

Proposal
Application Project III
Aplikasi Catatan Keuangan Pribadi



Disusun Oleh :

Dwi Lika A. (191117001)

SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA

STIKI MALANG

2021

1. Judul/ Nama Perangkat Lunak;

Aplikasi Catatan Keuangan Pribadi

2. Latar Belakang Ide Perangkat Lunak;

Perkembangan teknologi juga dapat di buktikan dengan terciptanya berbagai aplikasi baik berbasis web maupun mobile yang mempercepat dalam pertukaran informasi yang mengikuti perkembangan zaman. Cara untuk mencapai pribadi yang lebih sejahtera adalah dengan mengontrol keuangannya dengan bijak dan cermat. Salah satu upaya yang dapat dilakukan adalah dengan cara membuat catatan keuangan sehingga keuangan dapat terkontrol dengan baik. Manfaatnya adalah setiap waktu dapat mengevaluasi pemasukan dan pengeluaran sebagai bahan acuan untuk keuangan ke depannya. Pencatatan keuangan tersebut dapat membantu seseorang untuk tetap berada pada kondisi keuangan yang stabil sehingga dapat bisa melihat apakah perlu ada pengeluaran yang dikurangi atau menambah tabungan.

Namun pada umumnya pencatatan laporan keuangan masih dilakukan secara manual, yaitu dengan cara mencatat transaksi ke buku catatan dan menghitungnya secara manual data transaksi tersebut sehingga menghasilkan laporan keuangan dalam pembukuan sederhana. Akan tetapi hal tersebut sangatlah tidak efisien, banyaknya jumlah transaksi yang terjadi setiap harinya mengharuskan pencatatan, penghitungan transaksi dan pembuatan laporan memakan waktu yang tidak sedikit.

Melihat permasalahan dalam mengelola keuangan maka dibutuhkan sebuah aplikasi pengelolaan keuangan yang dapat digunakan oleh seseorang untuk membantunya dalam mengendalikan keuangan dengan cara melakukan pencatatan dan penghitungan pada pemasukan dan pengeluaran keuangan secara rinci sehingga dapat dengan mudah melihat rincian laporan keuangan dan kemudian laporan keuangan tersebut bisa dijadikan sebagai bahan evaluasi untuk ke depannya. Aplikasi ini dapat digunakan pada perangkat bergerak berbasis android yang dalam penggunaannya dapat dengan mudah dibawa ke mana-mana sehingga diharapkan dapat memudahkan pengguna menggunakan aplikasi ini dalam kehidupan sehari-hari. Aplikasi ini juga diharapkan mampu memudahkan seseorang dalam mengelola keuangannya sehingga dapat mewujudkan pribadi yang sejahtera.

3. Tujuan dan Manfaat Dikembangkannya Perangkat Lunak;

Adapun tujuan dan manfaat yang diperoleh dalam pembuatan aplikasi catatan keuangan ialah ;

- a. Aplikasi catatan keuangan dapat membantu untuk mengetahui detail pemasukan dan pengeluaran.
- b. Dapat membantu pola pengeluaran lebih stabil.
- c. Dapat membantu tujuan keuangan yang lebih efisien.

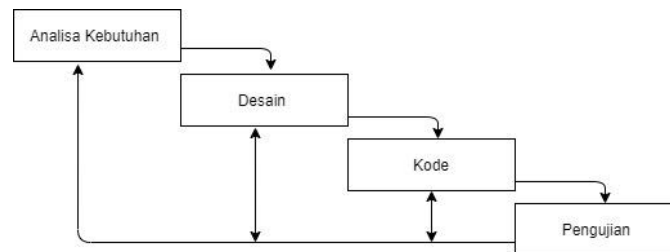
4. Batasan Perangkat Lunak yang Dikembangkan;

Adapun batasan dalam pembuatan aplikasi catatan keuangan ialah :

- a. Aplikasi ini hanya berjalan di smartphone berbasis Android.
- b. Aplikasi ini hanya menampilkan mengenai pengelolaan keuangan pribadi dan untuk pengalokasian keuangan dari kebutuhan setiap harinya.
- c. Pengguna hanya dapat melihat total pemasukan dan pengeluaran tiap bulannya.

