

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC002023123413, 4 Desember 2023

Pencipta

Nama : **Herman Yuliansyah, S.T., M.Eng., Ph.D., Dr. Surahma Asti Mulasari, S.Si., M.Kes. dkk**

Alamat : Perumahan Graha Nirmala F17, RT 000 RW 000, Tirtonirmolo, Kasihan, Bantul, DI Yogyakarta, 55181

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **UNIVERSITAS AHMAD DAHLAN**

Alamat : Jl. Pramuka 5F, Pandeyan, Umbulharjo, Yogyakarta, Di Yogyakarta 55161

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **Aplikasi Android Daur Ulang Sampah Mu Untuk Bank Sampah**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 26 November 2023, di Yogyakarta

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000556368

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
Direktur Hak Cipta dan Desain Industri



Anggoro Dasananto
NIP. 196412081991031002

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Herman Yuliansyah, S.T., M.Eng., Ph.D.	Perumahan Graha Nirmala F17, RT 000 RW 000, Tirtonirmolo, Kasihan, Bantul
2	Dr. Surahma Asti Mulasari, S.Si., M.Kes.	Perumahan Sidorejo Gang Nakula No. 161, RT 004 RW 000, Ngestiharjo, Kasihan, Bantul
3	Lu'lu' Nafiati, S.E., M.Sc.	Dongkelan No 39, RT 002 RW 000, Panggungharjo , Sewon, Bantul
4	Dr. Tri Wahyuni Sukesi, S.Si, M.PH.	Dusun Lojajar, RT 003 RW 022, Sinduharjo, Ngaglik, Sleman
5	Sulistiyawati, S.Si., M.PH., PhD.	Sepat, RT 005 RW 002, Ngoro-oro, Patuk, Gunung Kidul
6	Fanani Arief Ghozali, S.Pd., M.Pd.	Tegal Menukan, RT 004 RW 000, Bangungharjo, Sewon, Bantul
7	Dr. Fatwa Tentama, S.Psi., M.Si.	Sidorejo No. 2, RT 006 RW 000, Ngestiharjo, Kasihan, Bantul
8	Dr. Bambang Sudarsono, M.Pd.	Denggung, RT 002 RW 035, Tridadi, Sleman, Sleman





Daur Ulang

Sampah Mu

November, 2023

BUKU PANDUAN APLIKASI DAUR ULANG SAMPAH MU UNTUK PENGELOLA

Disusun oleh: Herman Yuliansyah, S.T., M.Eng., Ph.D., Dr. Surahma Asti Mulasari, S.Si, M.Kes., Lu'lu' Nafiati, S.E., M.Sc., Dr. Tri Wahyuni Sukesi, S.Si., M.PH., Sulistyawati, S.Si., M.PH., Ph.D., Fanani Arief Ghozali, S.Pd., M.Pd., Dr. Fatwa Tentama, S.Psi., M.Si., Dr. Bambang Sudarsono, M.Pd.

DAFTAR ISI

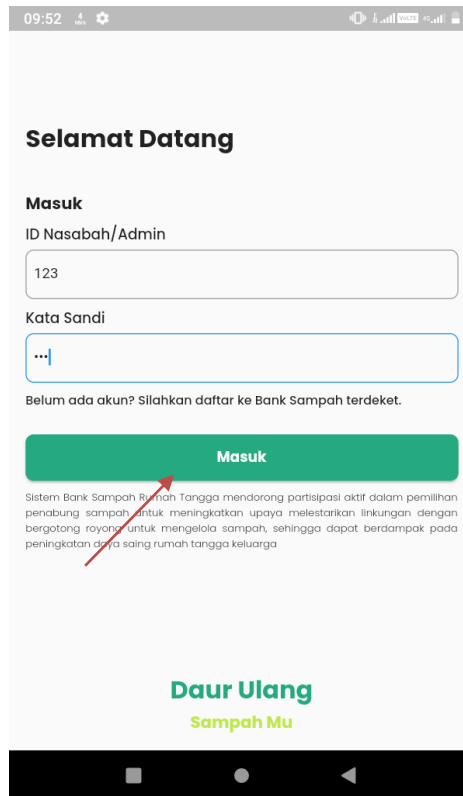
E - BANK SAMPAH

Log In.....	3
Dashboard.....	4
Kas Bank Sampah.....	5
Kredit.....	6
Debit.....	9
Nasabah.....	11

Log In

Untuk dapat menggunakan aplikasi Daur Ulang Sampah Mu, pengelola diharapkan melakukan log in terlebih dahulu dengan akun khusus pengelola.

