

PEMROGRAMAN C++ APLIKASI DANA CERDAS

(Makalah ini disusun untuk memenuhi tugas Praktikum Algoritma dan Pemrograman)

Dosen Pengampu: Agus Nugraha, S.T., M.Kom.

Asisten Laboratorium: Sevi Nurafni S.T., M.Si.



Oleh Kelompok 2:

1. Najlia Intani (2C2230010)
2. Nadine Valia Azzahra (2C2230007)
3. Farhan Darmawan (2C2230017)
4. Yusuf Sohibul Falah (2C2230013)

PROGRAM STUDI ALGORITMA DAN PEMROGRAMAN

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS KOPERASI INDONESIA

JATINANGOR

2024

KATA PENGANTAR

Dengan mengucapkan syukur kehadiran Tuhan YME, atas segala kebesaran dan limpahan nikmat yang diberikan-Nya, sehingga kami dapat menyelesaikan makalah “Pemrograman C++ *Aplikasi Dana Cerdas*”.

Adapun penulisan makalah ini bertujuan untuk memenuhi tugas praktikum laboratorium pada mata kuliah Algoritma dan Pemrograman di Ikopin University. Disamping dapat menuangkan gagasan dalam bentuk tulisan, melalui makalah ini kami juga dapat berlatih menjadi insan peneliti di masa depan.

Dalam penulisan makalah ini, berbagai hambatan telah kami alami. Oleh karena itu, terselesaikan makalah ini tentu saja bukan karena kemampuan kami semata-mata. Namun karena adanya dukungan dan bantuan dari pihak-pihak yang terkait.

Sehubungan dengan hal tersebut, perlu kiranya kami dengan ketulusan hati mengucapkan terima kasih kepada Bapak Agus Nugraha, ST., M. Kom. selaku Dosen pada mata kuliah Algoritma dan Pemrograman dan Ibu Sevi Nurafni selaku Asisten Laboratorium serta teman-teman kelompok 2 Prodi Sains Data di Ikopin University.

Dalam penyusunan makalah ini, kami menyadari pengetahuan dan pengalaman kami masih sangat terbatas. Oleh karena itu, kami sangat mengharapkan adanya kritik dan saran dari berbagai pihak agar makalah ini lebih baik dan bermanfaat.

Akhir kata, semoga makalah ini dapat bermanfaat dan berguna bagi banyak pihak terutama untuk pengembangan ilmu pengetahuan.

Bandung, 20 Januari 2024

Penulis

ABSTRAK

Penelitian ini dilatarbelakangi oleh kondisi dimana seseorang menyimpan uang di banyak platform, sehingga dibutuhkan sebuah alat yang dapat menjadi arsip keuangan yang datanya dapat terus di update dengan cepat dan tepat. Untuk merespons hal itu, maka peneliti melakukan penelitian yang bertujuan untuk membuat sebuah program aplikasi pengelola keuangan yang diberi nama 'Dana Cerdas'. Tujuan yang hendak dicapai pada penelitian ini antara lain: (1) Mengetahui bentuk penulisan bahasa program bahasa yang paling tepat untuk aplikasi 'Dana Cerdas' (2) Mengetahui implementasi keberhasilan program.

Jenis penelitian ini adalah metode analisis, desain, coding, testing, implementation, dan juga maintenance. Adapun hasil penelitian ini adalah: (1) Terbentuknya penulisan bahasa program bahasa untuk aplikasi 'Dana Cerdas' (2) Implementasi keberhasilan program aplikasi 'Dana Cerdas' berhasil melalui uji coba terhadap contoh kasus.

Kata Kunci: Program, Aplikasi, Keuangan.

ABSTRACT

This research is undermined by a condition where someone stores money on multiple platforms, so it requires a tool that can be a financial archive whose data can be kept updated quickly and accurately. To respond to that, the researchers undertook a research aimed at creating a financial manager's application program called 'Dana Cerdas'. The objectives of this research include: (1) Knowing the language writing form of the language program which is most appropriate for the application of 'Dana Cerdas' (2) Knowing successful implementation of the program.

This type of research is methods of analysis, design, coding, testing, implementation, and also maintenance. The results of this study are: (1) Formation of language program writing for the application 'Dana Cerdas' (2) Successful implementation of the application program 'Smarter Data' successfully through trial on examples of cases.

Keywords: Programs, Applications, Finance.

DAFTAR ISI

	Halaman
KATA PENGANTAR	i
ABSTRAK	iii
DAFTAR ISI	iii
DAFTAR TABEL	iv
DAFTAR GAMBAR	v
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
2.1 Batasan Masalah	1
BAB II DASAR TEORI	2
2.1 Konsep Dasar C++	2
2.2 Struktur C++	2
2.3 File Header	3
2.4 Tipe Data	4
2.5 Variabel	4
2.6 Label	4
2.7 Penyeleksian Kondisi	4
2.8 Fungsi	5
2.9 Operasi Perulangan (Looping)	5
2.10 Algoritma	6
2.11 Flowchart	7
2.12 Aplikasi	8
BAB III ALGORITMA DAN FLOWCHART	9
3.1 Algoritma Penelitian	9
3.2 Flowchart Penelitian	9
3.3 Algoritma Program	10
3.4 Flowchart Program	10
BAB 4 IMPLEMENTASI	11
4.1 Implementasi Sistem	11
4.2 Penulisan Program Aplikasi.....	11
4.3 Hasil Akhir Penulisan Program (Source Code)	11
4.4 Hasil Pengujian Bahasa Program	15
BAB 5 PENUTUP	22
5.1 Simpulan	22
5.2 Saran	22
DAFTAR PUSTAKA	23

DAFTAR TABEL

	Halaman
Tabel 2.1 Tipe Data	4
Tabel 4.1 Data Keuangan Putri	17
Tabel 4.2 Data Keuangan Pegawai Toko Bangunan Jaya Bakti	18