5. Metodologi Pengembangan Perangkat Lunak;

Model Sekuensial Linier Sering disebut siklus kehidupan klasik (*classic life cycle*) atau model air terjun (*waterfall model*). Sekuensial linier mengusulkan sebuah pendekatan pada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem melalui analisis, desain, pengkodean (coding) dan pengujian (*testing*).



Gambar 1.1 *Waterfall model*

a. Analisa Kebutuhan

Dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem dengan mendefinisikan kebutuhan yang dibutuhkan atau persyaratan terkait sistem yang akan dikembangkan. Spesifikasi kebutuhan melibatkan analisis perangkat keras (hardware) dan perangkat lunak (software).

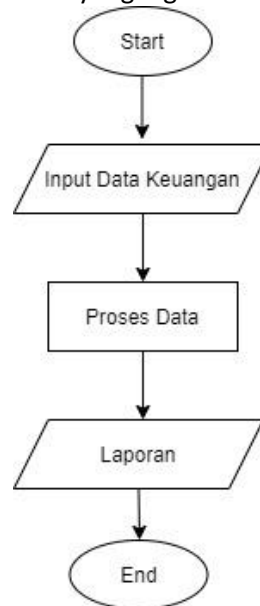
b. Desain Proses

Adapun Tugas-tugas yang dibutuhkan untuk membangun satu atau lebih representasi dari aplikasi catatan keuangan adalah sebagai berikut :

Desain sistem menggunakan

1.Flowchart

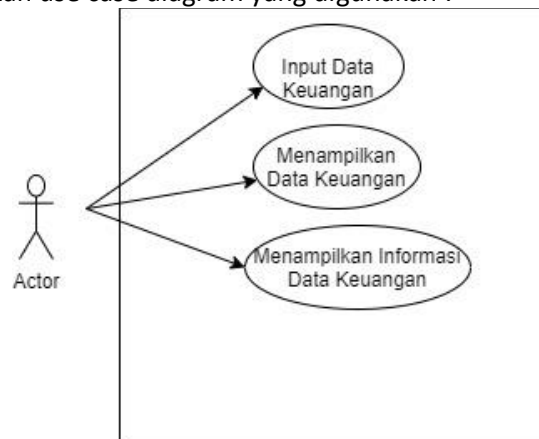
Berikut merupakan alur flowchart yang digunakan pada catatan keuangan :



Gambar 1.2 Flowchart Diagram

2. Use Case

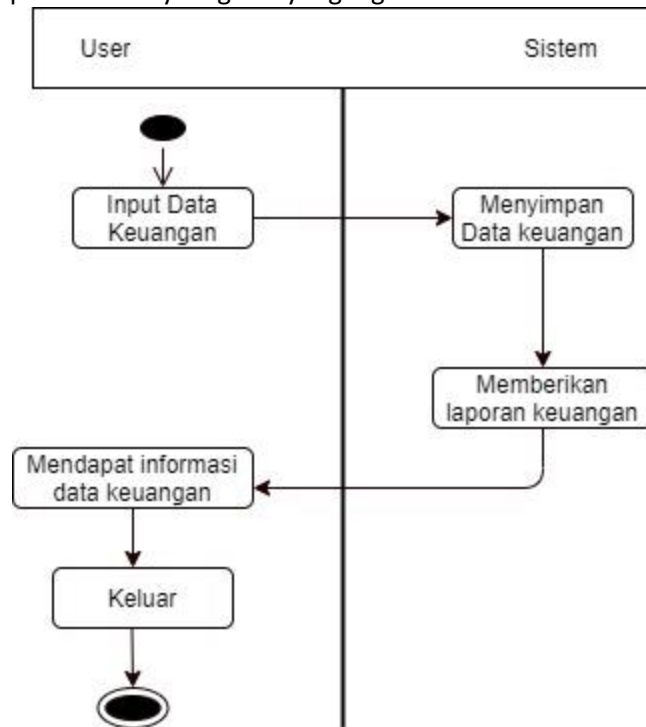
Berikut merupakan use case diagram yang digunakan :