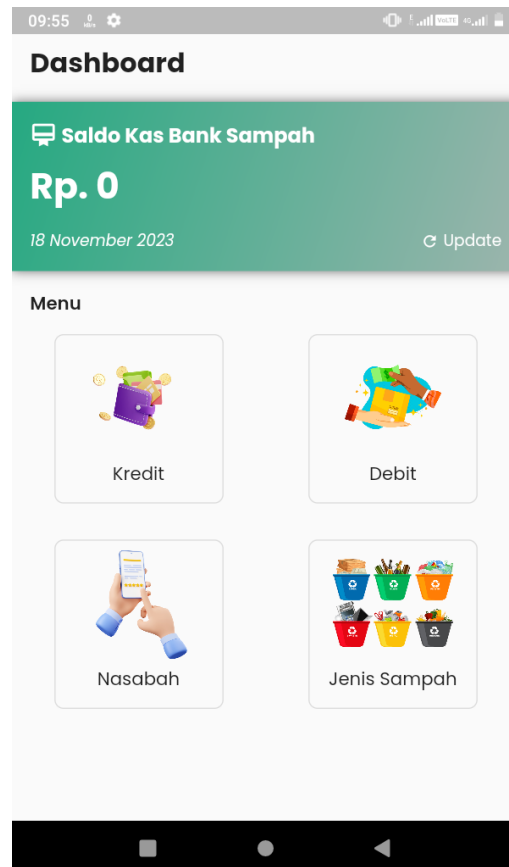
1. Buka Aplikasi Daur Ulang Sampah Mu.
2. Masukkan ID Admin (pengelola) dan Kata Sandi di kolom yang disediakan secara berurutan. ID dan Kata Sandi dibuatkan oleh *administrator system*.



3. Setelah itu, klik tombol Masuk.

Dashboard

Ketika pengelola berhasil Masuk/Log In, maka akan otomatis diarahkan ke laman dashboard pengelola. Dashboard merupakan tampilan utama dari laman pengelola. Dashboard

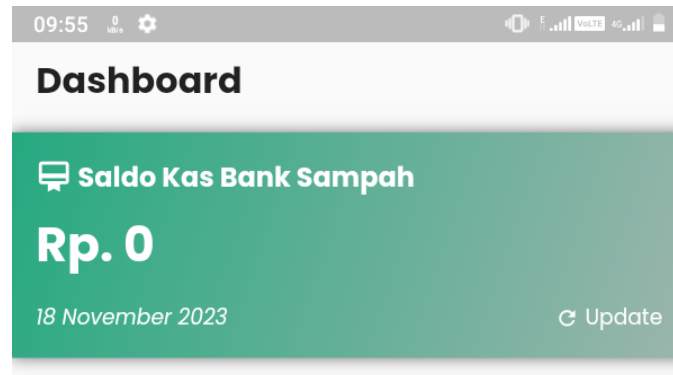


Terdapat beberapa menu yang disediakan, yaitu:

