiItem.rating')->find($id);
        return response()->json(new ProdukDetailResource($produk));
    }
```

7. Keranjang



```
public function show()
    {
        $user = Auth::user();
        $cari = TransaksiItem::where('id_user', $user->user_id)->where('id_transaksi', null)-
>get(); return response()->json(TransaksiItemResource::collection($cari));
    }
```

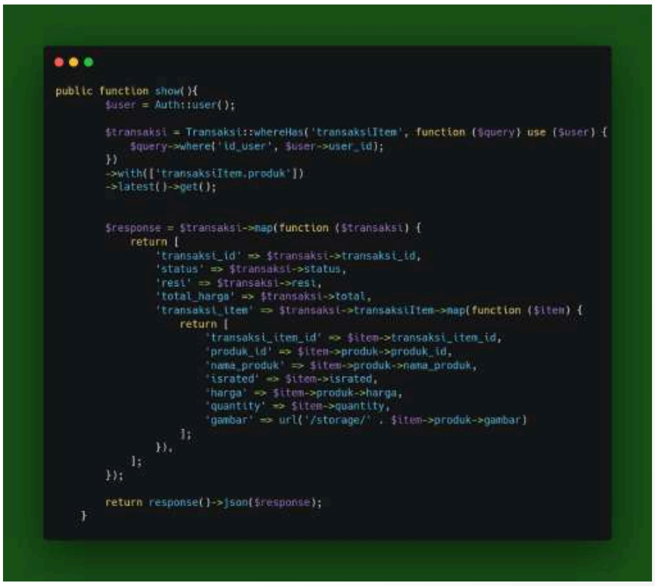
8. Checkout



```
public function store(Request $request){
        $user = Auth::user();
        $data = $request->validate([
            'total' => 'required',
            'id_alamat' => 'required'
        ]);

        foreach ($request->id_item as $id_item) {
            $cari = TransaksiItem::findOrFail($id_item);
            if($cari->id_transaksi){
                return response()->json([
                    'message' => 'Item sudah memiliki transaksi.'
                ],400);
            }
            $produk = $cari->produk;
            $stockBaru = $produk->stok - $cari->quantity;
            if($stockBaru < 0){
                return response()->json([
                    'message' => 'Ada barang yang stoknya habis'
                ], 400);
            }
            $produk->update([
                'stok' => $stockBaru
            ]);
        }

        Config::$serverKey = config('midtrans.serverKey');
        Config::$isProduction = config('midtrans.isProduction');
        Config::$isSanitized = config('midtrans.isSanitized');
        Config::$is3ds = config('midtrans.is3ds');

        $transaction_details = array(
            'order_id' => rand(),
            'gross_amount' => $request->total
        );
        $customer_details = array(
            'first_name' => $user->first_name,
            'last_name' => $user->last_name,
            'email' => $user->email
        );
        $params = array(
            'transaction_details' => $transaction_details,
            'customer_details' => $customer_details
        );

        $snapToken = Snap::getSnapToken($params);
        $transaksi = Transaksi::create([
            'total' => $request->total,
            'status' => 'pending',
            'id_alamat' => $request->id_alamat,
            'snaptoken' => $snapToken
        ]);
        foreach ($request->id_item as $id_item) {
            $cari = TransaksiItem::findOrFail($id_item);
            $cari->update([
                'id_transaksi' => $transaksi->transaksi_id
            ]);
        }
        return response()->json([
            'transaksi_id' => $transaksi->transaksi_id,
            'snaptoken' => $transaksi->snaptoken
        ]);
    }
```

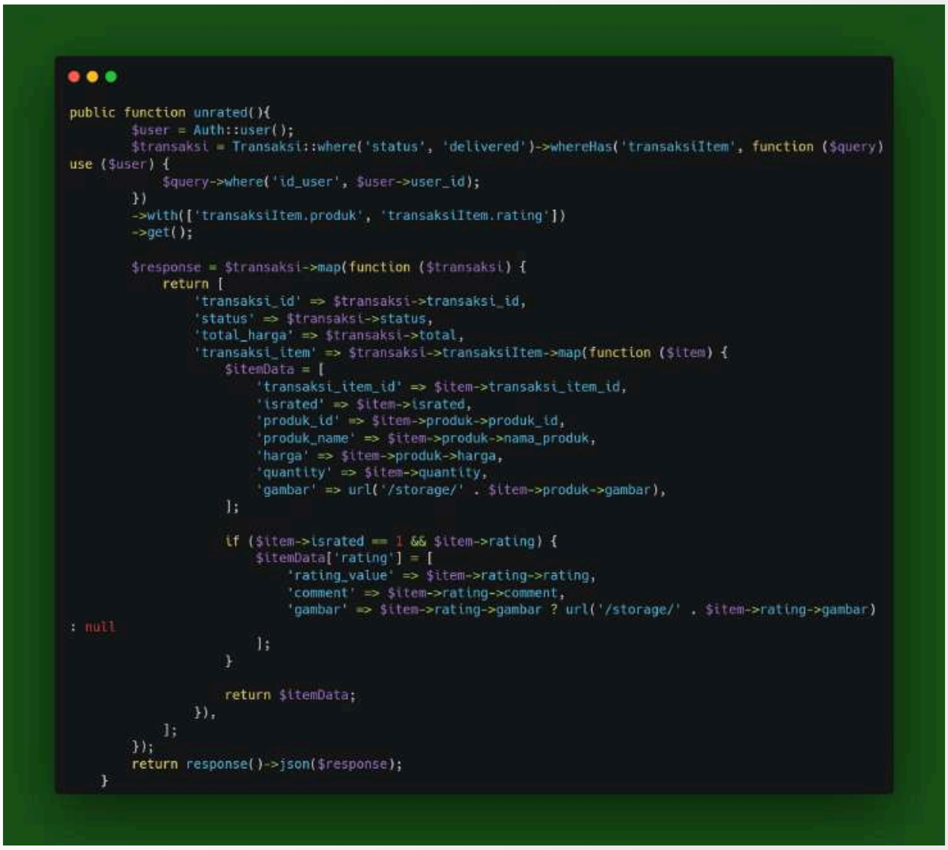
Source Code Website

9. Histori Transaksi



```
public function show( ){
        $user = Auth::user();

        $transaksi = Transaksi::whereHas('transaksiItem', function ($query) use ($user) {
            $query->where('id_user', $user->user_id);
        })
        ->with(['transaksiItem.produk'])
        ->latest()->get();


        $response = $transaksi->map(function ($transaksi) {
            return [
                'transaksi_id' => $transaksi->transaksi_id,
                'status' => $transaksi->status,
                'resi' => $transaksi->resi,
                'total_harga' => $transaksi->total,
                'transaksi_item' => $transaksi->transaksiItem->map(function ($item) {
                    return [
                        'transaksi_item_id' => $item->transaksi_item_id,
                        'produk_id' => $item->produk->produk_id,
                        'nama_produk' => $item->produk->nama_produk,
                        'israted' => $item->israted,
                        'harga' => $item->produk->harga,
                        'quantity' => $item->quantity,
                        'gambar' => url('/storage/' . $item->produk->gambar)
                    ];
                }),
            ];
        });

        return response()->json($response);
    }
```

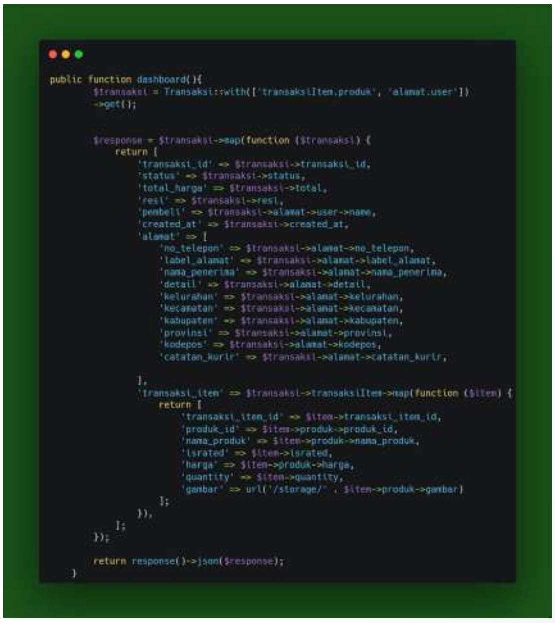
10. Ulasan



```
public function unrated(){
        $user = Auth::user();
        $transaksi = Transaksi::where('status', 'delivered')->whereHas('transaksiItem', function ($query)
use ($user) {
            $query->where('id_user', $user->user_id);
        })
        ->with(['transaksiItem.produk', 'transaksiItem.rating'])
        ->get();

        $response = $transaksi->map(function ($transaksi) {
            return [
                'transaksi_id' => $transaksi->transaksi_id,
                'status' => $transaksi->status,
                'total_harga' => $transaksi->total,
                'transaksi_item' => $transaksi->transaksiItem->map(function ($item) {
                    $itemData = [
                        'transaksi_item_id' => $item->transaksi_item_id,
                        'israted' => $item->israted,
                        'produk_id' => $item->produk->produk_id,
                        'produk_name' => $item->produk->nama_produk,
                        'harga' => $item->produk->harga,
                        'quantity' => $item->quantity,
                        'gambar' => url('/storage/' . $item->produk->gambar),
                    ];

                    if ($item->israted == 1 && $item->rating) {
                        $itemData['rating'] = [
                            'rating_value' => $item->rating->rating,
                            'comment' => $item->rating->comment,
                            'gambar' => $item->rating->gambar ? url('/storage/' . $item->rating->gambar)
: null
                        ];
                    }

                    return $itemData;
                }),
            ];
        });
        return response()->json($response);
    }
```

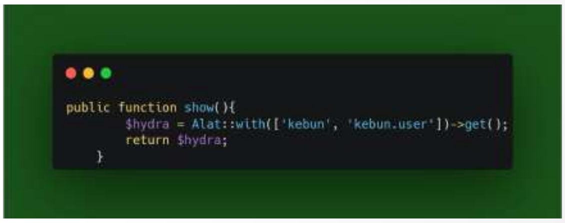
Source Code Website

11. Beranda Admin



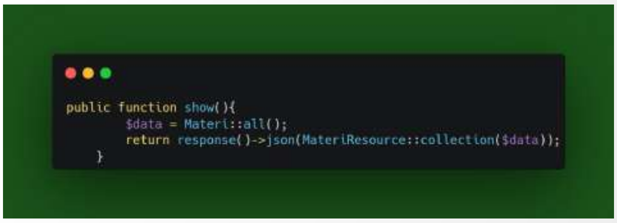
```
public function dashboard(){
        $transaksi = Transaksi::with(['transaksiItem.produk', 'alamat.user'])
        ->get();


        $response = $transaksi->map(function ($transaksi) {
            return [
                'transaksi_id' => $transaksi->transaksi_id,
                'status' => $transaksi->status,
                'total_harga' => $transaksi->total,
                'resi' => $transaksi->resi,
                'pembeli' => $transaksi->alamat->user->name,
                'created_at' => $transaksi->created_at,
                'alamat' => [
                    'no_telepon' => $transaksi->alamat->no_telepon,
                    'label_alamat' => $transaksi->alamat->label_alamat,
                    'nama_penerima' => $transaksi->alamat->nama_penerima,
                    'detail' => $transaksi->alamat->detail,
                    'kelurahan' => $transaksi->alamat->kelurahan,
                    'kecamatan' => $transaksi->alamat->kecamatan,
                    'kabupaten' => $transaksi->alamat->kabupaten,
                    'provinsi' => $transaksi->alamat->provinsi,
                    'kodepos' => $transaksi->alamat->kodepos,
                    'catatan_kurir' => $transaksi->alamat->catatan_kurir,

                ],
                'transaksi_item' => $transaksi->transaksiItem->map(function ($item) {
                    return [
                        'transaksi_item_id' => $item->transaksi_item_id,
                        'produk_id' => $item->produk->produk_id,
                        'nama_produk' => $item->produk->nama_produk,
                        'israted' => $item->israted,
                        'harga' => $item->produk->harga,
                        'quantity' => $item->quantity,
                        'gambar' => url('/storage/' . $item->produk->gambar)
                    ];
                }),
            ];
        });

        return response()->json($response);
    }
```

10. Alat



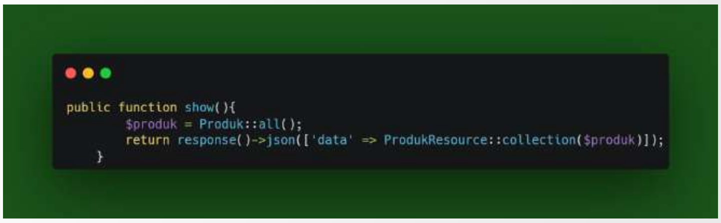
```
public function show(){
        $hydra = Alat::with(['kebun', 'kebun.user'])->get();
        return $hydra;
    }
```

11. Materi



```
public function show(){
        $data = Materi::all();
        return response()->json(MateriResource::collection($data));
    }
```

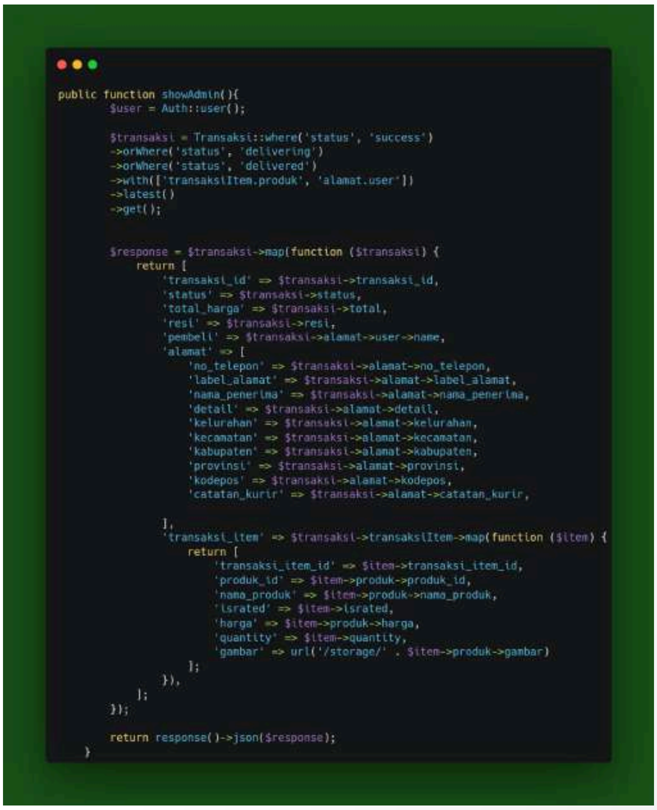
12. Produk



```
public function show(){
        $produk = Produk::all();
        return response()->json(['data' => ProdukResource::collection($produk)]);
    }
```

Source Code Website

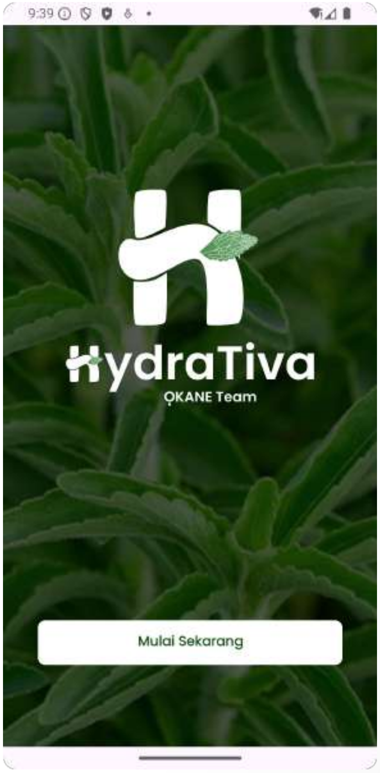
13. Pesanan



```
public function showAdmin(){
        $user = Auth::user();

        $transaksi = Transaksi::where('status', 'success')
        ->orWhere('status', 'delivering')
        ->orWhere('status', 'delivered')
        ->with(['transaksiItem.produk', 'alamat.user'])
        ->latest()
        ->get();


        $response = $transaksi->map(function ($transaksi) {
            return [
                'transaksi_id' => $transaksi->transaksi_id,
                'status' => $transaksi->status,
                'total_harga' => $transaksi->total,
                'resi' => $transaksi->resi,
                'pembeli' => $transaksi->alamat->user->name,
                'alamat' => [
                    'no_telepon' => $transaksi->alamat->no_telepon,
                    'label_alamat' => $transaksi->alamat->label_alamat,
                    'nama_penerima' => $transaksi->alamat->nama_penerima,
                    'detail' => $transaksi->alamat->detail,
                    'kelurahan' => $transaksi->alamat->kelurahan,
                    'kecamatan' => $transaksi->alamat->kecamatan,
                    'kabupaten' => $transaksi->alamat->kabupaten,
                    'provinsi' => $transaksi->alamat->provinsi,
                    'kodepos' => $transaksi->alamat->kodepos,
                    'catatan_kurir' => $transaksi->alamat->catatan_kurir,