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Bentuk Umum Struktur C++	2
Gambar 2.2 Label	4
Gambar 2.3 Bentuk Umum Looping For	5
Gambar 2.4 Bentuk Umum Looping While	6
Gambar 2.5 Bentuk Umum Looping While-Do	6
Gambar 2.6 Simbol-simbol Flowchart Yang Umum Digunakan	7
Gambar 3.1 Flowchart Penelitian	9
Gambar 3.2 Flowchart Program	10
Gambar 4.1 Output Program Aplikasi Dana Cerdas	15
Gambar 4.2 Menampilkan informasi saldo keuangan Bulanan Putri	19
Gambar 4.3 Menampilkan data bulanan Putri telah disimpan	19
Gambar 4.4 Menampilkan pengguna Putri telah keluar dari program	20
Gambar 4.5 Menampilkan informasi saldo keuangan Toko Jaya Bakti	20
Gambar 4.6 Menampilkan data Toko Jaya Bakti telah disimpan	20
Gambar 4.7 Menampilkan pegawai Toko Jaya Bakti telah keluar dari program ...	21

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang terus berkembang beriringan dengan semakin majunya perekonomian dunia, dimana kita saat ini telah memasuki era revolusi industri 5.0. Oleh sebab itu, penting bagi setiap manusia untuk dapat menguasai berbagai ilmu pengetahuan terutama yang memiliki ikatan dengan teknologi. Dengan adanya teknologi, digitalisasi menjadi hal yang terus ditingkatkan kecanggihannya dalam segala bidang untuk membantu kehidupan manusia menjadi lebih mudah, efisien, dan praktis.

Uang merupakan bagian terpenting dalam kehidupan manusia. Dengan uang, manusia memiliki alat transaksi untuk memenuhi segala kebutuhan hidupnya. Kecanggihan teknologipun masuk ke ranah ekonomi keuangan, dimana setiap Bank sudah memiliki aplikasi digital, dan sudah banyak orang yang menggunakan berbagai macam platform E-Money. Akan tetapi, setiap orang biasanya menyimpan uang di banyak platform, sehingga sulit untuk menghitung pemasukan dan pengeluaran. Dimana hal ini merupakan latar belakang dari permasalahan yang ada.

Aspek tersebutlah yang mendorong kami untuk menjadikannya sebagai sebuah tema dalam pembuatan rancangan program C++. Pembuatan sebuah program Aplikasi pengelolaan keuangan yang kami beri nama 'Dana Cerdas' untuk memudahkan setiap penggunaanya dalam mengelola arsip keuangan harian, bulanan, sampai jangka waktu yang diinginkan dan dapat terus di update datanya. Sehingga data-data yang diperoleh dapat menjadi sebuah evaluasi pengguna untuk penggunaan keuangan kedepannya ataupun sebagai arsip keuangan. Dengan ilmu yang dimiliki, kami berusaha untuk merancang program pengelola keuangan yang diharapkan dapat memudahkan pendataan keuangan menjadi lebih cepat dan tepat.

1.2 Batasan Masalah

Agar makalah ini dapat dipahami secara terfokus dan mendalam maka penulis memandang permasalahan yang ada harus diberikan sebuah batasan terhadap variable. Oleh karena itu, penulis membatasi hanya berkaitan dengan "Konsep dasar C++, Struktur C++, File Header, Type Data, Variabel, Label, Operasi Penyeleksian kondisi, Fungsi, Operasi Perulangan, Algoritma, Flowchart, Aplikasi". Penempatan sebuah masalah dan Profiling pada program dalam Borland C++.

BAB II

DASAR TEORI

2.1 Konsep Dasar C++

Menurut buku Cepat Menguasai C++ terbitan Jubilee Enterprise (2015: 10), C++ adalah bahasa pemrograman hasil pengembangan bahasa C. Jika dilihat dari sejarahnya, bahasa C++ ini muncul di tahun 1970-an. Bahasa ini dikembangkan oleh seseorang bernama Bjarne Stroustrup. C++ merupakan subset dari bahasa C, maka pustaka yang ditulis dalam bahasa C dapat dipergunakan dalam bahasa C++.

Bahasa C++ digolongkan ke dalam Bahasa tingkat menengah (Middle Level Programming Language). Kode C++ bersifat *case sensitive*, artinya membedakan antara huruf kapital dengan huruf kecil. Keistimewaan dari bahasa C++ adalah karena bahasa ini mendukung pemrograman berarah objek atau yang lebih dikenal dengan istilah *Objek Oriented Programming* (OOP). Dengan *Objek Oriented Programming* (OOP), mempermudah programmer menulis program, mempercepat proses pembuatan program, dan mempermudah pemeliharaan program.

2.2 Struktur C++

Struktur program C++ terdiri dari sejumlah blok fungsi, setiap fungsi terdiri dari satu atau beberapa pernyataan yang melaksanakan tugas tertentu.

Bentuk umum:

```
#include <file-header>

Main ()
{
    Pernyataan;
}
```

Gambar 2.1 Bentuk Umum Struktur C++

Preprocessor directive: Perintah dalam bahasa C++ yang akan membuat dan mengkompilasi file perantara dengan program utama.

Deklarasi global: Bagian program C++ tempat pendefinisian konstanta, variable, fungsi atau header fungsi, tipe data baru, atau struktur data yang memiliki sifat global.

Fungsi utama: Fungsi utama yang akan dieksekusi oleh compiler C++. Compiler C++ hanya akan mengeksekusi fungsi-fungsi lain yang dipanggil pada fungsi utama atau yang memiliki hubungan dengan fungsi utama secara tidak langsung.

Fungsi: Tempat pendefinisian header fungsi yang telah dideklarasikan pada bagian deklarasi global. Setiap fungsi dideklarasikan dengan diawali nama fungsi dan diikuti dengan memuat di dalamnya perintah/instruksi berupa statement-statement untuk melakukan pekerjaan pengolahan data yang dapat diberikan kepadanya, ditulis dengan menggunakan tanda awal { dan diakhiri dengan }.