- Dashboard
- Kas Bank Sampah
- Kredit
- Debit
- Nasabah
- Jenis Sampah

Kas Bank Sampah

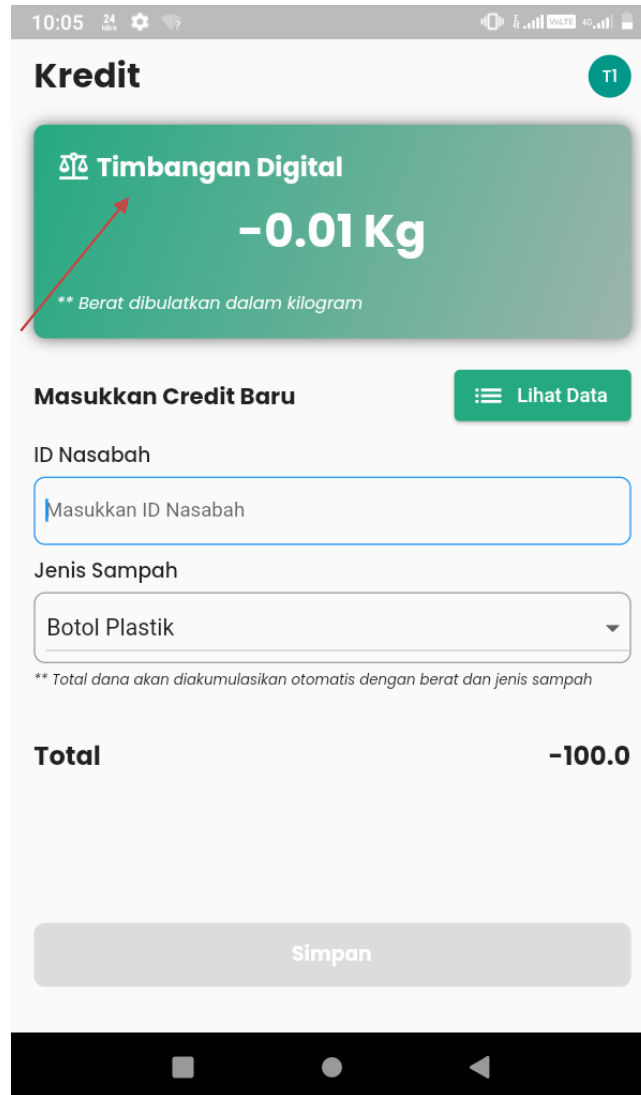
Kas bank sampah merupakan fitur yang menampilkan saldo dari Kas Bank Sampah yang telah di catat.



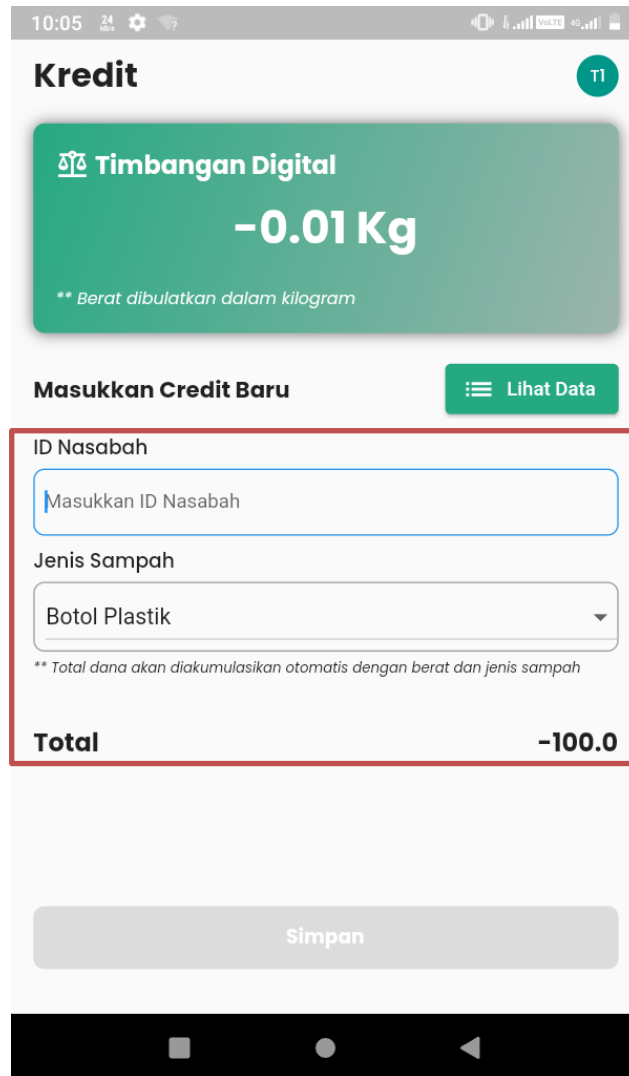
Kredit

Menu Kredit digunakan untuk menghitung jumlah sampah yang diterima oleh pengelola.