                ],
                'transaksi_item' => $transaksi->transaksiItem->map(function ($item) {
                    return [
                        'transaksi_item_id' => $item->transaksi_item_id,
                        'produk_id' => $item->produk->produk_id,
                        'nama_produk' => $item->produk->nama_produk,
                        'israted' => $item->israted,
                        'harga' => $item->produk->harga,
                        'quantity' => $item->quantity,
                        'gambar' => url('/storage/' . $item->produk->gambar)
                    ];
                }),
            ];
        });

        return response()->json($response);
    }
```

Tampilan Aplikasi

1. Splash Screen



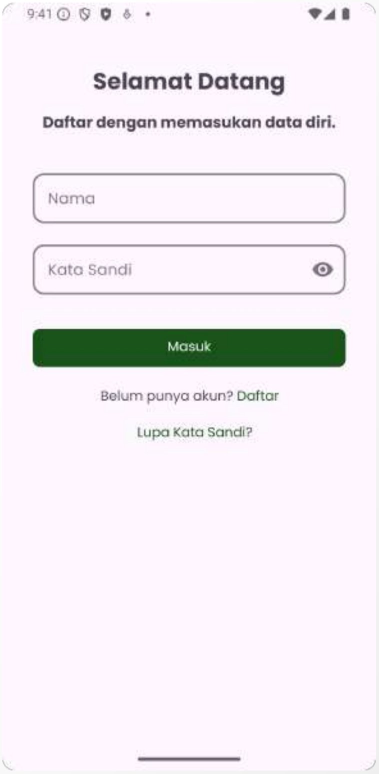
9:39

HydraTiva

QKANE Team

Mulai Sekarang

2. Login



9:41

Selamat Datang

Daftar dengan memasukan data diri.

Nama

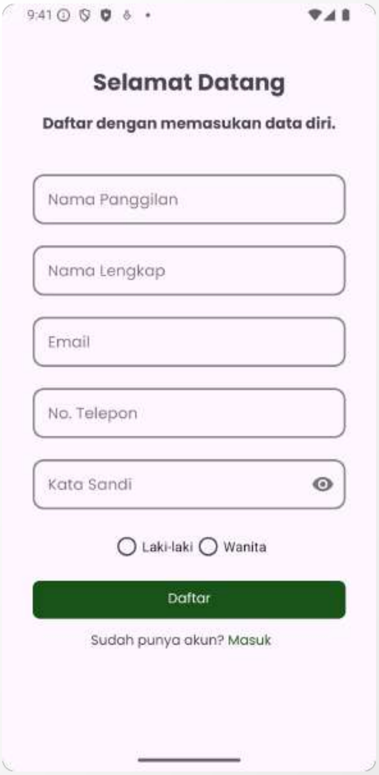
Kata Sandi

Masuk

Belum punya akun? Daftar

Lupa Kata Sandi?

3. Register



9:41

Selamat Datang

Daftar dengan memasukan data diri.

Nama Panggilan

Nama Lengkap

Email

No. Telepon

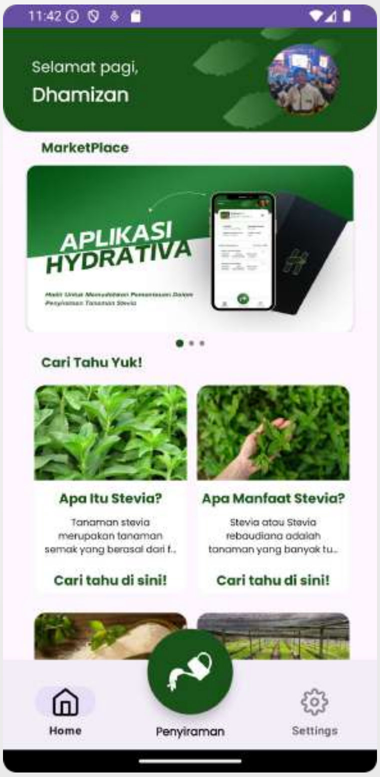
Kata Sandi

○ Laki-laki ○ Wanita

Daftar

Sudah punya akun? Masuk

4. Beranda



11:42

Selamat pagi,
Dhamizan

MarketPlace

APLIKASI HYDRATIVA

Hadir Untuk Memudahkan Pemantauan Dalam Penyiraman Tanaman Stevia

Cari Tahu Yuk!

Apa Itu Stevia?

Tanaman stevia merupakan tanaman semak yang berasal dari f...

Cari tahu di sini!

Apa Manfaat Stevia?

Stevia atau Stevia rebaudiana adalah tanaman yang banyak tu...

Cari tahu di sini!

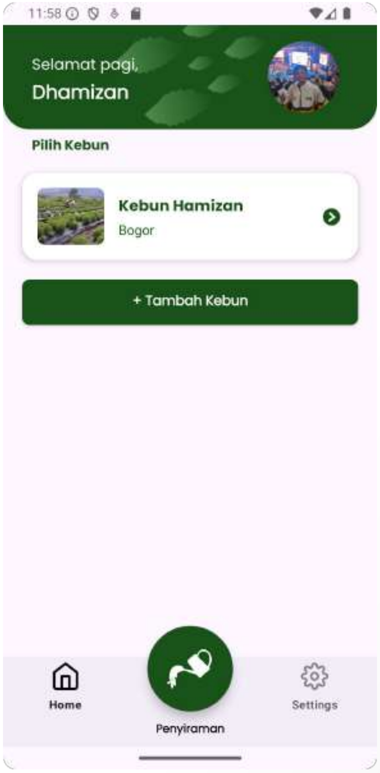
Home

Penyiraman

Settings

Tampilan Aplikasi

5. Kebun



11:58

Selamat pagi,
Dhamizan

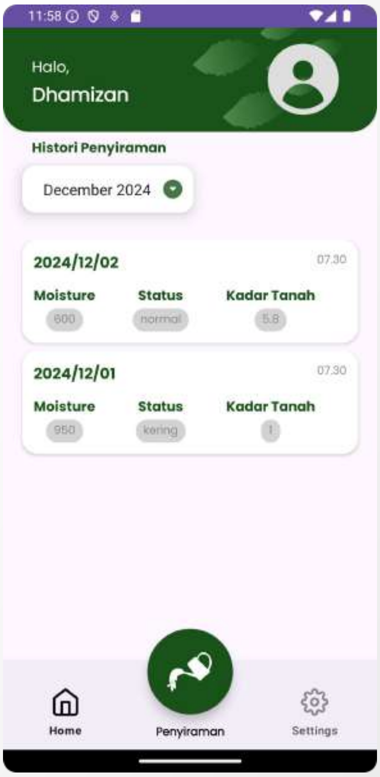
Pilih Kebun

Kebun Hamizan
Bogor

+ Tambah Kebun

Home
Penyiraman
Settings

7. Histori Penyiraman



11:58

Halo,
Dhamizan

Histori Penyiraman

December 2024

2024/12/02 07.30

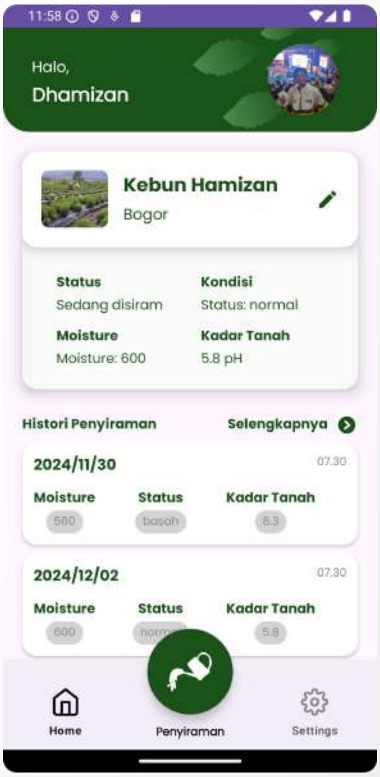
Moisture	Status	Kadar Tanah
600	normal	5.8

2024/12/01 07.30

Moisture	Status	Kadar Tanah
950	kering	1

Home
Penyiraman
Settings

6. Pemantauan Penyiraman



11:58

Halo,
Dhamizan

Kebun Hamizan
Bogor

Status Sedang disiram	Kondisi Status: normal
Moisture Moisture: 600	Kadar Tanah 5.8 pH

Histori Penyiraman
Selengkapnya

2024/11/30 07.30

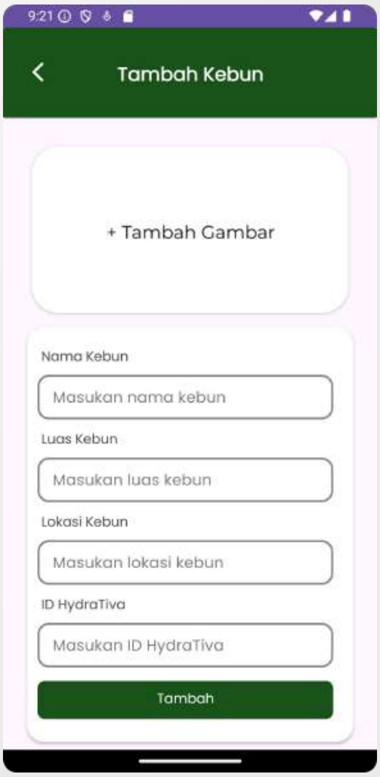
Moisture	Status	Kadar Tanah
560	basah	6.3

2024/12/02 07.30

Moisture	Status	Kadar Tanah
600	norm	5.8

Home
Penyiraman
Settings

8. Tambah Kebun



9:21

Tambah Kebun

+ Tambah Gambar

Nama Kebun
Masukan nama kebun

Luas Kebun
Masukan luas kebun

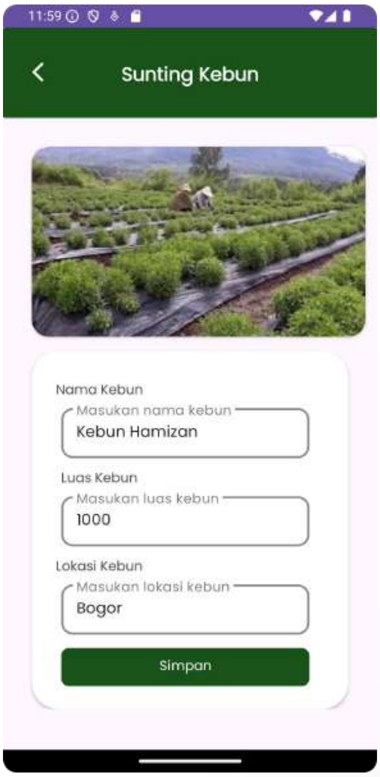
Lokasi Kebun
Masukan lokasi kebun

ID HydraTiva
Masukan ID HydraTiva

Tambah

Tampilan Aplikasi

9. Sunting Kebun



11:59

Sunting Kebun

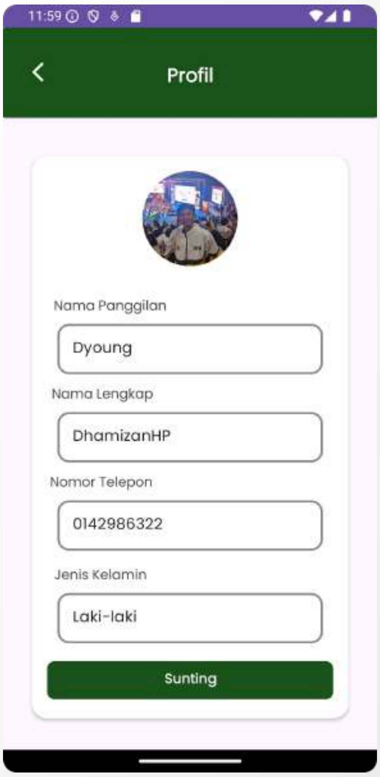
Nama Kebun
Masukan nama kebun: Kebun Hamizan

Luas Kebun
Masukan luas kebun: 1000

Lokasi Kebun
Masukan lokasi kebun: Bogor

Simpan

11. Profil Pengguna



11:59

Profil

Nama Panggilan: Dyoung

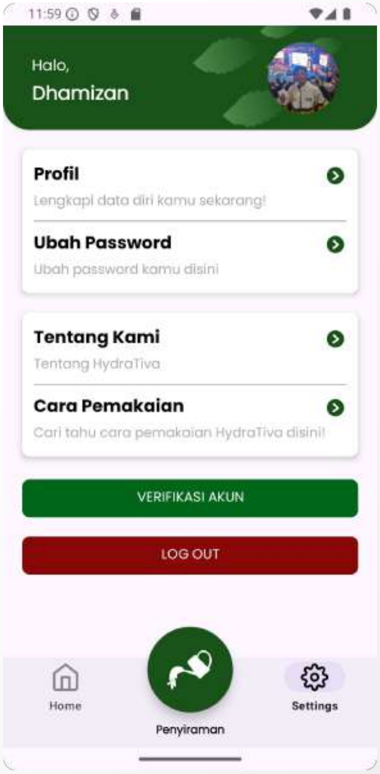
Nama Lengkap: DhamizanHP

Nomor Telepon: 0142986322

Jenis Kelamin: Laki-laki

Sunting

10. Pengaturan



11:59

Halo,
Dhamizan

Profil
Lengkapi data diri kamu sekarang!

Ubah Password
Ubah password kamu disini

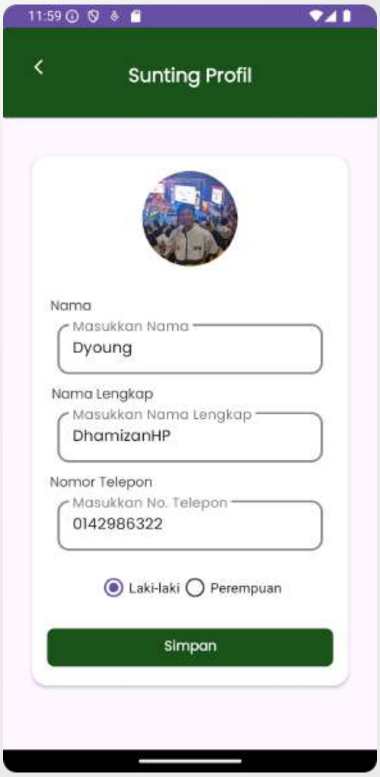
Tentang Kami
Tentang HydraTiva

Cara Pemakaian
Cari tahu cara pemakaian HydraTiva disini!

VERIFIKASI AKUN

LOG OUT

12. Sunting Profil



11:59

Sunting Profil

Nama
Masukkan Nama: Dyoung

Nama Lengkap
Masukkan Nama Lengkap: DhamizanHP

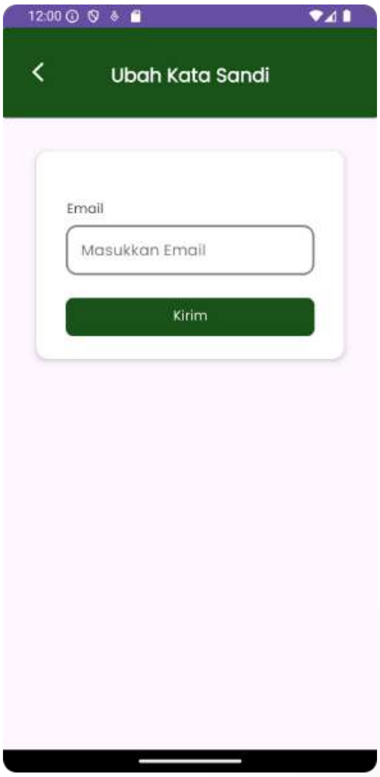
Nomor Telepon
Masukkan No. Telepon: 0142986322

● Laki-laki ○ Perempuan

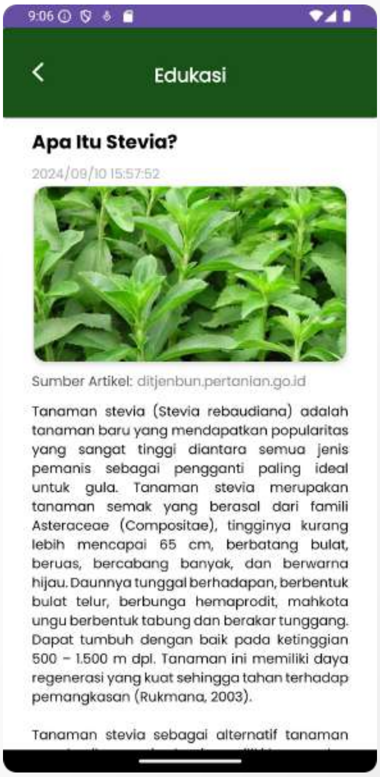
Simpan

Tampilan Aplikasi

13. Verifikasi Akun



14. Edukasi (Materi)



Apa Itu Stevia?

2024/09/10 15:57:52

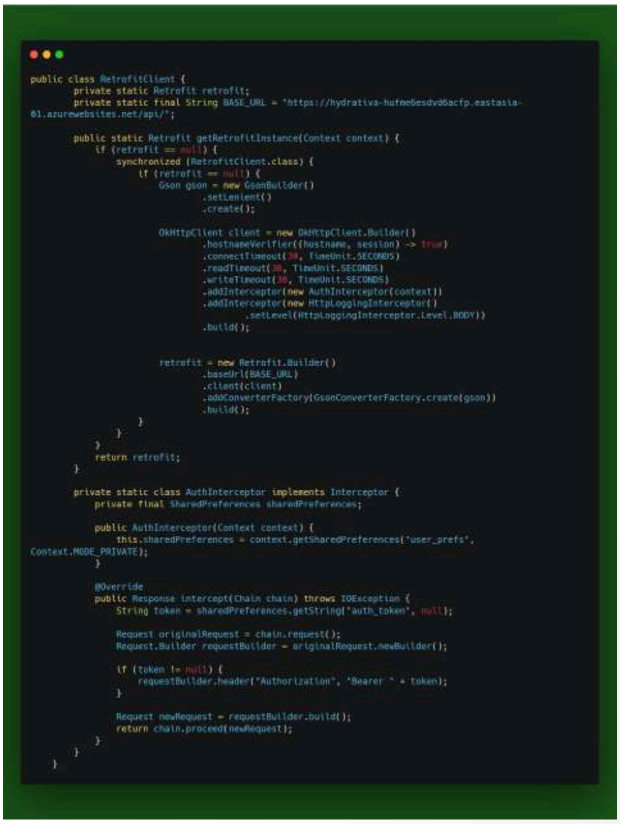
Sumber Artikel: ditjenbun.pertanian.go.id

Tanaman stevia (Stevia rebaudiana) adalah tanaman baru yang mendapatkan popularitas yang sangat tinggi diantara semua jenis pemanis sebagai pengganti paling ideal untuk gula. Tanaman stevia merupakan tanaman semak yang berasal dari famili Asteraceae (Compositae), tingginya kurang lebih mencapai 65 cm, berbatang bulat, beruas, bercabang banyak, dan berwarna hijau. Daunnya tunggal berhadapan, berbentuk bulat telur, berbunga hemaprodit, mahkota ungu berbentuk tabung dan berakar tunggang. Dapat tumbuh dengan baik pada ketinggian 500 – 1.500 m dpl. Tanaman ini memiliki daya regenerasi yang kuat sehingga tahan terhadap pemangkasan (Rukmana, 2003).

Tanaman stevia sebagai alternatif tanaman

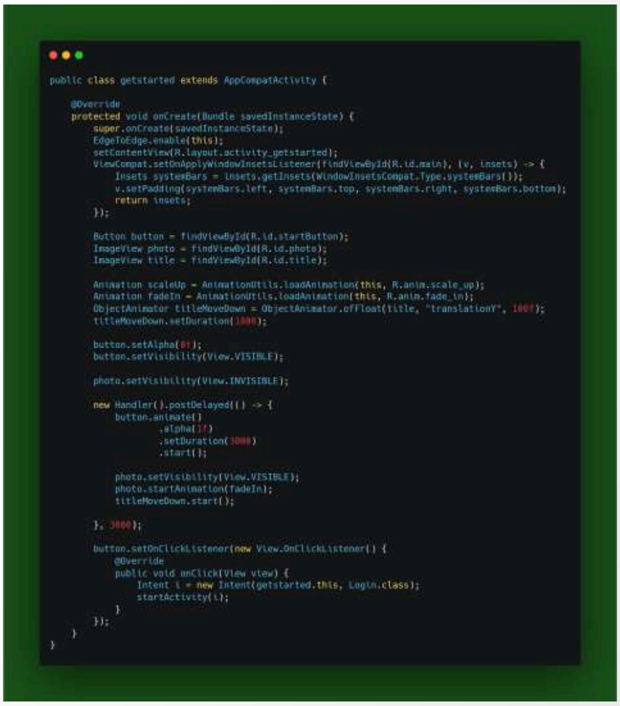
Source Code Aplikasi

1. Retrofit Client



```
public class RetrofitClient {
        private static Retrofit retrofit;
        private static final String BASE_URL = "https://hydrativa-hufme6esdvd6acfp.eastasia-
01.azurewebsites.net/api/";

        public static Retrofit getRetrofitInstance(Context context) {
            if (retrofit == null) {
                synchronized (RetrofitClient.class) {
                    if (retrofit == null) {
                        Gson gson = new GsonBuilder()
                                .setLenient()
                                .create();

                        OkHttpClient client = new OkHttpClient.Builder()
                                .hostnameVerifier((hostname, session) -> true)
                                .connectTimeout(30, TimeUnit.SECONDS)
                                .readTimeout(30, TimeUnit.SECONDS)
                                .writeTimeout(30, TimeUnit.SECONDS)
                                .addInterceptor(new AuthInterceptor(context))
                                .addInterceptor(new HttpLoggingInterceptor()
                                        .setLevel(HttpLoggingInterceptor.Level.BODY))
                                .build();


                        retrofit = new Retrofit.Builder()
                                .baseUrl(BASE_URL)
                                .client(client)
                                .addConverterFactory(GsonConverterFactory.create(gson))
                                .build();
                    }
                }
            }
            return retrofit;
        }

        private static class AuthInterceptor implements Interceptor {
            private final SharedPreferences sharedPreferences;

            public AuthInterceptor(Context context) {
                this.sharedPreferences = context.getSharedPreferences("user_prefs",
Context.MODE_PRIVATE);
            }

            @Override
            public Response intercept(Chain chain) throws IOException {
                String token = sharedPreferences.getString("auth_token", null);

                Request originalRequest = chain.request();
                Request.Builder requestBuilder = originalRequest.newBuilder();

                if (token != null) {
                    requestBuilder.header("Authorization", "Bearer " + token);
                }