Penggunaan Text Color:

1. Light young green; Kode angka: 10
2. Red; Kode angka: 12
3. Light young blue; Kode angka: 11
4. Yellow; Kode angka: 14

2.3 File Header

File header adalah file yang berisi deklarasi untuk berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar input/output maupun sebagai syntax. File header dalam C++ digunakan untuk memanggil library-library yang ada sehingga suatu fungsi dapat digunakan secara baik dan benar.

File header yang kami gunakan adalah:

1. `#include <iostream>`:

Ini digunakan agar program dapat melakukan operasi input dan output standar, seperti menampilkan pesan ke layar atau membaca input dari pengguna.

2. `#include <iomanip>`:

Digunakan untuk mengatur tampilan output, seperti membuat tabel dengan lebar kolom tertentu atau mengatur jumlah desimal pada angka.

3. `#include <vector>`:

Memungkinkan penggunaan struktur data vektor, yang mirip dengan daftar dinamis. Ini memudahkan pengelolaan banyak data.

4. `#include <fstream>`:

Berguna untuk bekerja dengan file, misalnya, menyimpan data ke file atau membaca data dari file.

2.4 Tipe Data

Tipe data adalah suatu media atau memori pada komputer yang digunakan untuk menampung. Berikut adalah jenis-jenis dari tipe data yang kami gunakan:

Tipe Data	Keterangan	Ukuran Nilai	Jangkauan
Int	Integer/Bilangan bulat	4 bytes/32 bit	Signed: -2147483648 s/d 2147483648 Unsigned: 0 s/d 4294967295
String	String	4 bytes/32 bit	%s
Void	Tidak bertipe	1 bytes/8 bit	-
Double	Pecahan presisi ganda	8 bytes	+/- 1.7e+/-308(~15 digits)

Tabel 2.1 Tipe Data

2.5 Variabel

Variabel adalah suatu tempat menampung data atau konstanta dimemori yang mempunyai nilai atau data yang dapat berubah-ubah selama proses program. Variabel dibagi menjadi dua macam, yaitu variable numerik dan variable teks.

2.6 Label

Label merupakan suatu pengenalan (identifier) yang diikuti dengan tanda titik dua (:), dimana label selalu berkaitan dengan *pernyataan goto* yang merupakan instruksi untuk mengarahkan eksekusi program ke pernyataan yang diawali dengan suatu label. Bentuk pemakaian goto sebagai berikut:

Goto label;

Gambar 2.2 Label

2.7 Penyeleksian Kondisi

Penyeleksian kondisi atau pernyataan percabangan digunakan untuk memecahkan persoalan untuk mengambil suatu keputusan diantara sekian pernyataan yang ada.

Untuk keperluan pengambilan keputusan, Borland C++ menyediakan beberapa perintah, yaitu:

1. If

Pernyataan if mempunyai pengertian “Jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan”.

2. If-Else

Pernyataan if-else mempunyai pengertian “Jika kondisi bernilai benar, maka perintah-1 akan dikerjakan dan jika tidak memenuhi syarat maka akan mengerjakan perintah-2”.

2.8 Fungsi

Fungsi digunakan untuk membagi program berdasarkan bagian-bagian tertentu yang memiliki tugasnya masing-masing. Fungsi tidak akan berjalan sendiri-sendiri namun bekerja atas dasar panggilan atau koordinasi fungsi utama dan fungsi main ().

2.9 Operasi Perulangan (Looping)

Dalam bahasa C++ looping digunakan untuk melakukan proses yang berulang-ulang sebanyak keinginan yang diinginkan menggunakan perintah. Struktur looping mempunyai bentuk yang bermacam-macam. Sebuah/kelompok instruksi diulang untuk jumlah pengulangan tertentu. Baik yang terdefinisi sebelumnya ataupun tidak. Struktur pengulangan terdiri atas Kondisi pengulangan yaitu Ekspresi Boolean yang harus dipenuhi untuk melaksanakan pengulangan dan Isi atau badan pengulangan yaitu satu atau lebih pernyataan (aksi) yang akan diulang.

1. Statement For

Struktur pengulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur pengulangan for tampaknya lebih efisien karena susunannya lebih simple dan sederhana. Pada umumnya looping yang digunakan oleh statement for telah diketahui batas awal, syarat looping dan perubahannya. Selama kondisi terpenuhi, maka pernyataan akan terus dieksekusi.

Bentuk umum looping for:

```
for (inisialisasiNilai;SyaratPerulangan;PerubahanNilai)
{
    Statement yang diulang;
}
```

Gambar 2.3 Bentuk Umum Looping For

2. Statement While

Perulangan while banyak digunakan pada program yang terstruktur. Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar (tidak sama dengan 0) dan akan berhenti bila kondisinya bernilai salah (sama dengan 0).

Bentuk umum looping while:

```
While (syarat pengulangan)
{
    Statement yang dijalankan;
    Statement control;
}
```

Gambar 2.4 Bentuk Umum Looping While

3. Statement Do-While

Pada do-while statement perulangannya dilakukan terlebih dahulu baru kemudian di cek kondisinya. Akibat dari hal ini adalah do-while minimal terdapat satu kali perulangan.

Bentuk umum looping Do-While:

```
do
{
    Blok Pernyataan;
}
while(kondisi);
```

Gambar 2.5 Bentuk Umum Looping While-Do

2.10 Algoritma

Algoritma menurut (Kani, 2020, 1.19) adalah suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu.