1. Sampah diletakkan ke dalam timbangan kemudian timbangan digital akan menampilkan berat dari sampah tersebut.



2. Masukkan ID Nasabah dan Jenis sampah yang ditimbang. Total saldo yang didapatkan akan otomatis terhitung dalam aplikasi.

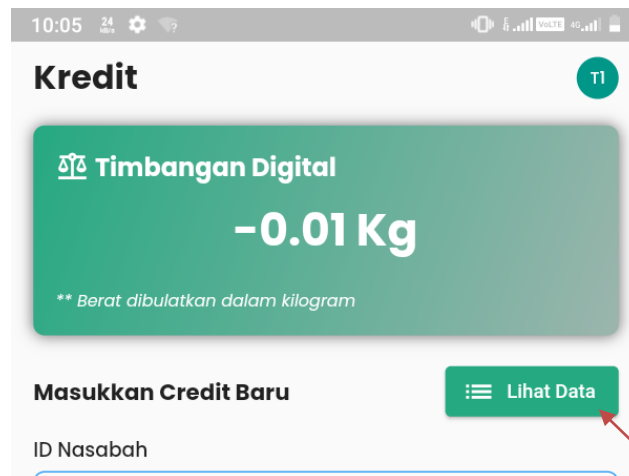


The screenshot shows the 'Kredit' app interface. At the top, the status bar displays the time 10:05, date 24 Nov, and various system icons. The app title 'Kredit' is at the top left, and a notification badge '11' is at the top right. A green card displays 'Timbangan Digital' (Digital Scale) with a reading of '-0.01 Kg' and a note: '** Berat dibulatkan dalam kilogram' (Weight rounded in kilograms). Below this, there are two buttons: 'Masukkan Credit Baru' (Add New Credit) and 'Lihat Data' (View Data). A red box highlights a form with the following fields: 'ID Nasabah' (Customer ID) with a placeholder 'Masukkan ID Nasabah', and 'Jenis Sampah' (Waste Type) with a dropdown menu showing 'Botol Plastik' (Plastic Bottle). A note below the dropdown reads: '** Total dana akan diakumulasikan otomatis dengan berat dan jenis sampah' (Total funds will be automatically accumulated with weight and waste type). At the bottom of the form, it shows 'Total' as '-100.0'. A 'Simpan' (Save) button is located below the form.

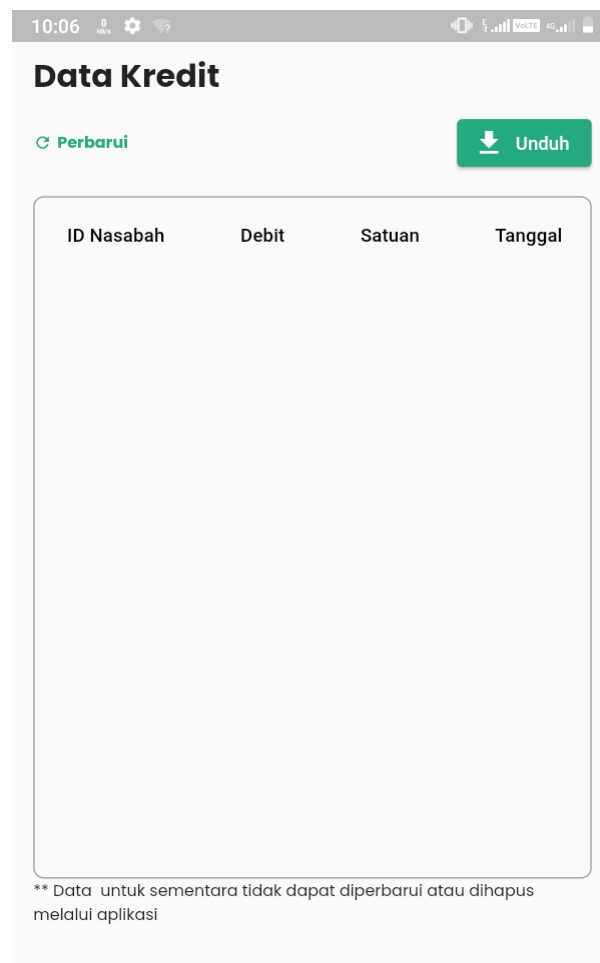
3. Klik tombol Simpan untuk menginputkan data kredit nasabah.

