

**PERANCANGAN DAN IMPLEMENTASI APLIKASI
SMART BUDGET PLANNER BERBASIS DEKSTOP
MENGGUNAKAN C# DAN VISUAL STUDIO**

Kelompok 4

Yazid Chairy Syahputra	2317120029
Suci Ade Pirana Sembiring	2317120021
Melanie Putri	231712015

KOM A1'23



**PROGRAM STUDI D-3 TEKNIK INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025**

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, dengan limpah karunia-Nya penulis dapat menyelesaikan penyusunan makalah ini dengan judul Aplikasi Smart Budget Planner. Penulis menyadari bahwa tanpa pertolongan Tuhan makalah ini tidak dapat terselesaikan dengan baik. Semoga karya ini dapat menjadi referensi dan memberikan manfaat bagi siapa pun yang membacanya.

Terima kasih penulis sampaikan kepada setiap pihak yang sudah mendukung selama berlangsungnya pembuatan makalah ini. Terkhusus lagi penulis sampaikan terima kasih Bapak Dr. Niskarto Zendrato, S.Kom., M.Kom. Selaku Dosen Pengampu Mata Kuliah Pemrograman Visual dan kepada Nadira Natalie Kuechler selaku asisten laboratorium, yang telah memberikan bimbingan, saran, dan dorongan selama proses penyusunan makalah ini. Penulis berharap makalah ini dapat memberikan manfaat dan menambah wawasan bagi pembaca.

Dengan segala kerendahan hati, penulis sangat mengharapkan kritik dan saran yang bersifat membangun demi kesempurnaan makalah ini di masa mendatang. Penulis meyakini bahwa masih terdapat kekurangan dalam penyusunan makalah ini, sehingga masukan dari berbagai pihak sangat berarti bagi peningkatan kualitas karya penulis pada tugas-tugas berikutnya..

Medan, 04 Desember 2025

Penulis

DAFTAR ISI

	Halaman
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
DAFTAR TABEL	vi
BAB 1 PENDAHULUAN	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	1
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Manfaat Penelitian.....	3
BAB 2 TINJAUAN PUSTAKA	
2.1 Komponen Penelitian	4
2.1.1 Visual Studio	4
2.1.2 Visual Basic.....	5
2.1.3 C#	5
2.1.4 NuGet Package	5
2.1.5 MySQL.....	6
2.2 Metode Pengerjaan	6
2.2.1 Menentukan Topik	6
2.2.2 Menentukan Tujuan.....	7
2.2.3 Perancangan Sistem.....	7
2.2.4 Implementasi Sistem	8
BAB 3 METODOLOGI PENELITIAN	
3.1 Diagram Penelitian	12
3.1.1 Flowchart.....	13
3.1.2 ERD	14
3.2 Jadwal Penelitian	15
BAB 4 HASIL	
4.1 Pengertian	16
4.2 Panduan Penggunaan.....	16
4.2.1 Register.....	17
4.2.2 Login	18
4.2.3 Dashboard.....	18
4.2.4 Halaman Input Pemasukan / Pengeluaran	19
4.2.5 Halaman Total Pengeluaran Berdasarkan Kategori	20
4.2.6 Halaman Riwayat Transaksi.....	21

4.2.7 Halaman Profil 22

BAB 5 PENUTUP

5.1 Kesimpulan.....	23
5.2 Saran.....	23

DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 3.1 Flowchart Pembuatan Project.....	13
Gambar 3.2 ERD.....	14
Gambar 4.1 Halaman Register.....	16
Gambar 4.2 Halaman Login.....	17
Gambar 4.3 Halaman Dashboard.....	18
Gambar 4.4 Halaman Input Pemasukan / Pengeluaran.....	19
Gambar 4.5 Halaman Total Pengeluaran.....	20
Gambar 4.6 Halaman Riwayat Transaksi.....	21
Gambar 4.7 Halaman Profil.....	21

DAFTAR TABEL

Tabel 3.1 Jadwal Penelitian.....14

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Pengelolaan keuangan merupakan aspek penting dalam kehidupan seseorang. Pengetahuan keuangan dan pengambilan keputusan finansial yang baik kini diakui secara luas sebagai penentu penting dari kemakmuran pribadi dan masyarakat (Kerner & Goodyear, 2022). Banyak individu mengalami kesulitan dalam mengatur pemasukan dan pengeluaran sehingga menyebabkan permasalahan keuangan seperti pemborosan, ketidakteraturan anggaran, hingga ketidakmampuan memenuhi kebutuhan bulanan.

Seiring berkembangnya teknologi digital, kebutuhan akan aplikasi keuangan menjadi semakin tinggi. Aplikasi keuangan digital memungkinkan pengguna mengelola keuangan mereka secara lebih efektif melalui fitur-fitur seperti pemantauan transaksi real-time, saran penganggaran yang dipersonalisasi, dan investasi otomatis (Dafri, 2023).

Aplikasi *Smart Budget Planner* hadir sebagai solusi untuk membantu pengguna mengatur keuangan dengan lebih efektif melalui fitur pencatatan pemasukan, pengeluaran, pengelompokan kategori belanja, hingga analisis anggaran bulanan. Dengan adanya aplikasi ini, pengguna dapat mengetahui detail arus keuangan secara lebih jelas, merencanakan anggaran dengan lebih akurat, serta meminimalkan risiko pemborosan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang aplikasi yang dapat membantu pengguna dalam mengatur, mencatat, dan memantau keuangan secara efektif?
2. Bagaimana aplikasi dapat menyajikan informasi anggaran yang jelas sehingga memudahkan pengguna memahami kondisi finansial mereka?

3. Bagaimana aplikasi dapat memberikan fitur perencanaan anggaran bulanan yang sesuai dengan kebutuhan pengguna?
4. Bagaimana merancang antarmuka aplikasi agar mudah digunakan oleh pengguna dari berbagai kalangan?