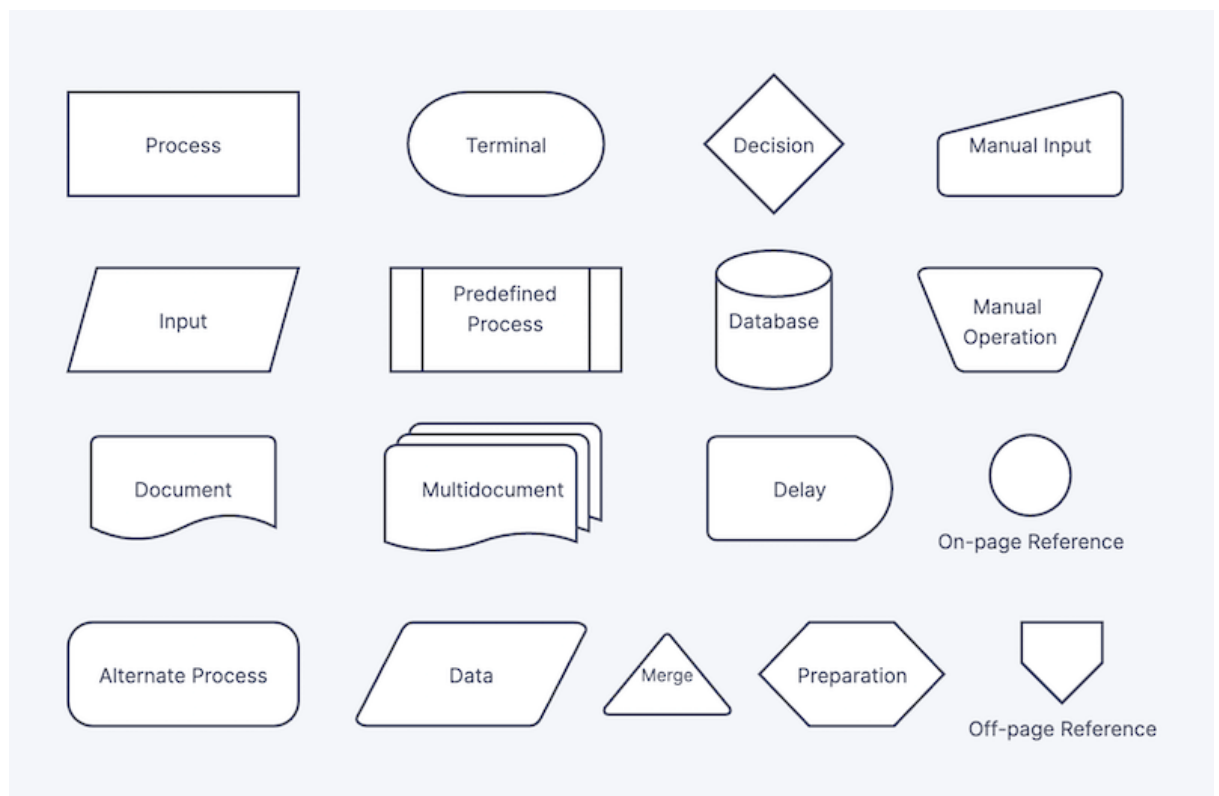
Algoritma berasal dari kata algoris dan ritmis yang pertama kali diperkenalkan oleh Abu Ja'far Muhammad Ibn Musa Al Khwarizmi pada 825 M di dalam buku Al-Jabr Wa-al Muqabla.

Dalam bidang pemrograman, algoritma merupakan sekumpulan instruksi atau langkah-langkah yang dituliskan secara sistematis dan digunakan untuk menyelesaikan masalah atau persoalan logika dan matematika dengan bantuan komputer (Sismoro, 2005, 29).

2.11 Flowchart

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. Fungsi utama dari flowchart adalah memberi gambaran jalannya sebuah program dari satu proses ke proses lainnya. Sehingga, alur program menjadi mudah dipahami. Dibagi menjadi lima jenis flowchart yakni dokumen, program, proses, system dan skematik.

Simbol-simbol flowchart:



Gambar 2.6 Simbol-simbol Flowchart Yang Umum Digunakan

Sumber: ZenFlowchart

Berikut simbol flowchart yang kami gunakan:

1. Oval (Lingkaran): Mewakili awal atau akhir dari program.
2. Rectangle (Persegi Panjang): Mewakili proses atau langkah dalam algoritma.
3. Diamond (Belah ketupat): Mewakili keputusan atau kondisi yang memerlukan pilihan.
4. Parallelogram (Jajar genjang): Mewakili input/output, seperti membaca data atau menampilkan informasi.
5. Panah: Menggambarkan arah aliran dari satu langkah ke langkah berikutnya.

2.12 Aplikasi

Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer (Wardana, 2010).