                Request newRequest = requestBuilder.build();
                return chain.proceed(newRequest);
            }
        }
    }
```

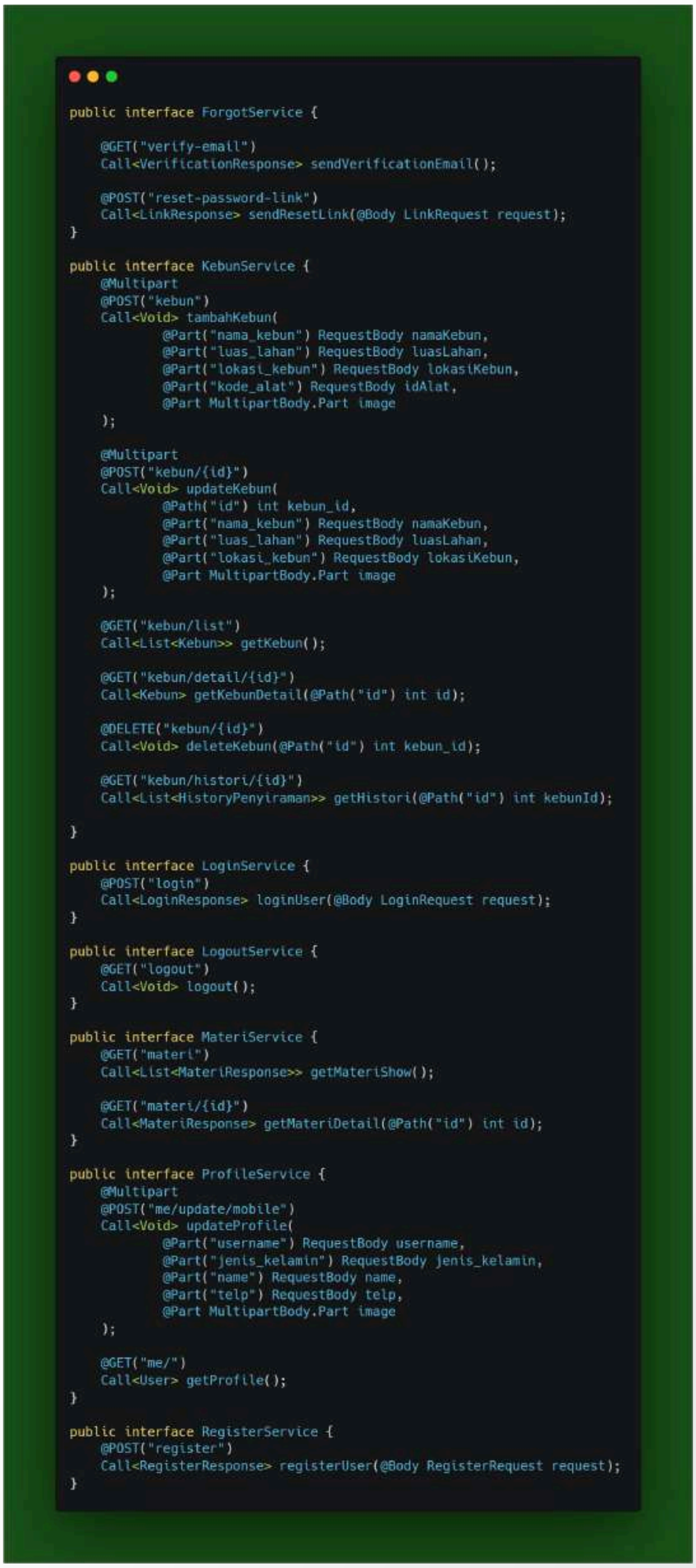
2. Splash Screen



```
public class getstarted extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_getstarted);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        Button button = findViewById(R.id.startButton);
        ImageView photo = findViewById(R.id.photo);
        ImageView title = findViewById(R.id.title);

        Animation scaleUp = AnimationUtils.loadAnimation(this, R.anim.scale_up);
        Animation fadeIn = AnimationUtils.loadAnimation(this, R.anim.fade_in);
        ObjectAnimator titleMoveDown = ObjectAnimator.ofFloat(title, "translationY", 100f);
        titleMoveDown.setDuration(1000);

        button.setAlpha(0f);
        button.setVisibility(View.VISIBLE);

        photo.setVisibility(View.INVISIBLE);

        new Handler().postDelayed(() -> {
            button.animate()
                    .alpha(1f)
                    .setDuration(3000)
                    .start();

            photo.setVisibility(View.VISIBLE);
            photo.startAnimation(fadeIn);
            titleMoveDown.start();

        }, 3000);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(getstarted.this, Login.class);
                startActivity(i);
            }
        });
    }
}
```

Source Code Aplikasi

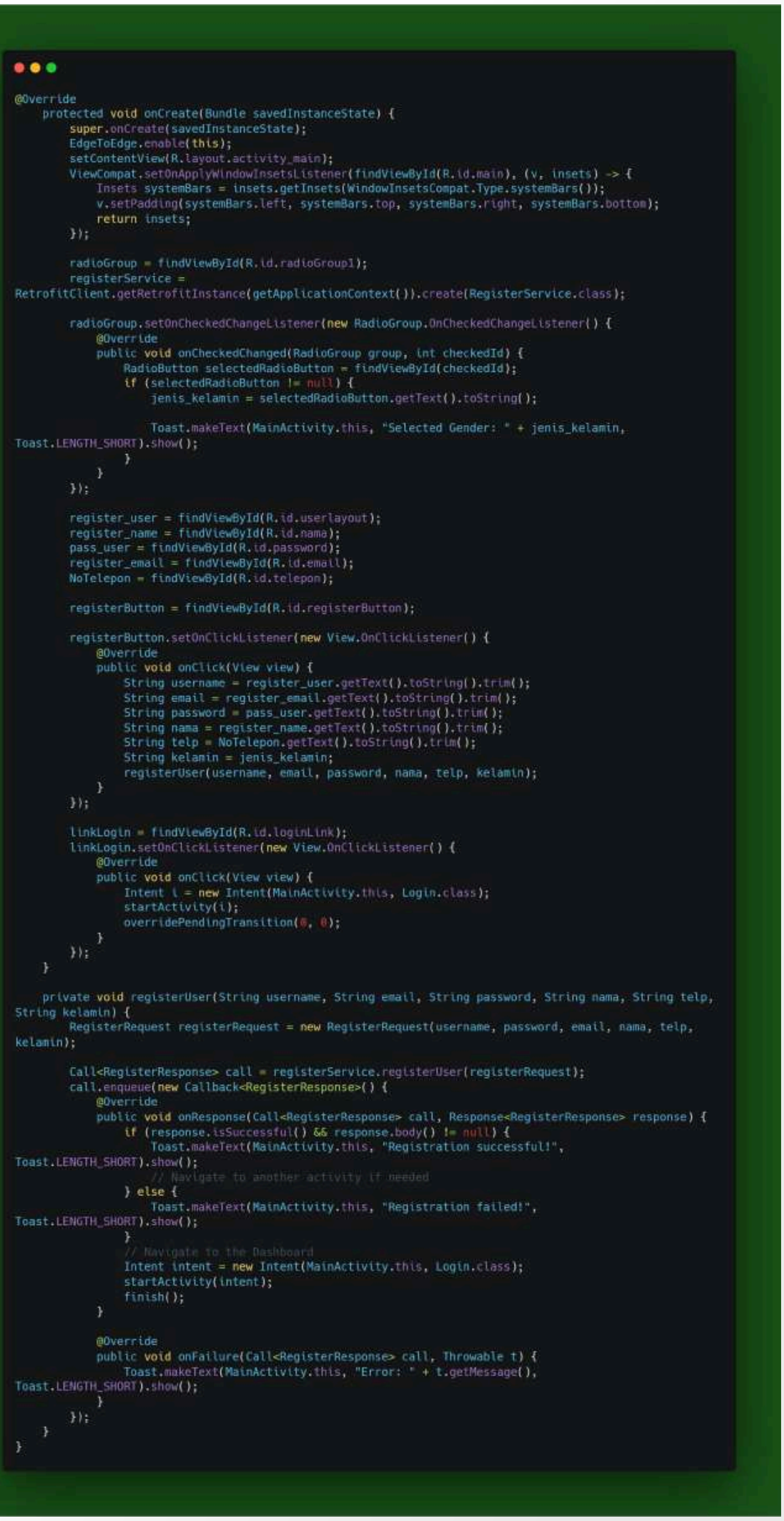
3. Kode Service



```java
public interface ForgotService {

    @GET("verify-email")
    Call<VerificationResponse> sendVerificationEmail();

    @POST("reset-password-link")
    Call<LinkResponse> sendResetLink(@Body LinkRequest request);
}

public interface KebunService {
    @Multipart
    @POST("kebun")
    Call<Void> tambahKebun(
            @Part("nama_kebun") RequestBody namaKebun,
            @Part("luas_lahan") RequestBody luasLahan,
            @Part("lokasi_kebun") RequestBody lokasiKebun,
            @Part("kode_alat") RequestBody idAlat,
            @Part MultipartBody.Part image
    );

    @Multipart
    @POST("kebun/{id}")
    Call<Void> updateKebun(
            @Path("id") int kebun_id,
            @Part("nama_kebun") RequestBody namaKebun,
            @Part("luas_lahan") RequestBody luasLahan,
            @Part("lokasi_kebun") RequestBody lokasiKebun,
            @Part MultipartBody.Part image
    );

    @GET("kebun/list")
    Call<List<Kebun>> getKebun();

    @GET("kebun/detail/{id}")
    Call<Kebun> getKebunDetail(@Path("id") int id);

    @DELETE("kebun/{id}")
    Call<Void> deleteKebun(@Path("id") int kebun_id);

    @GET("kebun/histori/{id}")
    Call<List<HistoryPenyiraman>> getHistori(@Path("id") int kebunId);

}

public interface LoginService {
    @POST("login")
    Call<LoginResponse> loginUser(@Body LoginRequest request);
}

public interface LogoutService {
    @GET("logout")
    Call<Void> logout();
}

public interface MateriService {
    @GET("materi")
    Call<List<MateriResponse>> getMateriShow();

    @GET("materi/{id}")
    Call<MateriResponse> getMateriDetail(@Path("id") int id);
}

public interface ProfileService {
    @Multipart
    @POST("me/update/mobile")
    Call<Void> updateProfile(
            @Part("username") RequestBody username,
            @Part("jenis_kelamin") RequestBody jenis_kelamin,
            @Part("name") RequestBody name,
            @Part("telp") RequestBody telp,
            @Part MultipartBody.Part image
    );

    @GET("me/")
    Call<User> getProfile();
}

public interface RegisterService {
    @POST("register")
    Call<RegisterResponse> registerUser(@Body RegisterRequest request);
}
```

Source Code Aplikasi

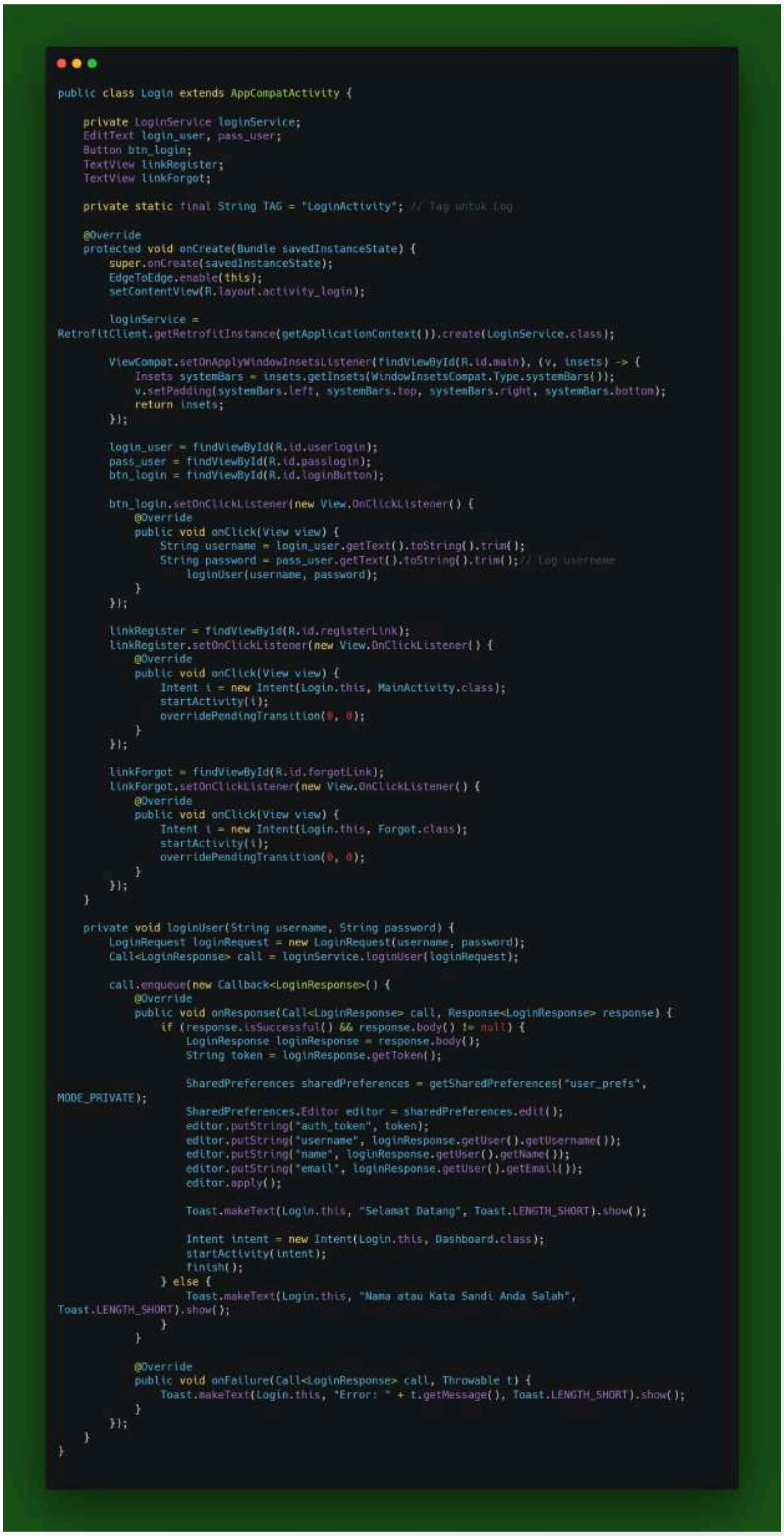
4. Register



```java
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        radioGroup = findViewById(R.id.radioGroup1);
        registerService =
RetrofitClient.getRetrofitInstance(getApplicationContext()).create(RegisterService.class);

        radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                RadioButton selectedRadioButton = findViewById(checkedId);
                if (selectedRadioButton != null) {
                    jenis_kelamin = selectedRadioButton.getText().toString();

                    Toast.makeText(MainActivity.this, "Selected Gender: " + jenis_kelamin,
Toast.LENGTH_SHORT).show();
                }
            }
        });

        register_user = findViewById(R.id.userlayout);
        register_name = findViewById(R.id.nama);
        pass_user = findViewById(R.id.password);
        register_email = findViewById(R.id.email);
        NoTelepon = findViewById(R.id.telepon);

        registerButton = findViewById(R.id.registerButton);

        registerButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String username = register_user.getText().toString().trim();
                String email = register_email.getText().toString().trim();
                String password = pass_user.getText().toString().trim();
                String nama = register_name.getText().toString().trim();
                String telp = NoTelepon.getText().toString().trim();
                String kelamin = jenis_kelamin;
                registerUser(username, email, password, nama, telp, kelamin);
            }
        });

        linkLogin = findViewById(R.id.loginLink);
        linkLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(MainActivity.this, Login.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });
    }

    private void registerUser(String username, String email, String password, String nama, String telp,
String kelamin) {
        RegisterRequest registerRequest = new RegisterRequest(username, password, email, nama, telp,
kelamin);

        Call<RegisterResponse> call = registerService.registerUser(registerRequest);
        call.enqueue(new Callback<RegisterResponse>() {
            @Override
            public void onResponse(Call<RegisterResponse> call, Response<RegisterResponse> response) {
                if (response.isSuccessful() && response.body() != null) {
                    Toast.makeText(MainActivity.this, "Registration successful!",
Toast.LENGTH_SHORT).show();
                    // Navigate to another activity if needed
                } else {
                    Toast.makeText(MainActivity.this, "Registration failed!",
Toast.LENGTH_SHORT).show();
                }
                // Navigate to the Dashboard
                Intent intent = new Intent(MainActivity.this, Login.class);
                startActivity(intent);
                finish();
            }

            @Override
            public void onFailure(Call<RegisterResponse> call, Throwable t) {
                Toast.makeText(MainActivity.this, "Error: " + t.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Source Code Aplikasi