Gambar 1.3 Use Case Diagram

3. Activity Diagram

Berikut merupakan activity diagram yang digunakan :



Gambar 1.4 Activity Diagram

c. Kode

Dalam Pembuatan perangkat lunak pengetikan kode program dan struktur logika aplikasi aplikasi Catatan Keuangan Pribadi penulis Menggunakan Bahasa Pemrograman Java dan Android SDK.

d. Pengujian

Adapun proses pengujian dilakukan untuk mengetahui kualitas program yang akan digunakan oleh pengguna nantinya dan menghindari kesalahan program.

6. Analisis Kebutuhan dan Desain Solusi Perangkat Lunak;

a. Analisis Kebutuhan

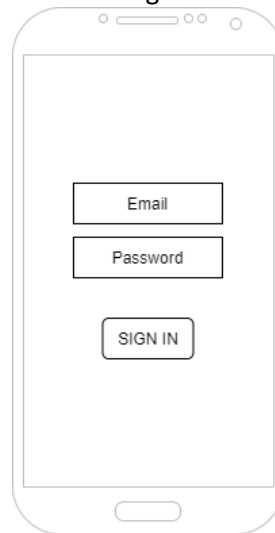
Untuk membantu dalam pembuatan aplikasi catatan keuangan , maka dibutuhkan komponen penunjang yang dibutuhkan berupa data keuangan yang nantinya akan menampilkan informasi keuangan pengeluaran dan pemasukan.

b. Desain Perangkat Lunak

Adapun tampilan desain perangkat lunak yang digunakan dalam pembuatan aplikasi catatan keuangan :

1. Tampilan Login

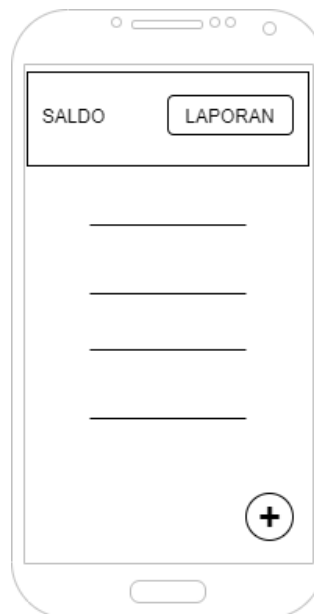
Berikut tampilan halaman login, sebelum login pengguna harus mendaftarkan email ke halaman register . Setelah email terdaftar pengguna dapat melakukan login.



Gambar 1.5 Tampilan Login

2. Tampilan home

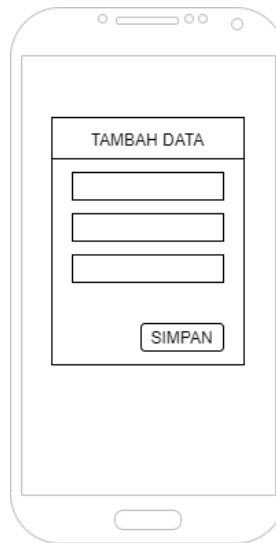
Berikut tampilan home saat pengguna sudah masuk kedalam aplikasi , akan muncul saldo sesuai jumlah pemasukan dan pengeluaran. Dan pengguna dapat menambahkan data. Dan melihat laporan.



Gambar 1.6 Tampilan Home

3. Tampilan tambah data

Berikut merupakan tampilan halaman tambah data yang nantinya digunakan pengguna untuk menambahkan data keuangan.



The image shows a mobile application interface for adding data. It features a central form titled "TAMBAH DATA". Inside the form, there are three empty rectangular input fields stacked vertically. At the bottom right of the form is a button labeled "SIMPAN". The entire form is centered on a light gray background that represents the mobile device screen.