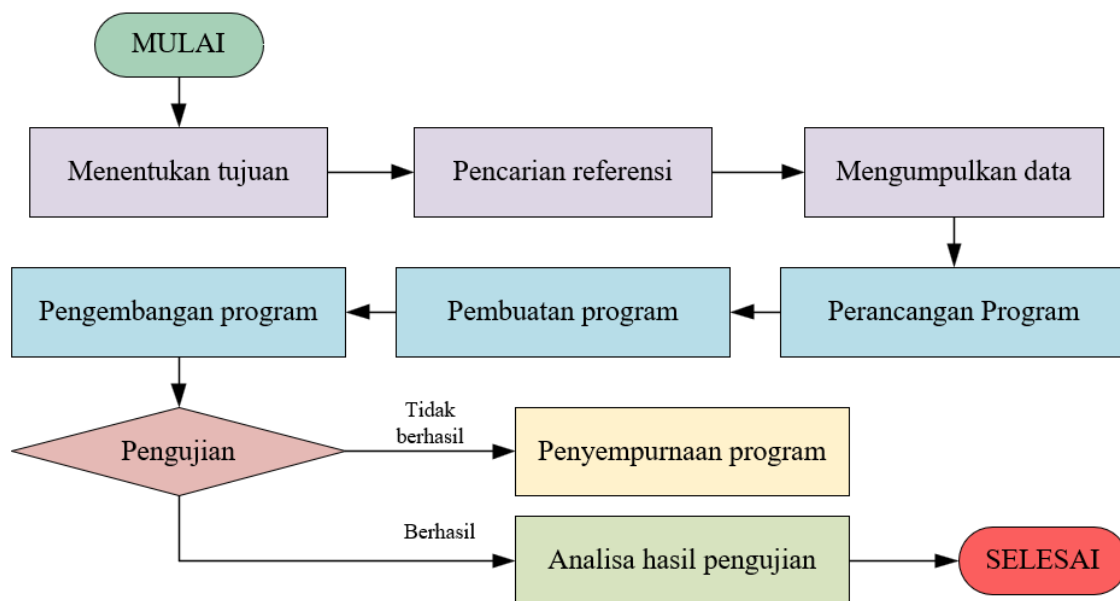
BAB III

ALGORITMA DAN FLOWCHART

3.1 Algoritma Penelitian

- Mulai.
- Menentukan tujuan pembuatan dari aplikasi pengelola keuangan seperti membantu mengelola keuangan pribadi menjadi terkelola dengan baik, teratur, dan valid.
- Peneliti mencari referensi melalui internet dan sumber lainnya, setelah mendapatkan referensi yang cukup, peneliti mengumpulkan data-data yang diperlukan.
- Peneliti melakukan perancangan program, setelah rancangan selesai, peneliti mulai melakukan pembuatan program.
- Peneliti mengembangkan program dengan menggunakan bahasa pemrograman yang sesuai.
- Ketika program telah selesai dibuat, peneliti melakukan uji coba program yang telah dibuat, apabila pengujian yang dilakukan berhasil, penelitian melakukan analisa hasil pengujian.
- Selesai

3.2 Flowchart Penelitian

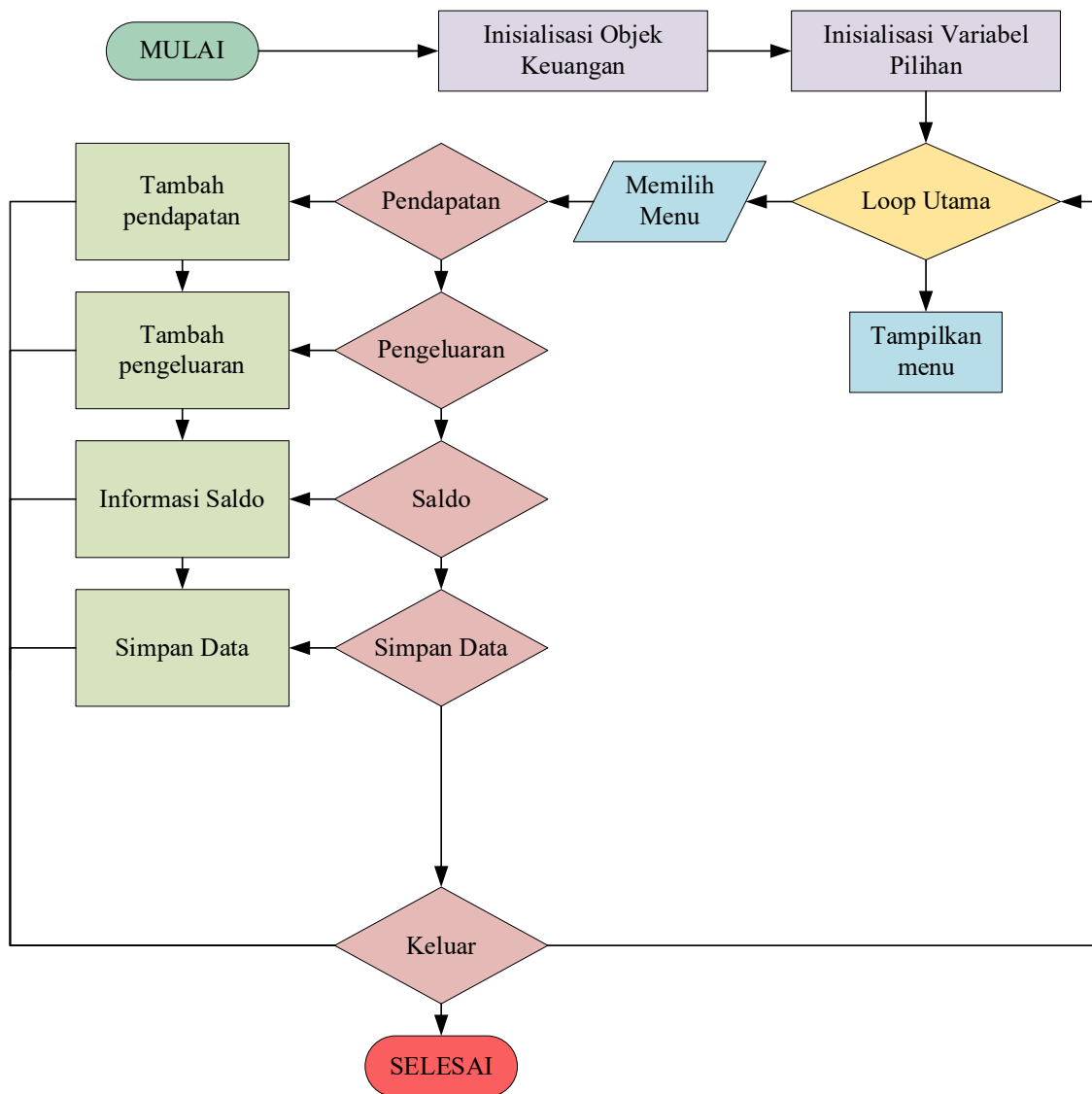


Gambar 3.1 Flowchart Penelitian

3.3 Algoritma Program

- Mulai.
- Inisialisasi Objek Keuangan.
- Inisialisasi Variabel Pilihan.
- Menentukan apakah program harus tetap berada dalam loop utama.
- Menampilkan opsi menu untuk pengguna.
- Meminta pengguna memilih menu.
- Memproses pilihan pengguna.
- Program akan terus berjalan dalam loop sampai pengguna memilih untuk keluar.
- Selesai.