Melihat Data Kredit

4. Untuk melihat Data Kredit, klik tombol Lihat Data



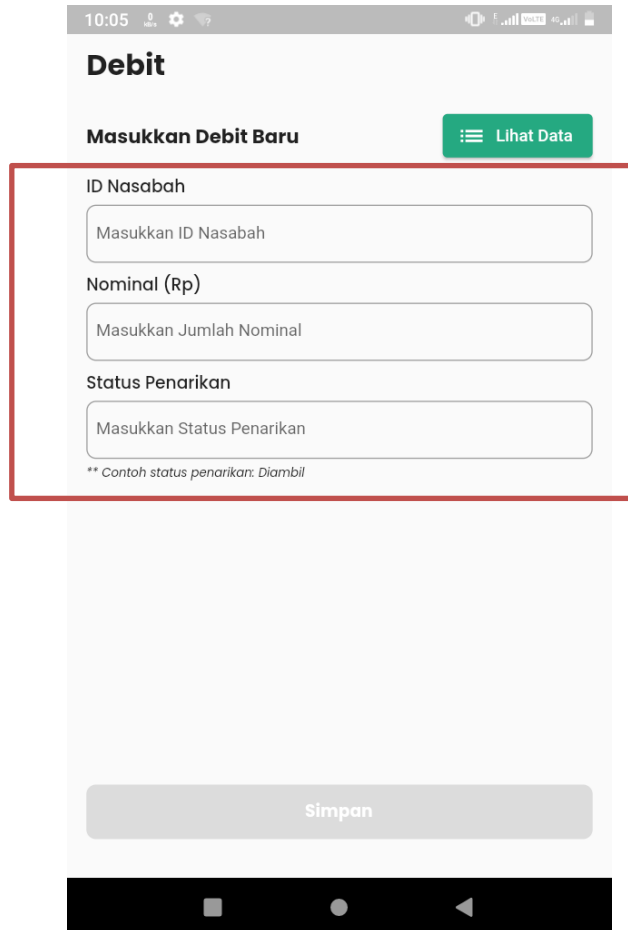
5. Tampilan aplikasi akan muncul seperti berikut



Debit

Menu Debit digunakan untuk penarikan saldo nasabah. Nasabah datang ke pengelola untuk meminta penarikan saldo.

1. Masukkan ID Nasabah, Nominal (Rp), dan Status Penarikan.

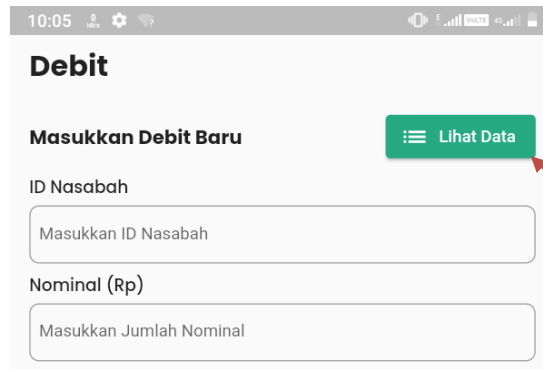


The screenshot shows a mobile application interface for recording a debit. At the top, the title "Debit" is displayed. Below it, there is a section titled "Masukkan Debit Baru" with a green button labeled "Lihat Data". The form contains three input fields: "ID Nasabah" with the placeholder "Masukkan ID Nasabah", "Nominal (Rp)" with the placeholder "Masukkan Jumlah Nominal", and "Status Penarikan" with the placeholder "Masukkan Status Penarikan". A red box highlights these three input fields. At the bottom of the form, there is a grey button labeled "Simpan". A note at the bottom of the form reads: "** Contoh status penarikan: Diambil".