1.3. Batasan Masalah

Agar penelitian lebih terarah dan sesuai dengan tujuan, maka batasan masalah pada penelitian ini adalah sebagai berikut:

1. Aplikasi hanya berfokus pada pengelolaan keuangan pribadi, bukan keuangan bisnis atau perusahaan.
2. Fitur yang dikembangkan meliputi pencatatan pemasukan, pencatatan pengeluaran, pengelompokan kategori, perencanaan anggaran, serta tampilan laporan sederhana.
3. Aplikasi tidak mencakup fitur investasi, perhitungan pajak, atau integrasi dengan rekening bank.
4. Penelitian hanya membahas pengembangan dari sisi fungsional dan desain antarmuka, tanpa membahas aspek keamanan tingkat lanjut atau proses komersialisasi aplikasi.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan sebuah aplikasi *Smart Budget Planner* yang dapat membantu pengguna dalam mengelola keuangan pribadi secara lebih efektif dan terstruktur. Penelitian ini bertujuan menghasilkan sistem yang mampu mencatat pemasukan dan pengeluaran secara otomatis dalam kategori tertentu sehingga pengguna dapat mengetahui pola belanja mereka. Tujuan lainnya adalah menyediakan fitur perencanaan anggaran yang dapat membantu pengguna menetapkan batas pengeluaran bulanan sesuai kebutuhan.

Selain itu, penelitian ini juga ditujukan untuk merancang antarmuka aplikasi yang mudah digunakan, informatif, dan nyaman bagi pengguna. Dengan adanya aplikasi ini, diharapkan pengguna mampu meningkatkan kemampuan pengelolaan keuangan mereka, menghindari pemborosan, serta mencapai tujuan finansial secara lebih terukur.

Tujuan lain dalam pengembangan aplikasi ini adalah menciptakan antarmuka yang nyaman digunakan oleh semua kalangan, baik pemula maupun pengguna yang sudah terbiasa menggunakan aplikasi keuangan. Aplikasi ini dirancang agar dapat menjadi alat bantu yang tidak hanya berfungsi sebagai pencatat keuangan, tetapi juga sebagai panduan yang mendorong pengguna untuk lebih disiplin dan sadar terhadap kebiasaan pengeluarannya.

1.5. Manfaat Penelitian

Penelitian mengenai pengembangan *Smart Budget Planner* memberikan beberapa manfaat, antara lain:

1. Bagi Pengguna: Membantu pengguna mengatur keuangan pribadi secara lebih rapi, terencana, dan sesuai kebutuhan sehingga dapat mengurangi pemborosan serta meningkatkan kebiasaan menabung.
2. Bagi Peneliti: Memberikan pengalaman dalam merancang dan mengembangkan aplikasi pengelolaan keuangan berbasis digital, sekaligus memperluas wawasan terkait analisis kebutuhan pengguna.
3. Bagi Dunia Pendidikan: Menjadi referensi atau contoh bagi mahasiswa dan peneliti lain yang ingin mengembangkan aplikasi dengan fokus pada manajemen keuangan pribadi.
4. Bagi Masyarakat Umum: Meningkatkan pemahaman mengenai pentingnya literasi finansial serta mendorong penerapan gaya hidup yang lebih teratur dalam aspek ekonomi.

BAB 2

TINJAUAN PUSTAKA

2.1 Komponen Penelitian

Dalam pelaksanaan proyek, diperlukan komponen guna kelancaran dan kesuksesan dalam pengerjaan proyek. Karena tanpa tools yang diperlukan, tidak akan ada hasil yang bisa didapat. Berikut komponen yang kami gunakan selama pengerjaan proyek.

2.1.1 Visual Studio

Visual Studio Adalah IDE (Integrated Development Environment) yang dapat anda gunakan untuk mengembangkan aplikasi-aplikasi Windows. Visual Studio dirancang untuk fokus pada produktivitas. Tool ini disebut juga *Rapid Application Development tools* (RAD tools) karena dirancang dan dilengkapi untuk meningkatkan produktivitas. Versi baru dari Visual Studio ini mudah digunakan dan mudah dipelajari. Fitur-fitur dalam Visual Studio versi terbaru dibuat lebih sederhana untuk mempermudah pengguna dalam mempelajarinya dan memenuhi kebutuhan para programmer. (Enterprise, 2015)

Di dalam Visual Studio, programmer dapat menggunakan berbagai bahasa pemrograman seperti C#, Visual Basic .NET, C++, Python (melalui ekstensi), JavaScript, TypeScript, dan banyak lagi. IDE ini juga dilengkapi fitur *IntelliSense* yang dapat memberikan saran kode otomatis, mendeteksi error lebih awal, serta mempermudah penulisan program bagi pemula maupun profesional. Namun, IDE ini lebih memprioritaskan pemrograman Bahasa C#. C# atau yang dibaca C sharp merupakan bahasa pemrograman yang berorientasi konsep objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka *.NET Framework*, sehingga bisa digunakan untuk membangun aplikasi berbasis desktop, program game, aplikasi mobile dan *server – client Web* . (Isman, 2017)

2.1.2 Visual Basic

Microsoft Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NET Framework*, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi Windows Forms, Aplikasi web berbasis ASP.NET, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti Microsoft Visual C++, Visual C#, atau Visual J#), atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio .NET. Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas *.NET Framework*. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft, dan versi baru ini tidak kompatibel dengan versi terdahulu. (Blazing, 2018)

2.1.3 C#

C# merupakan salah satu bahasa pemrograman komputer yang diciptakan oleh Microsoft (dikembangkan dibawah kepemimpinan Anders Hejlsberg yang notabene juga telah menciptakan berbagai macam bahasa pemrograman termasuk Borland Turbo C++ dan Borland Delphi). Bahasa C# juga telah di standarisasi secara internasional oleh ECMA. Seperti halnya 2 bahasa pemrograman yang lain, C# bisa digunakan untuk membangun berbagai macam jenis aplikasi. (Syakir, 2015)

2.1.4 NuGet Package

Paket NuGet berisi kode yang dapat digunakan kembali yang telah disediakan pengembang lain untuk Anda gunakan dalam proyek Anda. Anda dapat menginstal paket NuGet di proyek Microsoft Visual Studio dengan menggunakan *NuGet Package Manager*, *Package Manager Console*, atau .NET CLI. Artikel ini menunjukkan cara membuat proyek Windows Presentation Foundation (WPF) dengan paket Newtonsoft.Json populer. Proses yang sama berlaku untuk proyek .NET atau .NET Core lainnya.