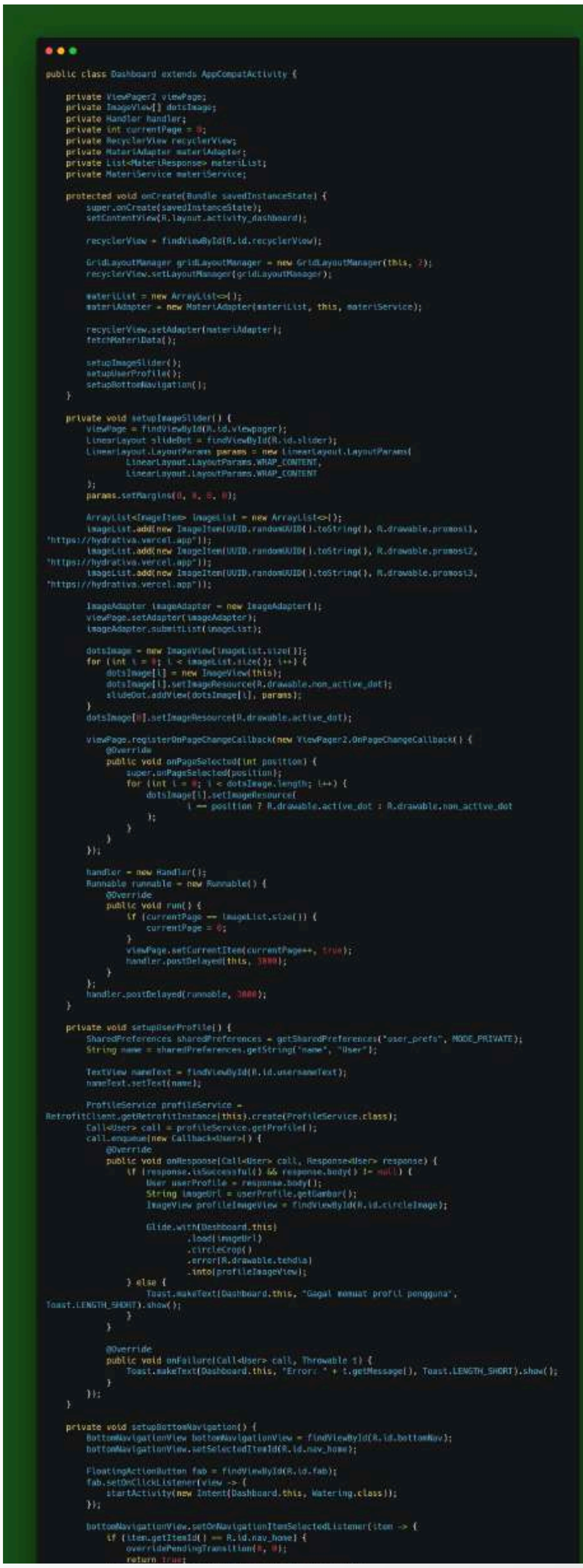
5. Login



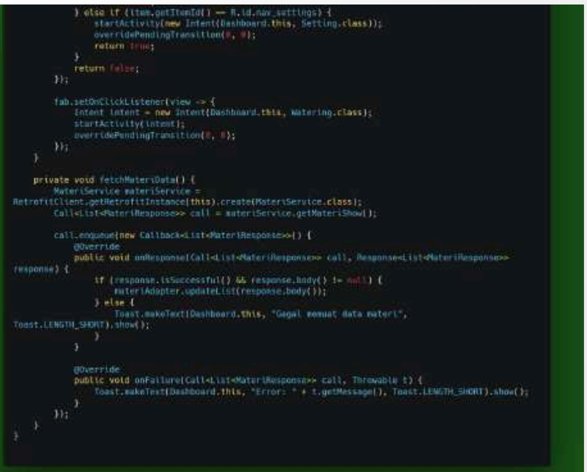
```
public class Login extends AppCompatActivity {

    private LoginService loginService;
    EditText login_user, pass_user;
    Button btn_login;
    TextView linkRegister;
    TextView linkForgot;

    private static final String TAG = "LoginActivity"; // Tag untuk Log

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_login);

        loginService =
RetrofitClient.getRetrofitInstance(getApplicationContext()).create(LoginService.class);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        login_user = findViewById(R.id.userlogin);
        pass_user = findViewById(R.id.passlogin);
        btn_login = findViewById(R.id.loginButton);

        btn_login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String username = login_user.getText().toString().trim();
                String password = pass_user.getText().toString().trim();// Log username
                    loginUser(username, password);
            }
        });

        linkRegister = findViewById(R.id.registerLink);
        linkRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Login.this, MainActivity.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });

        linkForgot = findViewById(R.id.forgotLink);
        linkForgot.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Login.this, Forgot.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });
    }

    private void loginUser(String username, String password) {
        LoginRequest loginRequest = new LoginRequest(username, password);
        Call<LoginResponse> call = loginService.loginUser(loginRequest);

        call.enqueue(new Callback<LoginResponse>() {
            @Override
            public void onResponse(Call<LoginResponse> call, Response<LoginResponse> response) {
                if (response.isSuccessful() && response.body() != null) {
                    LoginResponse loginResponse = response.body();
                    String token = loginResponse.getToken();

                    SharedPreferences sharedPreferences = getSharedPreferences("user_prefs",
MODE_PRIVATE);
                    SharedPreferences.Editor editor = sharedPreferences.edit();
                    editor.putString("auth_token", token);
                    editor.putString("username", loginResponse.getUser().getUsername());
                    editor.putString("name", loginResponse.getUser().getName());
                    editor.putString("email", loginResponse.getUser().getEmail());
                    editor.apply();

                    Toast.makeText(Login.this, "Selamat Datang", Toast.LENGTH_SHORT).show();

                    Intent intent = new Intent(Login.this, Dashboard.class);
                    startActivity(intent);
                    finish();
                } else {
                    Toast.makeText(Login.this, "Nama atau Kata Sandi Anda Salah",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<LoginResponse> call, Throwable t) {
                Toast.makeText(Login.this, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Source Code Aplikasi

6. Dashboard



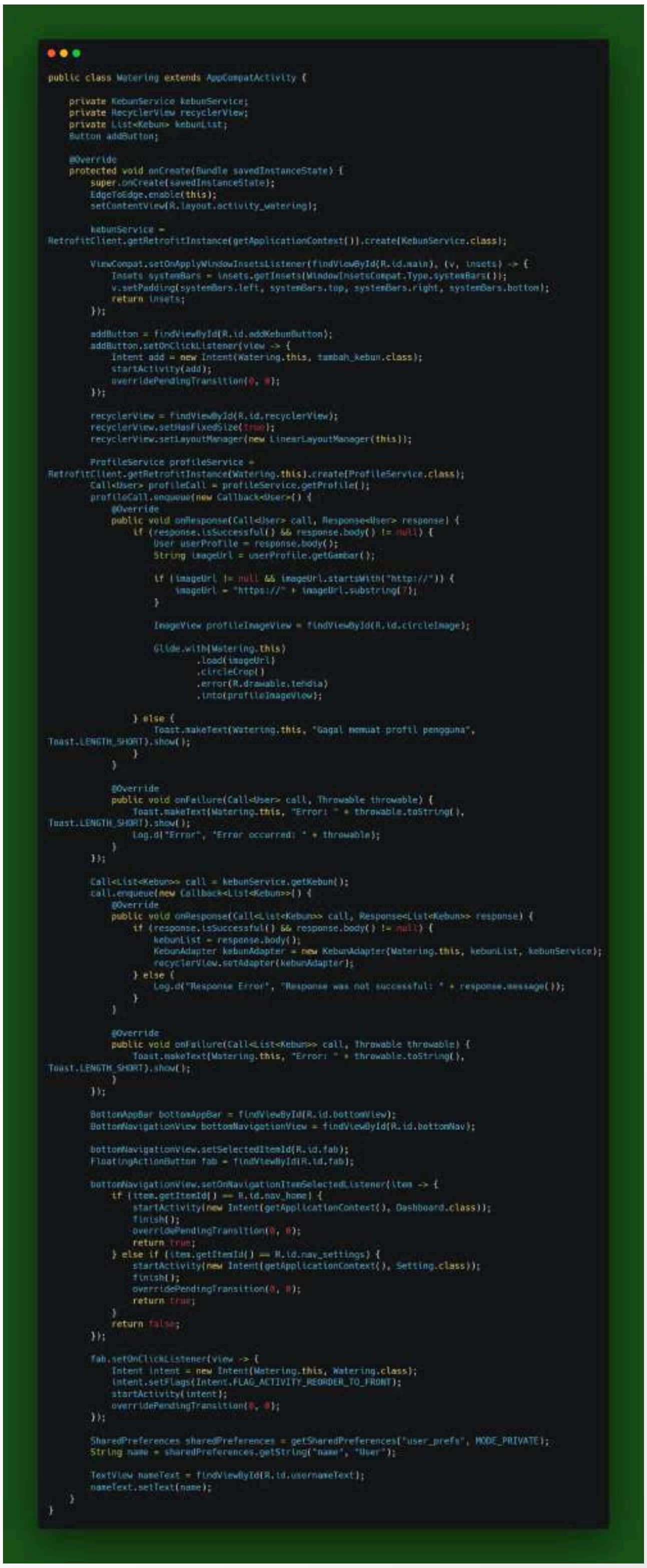
```java
public class Dashboard extends AppCompatActivity {

    private ViewPager2 viewPage;
    private ImageView[] dotsImage;
    private Handler handler;
    private int currentPage = 0;
    private RecyclerView recyclerView;
    private MateriAdapter materiAdapter;
    private List<MateriResponse> materiList;
    private MateriService materiService;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);

        recyclerView = findViewById(R.id.recyclerView);

        GridLayoutManager gridLayoutManager = new GridLayoutManager(this, 2);
        recyclerView.setLayoutManager(gridLayoutManager);

        materiList = new ArrayList<>();
        materiAdapter = new MateriAdapter(materiList, this, materiService);

        recyclerView.setAdapter(materiAdapter);
        fetchMateriData();

        setupImageSlider();
        setupUserProfile();
        setupBottomNavigation();
    }

    private void setupImageSlider() {
        viewPage = findViewById(R.id.viewpager);
        LinearLayout slideDot = findViewById(R.id.slider);
        LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.WRAP_CONTENT,
                LinearLayout.LayoutParams.WRAP_CONTENT
        );
        params.setMargins(8, 0, 8, 0);

        ArrayList<ImageItem> imageList = new ArrayList<>();
        imageList.add(new ImageItem(UUID.randomUUID().toString(), R.drawable.promosi1,
"https://hydrativa.vercel.app"));
        imageList.add(new ImageItem(UUID.randomUUID().toString(), R.drawable.promosi2,
"https://hydrativa.vercel.app"));
        imageList.add(new ImageItem(UUID.randomUUID().toString(), R.drawable.promosi3,
"https://hydrativa.vercel.app"));

        ImageAdapter imageAdapter = new ImageAdapter();
        viewPage.setAdapter(imageAdapter);
        imageAdapter.submitList(imageList);

        dotsImage = new ImageView[imageList.size()];
        for (int i = 0; i < imageList.size(); i++) {
            dotsImage[i] = new ImageView(this);
            dotsImage[i].setImageResource(R.drawable.non_active_dot);
            slideDot.addView(dotsImage[i], params);
        }
        dotsImage[0].setImageResource(R.drawable.active_dot);

        viewPage.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {
            @Override
            public void onPageSelected(int position) {
                super.onPageSelected(position);
                for (int i = 0; i < dotsImage.length; i++) {
                    dotsImage[i].setImageResource(
                            i == position ? R.drawable.active_dot : R.drawable.non_active_dot
                    );
                }
            }
        });

        handler = new Handler();
        Runnable runnable = new Runnable() {
            @Override
            public void run() {
                if (currentPage == imageList.size()) {
                    currentPage = 0;
                }
                viewPage.setCurrentItem(currentPage++, true);
                handler.postDelayed(this, 3000);
            }
        };
        handler.postDelayed(runnable, 3000);
    }

    private void setupUserProfile() {
        SharedPreferences sharedPreferences = getSharedPreferences("user_prefs", MODE_PRIVATE);
        String name = sharedPreferences.getString("name", "User");

        TextView nameText = findViewById(R.id.usernameText);
        nameText.setText(name);

        ProfileService profileService =
RetrofitClient.getRetrofitInstance(this).create(ProfileService.class);
        Call<User> call = profileService.getProfile();
        call.enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                if (response.isSuccessful() && response.body() != null) {
                    User userProfile = response.body();
                    String imageUrl = userProfile.getGambar();
                    ImageView profileImageView = findViewById(R.id.circleImage);

                    Glide.with(Dashboard.this)
                            .load(imageUrl)
                            .circleCrop()
                            .error(R.drawable.tehdia)
                            .into(profileImageView);
                } else {
                    Toast.makeText(Dashboard.this, "Gagal memuat profil pengguna",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<User> call, Throwable t) {
                Toast.makeText(Dashboard.this, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void setupBottomNavigation() {
        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNav);
        bottomNavigationView.setSelectedItemId(R.id.nav_home);

        FloatingActionButton fab = findViewById(R.id.fab);
        fab.setOnClickListener(view -> {
            startActivity(new Intent(Dashboard.this, Watering.class));
        });

        bottomNavigationView.setOnNavigationItemSelectedListener(item -> {
            if (item.getItemId() == R.id.nav_home) {
                overridePendingTransition(0, 0);
                return true;
```



```java
            } else if (item.getItemId() == R.id.nav_settings) {
                startActivity(new Intent(Dashboard.this, Setting.class));
                overridePendingTransition(0, 0);
                return true;
            }
            return false;
        });

        fab.setOnClickListener(view -> {
            Intent intent = new Intent(Dashboard.this, Watering.class);
            startActivity(intent);
            overridePendingTransition(0, 0);
        });
    }

    private void fetchMateriData() {
        MateriService materiService =
RetrofitClient.getRetrofitInstance(this).create(MateriService.class);
        Call<List<MateriResponse>> call = materiService.getMateriShow();

        call.enqueue(new Callback<List<MateriResponse>>() {
            @Override
            public void onResponse(Call<List<MateriResponse>> call, Response<List<MateriResponse>>
response) {
                if (response.isSuccessful() && response.body() != null) {
                    materiAdapter.updateList(response.body());
                } else {
                    Toast.makeText(Dashboard.this, "Gagal memuat data materi",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<List<MateriResponse>> call, Throwable t) {
                Toast.makeText(Dashboard.this, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Source Code Aplikasi

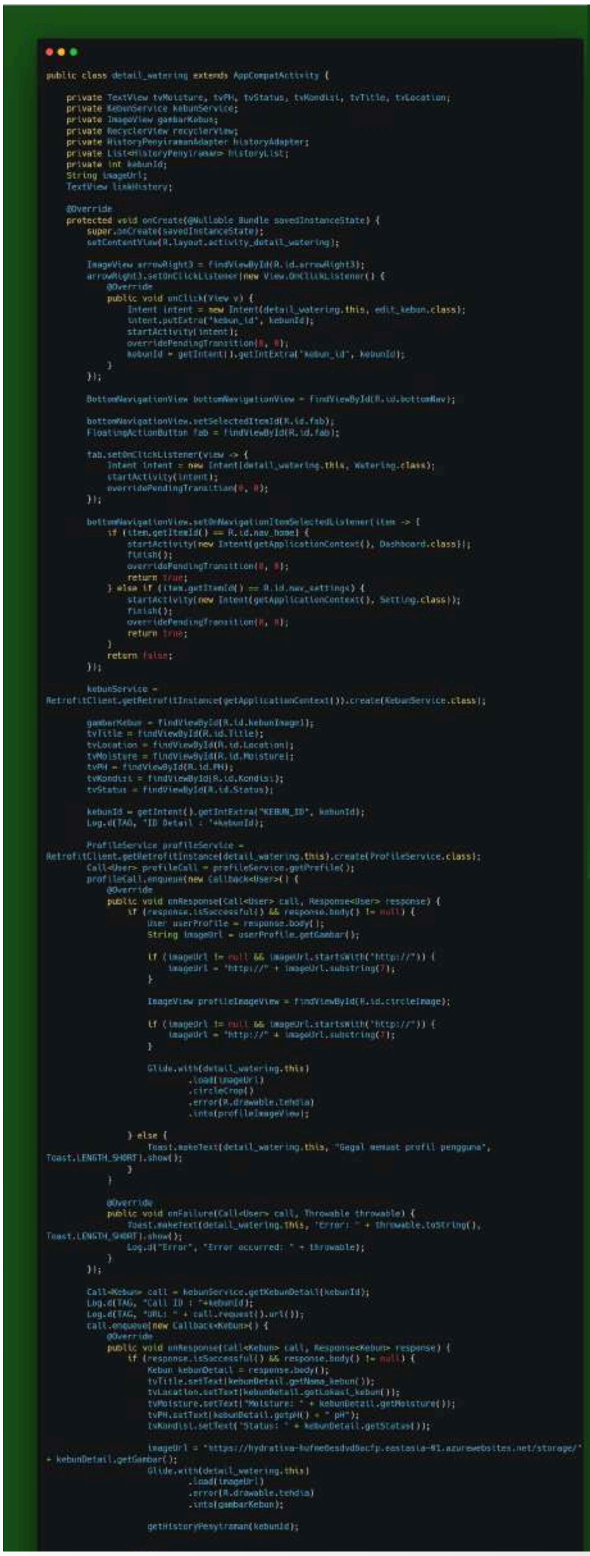
7. Kebun



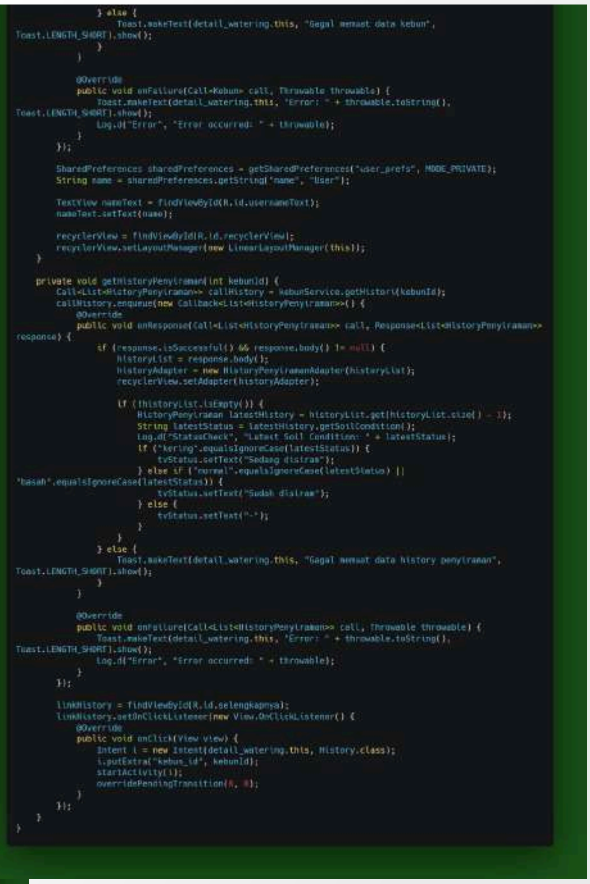
```java
public class Watering extends AppCompatActivity {

    private KebunService kebunService;
    private RecyclerView recyclerView;
    private List<Kebun> kebunList;
    Button addButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_watering);

        kebunService =
RetrofitClient.getRetrofitInstance(getApplicationContext()).create(KebunService.class);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        addButton = findViewById(R.id.addKebunButton);
        addButton.setOnClickListener(view -> {
            Intent add = new Intent(Watering.this, tambah_kebun.class);
            startActivity(add);
            overridePendingTransition(0, 0);
        });

        recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        ProfileService profileService =
RetrofitClient.getRetrofitInstance(Watering.this).create(ProfileService.class);
        Call<User> profileCall = profileService.getProfile();
        profileCall.enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                if (response.isSuccessful() && response.body() != null) {
                    User userProfile = response.body();
                    String imageUrl = userProfile.getGambar();

                    if (imageUrl != null && imageUrl.startsWith("http://")) {
                        imageUrl = "https://" + imageUrl.substring(7);
                    }

                    ImageView profileImageView = findViewById(R.id.circleImage);

                    Glide.with(Watering.this)
                            .load(imageUrl)
                            .circleCrop()
                            .error(R.drawable.tehdia)
                            .into(profileImageView);

                } else {
                    Toast.makeText(Watering.this, "Gagal memuat profil pengguna",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<User> call, Throwable throwable) {
                Toast.makeText(Watering.this, "Error: " + throwable.toString(),
Toast.LENGTH_SHORT).show();
                Log.d("Error", "Error occurred: " + throwable);
            }
        });

        Call<List<Kebun>> call = kebunService.getKebun();
        call.enqueue(new Callback<List<Kebun>>() {
            @Override
            public void onResponse(Call<List<Kebun>> call, Response<List<Kebun>> response) {
                if (response.isSuccessful() && response.body() != null) {
                    kebunList = response.body();
                    KebunAdapter kebunAdapter = new KebunAdapter(Watering.this, kebunList, kebunService);
                    recyclerView.setAdapter(kebunAdapter);
                } else {
                    Log.d("Response Error", "Response was not successful: " + response.message());
                }
            }

            @Override
            public void onFailure(Call<List<Kebun>> call, Throwable throwable) {
                Toast.makeText(Watering.this, "Error: " + throwable.toString(),
Toast.LENGTH_SHORT).show();
            }
        });

        BottomAppBar bottomAppBar = findViewById(R.id.bottomView);
        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNav);

        bottomNavigationView.setSelectedItemId(R.id.fab);
        FloatingActionButton fab = findViewById(R.id.fab);

        bottomNavigationView.setOnNavigationItemSelectedListener(item -> {
            if (item.getItemId() == R.id.nav_home) {
                startActivity(new Intent(getApplicationContext(), Dashboard.class));
                finish();
                overridePendingTransition(0, 0);
                return true;
            } else if (item.getItemId() == R.id.nav_settings) {
                startActivity(new Intent(getApplicationContext(), Setting.class));
                finish();
                overridePendingTransition(0, 0);
                return true;
            }
            return false;
        });

        fab.setOnClickListener(view -> {
            Intent intent = new Intent(Watering.this, Watering.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
            startActivity(intent);
            overridePendingTransition(0, 0);
        });

        SharedPreferences sharedPreferences = getSharedPreferences("user_prefs", MODE_PRIVATE);
        String name = sharedPreferences.getString("name", "User");

        TextView nameText = findViewById(R.id.usernameText);
        nameText.setText(name);
    }
}
```