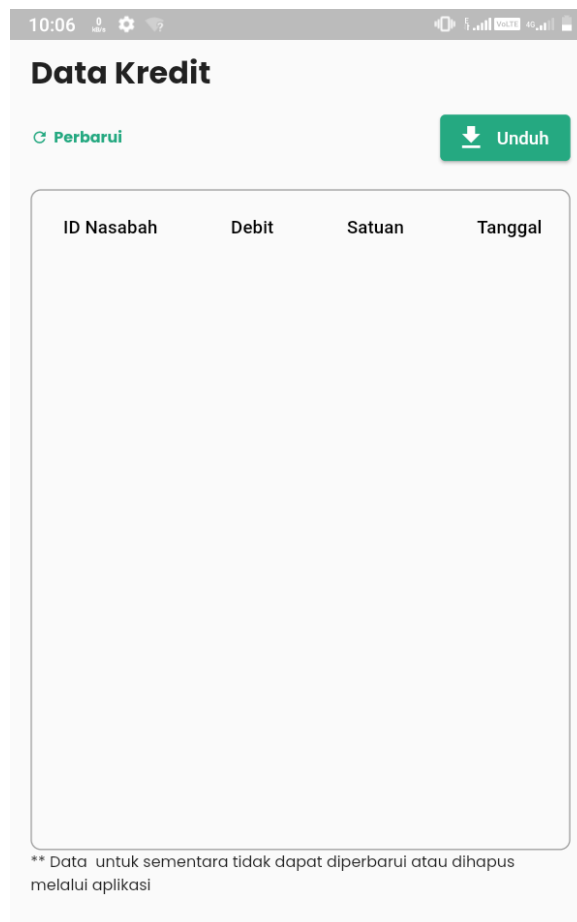
2. Klik tombol Simpan jika data sudah sesuai.

Melihat Data Debit

3. Untuk melihat Data Debit, klik tombol Lihat Data



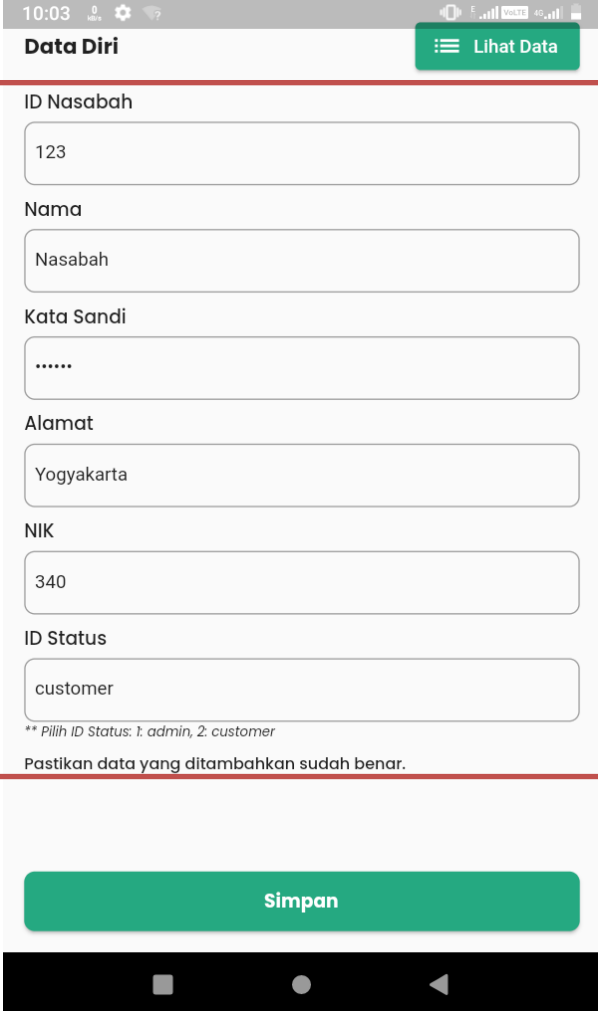
4. Tampilan aplikasi akan muncul seperti berikut.