Setelah menginstal paket NuGet, Anda kemudian dapat membuat referensi ke dalam kode Anda dengan pernyataan `using <namespace>`, di mana `<namespace>` adalah nama paket yang Anda gunakan. Setelah membuat referensi, Anda kemudian dapat memanggil paket melalui API-nya. (Microsoft, 2022)

2.1.5 MySQL

MySQL merupakan sistem manajemen database yang mempunyai *model relational database management system* (RDBMS) seperti *SQL Server*. MySQL sebagai sistem yang bersifat *open source* dapat digunakan dengan bebas dan dikembangkan sesuai keperluan (Sophian, 2014). MySQL menjadi pilihan utama dalam pengembangan web dan aplikasi berbasis web, dikarenakan MySQL dapat memproses jutaan permintaan dan ribuan transaksi sekaligus. Berdasarkan uraian diatas, dapat diketahui bahwa sistem manajemen database yang dapat mempermudah penyimpanan data mempunyai keunikan masing-masing. Maka kami melakukan penelitian ini untuk mengetahui efisiensi penggunaan database pada sumber daya komputer melalui studi kasus SQL Server dan MySQL. (Sidharta, et al, 2020)

2.2 Metode Pengerjaan

Dengan adanya komponen yang dibutuhkan, pengerjaan sudah bisa dilakukan dan dilaksanakan, namun sebelum kita langsung terjun ke pelaksanaan proyek pastinya kita juga butuh tahapan untuk pengerjaannya.

2.2.1 Menentukan Topik

Topik adalah suatu isu atau pokok persoalan yang sifatnya masih umum dan abstrak, pada dasarnya merupakan pokok pembicaraan dalam keseluruhan tulisan yang digarap dan sebagai landasan yang dapat dipergunakan oleh seorang penulis untuk menyampaikan maksudnya. Aktivitas menulis tidak mungkin dilakukan tanpa topik. Oleh karena itu, kegiatan pertama yang harus dilakukan pada tahap pra penulisan ialah memilih topik. Misalnya, isu mengenai Akuntansi, ini adalah topiknya, yang tentunya masih bersifat umum, permasalahan Akuntansi dalam hal apa masih belum jelas, oleh karenanya tadi dikatakan bahwa topik masih bersifat umum dan abstrak. Sehingga langkah selanjutnya untuk membuat karya tulis ilmiah setelah diketahui topiknya,

adalah pembatasan topik, kemudian menentukan tema dan judul. Dengan demikian dapat juga dikatakan bahwa judul merupakan perwujudan spesifik dari topik. (Sanggup, 2015: 28). Maka dari itu, kami telah mengambil judul proyek kami bernama Budget Planner.

2.2.2 Menentukan Tujuan

Penentuan tujuan dilakukan agar proses pengembangan aplikasi Smart Budget Planner memiliki arah yang jelas dan tidak keluar dari fokus penelitian. Dengan menetapkan tujuan sejak awal, tim dapat mengetahui apa yang ingin dicapai, fitur apa yang harus dikembangkan, serta bagaimana aplikasi harus bekerja untuk memenuhi kebutuhan pengguna. Penentuan tujuan juga berfungsi sebagai pedoman dalam setiap tahap pengerjaan mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian sehingga hasil akhir sesuai dengan manfaat yang diharapkan.

Berdasarkan hal tersebut, tujuan yang ingin dicapai dalam pengembangan aplikasi ini adalah sebagai berikut:

1. Membantu pengguna mengelola keuangan pribadi dengan lebih teratur dan terkontrol.
2. Memberikan sarana pencatatan pemasukan dan pengeluaran secara digital.
3. Menyediakan visualisasi grafik untuk memahami pola pengeluaran.
4. Menyediakan riwayat transaksi yang lengkap dan terurut.
5. Menyediakan fitur login dan register agar data keuangan setiap pengguna aman dan terpisah.
6. Menyimpan data secara permanen menggunakan database MySQL.
7. Memberikan saran keuangan otomatis berdasarkan pemasukan dan pengeluaran pengguna.

Dengan tujuan-tujuan tersebut, aplikasi yang dikembangkan diharapkan mampu menjadi alat bantu yang efektif untuk membantu pengguna mengatur keuangan secara lebih disiplin dan terstruktur.

2.2.3 Perancangan Sistem

Perancangan sistem adalah proses mendesain sistem baru setelah analisis dalam siklus pengembangan sistem (Parinsi, et al. 2021). Pada tahap ini, kebutuhan

fungsional sistem didefinisikan dan strukturnya digambarkan menggunakan diagram seperti *Data Flow Diagram* (DFD) serta *Entity Relationship Diagram* (ERD). Pada KYSR Store, perancangan sistem mencakup aliran informasi dari pelanggan, admin, produksi, hingga laporan dan penyimpanan data dalam database, memastikan kecepatan, ketepatan, dan relevansi informasi. (Candra, et al. 2024)

2.2.4 Implementasi Sistem

Implementasi merupakan tahap penerapan dari hasil perancangan yang telah dibuat sebelumnya. Implementasi bertujuan untuk menghasilkan sebuah sistem informasi yang sesuai dengan kebutuhan. Tahapan yang dilakukan adalah dengan mengimplementasikan basis data beserta tabel-tabel yang dibutuhkan oleh sistem. (Dorothy, et al. 2014)

Ini termasuk dengan Pembuatan Proyek, Penulisan Program, Pengintegrasian Database, Debugging dan Menjalankan Program.

1. Implementasi Koneksi Database

Koneksi database dibuat pada file `Database.cs` yang berfungsi sebagai kelas helper untuk membuka dan menutup koneksi ke MySQL. Kode ini memastikan setiap form dapat mengakses database dengan aman menggunakan objek yang sama.