Source Code Aplikasi

8. Pemantauan Penyiraman



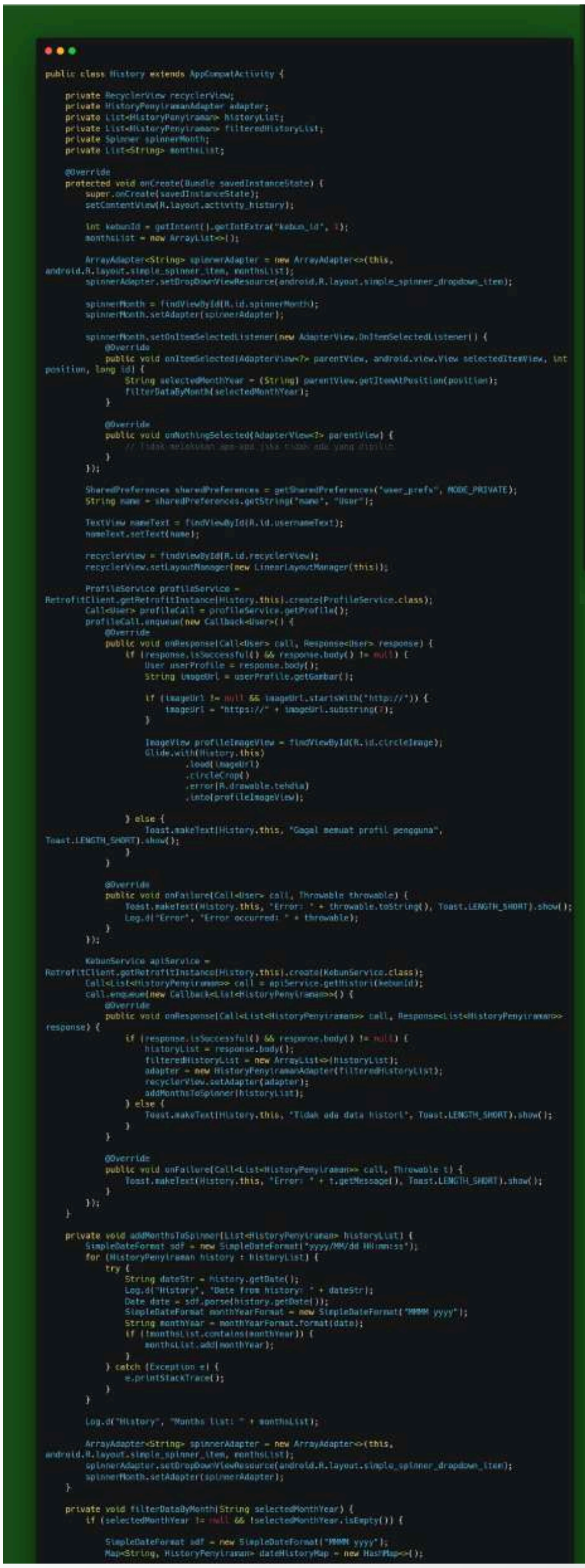
```java
public class detail_watering extends AppCompatActivity {

    private TextView tvMoisture, tvPH, tvStatus, tvKondisi, tvTitle, tvLocation;
    private KebunService kebunService;
    private ImageView gambarKebun;
    private RecyclerView recyclerView;
    private HistoryPenyiramanAdapter historyAdapter;
    private List<HistoryPenyiraman> historyList;
    private int kebunId;
    String imageUrl;
    TextView linkHistory;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_watering);

        ImageView arrowRight3 = findViewById(R.id.arrowRight3);
        arrowRight3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(detail_watering.this, edit_kebun.class);
                intent.putExtra("kebun_id", kebunId);
                startActivity(intent);
                overridePendingTransition(0, 0);
                kebunId = getIntent().getIntExtra("kebun_id", kebunId);
            }
        });

        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNav);

        bottomNavigationView.setSelectedItemId(R.id.fab);
        FloatingActionButton fab = findViewById(R.id.fab);

        fab.setOnClickListener(view -> {
            Intent intent = new Intent(detail_watering.this, Watering.class);
            startActivity(intent);
            overridePendingTransition(0, 0);
        });

        bottomNavigationView.setOnNavigationItemSelectedListener(item -> {
            if (item.getItemId() == R.id.nav_home) {
                startActivity(new Intent(getApplicationContext(), Dashboard.class));
                finish();
                overridePendingTransition(0, 0);
                return true;
            } else if (item.getItemId() == R.id.nav_settings) {
                startActivity(new Intent(getApplicationContext(), Setting.class));
                finish();
                overridePendingTransition(0, 0);
                return true;
            }
            return false;
        });

        kebunService =
RetrofitClient.getRetrofitInstance(getApplicationContext()).create(KebunService.class);

        gambarKebun = findViewById(R.id.kebunImage);
        tvTitle = findViewById(R.id.Title);
        tvLocation = findViewById(R.id.Location);
        tvMoisture = findViewById(R.id.Moisture);
        tvPH = findViewById(R.id.PH);
        tvKondisi = findViewById(R.id.Kondisi);
        tvStatus = findViewById(R.id.Status);

        kebunId = getIntent().getIntExtra("KEBUN_ID", kebunId);
        Log.d(TAG, "ID Detail : "+kebunId);

        ProfileService profileService =
RetrofitClient.getRetrofitInstance(detail_watering.this).create(ProfileService.class);
        Call<User> profileCall = profileService.getProfile();
        profileCall.enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                if (response.isSuccessful() && response.body() != null) {
                    User userProfile = response.body();
                    String imageUrl = userProfile.getGambar();

                    if (imageUrl != null && imageUrl.startsWith("http://")) {
                        imageUrl = "http://" + imageUrl.substring(7);
                    }

                    ImageView profileImageView = findViewById(R.id.circleImage);

                    if (imageUrl != null && imageUrl.startsWith("http://")) {
                        imageUrl = "http://" + imageUrl.substring(7);
                    }

                    Glide.with(detail_watering.this)
                            .load(imageUrl)
                            .circleCrop()
                            .error(R.drawable.tehdia)
                            .into(profileImageView);

                } else {
                    Toast.makeText(detail_watering.this, "Gagal memuat profil pengguna",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<User> call, Throwable throwable) {
                Toast.makeText(detail_watering.this, "Error: " + throwable.toString(),
Toast.LENGTH_SHORT).show();
                Log.d("Error", "Error occurred: " + throwable);
            }
        });

        Call<Kebun> call = kebunService.getKebunDetail(kebunId);
        Log.d(TAG, "Call ID : "+kebunId);
        Log.d(TAG, "URL: " + call.request().url());
        call.enqueue(new Callback<Kebun>() {
            @Override
            public void onResponse(Call<Kebun> call, Response<Kebun> response) {
                if (response.isSuccessful() && response.body() != null) {
                    Kebun kebunDetail = response.body();
                    tvTitle.setText(kebunDetail.getNama_kebun());
                    tvLocation.setText(kebunDetail.getLokasi_kebun());
                    tvMoisture.setText("Moisture: " + kebunDetail.getMoisture());
                    tvPH.setText(kebunDetail.getpH() + " pH");
                    tvKondisi.setText("Status: " + kebunDetail.getStatus());

                    imageUrl = "https://hydrativa-hufme0esdvd0ecfp.eastasia-01.azurewebsites.net/storage/"
+ kebunDetail.getGambar();
                    Glide.with(detail_watering.this)
                            .load(imageUrl)
                            .error(R.drawable.tehdia)
                            .into(gambarKebun);

                    getHistoryPenyiraman(kebunId);
```



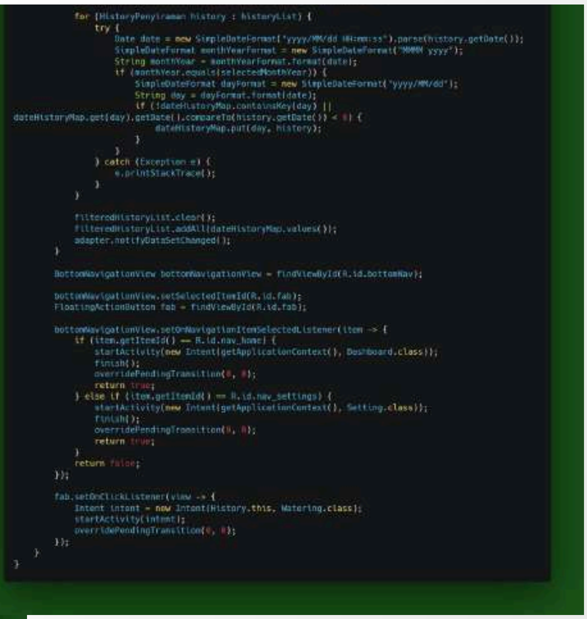
```java
                } else {
                    Toast.makeText(detail_watering.this, "Gagal memuat data kebun",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<Kebun> call, Throwable throwable) {
                Toast.makeText(detail_watering.this, "Error: " + throwable.toString(),
Toast.LENGTH_SHORT).show();
                Log.d("Error", "Error occurred: " + throwable);
            }
        });

        SharedPreferences sharedPreferences = getSharedPreferences("user_prefs", MODE_PRIVATE);
        String name = sharedPreferences.getString("name", "User");

        TextView nameText = findViewById(R.id.usernameText);
        nameText.setText(name);

        recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
    }

    private void getHistoryPenyiraman(int kebunId) {
        Call<List<HistoryPenyiraman>> callHistory = kebunService.getHistori(kebunId);
        callHistory.enqueue(new Callback<List<HistoryPenyiraman>>() {
            @Override
            public void onResponse(Call<List<HistoryPenyiraman>> call, Response<List<HistoryPenyiraman>>
response) {
                if (response.isSuccessful() && response.body() != null) {
                    historyList = response.body();
                    historyAdapter = new HistoryPenyiramanAdapter(historyList);
                    recyclerView.setAdapter(historyAdapter);

                    if (!historyList.isEmpty()) {
                        HistoryPenyiraman latestHistory = historyList.get(historyList.size() - 1);
                        String latestStatus = latestHistory.getSoilCondition();
                        Log.d("StatusCheck", "Latest Soil Condition: " + latestStatus);
                        if ("kering".equalsIgnoreCase(latestStatus)) {
                            tvStatus.setText("Sedang disiram");
                        } else if ("normal".equalsIgnoreCase(latestStatus) ||
"basah".equalsIgnoreCase(latestStatus)) {
                            tvStatus.setText("Sudah disiram");
                        } else {
                            tvStatus.setText("-");
                        }
                    }
                } else {
                    Toast.makeText(detail_watering.this, "Gagal memuat data history penyiraman",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<List<HistoryPenyiraman>> call, Throwable throwable) {
                Toast.makeText(detail_watering.this, "Error: " + throwable.toString(),
Toast.LENGTH_SHORT).show();
                Log.d("Error", "Error occurred: " + throwable);
            }
        });

        linkHistory = findViewById(R.id.selengkapnya);
        linkHistory.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(detail_watering.this, History.class);
                i.putExtra("kebun_id", kebunId);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });
    }
}
```

Source Code Aplikasi

9. History Penyiraman



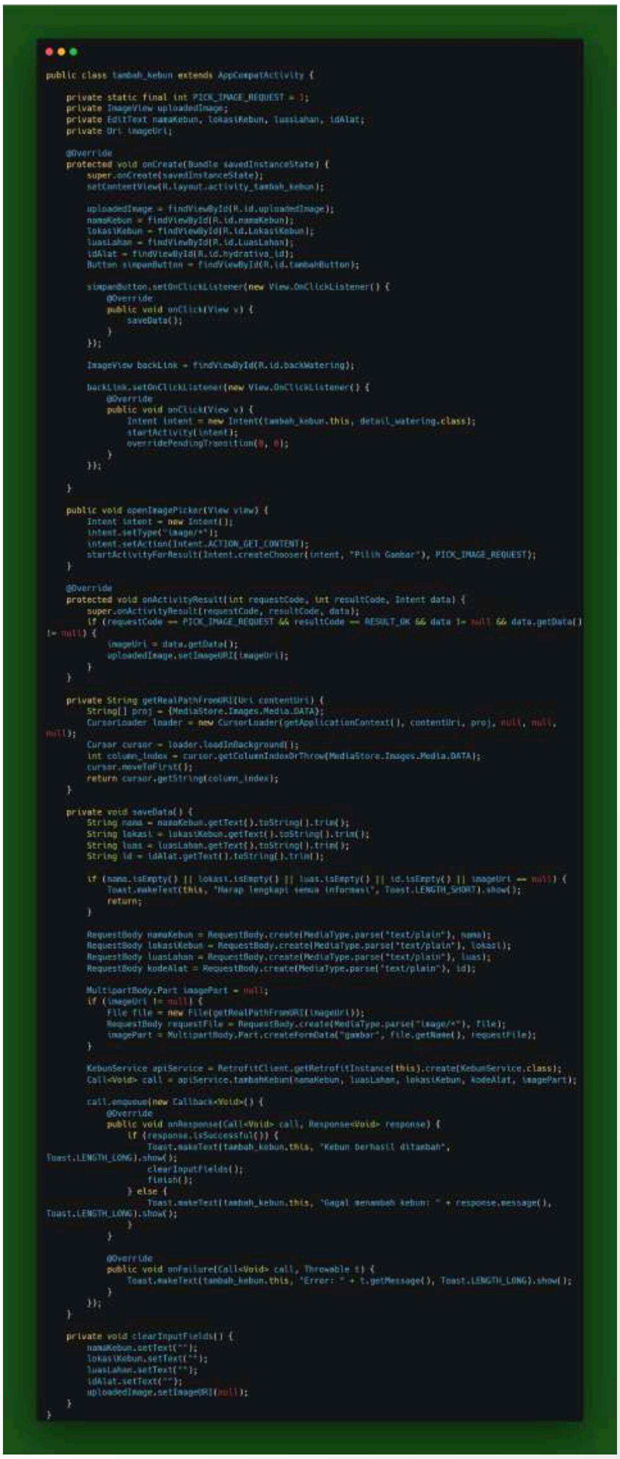
```
public class History extends AppCompatActivity {

    private RecyclerView recyclerView;
    private HistoryPenyiramanAdapter adapter;
    private List<HistoryPenyiraman> historyList;
    private List<HistoryPenyiraman> filteredHistoryList;
    private Spinner spinnerMonth;
    private List<String> monthsList;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history);

        int kebunId = getIntent().getIntExtra("kebun_id", -1);
        monthsList = new ArrayList<>();

        ArrayAdapter<String> spinnerAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item, monthsList);
        spinnerAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

        spinnerMonth = findViewById(R.id.spinnerMonth);
        spinnerMonth.setAdapter(spinnerAdapter);

        spinnerMonth.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parentView, android.view.View selectedItemView, int
position, long id) {
                String selectedMonthYear = (String) parentView.getItemAtPosition(position);
                filterDataByMonth(selectedMonthYear);
            }

            @Override
            public void onNothingSelected(AdapterView<?> parentView) {
                // Tidak melakukan apa-apa jika tidak ada yang dipilih
            }
        });

        SharedPreferences sharedPreferences = getSharedPreferences("user_prefs", MODE_PRIVATE);
        String name = sharedPreferences.getString("name", "User");

        TextView nameText = findViewById(R.id.usernameText);
        nameText.setText(name);

        recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        ProfileService profileService =
RetrofitClient.getRetrofitInstance(History.this).create(ProfileService.class);
        Call<User> profileCall = profileService.getProfile();
        profileCall.enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                if (response.isSuccessful() && response.body() != null) {
                    User userProfile = response.body();
                    String imageUrl = userProfile.getGambar();

                    if (imageUrl != null && imageUrl.startsWith("http://")) {
                        imageUrl = "https://" + imageUrl.substring(7);
                    }

                    ImageView profileImageView = findViewById(R.id.circleImage);
                    Glide.with(History.this)
                            .load(imageUrl)
                            .circleCrop()
                            .error(R.drawable.tehdia)
                            .into(profileImageView);

                } else {
                    Toast.makeText(History.this, "Gagal memuat profil pengguna",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<User> call, Throwable throwable) {
                Toast.makeText(History.this, "Error: " + throwable.toString(), Toast.LENGTH_SHORT).show();
                Log.d("Error", "Error occurred: " + throwable);
            }
        });

        KebunService apiService =
RetrofitClient.getRetrofitInstance(History.this).create(KebunService.class);
        Call<List<HistoryPenyiraman>> call = apiService.getHistori(kebunId);
        call.enqueue(new Callback<List<HistoryPenyiraman>>() {
            @Override
            public void onResponse(Call<List<HistoryPenyiraman>> call, Response<List<HistoryPenyiraman>>
response) {
                if (response.isSuccessful() && response.body() != null) {
                    historyList = response.body();
                    filteredHistoryList = new ArrayList<>(historyList);
                    adapter = new HistoryPenyiramanAdapter(filteredHistoryList);
                    recyclerView.setAdapter(adapter);
                    addMonthsToSpinner(historyList);
                } else {
                    Toast.makeText(History.this, "Tidak ada data histori", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<List<HistoryPenyiraman>> call, Throwable t) {
                Toast.makeText(History.this, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void addMonthsToSpinner(List<HistoryPenyiraman> historyList) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
        for (HistoryPenyiraman history : historyList) {
            try {
                String dateStr = history.getDate();
                Log.d("History", "Date from history: " + dateStr);
                Date date = sdf.parse(history.getDate());
                SimpleDateFormat monthYearFormat = new SimpleDateFormat("MMMM yyyy");
                String monthYear = monthYearFormat.format(date);
                if (!monthsList.contains(monthYear)) {
                    monthsList.add(monthYear);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        Log.d("History", "Months List: " + monthsList);

        ArrayAdapter<String> spinnerAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item, monthsList);
        spinnerAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerMonth.setAdapter(spinnerAdapter);
    }

    private void filterDataByMonth(String selectedMonthYear) {
        if (selectedMonthYear != null && !selectedMonthYear.isEmpty()) {

            SimpleDateFormat sdf = new SimpleDateFormat("MMMM yyyy");
            Map<String, HistoryPenyiraman> dateHistoryMap = new HashMap<>();
```



```
            for (HistoryPenyiraman history : historyList) {
                try {
                    Date date = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").parse(history.getDate());
                    SimpleDateFormat monthYearFormat = new SimpleDateFormat("MMMM yyyy");
                    String monthYear = monthYearFormat.format(date);
                    if (monthYear.equals(selectedMonthYear)) {
                        SimpleDateFormat dayFormat = new SimpleDateFormat("yyyy/MM/dd");
                        String day = dayFormat.format(date);
                        if (!dateHistoryMap.containsKey(day) ||
dateHistoryMap.get(day).getDate().compareTo(history.getDate()) < 0) {
                            dateHistoryMap.put(day, history);
                        }
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }

            filteredHistoryList.clear();
            filteredHistoryList.addAll(dateHistoryMap.values());
            adapter.notifyDataSetChanged();
        }

        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNav);

        bottomNavigationView.setSelectedItemId(R.id.fab);
        FloatingActionButton fab = findViewById(R.id.fab);

        bottomNavigationView.setOnNavigationItemSelectedListener(item -> {
            if (item.getItemId() == R.id.nav_home) {
                startActivity(new Intent(getApplicationContext(), Dashboard.class));
                finish();
                overridePendingTransition(0, 0);
                return true;
            } else if (item.getItemId() == R.id.nav_settings) {
                startActivity(new Intent(getApplicationContext(), Setting.class));
                finish();
                overridePendingTransition(0, 0);
                return true;
            }
            return false;
        });

        fab.setOnClickListener(view -> {
            Intent intent = new Intent(History.this, Watering.class);
            startActivity(intent);
            overridePendingTransition(0, 0);
        });
    }
}
```