Nasabah

Menu nasabah merupakan menu dimana pengelola dapat melakukan penambahan akun nasabah.

1. Masukkan data nasabah seperti ID Nasabah, Nama, Kata Sandi, Alamat, NIK, ID Status

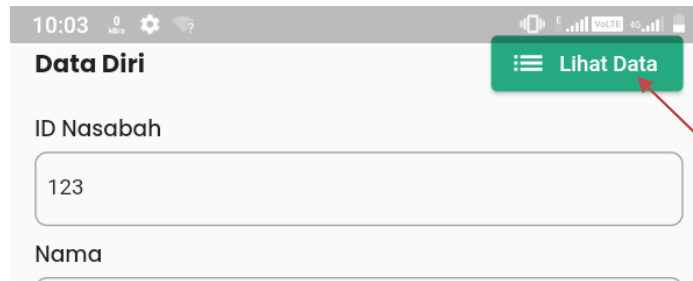


The screenshot shows a mobile application interface for adding a customer. At the top, the status bar shows the time 10:03 and various icons. Below it, the title 'Data Diri' is displayed in a green header bar, with a 'Lihat Data' button on the right. The main form area contains several input fields: 'ID Nasabah' (123), 'Nama' (Nasabah), 'Kata Sandi' (masked with dots), 'Alamat' (Yogyakarta), 'NIK' (340), and 'ID Status' (customer). A note below the fields reads: '** Pilih ID Status: 1. admin, 2. customer'. At the bottom of the form, there is a green 'Simpan' button. Below the form, there is a black navigation bar with standard Android icons.