Kode Implementasi:

```
String connString =  
"server=localhost;uid=root;pwd=;database=budgetplanner;";  
connection = new MySqlConnection(connString);
```

Kelas ini menyediakan tiga fungsi utama:

- a. `GetConnection()`: mengambil objek koneksi
- b. `OpenConnection()`: membuka koneksi
- c. `CloseConnection()`: menutup koneksi

Kelas ini digunakan oleh seluruh form di aplikasi agar koneksi database lebih terstruktur dan tidak ditulis berulang.

2. Fitur Login

Fitur login digunakan untuk mengautentikasi pengguna berdasarkan email dan password yang tersimpan di database. Pada tahap implementasi, proses login mencakup pengambilan data user, verifikasi password hash, dan pengalihan ke halaman dashboard.

a. Mengambil Data User dari Database

```
string query = "SELECT * FROM users WHERE email = @email";
MySqlCommand cmd = new MySqlCommand(query, conn);
cmd.Parameters.AddWithValue("@email", email);
MySqlDataReader reader = cmd.ExecuteReader();
Kode ini menjalankan query untuk mencari user berdasarkan email.
Parameter digunakan agar data lebih aman dan menghindari SQL injection.
```

b. Verifikasi Password (Menggunakan BCrypt)

```
string hashedPassword = reader["password"].ToString();

if (BCrypt.Net.BCrypt.Verify(password, hashedPassword))
{
    // login berhasil
}
Password yang tersimpan di database berbentuk hash.
BCrypt digunakan untuk memverifikasi apakah password yang dimasukkan pengguna cocok dengan password terenkripsi yang tersimpan di database.
Metode ini meningkatkan keamanan sistem.
```

c. Penanganan Kesalahan

```
MessageBox.Show("Email not registered!");
MessageBox.Show("Incorrect password!");
Aplikasi memberikan pesan error apabila: Email tidak terdaftar, Password salah, Atau terjadi error pada koneksi database.
```

Fitur ini penting untuk memberi umpan balik kepada pengguna.

3. Fitur registrasi dibuat pada file Register.cs, yang berfungsi untuk membuat akun baru dengan validasi:

- a. Semua field wajib diisi
- b. Username dan email tidak boleh duplikat
- c. Password disimpan dalam bentuk hash BCrypt

Kode Validasi Input:

```
if (name == "" || username == "" || email == "" || password == "")
{
    MessageBox.Show("Please fill in all fields.");
    return;
}
```

Hash Password:

```
string hashedPassword = BCrypt.Net.BCrypt.HashPassword(password);
```

Cek username/email sudah dipakai:

```
string checkQuery =
    "SELECT COUNT(*) FROM users WHERE username = @username OR
email = @email";
```

4. Input Transaksi

Aplikasi menyediakan dua jenis input, yaitu pemasukan dan pengeluaran.

a. Input Pemasukan

```
mainForm.AddPemasukan(inputPemasukan.Text,
mainForm.currentUserId);
```

Ketika tombol “Tambah Pemasukan” diklik, aplikasi mengambil nilai jumlah pemasukan dari *textbox* dan mengirimnya ke *Form1* untuk disimpan ke database berdasarkan ID pengguna.

b. Input Pengeluaran

```
mainForm.AddPengeluaran(
    inputPengeluaran.Text,
    comboBox1.Text,
    DeskripsiPengeluaran.Text,
    mainForm.currentUserId
);
```

Pada pengeluaran, data yang dikirim mencakup jumlah, kategori, dan deskripsi. Semua informasi ini kemudian disimpan sebagai satu transaksi pengeluaran.

5. Perhitungan Total Pemasukan dan Pengeluaran

```
int totalPemasukan = pemasukanList.Sum(x => int.Parse(x.Jumlah));
int totalPengeluaran = pengeluaranList.Sum(x =>
int.Parse(x.Jumlah));
```

Aplikasi mengumpulkan seluruh transaksi pengguna, kemudian menjumlahkannya untuk menampilkan total pemasukan dan pengeluaran di dashboard. Nilai ini diperbarui otomatis setiap ada transaksi baru.

6. Sistem Saran Otomatis

```
if (totalPengeluaran > totalPemasukan)
    txtBoxSaran.Text = "Pengeluaran lebih besar dari pemasukan.";
Bagian ini menganalisis kondisi keuangan pengguna. Jika pengeluaran lebih besar dari pemasukan, aplikasi menampilkan peringatan. Jika pemasukan lebih tinggi, aplikasi memberikan rekomendasi menabung. Fitur ini mendukung pengguna dalam mengambil keputusan keuangan.
```

7. Implementasi Penyimpanan Transaksi ke Database

Pada `Form1`, data pemasukan disimpan menggunakan *query* berikut:

```
string query = "INSERT INTO transaksi (jenis, jumlah, kategori,
deskripsi, tanggal, user_id) " +
    "VALUES ('Pemasukan', @jumlah, '', '', NOW(),
@user_id);
```

Sedangkan pengeluaran menggunakan:

```
string query = "INSERT INTO transaksi (jenis, jumlah, kategori,
deskripsi, tanggal, user_id) " +
    "VALUES ('Pengeluaran', @jumlah, @kategori,
@deskripsi, NOW(), @user_id);
```

Kedua proses menggunakan *parameterized query* agar lebih aman dan tahan terhadap SQL Injection.

BAB 3

METODOLOGI PENELITIAN

3.1 Diagram Penelitian

Diagram penelitian menjelaskan alur penggerjaan proyek mulai dari perencanaan, pengumpulan kebutuhan, perancangan, implementasi, pengujian, hingga pelaporan. Setiap tahap dilakukan secara berurutan agar penelitian berjalan terarah. Hal ini sesuai dengan pendapat Creswell & Creswell (2018) bahwa penelitian harus dirancang secara runtut agar menghasilkan temuan yang akurat.

1. Perencanaan

Pada tahap ini peneliti menetapkan masalah yang akan diselesaikan dan strategi pelaksanaan penelitian. Menurut Kumar (2019), perencanaan penelitian berfungsi sebagai fondasi agar penelitian memiliki arah yang jelas dan sistematis.