Gambar 1.7 Tampilan Tambah Data

4. Tampilan laporan

Berikut tampilan laporan , ketika pengguna memilih menu masuk maka akan muncul data list keuangan pemasukan sedangkan ketika pengguna memilih menu keluar maka akan menampilkan data list keuangan pengeluaran. Dan menampilkan masing" total pemasukan dan pengeluaran.



The image shows a mobile application interface for a report. At the top, there are two tabs: "MASUK" and "KELUAR". Below the tabs is a list of four data items, each in a separate box labeled "Data 1", "Data 2", "Data 3", and "Data 4". At the bottom of the list is a button labeled "Total :". The entire interface is centered on a light gray background that represents the mobile device screen.

Gambar 1.8 Tampilan laporan

7. Implementasi Perangkat Lunak;

Loginactivity.java

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

    mEmail = findViewById(R.id.emailIn);
    mPass = findViewById(R.id.passwordIn);
    signInBtn = findViewById(R.id.btn_signin);
    mTextView = findViewById(R.id.textViewIn);

    mAuth = FirebaseAuth.getInstance();

    if (mAuth.getCurrentUser() != null) {
        startActivity(new Intent( packageContext: LoginActivity.this, MainActivity.class));
        finish();
    }

    TextView tvForgotPassword = findViewById(R.id.tv_forgot_password);
    tvForgotPassword.setOnClickListener((view) -> {
        String email = mEmail.getText().toString();

        if (email.isEmpty()) {
            mEmail.setError("Empty Fields Are not Allowed");
            return;
        }

        mAuth.sendPasswordResetEmail(email).addOnSuccessListener((OnSuccessListener) (unused) -> {
            Toast.makeText( context: LoginActivity.this, text: "Email sent. Please check your inbox", Toast.LENGTH_LONG).show();
        })
        .addOnFailureListener((e) -> {
            Log.e( tag: "gagallllll", e.getLocalizedMessage(), e);
        });
    });

    mTextView.setOnClickListener((v) -> {
        startActivity(new Intent( packageContext: LoginActivity.this, RegisterActivity.class));
    });

    signInBtn.setOnClickListener((v) -> { loginUser(); });

    if (getSupportActionBar() != null) {
        getSupportActionBar().setTitle("Login");
    }
}
```



```

private void loginUser() {
    String email = mEmail.getText().toString();
    String pass = mPass.getText().toString();

    if (!email.isEmpty() && Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        if (!pass.isEmpty()) {
            mAuth.signInWithEmailAndPassword(email, pass)
                .addOnSuccessListener((OnSuccessListener) (authResult) → {
                    Toast.makeText( context: LoginActivity.this, text: "Login Sukses", Toast.LENGTH_SHORT).show();
                    startActivity(new Intent( packageContext: LoginActivity.this, MainActivity.class));
                    finish();
                }).addOnFailureListener((e) → {

                    Toast.makeText( context: LoginActivity.this, text: "Login Failed", Toast.LENGTH_SHORT).show();

                });
        } else {
            mPass.setError("Empty Fields Are not Allowed");
        }
    } else if (email.isEmpty()) {
        mEmail.setError("Empty Fields Are not Allowed");
    } else {
        mEmail.setError("Please Enter Correct Email");
    }
}
}

```

MainActivity.jav

```

public class MainActivity extends AppCompatActivity {
    |
    FloatingActionButton tbhData;
    RecyclerView recyclerView;
    TextView tvSaldo;
    Button btnLaporan;
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    FirebaseAuth auth = FirebaseAuth.getInstance();
    DatabaseReference myRef = database.getReference( path: "Transaksiku").child(auth.getUid());
    List<Transaksiku> list = new ArrayList<>();
    AdapterTransaksiku adapterTransaksiku;

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

        tbhData = findViewById(R.id.tbl_data);
        recyclerView = findViewById(R.id.resikel_view);
        recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
        tvSaldo = findViewById(R.id.uang_dompot);

        btnLaporan = findViewById(R.id.button_laporan);
        btnLaporan.setOnClickListener((view) → {
            startActivity(new Intent( packageContext: MainActivity.this, ReportActivity.class));
        });
    }
}

```