3.4 Flowchart Program



Gambar 3.2 Flowchart Program

BAB IV

IMPLEMENTASI

4.1 Implementasi Sistem

Implementasi sistem adalah prosedur-prosedur yang dilakukan dalam mencoba hasil konsep desain pemrograman aplikasi yang telah dirancang sebelumnya. Tahap ini bertujuan untuk menguji hasil sistem yang telah selesai dibuat, disamping itu akan dihasilkan analisis yang berkaitan dengan hasil pengujian sistem secara keseluruhan.

4.2 Penulisan Program Aplikasi

Penulisan program aplikasi Dana Cerdas bertujuan sebagai pengelola keuangan secara cepat dan tepat. Pada rangkaian programnya, aplikasi Dana Cerdas dapat menghitung pemasukan dan pengeluaran secara rinci lengkap dengan sumber/tujuan dan keterangan nilai besarannya. Program ini juga bisa menginput banyak data sekaligus dengan hasil akhir pengguna bisa melihat informasi saldo.

4.3 Hasil Akhir Penulisan Program (Source Code)

```
1  #include <iostream>
2  #include <iomanip>
3  #include <vector>
4  #include <fstream>
5
6  using namespace std;
7
8  // Struktur untuk menyimpan informasi transaksi keuangan
9  struct Transaction {
10     string type;        // "Pendapatan" atau "Pengeluaran"
11     string name;        // Nama transaksi
12     double amount;      // Nominal transaksi
13     string category;    // Kategori transaksi
14 };
15
16 // Kelas untuk merepresentasikan manajemen keuangan
17 class FinancialManagement {
18 private:
19     double totalIncome;
20     double totalExpense;
21     double balance;
22     vector<Transaction> transactions;
23
24 public:
25     // Konstruktor untuk inisialisasi manajemen keuangan
```

```

26     FinancialManagement() : totalIncome(0.0), totalExpense(0.0),
    balance(0.0) {}
27
28     // Fungsi untuk menambah transaksi pendapatan
29     void addIncome() {
30         Transaction transaction;
31         transaction.type = "Pendapatan";
32         cout << "Masukkan nama pendapatan: ";
33         cin.ignore();
34         getline(cin, transaction.name);
35         cout << "Masukkan nominal pendapatan: ";
36         cin >> transaction.amount;
37         cout << "Masukkan kategori pendapatan: ";
38         cin.ignore();
39         getline(cin, transaction.category);
40         transactions.push_back(transaction);
41         totalIncome += transaction.amount;
42         balance += transaction.amount;
43     }
44
45     // Fungsi untuk menambah transaksi pengeluaran
46     void addExpense() {
47         Transaction transaction;
48         transaction.type = "Pengeluaran";
49         cout << "Masukkan nama pengeluaran: ";
50         cin.ignore();
51         getline(cin, transaction.name);
52         cout << "Masukkan nominal pengeluaran: ";
53         cin >> transaction.amount;
54         cout << "Masukkan kategori pengeluaran: ";
55         cin.ignore();
56         getline(cin, transaction.category);
57         transactions.push_back(transaction);
58         totalExpense += transaction.amount;
59         balance -= transaction.amount; // Ubah menjadi negatif
60     }
61
62     // Fungsi untuk mencetak data keuangan dalam bentuk tabel
63     void printFinancialData() const {
64         cout << setw(15) << "Jenis" << setw(20) << "Nama" << setw(15)
    << "Nominal" << setw(20) << "Kategori" << endl;
65         cout << setfill('-') << setw(70) << "-" << setfill(' ') <<
    endl;
66
67         for (const Transaction& transaction : transactions) {
68             cout << setw(15) << transaction.type << setw(20) <<
    transaction.name << setw(15) << transaction.amount
69             << setw(20) << transaction.category << endl;

```

```

70     }
71
72     cout << setfill('-') << setw(70) << "-" << setfill(' ') <<
endl;
73     cout << setw(35) << "Total Pendapatan:" << setw(15) <<
totalIncome << endl;
74     cout << setw(35) << "Total Pengeluaran:" << setw(15) <<
totalExpense << endl; // Ubah menjadi negatif
75     cout << setw(35) << "Saldo Akhir:" << setw(15) << balance <<
endl;
76 }
77
78 // Fungsi untuk menyimpan data keuangan ke dalam file
79 void saveData() const {
80     ofstream file("financial_data.txt");
81
82     if (file.is_open()) {
83         file << "Jenis\tNama\tNominal\tKategori\n";
84         file << setfill('-') << setw(70) << "-" << setfill(' ') <<
endl;
85
86         for (const Transaction& transaction : transactions) {
87             file << transaction.type << "\t" << transaction.name <<
"\t" << transaction.amount << "\t"
88                 << transaction.category << "\n";
89         }
90
91         file << setfill('-') << setw(70) << "-" << setfill(' ') <<
endl;
92         file << "Total Pendapatan:\t" << totalIncome << "\n";
93         file << "Total Pengeluaran:\t" << totalExpense << "\n"; //
Ubah menjadi negatif
94         file << "Saldo Akhir:\t" << balance << "\n";
95
96         cout << "Data berhasil disimpan ke dalam file
'financial_data.txt'.\n";
97         file.close();
98     } else {
99         cout << "Gagal membuka file untuk menyimpan data.\n";
100     }
101 }
102};
103
104int main() {
105     FinancialManagement financeManager;
106
107     char choice;
108