2. Pengumpulan Kebutuhan

Pada tahap ini dilakukan pengumpulan data mengenai kebutuhan sistem dan pengguna. Analisis kebutuhan penting karena menentukan bagaimana sistem seharusnya berfungsi (Sommerville, 2016).

3. Perancangan

Tahap perancangan mencakup pembuatan alur sistem, perancangan antarmuka, serta model database. Pressman & Maxim (2015) menjelaskan bahwa desain sistem adalah proses menerjemahkan kebutuhan menjadi representasi teknis yang siap diimplementasikan.

4. Implementasi

Implementasi merupakan proses mengubah desain menjadi kode program serta menghubungkannya dengan database. Tahap ini menghasilkan produk perangkat lunak yang dapat digunakan (Sommerville, 2016).

5. Pengujian

Tahap pengujian dilakukan untuk memastikan aplikasi berjalan sesuai dengan kebutuhan dan bebas dari kesalahan. Pengujian mencakup verifikasi logika sistem, pengecekan fungsi, dan evaluasi kualitas. Menurut Kanwal & Shahnawaz (2017), pengujian perangkat lunak bertujuan untuk mendeteksi cacat dan memastikan

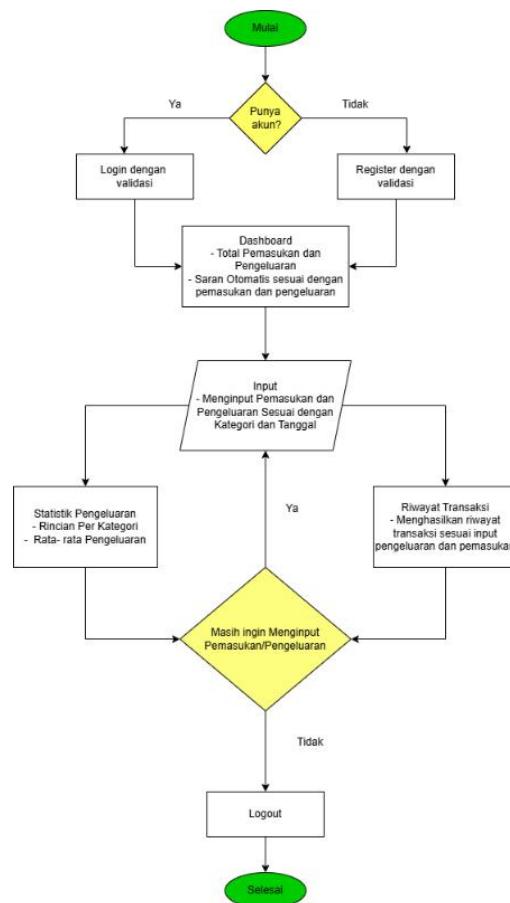
bahwa sistem memenuhi standar kualitas serta berfungsi sesuai spesifikasi yang ditetapkan.

6. Pelaporan

Tahap terakhir adalah penyusunan laporan yang berisi seluruh proses penelitian dan hasil yang diperoleh. Laporan digunakan sebagai bentuk pertanggungjawaban akademik. Menurut Creswell & Creswell (2018), pelaporan penelitian harus disusun secara sistematis agar mudah dipahami oleh pembaca dan dapat menjadi rujukan untuk penelitian berikutnya.

3.1.1 Flowchart

Flowchart ini menyajikan diagram alir proses utama (*User Flow*) dari sebuah sistem aplikasi, yang berfokus pada manajemen akun pengguna dan fungsionalitas inti pencatatan keuangan.



Gambar 3.1 Flowchart Pembuatan Project

3.1.2 ERD

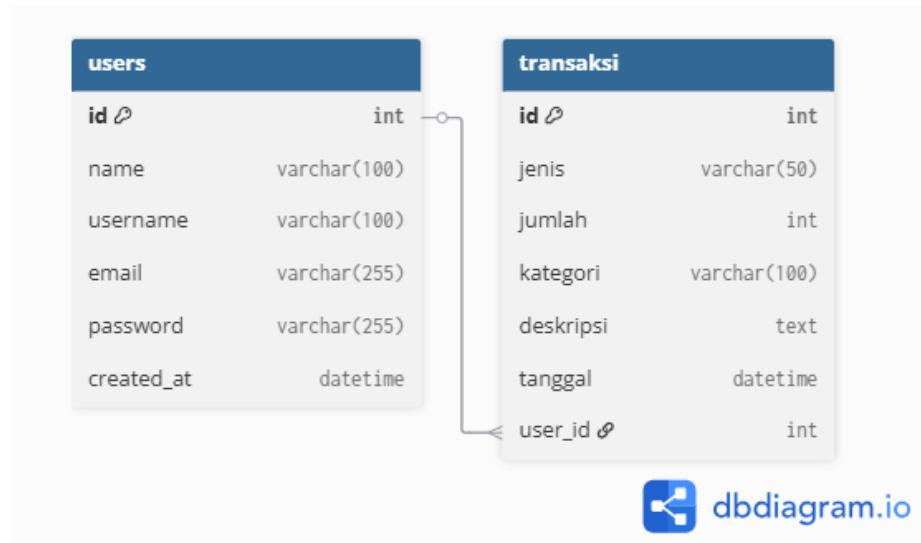
Struktur basis data pada aplikasi ini terdiri dari dua tabel utama, yaitu tabel users dan tabel transaksi, yang saling terhubung melalui relasi *one-to-many*. Tujuan perancangan ini adalah untuk memisahkan data pengguna dengan data transaksi sehingga sistem menjadi lebih terstruktur, efisien, dan mudah dikembangkan.

1. Tabel Users

Tabel users berfungsi untuk menyimpan informasi seluruh pengguna aplikasi. Setiap pengguna memiliki satu identitas unik berupa *primary key* id. Selain itu, tabel ini juga menyimpan data penting seperti nama, username, email, password, serta waktu pembuatan akun. Penyimpanan informasi ini bertujuan untuk mengelola proses autentikasi dan otorisasi pengguna dalam aplikasi.