Source Code Aplikasi

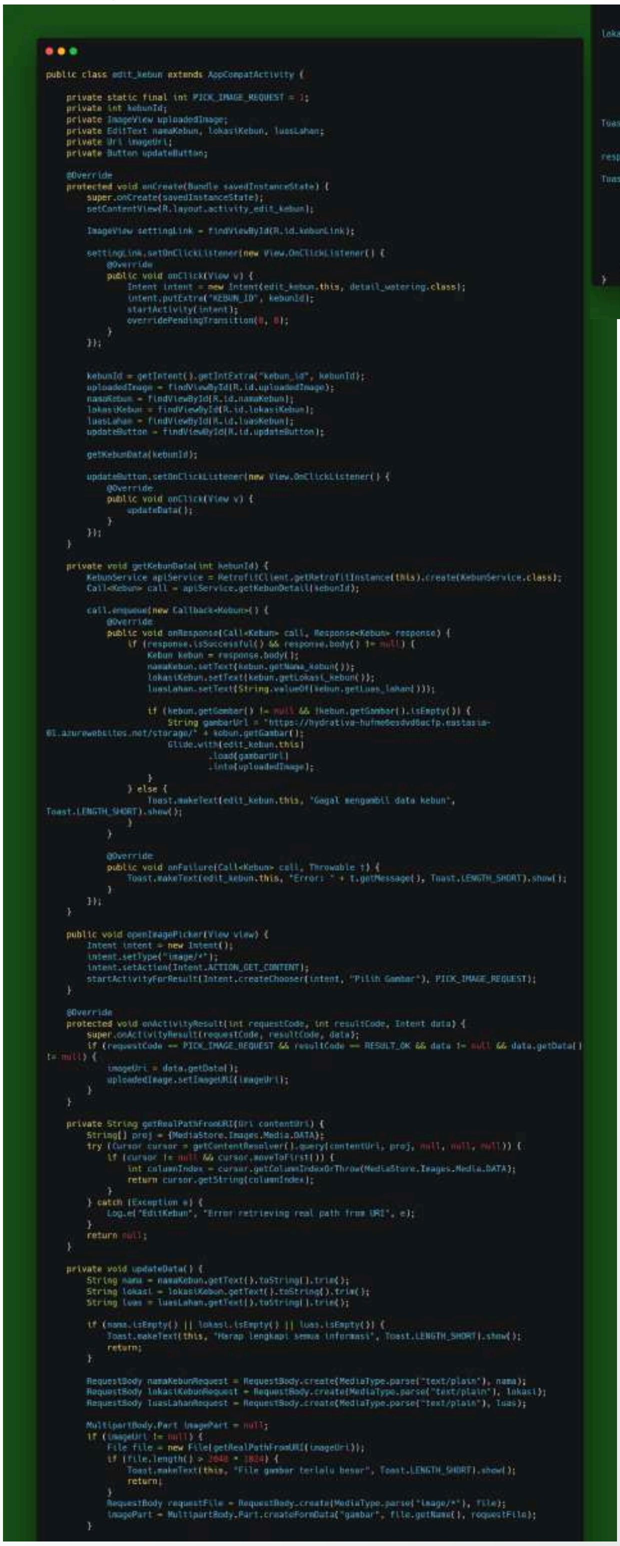
10. Tambah Kebun



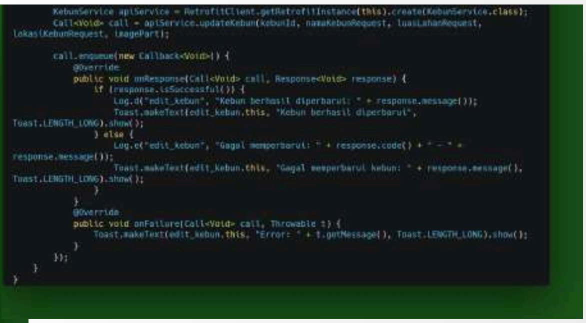
```
public class tambah_kebun extends AppCompatActivity {

    private static final int PICK_IMAGE_REQUEST = 1;
    private ImageView uploadedImage;
    private EditText namaKebun, lokasiKebun, luasLahan, idAlat;
    private Uri imageUri;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tambah_kebun);

        uploadedImage = findViewById(R.id.uploadedImage);
        namaKebun = findViewById(R.id.namaKebun);
        lokasiKebun = findViewById(R.id.LokasiKebun);
        luasLahan = findViewById(R.id.LuasLahan);
        idAlat = findViewById(R.id.hydrativa_id);
        Button simpanButton = findViewById(R.id.tambahButton);

        simpanButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                saveData();
            }
        });

        ImageView backLink = findViewById(R.id.backWatering);

        backLink.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(tambah_kebun.this, detail_watering.class);
                startActivity(intent);
                overridePendingTransition(0, 0);
            }
        });

    }

    public void openImagePicker(View view) {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Pilih Gambar"), PICK_IMAGE_REQUEST);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data != null && data.getData()
!= null) {
            imageUri = data.getData();
            uploadedImage.setImageURI(imageUri);
        }
    }

    private String getRealPathFromURI(Uri contentUri) {
        String[] proj = {MediaStore.Images.Media.DATA};
        CursorLoader loader = new CursorLoader(getApplicationContext(), contentUri, proj, null, null,
null);
        Cursor cursor = loader.loadInBackground();
        int column_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        return cursor.getString(column_index);
    }

    private void saveData() {
        String nama = namaKebun.getText().toString().trim();
        String lokasi = lokasiKebun.getText().toString().trim();
        String luas = luasLahan.getText().toString().trim();
        String id = idAlat.getText().toString().trim();

        if (nama.isEmpty() || lokasi.isEmpty() || luas.isEmpty() || id.isEmpty() || imageUri == null) {
            Toast.makeText(this, "Harap lengkapi semua informasi", Toast.LENGTH_SHORT).show();
            return;
        }

        RequestBody namaKebun = RequestBody.create(MediaType.parse("text/plain"), nama);
        RequestBody lokasiKebun = RequestBody.create(MediaType.parse("text/plain"), lokasi);
        RequestBody luasLahan = RequestBody.create(MediaType.parse("text/plain"), luas);
        RequestBody kodeAlat = RequestBody.create(MediaType.parse("text/plain"), id);

        MultipartBody.Part imagePart = null;
        if (imageUri != null) {
            File file = new File(getRealPathFromURI(imageUri));
            RequestBody requestFile = RequestBody.create(MediaType.parse("image/*"), file);
            imagePart = MultipartBody.Part.createFormData("gambar", file.getName(), requestFile);
        }

        KebunService apiService = RetrofitClient.getRetrofitInstance(this).create(KebunService.class);
        Call<Void> call = apiService.tambahKebun(namaKebun, luasLahan, lokasiKebun, kodeAlat, imagePart);

        call.enqueue(new Callback<Void>() {
            @Override
            public void onResponse(Call<Void> call, Response<Void> response) {
                if (response.isSuccessful()) {
                    Toast.makeText(tambah_kebun.this, "Kebun berhasil ditambah",
Toast.LENGTH_LONG).show();
                    clearInputFields();
                    finish();
                } else {
                    Toast.makeText(tambah_kebun.this, "Gagal menambah kebun: " + response.message(),
Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onFailure(Call<Void> call, Throwable t) {
                Toast.makeText(tambah_kebun.this, "Error: " + t.getMessage(), Toast.LENGTH_LONG).show();
            }
        });
    }

    private void clearInputFields() {
        namaKebun.setText("");
        lokasiKebun.setText("");
        luasLahan.setText("");
        idAlat.setText("");
        uploadedImage.setImageURI(null);
    }
}
```

Source Code Aplikasi

11. Sunting Kebun



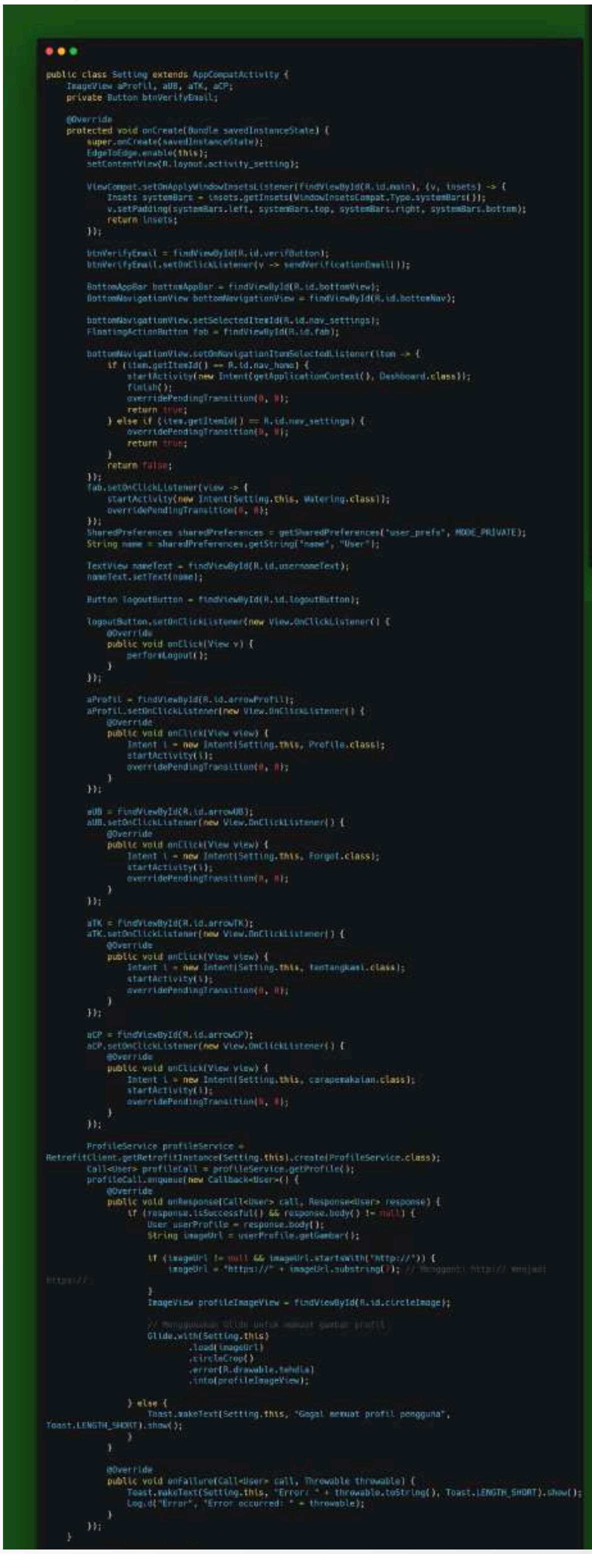
```
public class edit_kebun extends AppCompatActivity {

    private static final int PICK_IMAGE_REQUEST = 1;
    private int kebunId;
    private ImageView uploadedImage;
    private EditText namaKebun, lokasiKebun, luasLahan;
    private Uri imageUri;
    private Button updateButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_kebun);

        ImageView settingLink = findViewById(R.id.kebunLink);

        settingLink.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(edit_kebun.this, detail_watering.class);
                intent.putExtra("KEBUN_ID", kebunId);
                startActivity(intent);
                overridePendingTransition(0, 0);
            }
        });


        kebunId = getIntent().getIntExtra("kebun_id", kebunId);
        uploadedImage = findViewById(R.id.uploadedImage);
        namaKebun = findViewById(R.id.namaKebun);
        lokasiKebun = findViewById(R.id.lokasiKebun);
        luasLahan = findViewById(R.id.luasKebun);
        updateButton = findViewById(R.id.updateButton);

        getKebunData(kebunId);

        updateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                updateData();
            }
        });
    }

    private void getKebunData(int kebunId) {
        KebunService apiService = RetrofitClient.getRetrofitInstance(this).create(KebunService.class);
        Call<Kebun> call = apiService.getKebunDetail(kebunId);

        call.enqueue(new Callback<Kebun>() {
            @Override
            public void onResponse(Call<Kebun> call, Response<Kebun> response) {
                if (response.isSuccessful() && response.body() != null) {
                    Kebun kebun = response.body();
                    namaKebun.setText(kebun.getNama_kebun());
                    lokasiKebun.setText(kebun.getLokasi_kebun());
                    luasLahan.setText(String.valueOf(kebun.getLuas_lahan()));

                    if (kebun.getGambar() != null && !kebun.getGambar().isEmpty()) {
                        String gambarUrl = "https://hydrativa-hufme6esdvd6ecfp.eastasia-
01.azurewebsites.net/storage/" + kebun.getGambar();
                        Glide.with(edit_kebun.this)
                                .load(gambarUrl)
                                .into(uploadedImage);
                    }
                } else {
                    Toast.makeText(edit_kebun.this, "Gagal mengambil data kebun",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<Kebun> call, Throwable t) {
                Toast.makeText(edit_kebun.this, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    public void openImagePicker(View view) {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Pilih Gambar"), PICK_IMAGE_REQUEST);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data != null && data.getData()
!= null) {
            imageUri = data.getData();
            uploadedImage.setImageURI(imageUri);
        }
    }

    private String getRealPathFromURI(Uri contentUri) {
        String[] proj = {MediaStore.Images.Media.DATA};
        try (Cursor cursor = getContentResolver().query(contentUri, proj, null, null, null)) {
            if (cursor != null && cursor.moveToFirst()) {
                int columnIndex = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
                return cursor.getString(columnIndex);
            }
        } catch (Exception e) {
            Log.e("EditKebun", "Error retrieving real path from URI", e);
        }
        return null;
    }

    private void updateData() {
        String nama = namaKebun.getText().toString().trim();
        String lokasi = lokasiKebun.getText().toString().trim();
        String luas = luasLahan.getText().toString().trim();

        if (nama.isEmpty() || lokasi.isEmpty() || luas.isEmpty()) {
            Toast.makeText(this, "Harap lengkapi semua informasi", Toast.LENGTH_SHORT).show();
            return;
        }

        RequestBody namaKebunRequest = RequestBody.create(MediaType.parse("text/plain"), nama);
        RequestBody lokasiKebunRequest = RequestBody.create(MediaType.parse("text/plain"), lokasi);
        RequestBody luasLahanRequest = RequestBody.create(MediaType.parse("text/plain"), luas);

        MultipartBody.Part imagePart = null;
        if (imageUri != null) {
            File file = new File(getRealPathFromURI(imageUri));
            if (file.length() > 2048 * 1024) {
                Toast.makeText(this, "File gambar terlalu besar", Toast.LENGTH_SHORT).show();
                return;
            }
            RequestBody requestFile = RequestBody.create(MediaType.parse("image/*"), file);
            imagePart = MultipartBody.Part.createFormData("gambar", file.getName(), requestFile);
        }
```



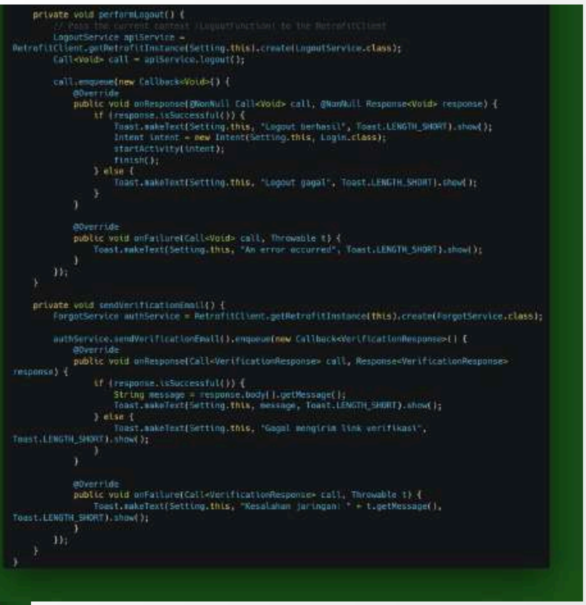
```
        KebunService apiService = RetrofitClient.getRetrofitInstance(this).create(KebunService.class);
        Call<Void> call = apiService.updateKebun(kebunId, namaKebunRequest, luasLahanRequest,
lokasiKebunRequest, imagePart);

        call.enqueue(new Callback<Void>() {
            @Override
            public void onResponse(Call<Void> call, Response<Void> response) {
                if (response.isSuccessful()) {
                    Log.d("edit_kebun", "Kebun berhasil diperbarui: " + response.message());
                    Toast.makeText(edit_kebun.this, "Kebun berhasil diperbarui",
Toast.LENGTH_LONG).show();
                } else {
                    Log.e("edit_kebun", "Gagal memperbarui: " + response.code() + " - " +
response.message());
                    Toast.makeText(edit_kebun.this, "Gagal memperbarui kebun: " + response.message(),
Toast.LENGTH_LONG).show();
                }
            }
            @Override
            public void onFailure(Call<Void> call, Throwable t) {
                Toast.makeText(edit_kebun.this, "Error: " + t.getMessage(), Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Source Code Aplikasi

12. Pengaturan



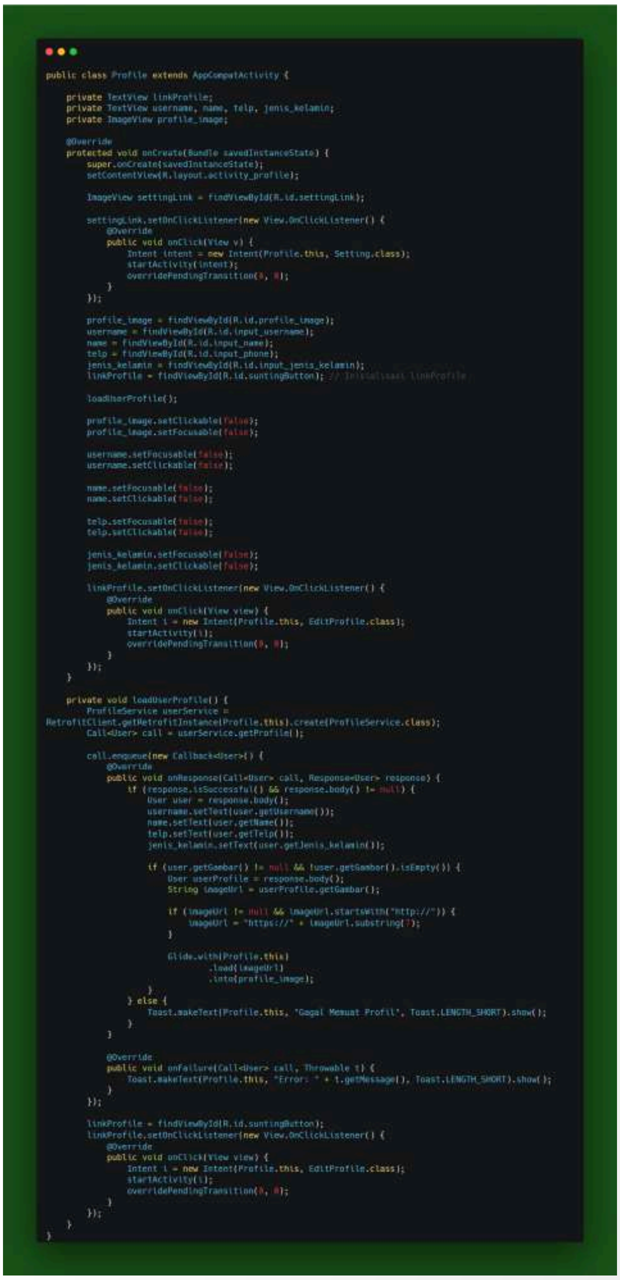
```java
public class Setting extends AppCompatActivity {
    ImageView aProfil, aUB, aTK, aCP;
    private Button btnVerifyEmail;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_setting);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        btnVerifyEmail = findViewById(R.id.verifButton);
        btnVerifyEmail.setOnClickListener(v -> sendVerificationEmail());