2. Kemudian klik Simpan

Melihat Data Nasabah

3. Untuk melihat Data Nasabah, klik tombol Lihat Data



4. Tampilan aplikasi akan muncul seperti berikut.



Source Code

```
import 'dart:convert';
import 'package:http/http.dart' as http;

class BaseRepository {
  // static const baseUrl =
  // "https://app-574b3b2b-05f4-4a14-a811-fd215c1e4fdf.cleverapps.io";
  static const baseUrl =
    "https://asnproject.site/api";

  static Future<Map<String, dynamic>?> fetchData(String idUser) async {
    try {
      final response = await http.get(Uri.parse('$baseUrl/customers/$idUser'));

      if (response.statusCode == 200) {
        final data = jsonDecode(response.body);
        return data;
      } else {
        return null;
      }
    } catch (e) {
      return null;
    }
  }

  static Future<Map<String, dynamic>?> fetchDataCustomerAll() async {
    try {
      final response = await http.get(Uri.parse('$baseUrl/customers'));

      if (response.statusCode == 200) {
        final data = jsonDecode(response.body);
        return data;
      } else {
        return null;
      }
    } catch (e) {
      return null;
    }
  }

  static Future<bool> deleteCustomer(String idUser) async {
    try {
      final response = await http.delete(Uri.parse('$baseUrl/customers/$idUser'));

      if (response.statusCode == 200) {
        return true;
      } else {
        return false;
      }
    } catch (e) {
      return false;
    }
  }
}
```

```
}  
}  
  
static Future<Map<String, dynamic>?> fetchDebit() async {  
  try {  
    final response = await http.get(Uri.parse('$baseUrl/debits'));  
  
    if (response.statusCode == 200) {  
      final data = jsonDecode(response.body);  
      return data;  
    } else {  
      return null;  
    }  
  } catch (e) {  
    return null;  
  }  
}  
  
static Future<Map<String, dynamic>?> fetchDebitId(String idUser) async {  
  try {  
    final response = await http.get(Uri.parse('$baseUrl/debits/$idUser'));  
  
    if (response.statusCode == 200) {  
      final data = jsonDecode(response.body);  
      return data;  
    } else {  
      return null;  
    }  
  } catch (e) {  
    return null;  
  }  
}  
  
static Future<Map<String, dynamic>?> fetchCredit() async {  
  try {  
    final response = await http.get(Uri.parse('$baseUrl/credits'));  
  
    if (response.statusCode == 200) {  
      final data = jsonDecode(response.body);  
      return data;  
    } else {  
      return null;  
    }  
  } catch (e) {  
    return null;  
  }  
}  
  
static Future<Map<String, dynamic>?> fetchCreditId(String idUser) async {  
  try {  
    final response = await http.get(Uri.parse('$baseUrl/credits/$idUser'));  
  
    if (response.statusCode == 200) {  
      final data = jsonDecode(response.body);
```




```
        return data;
    } else {
        return null;
    }
} catch (e) {
    return null;
}
}

static Future<bool> addCustomer({
  required String idUser,
  required String password,
  required String name,
  required String address,
  required String idNumber,
  required String idStatus,
  required String idLoad,
}) async {
  try {
    final response =
      await http.post(Uri.parse('${baseUrl}/customers'), headers: {
        'Content-Type': 'application/x-www-form-urlencoded',
      }, body: {
        'id_user': idUser,
        'password': password,
        'name': name,
        'address': address,
        'id_number': idNumber,
        'id_status': idStatus,
        'id_load': idLoad,
      });
    if (response.statusCode == 201) {
      return true;
    } else {
      return false;
    }
  } catch (e) {
    return false;
  }
}

static Future<bool> addWasteType({
  required String idType,
  required String type,
  required String price,
}) async {
  try {
    final response =
      await http.post(Uri.parse('${baseUrl}/type-wastes'), headers: {
        'Content-Type': 'application/x-www-form-urlencoded',
      }, body: {
        'id_type': idType,
```

```
'type': type,
'price': price,
});
if (response.statusCode == 201) {
  return true;
} else {
  return false;
}
} catch (e) {
  return false;
}
}

static Future<Map<String, dynamic>?> fetchDataTypeWasteAll() async {
  try {
    final response = await http.get(Uri.parse('$baseUrl/type-wastes'));

    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      return data;
    } else {
      return null;
    }
  } catch (e) {
    return null;
  }
}

static Future<Map<String, dynamic>?> fetchDataTypeWasteId(String id) async {
  try {
    final response = await http.get(Uri.parse('$baseUrl/type-wastes/$id'));

    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      return data;
    } else {
      return null;
    }
  } catch (e) {
    return null;
  }
}

static Future<bool> deleteTypeWaste(String idType) async {
  try {
    final response = await http.delete(Uri.parse('$baseUrl/type-wastes/$idType'));

    if (response.statusCode == 200) {
      return true;
    } else {
      return false;
    }
  }
}
```



```
} catch (e) {
    return false;
}
}

static Future<Map<String, dynamic>?> fetchLoadId(String code) async {
    try {
        final response = await http.get(Uri.parse('$baseUrl/loads/$code'));

        if (response.statusCode == 200) {
            final data = jsonDecode(response.body);
            return data;
        } else {
            return null;
        }
    } catch (e) {
        return null;
    }
}

static Future<bool> addCredit({
    required String userId,
    required String typeId,
    required String weight,
    required String credit,
}) async {
    try {
        final response =
            await http.post(Uri.parse('$baseUrl/credits'), headers: {
                'Content-Type': 'application/x-www-form-urlencoded',
            }, body: {
                'id_user': userId,
                'id_type': typeId,
                'weight': weight,
                'credit': credit,
            });
        if (response.statusCode == 201) {
            return true;
        } else {
            return false;
        }
    } catch (e) {
        return false;
    }
}

static Future<bool> addDebit({
    required String userId,
    required String debit,
    required String status,
}) async {
    try {
```



```
final response =
await http.post(Uri.parse('$baseUrl/debits'), headers: {
  'Content-Type': 'application/x-www-form-urlencoded',
}, body: {
  'id_user': userId,
  'debit': debit,
  'status_withdrawal': status,
});
if (response.statusCode == 201) {
  return true;
} else {
  return false;
}
} catch (e) {
  return false;
}
}
```