2. Tabel Transaksi

Tabel transaksi digunakan untuk menyimpan seluruh catatan pemasukan dan pengeluaran yang dilakukan oleh pengguna. Setiap transaksi memiliki *primary key* id serta beberapa atribut seperti jenis transaksi, jumlah uang, kategori, deskripsi, dan tanggal transaksi. Tabel ini juga memiliki kolom user_id sebagai *foreign key* yang menghubungkan data transaksi dengan pengguna yang melakukan transaksi tersebut.



Gambar 3.2 ERD

3.2 Jadwal Penelitian

Untuk Jadwal penelitian dapat dilihat pada tabel dibawah.

Tabel 3.1 Jadwal Penelitian

BAB 4

HASIL

4.1 Pengertian

Aplikasi *Smart Budget Planner* yang dikembangkan pada penelitian ini merupakan sebuah sistem pengelolaan keuangan pribadi yang berfungsi untuk membantu pengguna melakukan pencatatan transaksi secara lebih terstruktur, cepat, dan mudah. Hasil dari implementasi aplikasi ini menunjukkan bahwa sistem mampu menyediakan fitur utama yang dibutuhkan dalam pengelolaan keuangan harian, mulai dari pencatatan pemasukan, pencatatan pengeluaran, pengelompokan kategori, hingga penyajian laporan keuangan sederhana.

Secara fungsional, aplikasi ini mampu menyimpan data transaksi pengguna ke dalam basis data MySQL yang terintegrasi melalui Visual Studio. Setiap transaksi yang dicatat akan tersimpan berdasarkan kategori, tanggal, dan jumlah nominal, sehingga pengguna dapat dengan mudah memahami pola pengeluaran dan pemasukan mereka. Selain itu, aplikasi juga menampilkan informasi keuangan dalam bentuk daftar transaksi serta visualisasi sederhana seperti grafik atau summary keuangan untuk mendukung proses pengambilan keputusan finansial.

4.2 Panduan Penggunaan

Panduan penggunaan aplikasi *Smart Budget Planner* menjelaskan langkah-langkah yang harus dilakukan pengguna untuk mengoperasikan seluruh fitur aplikasi, mulai dari proses pendaftaran akun, login, mengakses dashboard, menambahkan transaksi, melihat total pengeluaran per kategori, hingga mengelola profil pengguna. Penjelasan berikut disusun untuk membantu pengguna memahami alur penggunaan aplikasi secara menyeluruh.

4.2.1 Register

Halaman ini berfungsi sebagai "gerbang masuk" bagi pengguna baru. Tujuannya adalah mengumpulkan data identitas dasar pengguna yang belum memiliki akun agar mereka bisa mulai menggunakan aplikasi.

The image shows a registration form with four input fields: 'Name', 'Username', 'Email', and 'Password'. Below the fields is a large blue 'Register' button. Underneath the button is a small link 'Already Registered?'. The entire form is set against a light gray background.

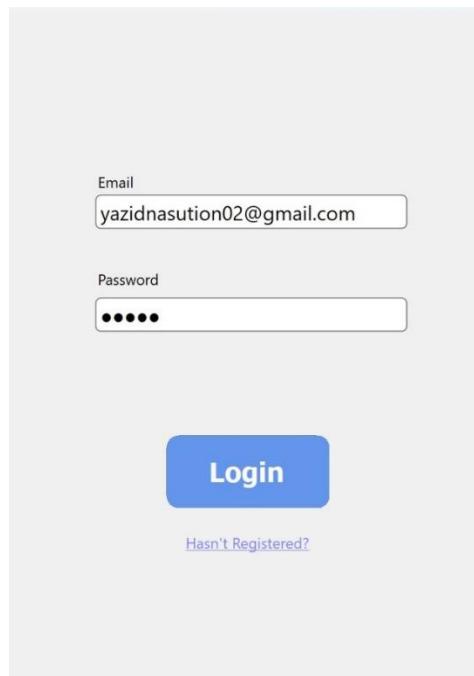
Gambar 4.1 Halaman Register

Tampilannya dibuat sangat sederhana biar pengguna enggak bingung. Ada empat hal penting yang harus diisi berurutan: Nama Lengkap, Username (nama panggilan unik untuk login), Email (buat urusan kontak dan keamanan), dan yang paling penting, Password (kata sandi rahasia).

Setelah semua diisi, pengguna tinggal menekan tombol besar berwarna biru, yaitu "Register". Tombol ini akan mengirim data ke sistem untuk diverifikasi. Kalau data oke semua, akun langsung jadi dan pengguna bisa langsung loncat ke menu utama aplikasi. Di bagian bawah ada tulisan kecil "Already Registered?" Ini adalah jalan pintas. Kalau pengguna tiba-tiba sadar, "Eh, ternyata aku sudah punya akun", mereka tinggal klik itu dan akan langsung dibawa kembali ke halaman Login tanpa perlu panik.

4.2.2 Login

Halaman ini adalah tempat pengguna yang sudah punya akun datang untuk mengakses semua fitur aplikasi. Ini adalah langkah validasi kedua setelah pendaftaran.



Gambar 4.2 Halaman Login

Halaman Login ini fungsinya pintu masuk utama buat pengguna lama. Kita cuma minta dua hal penting dari mereka: Email (atau kadang Username) dan Password. Begitu keduanya dimasukkan dan tombol Login ditekan, sistem langsung mencocokkan data. Kalau cocok, mereka langsung diantar masuk ke dalam aplikasi. Kalau ada yang belum punya akun, ada tulisan kecil di bawah, "Hasn't Registered?", yang bisa diklik untuk balik ke halaman Pendaftaran. halaman ini memastikan yang masuk cuma pengguna yang sah.