        BottomAppBar bottomAppBar = findViewById(R.id.bottomView);
        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNav);

        bottomNavigationView.setSelectedItemId(R.id.nav_settings);
        FloatingActionButton fab = findViewById(R.id.fab);

        bottomNavigationView.setOnNavigationItemSelectedListener(item -> {
            if (item.getItemId() == R.id.nav_home) {
                startActivity(new Intent(getApplicationContext(), Dashboard.class));
                finish();
                overridePendingTransition(0, 0);
                return true;
            } else if (item.getItemId() == R.id.nav_settings) {
                overridePendingTransition(0, 0);
                return true;
            }
            return false;
        });
        fab.setOnClickListener(view -> {
            startActivity(new Intent(Setting.this, Watering.class));
            overridePendingTransition(0, 0);
        });
        SharedPreferences sharedPreferences = getSharedPreferences("user_prefs", MODE_PRIVATE);
        String name = sharedPreferences.getString("name", "User");

        TextView nameText = findViewById(R.id.usernameText);
        nameText.setText(name);

        Button logoutButton = findViewById(R.id.logoutButton);

        logoutButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performLogout();
            }
        });

        aProfil = findViewById(R.id.arrowProfil);
        aProfil.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Setting.this, Profile.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });

        aUB = findViewById(R.id.arrowUB);
        aUB.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Setting.this, Forgot.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });

        aTK = findViewById(R.id.arrowTK);
        aTK.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Setting.this, tentangkami.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });

        aCP = findViewById(R.id.arrowCP);
        aCP.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Setting.this, carapemakaian.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });

        ProfileService profileService =
RetrofitClient.getRetrofitInstance(Setting.this).create(ProfileService.class);
        Call<User> profileCall = profileService.getProfile();
        profileCall.enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                if (response.isSuccessful() && response.body() != null) {
                    User userProfile = response.body();
                    String imageUrl = userProfile.getGambar();

                    if (imageUrl != null && imageUrl.startsWith("http://")) {
                        imageUrl = "https://" + imageUrl.substring(7); // Mengganti http:// menjadi
https://
                    }
                    ImageView profileImageView = findViewById(R.id.circleImage);

                    // Menggunakan Glide untuk memuat gambar profil
                    Glide.with(Setting.this)
                            .load(imageUrl)
                            .circleCrop()
                            .error(R.drawable.tehdia)
                            .into(profileImageView);

                } else {
                    Toast.makeText(Setting.this, "Gagal memuat profil pengguna",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<User> call, Throwable throwable) {
                Toast.makeText(Setting.this, "Error: " + throwable.toString(), Toast.LENGTH_SHORT).show();
                Log.d("Error", "Error occurred: " + throwable);
            }
        });
    }
```



```java
    private void performLogout() {
        // Pass the current context (LogoutFunction) to the RetrofitClient
        LogoutService apiService =
RetrofitClient.getRetrofitInstance(Setting.this).create(LogoutService.class);
        Call<Void> call = apiService.logout();

        call.enqueue(new Callback<Void>() {
            @Override
            public void onResponse(@NonNull Call<Void> call, @NonNull Response<Void> response) {
                if (response.isSuccessful()) {
                    Toast.makeText(Setting.this, "Logout berhasil", Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(Setting.this, Login.class);
                    startActivity(intent);
                    finish();
                } else {
                    Toast.makeText(Setting.this, "Logout gagal", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<Void> call, Throwable t) {
                Toast.makeText(Setting.this, "An error occurred", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void sendVerificationEmail() {
        ForgotService authService = RetrofitClient.getRetrofitInstance(this).create(ForgotService.class);

        authService.sendVerificationEmail().enqueue(new Callback<VerificationResponse>() {
            @Override
            public void onResponse(Call<VerificationResponse> call, Response<VerificationResponse>
response) {
                if (response.isSuccessful()) {
                    String message = response.body().getMessage();
                    Toast.makeText(Setting.this, message, Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(Setting.this, "Gagal mengirim link verifikasi",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<VerificationResponse> call, Throwable t) {
                Toast.makeText(Setting.this, "Kesalahan jaringan: " + t.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Source Code Aplikasi

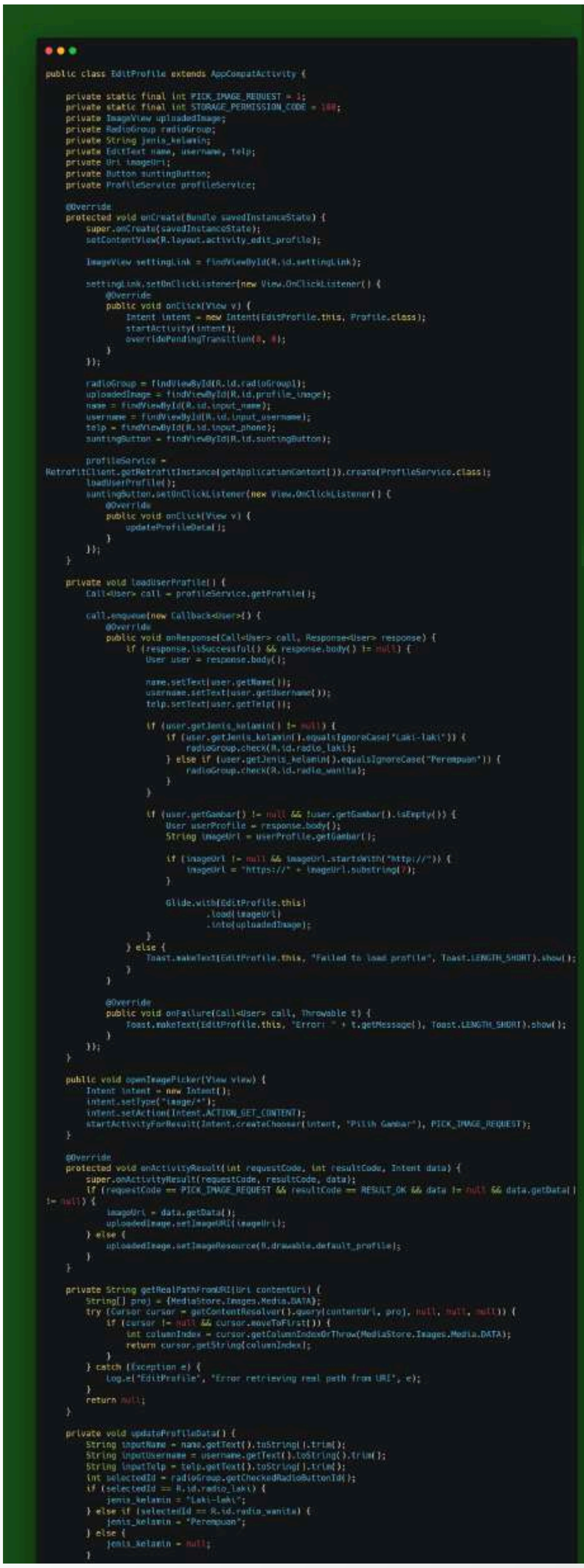
13. Profil Pengguna



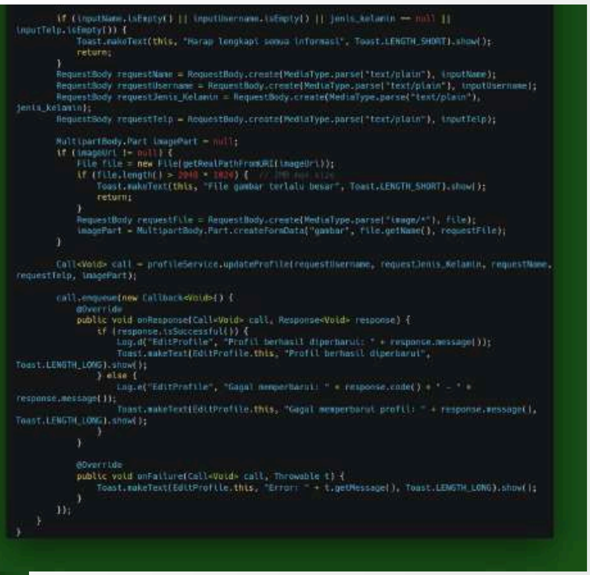
```java
public class Profile extends AppCompatActivity {

    private TextView linkProfile;
    private TextView username, name, telp, jenis_kelamin;
    private ImageView profile_image;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);

        ImageView settingLink = findViewById(R.id.settingLink);

        settingLink.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Profile.this, Setting.class);
                startActivity(intent);
                overridePendingTransition(0, 0);
            }
        });

        profile_image = findViewById(R.id.profile_image);
        username = findViewById(R.id.input_username);
        name = findViewById(R.id.input_name);
        telp = findViewById(R.id.input_phone);
        jenis_kelamin = findViewById(R.id.input_jenis_kelamin);
        linkProfile = findViewById(R.id.suntingButton); // Inisialisasi linkProfile

        loadUserProfile();

        profile_image.setClickable(false);
        profile_image.setFocusable(false);

        username.setFocusable(false);
        username.setClickable(false);

        name.setFocusable(false);
        name.setClickable(false);

        telp.setFocusable(false);
        telp.setClickable(false);

        jenis_kelamin.setFocusable(false);
        jenis_kelamin.setClickable(false);

        linkProfile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Profile.this, EditProfile.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });
    }

    private void loadUserProfile() {
        ProfileService userService =
RetrofitClient.getRetrofitInstance(Profile.this).create(ProfileService.class);
        Call<User> call = userService.getProfile();

        call.enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                if (response.isSuccessful() && response.body() != null) {
                    User user = response.body();
                    username.setText(user.getUsername());
                    name.setText(user.getName());
                    telp.setText(user.getTelp());
                    jenis_kelamin.setText(user.getJenis_kelamin());

                    if (user.getGambar() != null && !user.getGambar().isEmpty()) {
                        User userProfile = response.body();
                        String imageUrl = userProfile.getGambar();

                        if (imageUrl != null && imageUrl.startsWith("http://")) {
                            imageUrl = "https://" + imageUrl.substring(7);
                        }

                        Glide.with(Profile.this)
                                .load(imageUrl)
                                .into(profile_image);
                    }
                } else {
                    Toast.makeText(Profile.this, "Gagal Memuat Profil", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<User> call, Throwable t) {
                Toast.makeText(Profile.this, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });

        linkProfile = findViewById(R.id.suntingButton);
        linkProfile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Profile.this, EditProfile.class);
                startActivity(i);
                overridePendingTransition(0, 0);
            }
        });
    }
}
```

Source Code Aplikasi

14. Sunting Profil



```java
public class EditProfile extends AppCompatActivity {

    private static final int PICK_IMAGE_REQUEST = 1;
    private static final int STORAGE_PERMISSION_CODE = 100;
    private ImageView uploadedImage;
    private RadioGroup radioGroup;
    private String jenis_kelamin;
    private EditText name, username, telp;
    private Uri imageUri;
    private Button suntingButton;
    private ProfileService profileService;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_profile);

        ImageView settingLink = findViewById(R.id.settingLink);

        settingLink.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(EditProfile.this, Profile.class);
                startActivity(intent);
                overridePendingTransition(0, 0);
            }
        });

        radioGroup = findViewById(R.id.radioGroup1);
        uploadedImage = findViewById(R.id.profile_image);
        name = findViewById(R.id.input_name);
        username = findViewById(R.id.input_username);
        telp = findViewById(R.id.input_phone);
        suntingButton = findViewById(R.id.suntingButton);

        profileService =
RetrofitClient.getRetrofitInstance(getApplicationContext()).create(ProfileService.class);
        loadUserProfile();
        suntingButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                updateProfileData();
            }
        });
    }

    private void loadUserProfile() {
        Call<User> call = profileService.getProfile();

        call.enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                if (response.isSuccessful() && response.body() != null) {
                    User user = response.body();

                    name.setText(user.getName());
                    username.setText(user.getUsername());
                    telp.setText(user.getTelp());

                    if (user.getJenis_kelamin() != null) {
                        if (user.getJenis_kelamin().equalsIgnoreCase("Laki-laki")) {
                            radioGroup.check(R.id.radio_laki);
                        } else if (user.getJenis_kelamin().equalsIgnoreCase("Perempuan")) {
                            radioGroup.check(R.id.radio_wanita);
                        }
                    }

                    if (user.getGambar() != null && !user.getGambar().isEmpty()) {
                        User userProfile = response.body();
                        String imageUrl = userProfile.getGambar();

                        if (imageUrl != null && imageUrl.startsWith("http://")) {
                            imageUrl = "https://" + imageUrl.substring(7);
                        }

                        Glide.with(EditProfile.this)
                                .load(imageUrl)
                                .into(uploadedImage);
                    }
                } else {
                    Toast.makeText(EditProfile.this, "Failed to load profile", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<User> call, Throwable t) {
                Toast.makeText(EditProfile.this, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    public void openImagePicker(View view) {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Pilih Gambar"), PICK_IMAGE_REQUEST);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data != null && data.getData()
!= null) {
            imageUri = data.getData();
            uploadedImage.setImageURI(imageUri);
        } else {
            uploadedImage.setImageResource(R.drawable.default_profile);
        }
    }

    private String getRealPathFromURI(Uri contentUri) {
        String[] proj = {MediaStore.Images.Media.DATA};
        try (Cursor cursor = getContentResolver().query(contentUri, proj, null, null, null)) {
            if (cursor != null && cursor.moveToFirst()) {
                int columnIndex = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
                return cursor.getString(columnIndex);
            }
        } catch (Exception e) {
            Log.e("EditProfile", "Error retrieving real path from URI", e);
        }
        return null;
    }

    private void updateProfileData() {
        String inputName = name.getText().toString().trim();
        String inputUsername = username.getText().toString().trim();
        String inputTelp = telp.getText().toString().trim();
        int selectedId = radioGroup.getCheckedRadioButtonId();
        if (selectedId == R.id.radio_laki) {
            jenis_kelamin = "Laki-laki";
        } else if (selectedId == R.id.radio_wanita) {
            jenis_kelamin = "Perempuan";
        } else {
            jenis_kelamin = null;
        }
```



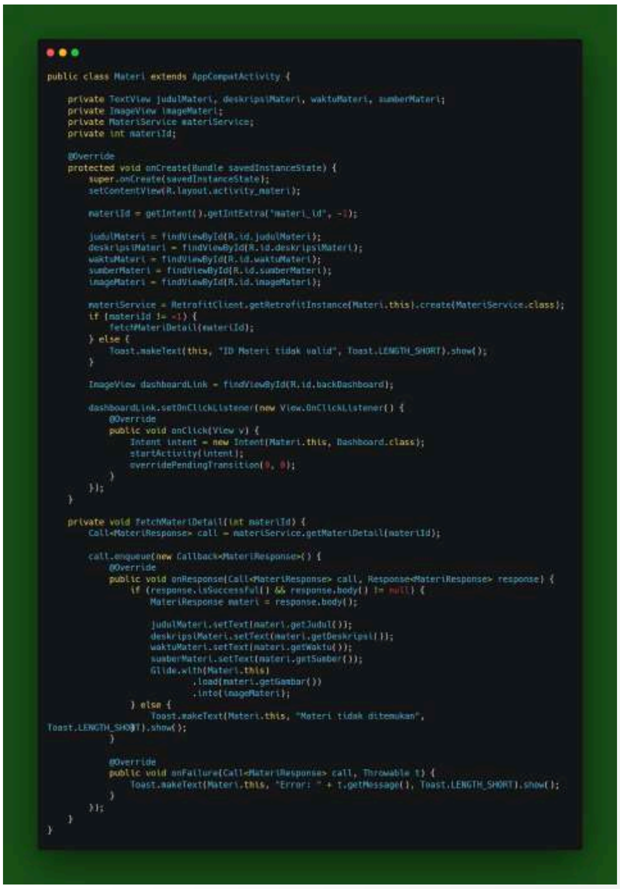
```java
        if (inputName.isEmpty() || inputUsername.isEmpty() || jenis_kelamin == null ||
inputTelp.isEmpty()) {
            Toast.makeText(this, "Harap lengkapi semua informasi", Toast.LENGTH_SHORT).show();
            return;
        }
        RequestBody requestName = RequestBody.create(MediaType.parse("text/plain"), inputName);
        RequestBody requestUsername = RequestBody.create(MediaType.parse("text/plain"), inputUsername);
        RequestBody requestJenis_Kelamin = RequestBody.create(MediaType.parse("text/plain"),
jenis_kelamin);
        RequestBody requestTelp = RequestBody.create(MediaType.parse("text/plain"), inputTelp);

        MultipartBody.Part imagePart = null;
        if (imageUri != null) {
            File file = new File(getRealPathFromURI(imageUri));
            if (file.length() > 2048 * 1024) { // 2MB max size
                Toast.makeText(this, "File gambar terlalu besar", Toast.LENGTH_SHORT).show();
                return;
            }
            RequestBody requestFile = RequestBody.create(MediaType.parse("image/*"), file);
            imagePart = MultipartBody.Part.createFormData("gambar", file.getName(), requestFile);
        }

        Call<Void> call = profileService.updateProfile(requestUsername, requestJenis_Kelamin, requestName,
requestTelp, imagePart);

        call.enqueue(new Callback<Void>() {
            @Override
            public void onResponse(Call<Void> call, Response<Void> response) {
                if (response.isSuccessful()) {
                    Log.d("EditProfile", "Profil berhasil diperbarui: " + response.message());
                    Toast.makeText(EditProfile.this, "Profil berhasil diperbarui",
Toast.LENGTH_LONG).show();
                } else {
                    Log.e("EditProfile", "Gagal memperbarui: " + response.code() + " - " +
response.message());
                    Toast.makeText(EditProfile.this, "Gagal memperbarui profil: " + response.message(),
Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onFailure(Call<Void> call, Throwable t) {
                Toast.makeText(EditProfile.this, "Error: " + t.getMessage(), Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Source Code Aplikasi

15. Verifikasi Akun

Source Code Aplikasi

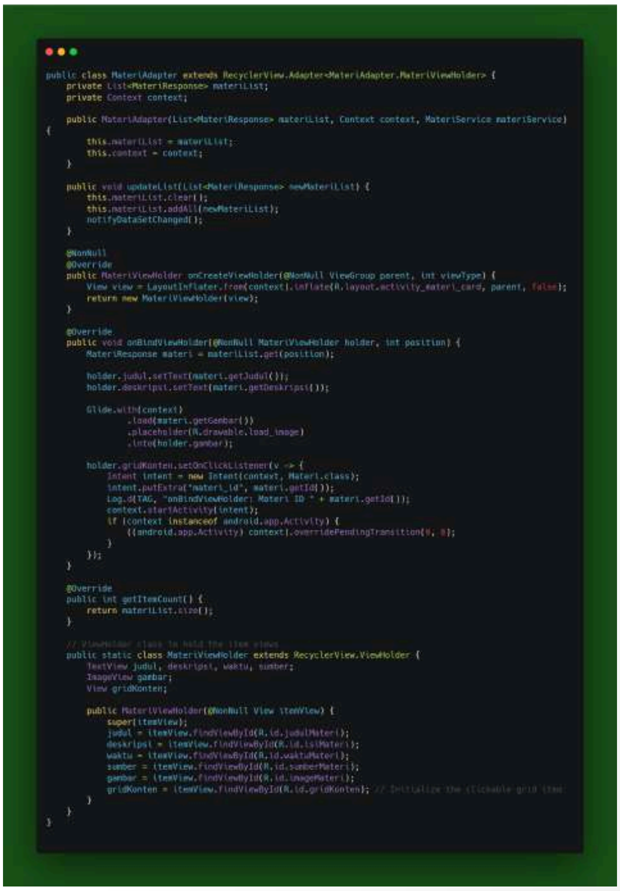
16. Edukasi (Materi)