```

        tbhData.setOnClickListener((v) → { showDialogTambahData(); });

        bacaData();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == R.id.menu_logout) {
            auth.signOut();
            startActivity(new Intent( packageContext: MainActivity.this, LoginActivity.class));
            finish();
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    private void bacaData() {
        // Read from the database
        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                list.clear();
                // This method is called once with the initial value and again
                // whenever data at this location is updated.
                for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                    Transaksiku value = snapshot.getValue(Transaksiku.class);
                    list.add(value);
                }
                adapterTransaksiku = new AdapterTransaksiku( context: MainActivity.this,list);
                recyclerView.setAdapter(adapterTransaksiku);
                // recyclerView.setAdapter(new AdapterTransaksiku(MainActivity.this,list));

                int totalMasuk = 0;
                int totalKeluar = 0;
                int saldo = 0;
                for (Transaksiku item : list) {
                    if (item.getTipe().equals("Pemasukan")) {
                        totalMasuk += item.getJumlah();
                    }
                    if (item.getTipe().equals("Pengeluaran")) {
                        totalKeluar += item.getJumlah();
                    }
                }
            }
        });
    }

```

8. Screenshot Mockup Interface Perangkat Lunak;
a. Halaman Login



Enter email


Enter Password

[forgot password?](#)

SIGN IN

[Do not have account? Register Here](#)

- b. Halaman Register



Enter email

Enter Password

SIGN UP

[Already Registered? Sign In](#)

c. Halaman Menu

Saldo: Rp 000.000,- Laporan

tipe
jumlah
keterangan
tanggal

tipe
jumlah
keterangan
tanggal

tipe
jumlah
keterangan
tanggal

tipe
jumlah
keterangan
tanggal

tipe
jumlah

d. Halaman Tambah Data

X Tambah Data

Keterangan

Jumlah

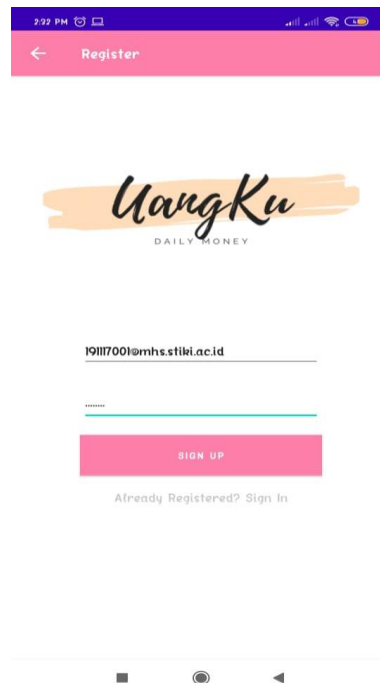
TAMBAH

e. Halaman Laporan



9. Dokumentasi Cara Penggunaan Perangkat Lunak;

- a. Pengguna mendaftarkan email dan memasukkan password yang akan digunakan untuk login.



- b. Pengguna melakukan Login dengan email dan password yang telah terdaftar



191117001@mhs.stiki.ac.id

[forgot password?](#)

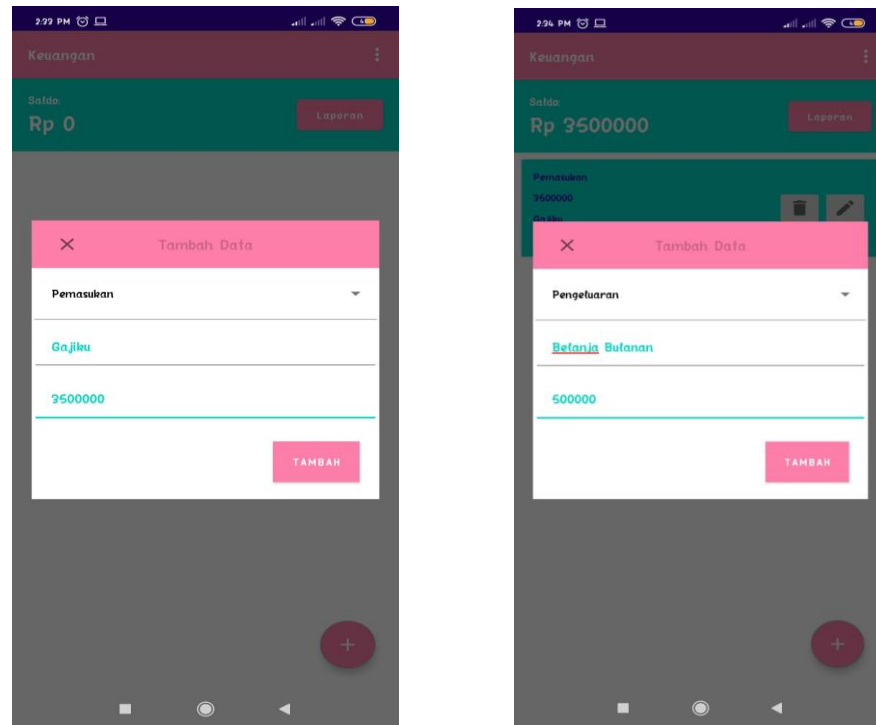
SIGN IN

[Do not have account? Register Here](#)

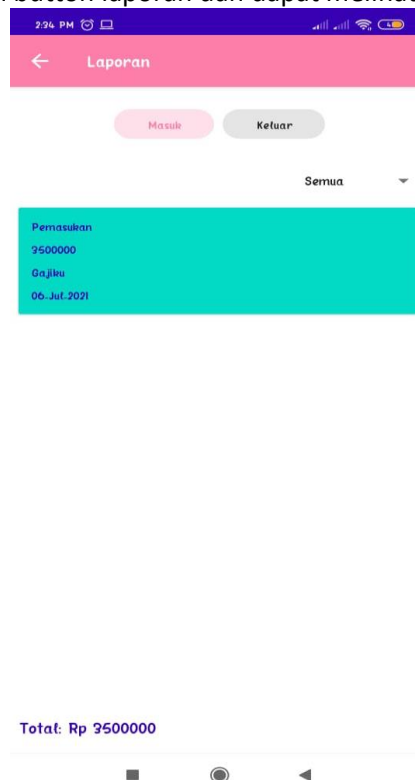
- c. Jika Login berhasil , pengguna akan masuk ke halaman pertama (dashboard)



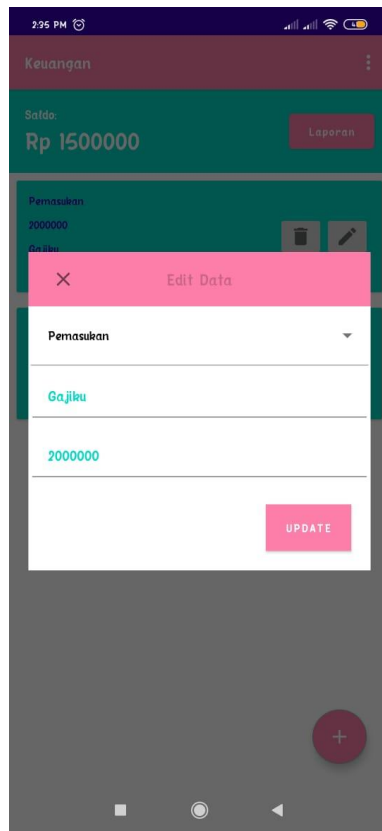
- d. Pengguna dapat menambahkan data pemasukan dan pengeluaran. Dan Pengguna juga dapat melihat total sisa saldo .



- e. Pengguna dapat memilih button laporan dan dapat melihat total pengeluaran dan pemasukan.



f. Pengguna dapat melihat melakukan edit data.



g. Pengguna dapat mereset password jika lupa dengan password emailnya.