4.2.3 Dashboard

Dashboard merupakan halaman utama setelah pengguna berhasil login. Pada halaman ini, pengguna dapat melihat:

1. Total Pemasukan
2. Total Pengeluaran

3. Saran Otomatis berdasarkan kondisi keuangan (misalnya: pengeluaran lebih besar dari pemasukan)
4. Tombol (+) untuk membuka halaman input transaksi

Dashboard berfungsi sebagai ringkasan keuangan yang menampilkan kondisi finansial pengguna secara cepat dan informatif. Tombol + dapat ditekan untuk menambahkan pemasukan atau pengeluaran baru.



Gambar 4.3 Halaman Dashboard

4.2.4 Halaman Input Pemasukan / Pengeluaran

Halaman ini digunakan untuk menambahkan transaksi baru ke dalam aplikasi. Pengguna dapat memilih apakah transaksi tersebut merupakan pemasukan atau pengeluaran. Selain itu, halaman ini menyediakan beberapa kolom input, yaitu:

1. Jenis transaksi (Pemasukan / Pengeluaran)
2. Jumlah Nominal
3. Tanggal transaksi
4. Deskripsi transaksi

Setelah data diisi dengan lengkap, pengguna dapat menekan tombol tambah, dan transaksi akan disimpan ke dalam database serta ditampilkan pada dashboard dan riwayat transaksi.

Pemasukan

Input Pengeluaran

Tanggal
Sunday , November 23, 2025

Kategori
Pilih Kategori

Deskripsi

Tambah

Gambar 4.4 Halaman Input Pemasukan / Pengeluaran

4.2.5 Halaman Total Pengeluaran Berdasarkan Kategori

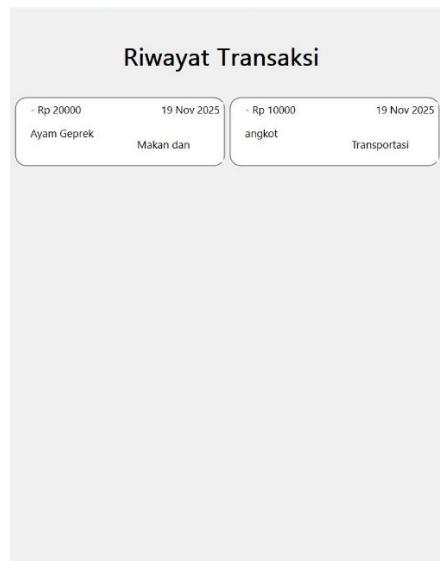
Halaman ini menampilkan total pengeluaran yang dikelompokkan berdasarkan kategori, seperti makan dan minum, belanja, hiburan, dan kategori lainnya yang tersedia dalam aplikasi. Setiap kategori menunjukkan jumlah total pengeluaran yang telah dicatat oleh pengguna. Halaman ini membantu pengguna mengetahui kategori mana yang paling banyak menghabiskan anggaran sehingga dapat mengatur keuangan lebih baik.



Gambar 4.5 Halaman Total Pengeluaran

4.2.6 Halaman Riwayat Transaksi

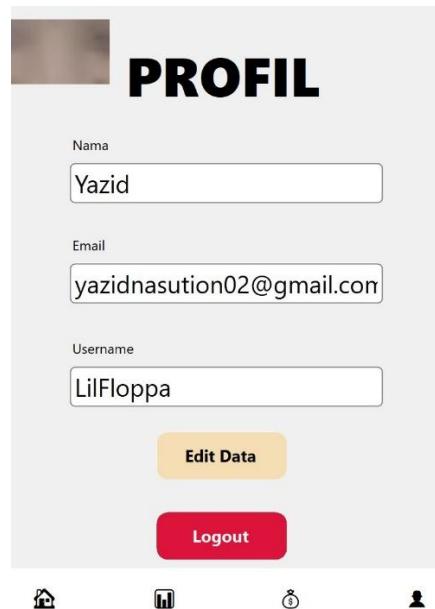
Riwayat transaksi menampilkan seluruh data pemasukan dan pengeluaran yang telah dicatat oleh pengguna dalam bentuk daftar kronologis. Pada halaman ini, pengguna dapat melihat detail transaksi seperti jenis transaksi, tanggal, kategori, deskripsi, dan nominal. Halaman ini memudahkan pengguna memantau aktivitas keuangan yang pernah dilakukan.



Gambar 4.6 Halaman Riwayat Transaksi

4.2.7 Halaman Profil

Halaman profil menampilkan informasi pengguna seperti nama, email, dan username. Selain itu, halaman ini menyediakan fitur edit profil, yang memungkinkan pengguna memperbarui data dirinya jika terdapat kesalahan atau ingin mengganti informasi tertentu.



Gambar 4.7 Halaman Profil

BAB 5

PENUTUP

5.1 Kesimpulan

Visual Studio merupakan salah satu *Integrated Development Environment* (IDE) yang sangat powerful dan banyak digunakan oleh pengembang perangkat lunak untuk membuat berbagai jenis aplikasi, mulai dari aplikasi desktop, web, hingga mobile. Dengan fitur lengkap seperti editor kode yang cerdas, *debugger* terintegrasi, *designer* antarmuka, serta dukungan berbagai bahasa pemrograman seperti C#, VB.NET, C++, dan Python, Visual Studio mampu mempercepat proses pengembangan aplikasi.

Kemampuan Visual Studio dalam mengintegrasikan framework modern seperti .NET, ASP.NET, WinForms, dan WPF membuat proses pembuatan aplikasi menjadi lebih terstruktur dan efisien. Selain itu, fitur seperti IntelliSense, *source control*, *extension marketplace*, dan integrasi dengan Azure membantu pengembang menghadirkan aplikasi berkualitas tinggi. Secara keseluruhan, Visual Studio berperan penting dalam mempermudah proses coding, pengujian, serta *deployment* aplikasi sehingga hasil akhir lebih stabil, aman, dan memenuhi kebutuhan pengguna.

5.2 Saran

Dalam menggunakan Visual Studio, sebaiknya pengguna mempelajari fitur-fitur dasarnya terlebih dahulu agar proses pengembangan aplikasi dapat berjalan lebih lancar. Manfaatkan fitur seperti IntelliSense, template project, dan *toolbox* untuk mempercepat pembuatan antarmuka maupun penulisan kode. Visual Studio juga menyediakan berbagai ekstensi yang bisa membantu meningkatkan produktivitas, sehingga pengguna dianjurkan untuk mengeksplorasinya melalui Visual Studio Marketplace.