```

```

109     do {
110         cout << "Menu:\n";
111         cout << "1. Tambah Pendapatan\n";
112         cout << "2. Tambah Pengeluaran\n";
113         cout << "3. Informasi Saldo\n";
114         cout << "4. Simpan Data\n";
115         cout << "5. Exit\n";
116         cout << "Pilih: ";
117         cin >> choice;
118
119         switch (choice) {
120             case '1':
121                 financeManager.addIncome();
122                 break;
123
124             case '2':
125                 financeManager.addExpense();
126                 break;
127
128             case '3':
129                 financeManager.printFinancialData();
130                 break;
131
132             case '4':
133                 financeManager.saveData();
134                 break;
135
136             case '5':
137                 cout << "Keluar dari program.\n";
138                 break;
139
140             default:
141                 cout << "Pilihan tidak valid.\n";
142         }
143     } while (choice != '5');
144
145     return 0;
146 }

```

4.4 Hasil Pengujian Bahasa Program

```
Menu:  
1. Tambah Pendapatan  
2. Tambah Pengeluaran  
3. Informasi Saldo  
4. Simpan Data  
5. Exit  
Pilih:
```

Gambar 4.1 Ouput Program Aplikasi Dana Cerdas

Cara kerja Aplikasi Dana Cerdas:

1. Program dimulai.
2. Objek keuangan diinisialisasi.
3. Variabel pilihan diinisialisasi.
4. Menentukan apakah program harus tetap berada dalam loop utama.
 - a. Menampilkan Menu Utama: Menampilkan opsi menu untuk pengguna.
 - b. Meminta pengguna memilih menu: Meminta pengguna memilih menu.
 - c. Memproses pilihan pengguna menggunakan switch case.
 - Jika pilihan 1: Panggil metode tambah pendapatan.
 - Jika pilihan 2: Panggil metode tambah pengeluaran.
 - Jika pilihan 3: Panggil metode informasi saldo.
 - Jika pilihan 4: Panggil metode simpan data.
 - Jika pilihan 5: Keluar dari loop.
 - Jika pilihan tidak valid: Tampilkan pesan “Pilihan tidak valid”.
5. Program akan terus berjalan dalam loop sampai pengguna memilih untuk keluar.
6. Program selesai.

Dalam pemrograman berorientasi objek, objek Keuangan dapat direpresentasikan sebagai instance dari sebuah kelas. Kelas Keuangan dapat memiliki metode-metode untuk menambah pendapatan, menambah pengeluaran, memberikan informasi saldo, menampilkan laporan, dan menyimpan data. Program ini menggunakan loop utama untuk terus berjalan hingga pengguna memilih untuk keluar. Loop ini memungkinkan program menjalankan berbagai operasi berulang kali tanpa harus menulis ulang kode yang sama.

Struktur switch case digunakan untuk memproses pilihan pengguna. Ini memungkinkan program memilih tindakan yang sesuai berdasarkan input pengguna. Dalam hal ini, pilihan 1-6 mencakup berbagai operasi yang dapat dilakukan oleh pengguna.

Metode Tambah Pendapatan dan Tambah Pengeluaran ini digunakan untuk menambahkan pendapatan dan pengeluaran ke dalam objek Keuangan. Ini mencerminkan konsep dasar penambahan data ke dalam suatu entitas.

Metode Info Saldo dan Tampilkan Laporan ini memberikan informasi kepada pengguna, seperti saldo saat ini atau laporan keuangan. Mereka memberikan output yang relevan berdasarkan data yang disimpan dalam objek Keuangan. Selanjutnya, metode simpan bertanggung jawab untuk menyimpan data keuangan ke dalam suatu tempat penyimpanan, seperti file atau database. Ini mencerminkan kebutuhan untuk menyimpan dan memulihkan data di antara sesi-sesi program.

Dengan variabel pilihan, pengguna diminta untuk memasukkan pilihan mereka, yang kemudian diproses dalam struktur switch case. Variabel pilihan ini digunakan untuk mengarahkan alur program ke operasi yang sesuai. Dalam switch case, terdapat penanganan default untuk menangani situasi di mana pengguna memasukkan pilihan yang tidak valid. Pesan "Pilihan tidak valid" akan ditampilkan untuk memberikan umpan balik kepada pengguna.

Supaya hasil pengujian bahasa program dapat dijalankan, maka perlu adanya sebuah permasalahan atau kasus untuk menjadi variabel pembuktiannya. Berikut kami berikan dua contoh kasus:

1. Putri adalah seorang mahasiswi Universitas Koperasi Indonesia yang akan segera masuk kuliah dan tinggal dengan sewa kos di jalan GKPN. Karena ini adalah pertama kalinya putri sewa kos, maka untuk satu bulan pertama putri ingin melakukan pencatatan keuangan atas setiap pemasukan dan pengeluaran sebagai bahan evaluasi untuk bulan-bulan selanjutnya.

Setelah satu bulan, putri telah mencatat data seluruh pengeluaran dan pemasukannya sebagai berikut:

Tanggal	Keterangan	Pemasukan	Pengeluaran
1/1/24	Uang bulanan	Rp. 5.000.000	
	Peralatan kuliah		Rp. 100.000
	Stok makanan minggu pertama		Rp. 200.000
2/1/24	Hangout Bersama teman		Rp. 50.000