```java
public class Materi extends AppCompatActivity {

    private TextView judulMateri, deskripsiMateri, waktuMateri, sumberMateri;
    private ImageView imageMateri;
    private MateriService materiService;
    private int materiId;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_materi);

        materiId = getIntent().getIntExtra("materi_id", -1);

        judulMateri = findViewById(R.id.judulMateri);
        deskripsiMateri = findViewById(R.id.deskripsiMateri);
        waktuMateri = findViewById(R.id.waktuMateri);
        sumberMateri = findViewById(R.id.sumberMateri);
        imageMateri = findViewById(R.id.imageMateri);

        materiService = RetrofitClient.getRetrofitInstance(Materi.this).create(MateriService.class);
        if (materiId != -1) {
            fetchMateriDetail(materiId);
        } else {
            Toast.makeText(this, "ID Materi tidak valid", Toast.LENGTH_SHORT).show();
        }

        ImageView dashboardLink = findViewById(R.id.backDashboard);

        dashboardLink.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Materi.this, Dashboard.class);
                startActivity(intent);
                overridePendingTransition(0, 0);
            }
        });
    }

    private void fetchMateriDetail(int materiId) {
        Call<MateriResponse> call = materiService.getMateriDetail(materiId);

        call.enqueue(new Callback<MateriResponse>() {
            @Override
            public void onResponse(Call<MateriResponse> call, Response<MateriResponse> response) {
                if (response.isSuccessful() && response.body() != null) {
                    MateriResponse materi = response.body();

                    judulMateri.setText(materi.getJudul());
                    deskripsiMateri.setText(materi.getDeskripsi());
                    waktuMateri.setText(materi.getWaktu());
                    sumberMateri.setText(materi.getSumber());
                    Glide.with(Materi.this)
                            .load(materi.getGambar())
                            .into(imageMateri);
                } else {
                    Toast.makeText(Materi.this, "Materi tidak ditemukan",
Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onFailure(Call<MateriResponse> call, Throwable t) {
                Toast.makeText(Materi.this, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Source Code Aplikasi

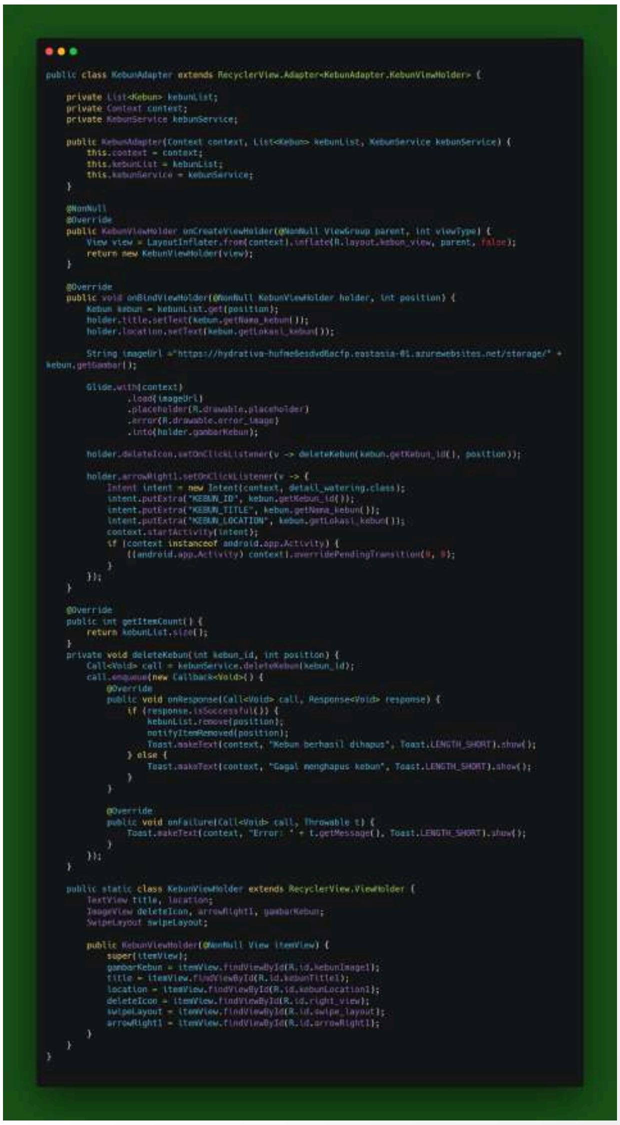
17. Adapter Materi



```
public class MateriAdapter extends RecyclerView.Adapter<MateriAdapter.MateriViewHolder> {
    private List<MateriResponse> materiList;
    private Context context;

    public MateriAdapter(List<MateriResponse> materiList, Context context, MateriService materiService)
{
        this.materiList = materiList;
        this.context = context;
    }

    public void updateList(List<MateriResponse> newMateriList) {
        this.materiList.clear();
        this.materiList.addAll(newMateriList);
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public MateriViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.activity_materi_card, parent, false);
        return new MateriViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MateriViewHolder holder, int position) {
        MateriResponse materi = materiList.get(position);

        holder.judul.setText(materi.getJudul());
        holder.deskripsi.setText(materi.getDeskripsi());

        Glide.with(context)
                .load(materi.getGambar())
                .placeholder(R.drawable.load_image)
                .into(holder.gambar);

        holder.gridKonten.setOnClickListener(v -> {
            Intent intent = new Intent(context, Materi.class);
            intent.putExtra("materi_id", materi.getId());
            Log.d(TAG, "onBindViewHolder: Materi ID " + materi.getId());
            context.startActivity(intent);
            if (context instanceof android.app.Activity) {
                ((android.app.Activity) context).overridePendingTransition(0, 0);
            }
        });
    }

    @Override
    public int getItemCount() {
        return materiList.size();
    }

    // ViewHolder class to hold the item views
    public static class MateriViewHolder extends RecyclerView.ViewHolder {
        TextView judul, deskripsi, waktu, sumber;
        ImageView gambar;
        View gridKonten;

        public MateriViewHolder(@NonNull View itemView) {
            super(itemView);
            judul = itemView.findViewById(R.id.judulMateri);
            deskripsi = itemView.findViewById(R.id.isiMateri);
            waktu = itemView.findViewById(R.id.waktuMateri);
            sumber = itemView.findViewById(R.id.sumberMateri);
            gambar = itemView.findViewById(R.id.imageMateri);
            gridKonten = itemView.findViewById(R.id.gridKonten); // Initialize the clickable grid item
        }
    }
}
```

Source Code Aplikasi

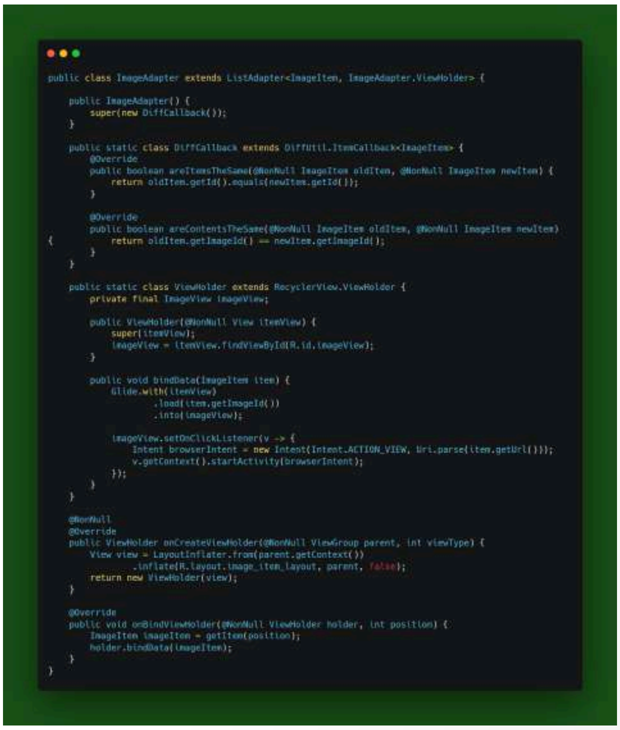
18. Adapter Kebun



```java
public class KebunAdapter extends RecyclerView.Adapter<KebunAdapter.KebunViewHolder> {

    private List<Kebun> kebunList;
    private Context context;
    private KebunService kebunService;

    public KebunAdapter(Context context, List<Kebun> kebunList, KebunService kebunService) {
        this.context = context;
        this.kebunList = kebunList;
        this.kebunService = kebunService;
    }

    @NonNull
    @Override
    public KebunViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.kebun_view, parent, false);
        return new KebunViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull KebunViewHolder holder, int position) {
        Kebun kebun = kebunList.get(position);
        holder.title.setText(kebun.getNama_kebun());
        holder.location.setText(kebun.getLokasi_kebun());

        String imageUrl ="https://hydrativa-hufme6esdvd8acfp.eastasia-01.azurewebsites.net/storage/" +
kebun.getGambar();

        Glide.with(context)
                .load(imageUrl)
                .placeholder(R.drawable.placeholder)
                .error(R.drawable.error_image)
                .into(holder.gambarKebun);

        holder.deleteIcon.setOnClickListener(v -> deleteKebun(kebun.getKebun_id(), position));

        holder.arrowRight1.setOnClickListener(v -> {
            Intent intent = new Intent(context, detail_watering.class);
            intent.putExtra("KEBUN_ID", kebun.getKebun_id());
            intent.putExtra("KEBUN_TITLE", kebun.getNama_kebun());
            intent.putExtra("KEBUN_LOCATION", kebun.getLokasi_kebun());
            context.startActivity(intent);
            if (context instanceof android.app.Activity) {
                ((android.app.Activity) context).overridePendingTransition(0, 0);
            }
        });
    }

    @Override
    public int getItemCount() {
        return kebunList.size();
    }
    private void deleteKebun(int kebun_id, int position) {
        Call<Void> call = kebunService.deleteKebun(kebun_id);
        call.enqueue(new Callback<Void>() {
            @Override
            public void onResponse(Call<Void> call, Response<Void> response) {
                if (response.isSuccessful()) {
                    kebunList.remove(position);
                    notifyItemRemoved(position);
                    Toast.makeText(context, "Kebun berhasil dihapus", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(context, "Gagal menghapus kebun", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(Call<Void> call, Throwable t) {
                Toast.makeText(context, "Error: " + t.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    public static class KebunViewHolder extends RecyclerView.ViewHolder {
        TextView title, location;
        ImageView deleteIcon, arrowRight1, gambarKebun;
        SwipeLayout swipeLayout;

        public KebunViewHolder(@NonNull View itemView) {
            super(itemView);
            gambarKebun = itemView.findViewById(R.id.kebunImage1);
            title = itemView.findViewById(R.id.kebunTitle1);
            location = itemView.findViewById(R.id.kebunLocation1);
            deleteIcon = itemView.findViewById(R.id.right_view);
            swipeLayout = itemView.findViewById(R.id.swipe_layout);
            arrowRight1 = itemView.findViewById(R.id.arrowRight1);
        }
    }
}
```

Source Code Aplikasi

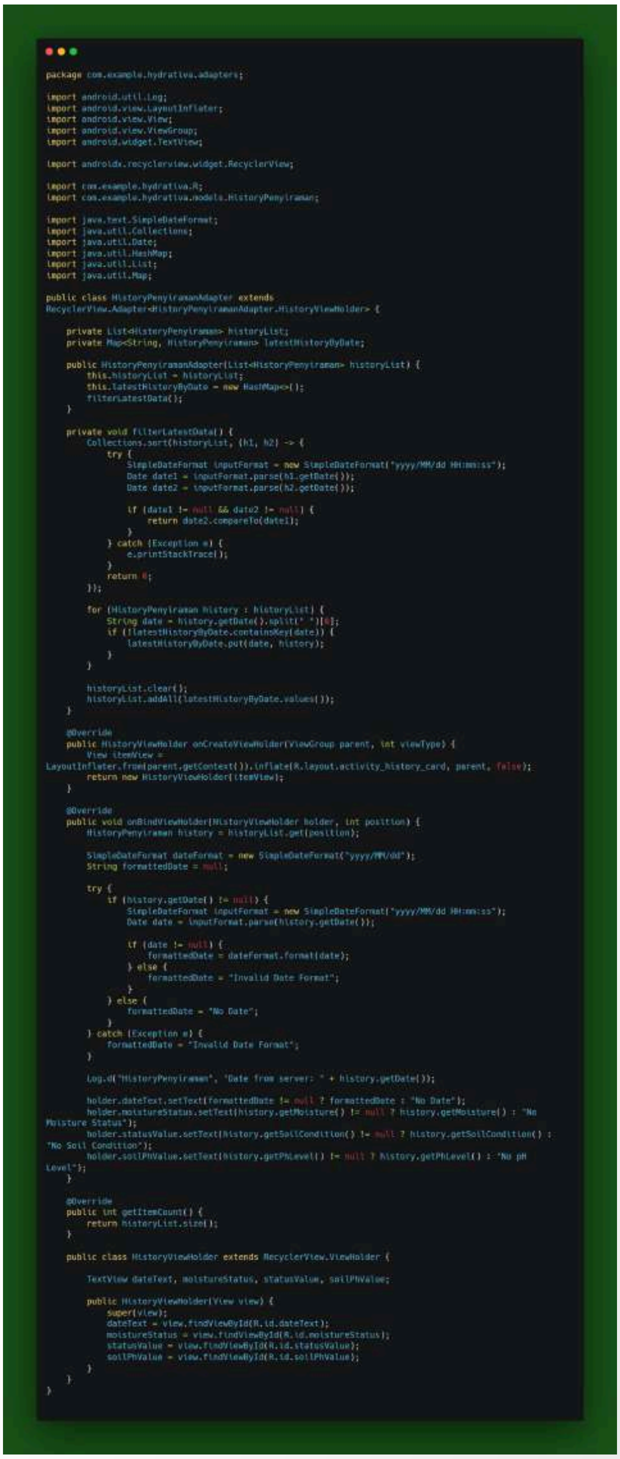
19. Adapter Gambar



```
public class ImageAdapter extends ListAdapter<ImageItem, ImageAdapter.ViewHolder> {

    public ImageAdapter() {
        super(new DiffCallback());
    }

    public static class DiffCallback extends DiffUtil.ItemCallback<ImageItem> {
        @Override
        public boolean areItemsTheSame(@NonNull ImageItem oldItem, @NonNull ImageItem newItem) {
            return oldItem.getId().equals(newItem.getId());
        }

        @Override
        public boolean areContentsTheSame(@NonNull ImageItem oldItem, @NonNull ImageItem newItem)
{
            return oldItem.getImageId() == newItem.getImageId();
        }
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        private final ImageView imageView;

        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            imageView = itemView.findViewById(R.id.imageView);
        }

        public void bindData(ImageItem item) {
            Glide.with(itemView)
                    .load(item.getImageId())
                    .into(imageView);

            imageView.setOnClickListener(v -> {
                Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(item.getUrl()));
                v.getContext().startActivity(browserIntent);
            });
        }
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.image_item_layout, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        ImageItem imageItem = getItem(position);
        holder.bindData(imageItem);
    }
}
```

Source Code Aplikasi

20. Adapter Histori

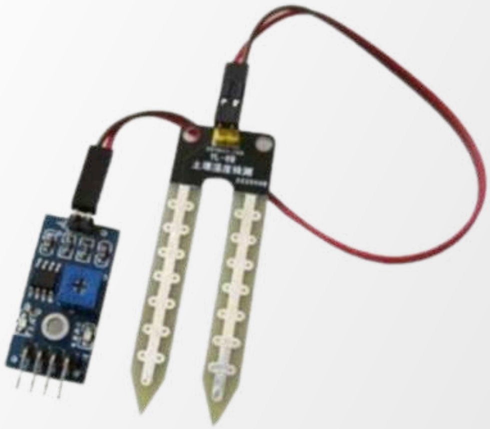


```java
package com.example.hydrativa.adapters;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.example.hydrativa.R;
import com.example.hydrativa.models.HistoryPenyiraman;

import java.text.SimpleDateFormat;
import java.util.Collections;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class HistoryPenyiramanAdapter extends
RecyclerView.Adapter<HistoryPenyiramanAdapter.HistoryViewHolder> {

    private List<HistoryPenyiraman> historyList;
    private Map<String, HistoryPenyiraman> latestHistoryByDate;

    public HistoryPenyiramanAdapter(List<HistoryPenyiraman> historyList) {
        this.historyList = historyList;
        this.latestHistoryByDate = new HashMap<>();
        filterLatestData();
    }

    private void filterLatestData() {
        Collections.sort(historyList, (h1, h2) -> {
            try {
                SimpleDateFormat inputFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
                Date date1 = inputFormat.parse(h1.getDate());
                Date date2 = inputFormat.parse(h2.getDate());

                if (date1 != null && date2 != null) {
                    return date2.compareTo(date1);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
            return 0;
        });

        for (HistoryPenyiraman history : historyList) {
            String date = history.getDate().split(" ")[0];
            if (!latestHistoryByDate.containsKey(date)) {
                latestHistoryByDate.put(date, history);
            }
        }

        historyList.clear();
        historyList.addAll(latestHistoryByDate.values());
    }

    @Override
    public HistoryViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.activity_history_card, parent, false);
        return new HistoryViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(HistoryViewHolder holder, int position) {
        HistoryPenyiraman history = historyList.get(position);

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");
        String formattedDate = null;

        try {
            if (history.getDate() != null) {
                SimpleDateFormat inputFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
                Date date = inputFormat.parse(history.getDate());

                if (date != null) {
                    formattedDate = dateFormat.format(date);
                } else {
                    formattedDate = "Invalid Date Format";
                }
            } else {
                formattedDate = "No Date";
            }
        } catch (Exception e) {
            formattedDate = "Invalid Date Format";
        }

        Log.d("HistoryPenyiraman", "Date from server: " + history.getDate());

        holder.dateText.setText(formattedDate != null ? formattedDate : "No Date");
        holder.moistureStatus.setText(history.getMoisture() != null ? history.getMoisture() : "No
Moisture Status");
        holder.statusValue.setText(history.getSoilCondition() != null ? history.getSoilCondition() :
"No Soil Condition");
        holder.soilPhValue.setText(history.getPhLevel() != null ? history.getPhLevel() : "No pH
Level");
    }

    @Override
    public int getItemCount() {
        return historyList.size();
    }

    public class HistoryViewHolder extends RecyclerView.ViewHolder {

        TextView dateText, moistureStatus, statusValue, soilPhValue;

        public HistoryViewHolder(View view) {
            super(view);
            dateText = view.findViewById(R.id.dateText);
            moistureStatus = view.findViewById(R.id.moistureStatus);
            statusValue = view.findViewById(R.id.statusValue);
            soilPhValue = view.findViewById(R.id.soilPhValue);
        }
    }
}
```

Alat Hydrativa

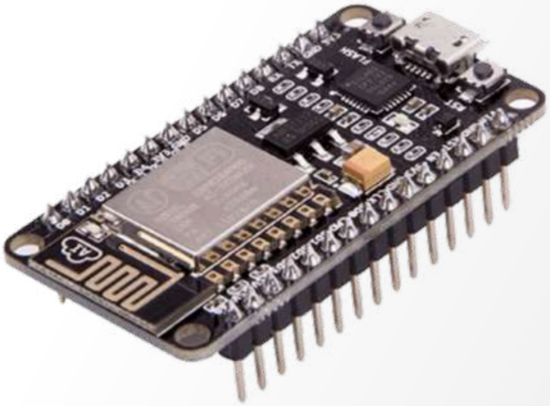
Komponen Alat

Relay



Soil Moisture Sensor

Komponen Alat



NodeMCU ESP8266

Mini Water Pump

Terima Kasih