Selain itu, penggunaan fitur debugging perlu dilakukan secara rutin agar setiap kesalahan dalam kode dapat ditemukan dan diperbaiki sejak awal. Jika bekerja dalam tim, Visual Studio menyediakan integrasi Git dan fitur kolaborasi seperti *Live Share* yang sangat membantu dalam menyelaraskan pekerjaan antaranggota.

DAFTAR PUSTAKA

- https://github.com/Asisten-Laboratorium-Teknik-Informatika/Semester5_Vispro_2025-2026/tree/main/4_A1
- Blazing, A. (2018). *Pemrograman Windows Dengan Visual Basic .NET: Praktikum Pemrograman VB.NET*. kliksolusi.com.
- Link: <https://id.scribd.com/document/402348057/CBR-B-INDO>
- Candra, D. G. A., & Pardika, P. P. (2024). Analisa dan perancangan sistem informasi manajemen pengolahan data pesanan sablon di KYSR Store menggunakan metode waterfall. *JAMI: Jurnal Ahli Muda Indonesia*, 5(2), 134–147.
- Link: <https://journal.akb.ac.id/index.php/jami/article/view/306>
- Creswell, J. W., & Creswell, J. D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches*. SAGE Publications.
- Link: https://www.ucg.ac.me/skladiste/blog_609332/objava_105202/fajlovi/Creswell.pdf
- Dafri, A. (2023). Digital finance applications and financial well-being. *International Journal of Research and Innovation in Social Science (IJRISS)*, 8(9).
- Link: <https://rsisinternational.org/journals/ijriss/articles/usage-of-digital-finance-applications-and-its-impact-on-financial-well-being-a-conceptual-framework/>
- Dorothy, L., Satoto, K. I., & Nurhayati, O. D. (2014). Perancangan dan implementasi sistem informasi perpustakaan di Program Studi Teknik Lingkungan Fakultas Teknik UNDIP. *Jurnal Teknologi dan Sistem Komputer*, 2(4), 209–222.
- Link: <https://jtsiskom.undip.ac.id/article/view/12593>
- Enterprise, J. (2015). *Pengenalan Visual Studio 2013*. Elex Media Komputindo.
- Link: <https://elexmedia.id/produk/detail/komputer/jubilee-enterprise/pengenalan-visual-studio-2013/9786020270487>
- Isman, R. K. (2017). Rancang bangun aplikasi barang jasa menggunakan C#, WPF dan SQL Server 2012. *Jurnal Teknologi Informatika dan Terapan*, 8.
- Link: <https://jtit.polije.ac.id/index.php/jtit/article/view/22>
- Kanwal, R., & Shahnawaz, M. (2017). Software testing techniques and strategies.
- Link: https://www.researchgate.net/publication/321883135_Software_Testing_Techniques_and_Strategies

- Kerner, A., & Goodyear, K. (2022). Motivations for personal financial management: A Self-Determination Theory perspective. *Frontiers in Psychology*, 13, 977818.
Link:<https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2022.977818/full>
- Khamis, M., & Mohamed, N. (2024). Usage of digital finance applications and its impact on financial well-being: A conceptual framework. *International Journal of Research and Innovation in Social Science (IJRISS)*, 8(9).
Link:<https://rsisinternational.org/journals/ijriss/articles/usage-of-digital-finance-applications-and-its-impact-on-financial-well-being-a-conceptual-framework/>
- Kumar, R. (2019). *Research methodology: A step-by-step guide for beginners* (5th ed.). SAGE Publications.
Link: <https://uk.sagepub.com/en-gb/eur/research-methodology/book257990>
- Mathur, A. P., & Malik, R. (2018). *Fundamentals of software testing*. Pearson Education.
Link:<https://www.cs.purdue.edu/homes/apm/FoundationsBookSecondEdition/Slides/ConsolidatedSlides.pdf>
- Parinsi, M. T., Mewengkang, A., & Rantung, T. (2021). Perancangan sistem informasi sekolah di Sekolah Menengah Kejuruan. *Edutik: Jurnal Pendidikan Teknologi Informasi dan Komunikasi*, 1(3), 227–240.
Link:https://www.researchgate.net/publication/367772512_PERANCANGAN_SISTEM_INFORMASI_SEKOLAH_DI_SEKOLAH_MENENGAH_KEJURUAN
- Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner's approach*. McGraw-Hill Education.
Link:<https://whyphi.staff.telkomuniversity.ac.id/files/2016/01/ebook-pressman-sw-engineering.pdf>
- Sanggup, B. (2015). *Pendidikan Bahasa Indonesia*. Medan: Unimed Press.
Link: <https://id.scribd.com/document/402348057/CBR-B-INDO>
- Sidharta, K. (2020). Studi efisiensi sumber daya terhadap efektivitas penggunaan database: Studi kasus SQL Server dan MySQL. *Conference on Business*,

Social Sciences and Innovation Technology, 1(1), 508–515.

Link: <https://journal.uib.ac.id/index.php/cbssit/article/view/1455>

Sophian, S. (2014). Pengimplementasian dan perancangan sistem informasi penjualan dan pengendalian stok barang pada Toko Swastika Servis (SS) dengan Visual Basic 6.0 dan MySQL. *Jurnal Momentum*, 16(2), 34–44.

Link: <https://download.garuda.kemdikbud.go.id/article.php?article=178673&val=5448>

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

Link: <https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf>

Syakir, A. A. (2015). Bahasa pemrograman C# berbasis Windows Application menggunakan SharpDevelop 4.4. STMIK PGRI.

Link: <https://abidalfansyakir.wordpress.com/wp-content/uploads/2015/12/jurnal-ilmiah.pdf>

Tutorial Resmi NuGet. (n.d.). *Install and use a package in Visual Studio*.

Link: <https://learn.microsoft.com/id-id/nuget/quickstart/install-and-use-a-package-in-visual-studio>