3/1/24	Bayar kartu perpustakaan		Rp. 25.000
	Makan di kantin		Rp. 25.000
4/1/24	Jajan		Rp. 100.00
5/1/24	Grocery		Rp. 150.000
6/1/24	Sprei		Rp. 100.000
7/1/24	Stok makanan minggu kedua		Rp. 200.000
	Top Up Gopay untuk gojek		Rp. 400.000
	Top Up Shopeepay untuk beli baju		Rp. 120.000
	Top Up Dana untuk bayar baju HIMA		Rp. 110.000
8/1/24	Restock Skincare dan Makeup		Rp. 778.000
10/1/24	Makan di kantin		Rp. 20.000
12/1/24	Jajan		Rp. 150.000
13/1/24	Jajan		Rp. 500.00
14/1/24	Stok makanan minggu ketiga		Rp. 200.000
15/1/24	Hangout Bersama teman		Rp. 100.000
16/1/24	Bajar kunjin		Rp. 750.00
17/1/24	Jajan		Rp. 50.000
18/1/24	Jajan		Rp. 30.000
19/1/24	Grill		Rp. 200.000
20/1/24	Beli kado untuk teman		Rp. 175.000
21/1/24	Stok makanan minggu keempat		Rp. 200.000
22/1/24	Grocery		Rp. 200.000
23/1/24	Beli lunch box set		Rp. 100.000
24/1/24	Grocery		Rp. 68.000
25/1/24	Makan di kantin		Rp. 12.000
26/1/24	Jajan		Rp. 20.000
27/1/24	Street food bersama teman		Rp. 120.000
28/1/24	Bayar kas		Rp. 5.000
29/1/24	Shopping		Rp. 500.000
30/1/24	Hangout Bersama teman		Rp. 250.000

Tabel 4.1 Data Keuangan Putri

2. Pegawai toko bangunan Jaya Bakti ingin menghitung pendapatannya selama satu bulan sebagai laporan keuangan kepada bosnya. Berikut daftar transaksi yang telah direkap oleh pegawai:

Tanggal	Keterangan	Pemasukan	Pengeluaran
1/8/21	Saldo bulan lalu	Rp. 20.000.000	
2/8/21	Penjualan cat 2,5L (2 Buah)	Rp. 380.000	
3/8/21	Penjualan semen 40kg (2 Buah)	Rp. 120.000	
3/8/21	Penjualan pipa PVC $\frac{3}{4}$ inch (100 Buah)	Rp. 3.500.000	
4/8/21	Pembayaran invoice stok tanki 600L (2 Buah)		Rp. 4.500.000

4/8/21	Penjualan pintu geser aluminium (1 Buah)	Rp. 1.200.000	
4/8/21	Penjualan lantai kayu (5 Meter)	Rp. 1.000.000	
6/8/21	Penjualan sekrup (2 Dus)	Rp. 280.000	
6/8/21	Pembelian ATK		Rp. 150.000
7/8/21	Penjualan kusen jendela aluminium (1 set uk. 120x120)	Rp. 1.850.000	
9/8/21	Penjualan kuas cat 5 inch (6 Buah)	Rp. 36.000	
10/8/21	Penjualan kran air plastik (2 Buah)	Rp. 15.000	
11/8/21	Penjualan baut	Rp. 70.000	
13/8/21	Pembayaran invoice stok pipa ¾ inch (50 Buah)		Rp. 1.350.000
13/8/21	Penjualan pipa pvc 2 inch (25 Buah)	Rp. 2.000.000	
14/8/21	Penjualan stop kontak	Rp. 140.000	
14/8/21	Penjualan keramik 40x40 (4 Dus)	Rp. 375.000	
15/8/21	Penjualan cat 5kg (5 Buah)	Rp. 675.000	
15/8/21	Pembelian meja kantor		Rp. 700.000
15/8/21	Ongkos pengiriman barang		Rp. 125.000
16/8/21	Penjualan lampu LED Philips 40 watt (1 Buah)	Rp. 188.000	
16/8/21	Bayar cicilan truk		Rp. 2.500.000
17/8/21	Penjualan tangga lipat 2 meter (2 Buah)	Rp. 1.370.000	
17/8/21	Penjualan keramik 40x40 (6 Dus)	Rp. 480.000	
18/8/21	Penjualan kanopi baja ringan 12 meter	Rp. 1.200.000	
18/8/21	Penjualan paku 10cm 1kg	Rp. 16.000	
19/8/21	Penjualan jendela kayu	Rp. 500.000	
19/8/21	Penjualan lem pipa pvc dextone (1 Kaleng)	Rp. 40.000	
20/8/21	Pembayaran invoice stok cat 5kg (50 buah)		Rp. 2.500.000
20/8/21	Pembayaran invoice keramik 40x40 (10 dus)		Rp. 800.000
20/8/21	Penjualan cat 5kg (5 Buah)	Rp. 405.000	
21/8/21	Penjualan pasir 1 colt	Rp. 560.000	
22/8/21	Penjualan saklar listrik (3 Buah)	Rp. 45.000	
23/8/21	Penjualan roller cat (1 Buah)	Rp. 46.000	
23/8/21	Penjualan cat 2,5L (2 Buah)	Rp. 380.000	
24/8/21	Penjualan kran plastik (3 Buah)	Rp. 22.500	
25/8/21	Pembayaran sewa gedung		Rp. 5.000.000
25/8/21	Pembayaran listrik, air, internet		Rp. 1.250.000
25/8/21	Pembayaran gaji karyawan		Rp. 4.500.000

Tabel 4.2 Data Keuangan Pegawai Toko Bangunan Jaya Bakti

Implementasi Kasus 1:

Masukkan kategori pengeluaran: Pengeluaran

Menu:

1. Tambah Pendapatan
2. Tambah Pengeluaran
3. Informasi Saldo
4. Simpan Data
5. Exit

Pilih: 3

	Jenis	Nama	Nominal	Kategori
	Pendapatan	Uang Bulanan	5e+06	Pemasukan
	Pengeluaran	Peralatan kuliah	100000	Pengeluaran
	Pengeluaran	Stok makan minggu 1	200000	Pengeluaran
	Pengeluaran	Hangout	50000	Pengeluaran
	Pengeluaran	Bayar kartu perpustakaan	25000	Pengeluaran
	Pengeluaran	Makan di kantin	25000	Pengeluaran
	Pengeluaran	Jajan	10000	Pengeluaran
	Pengeluaran	Grocery	150000	Pengeluaran
	Pengeluaran	Sprei	100000	Pengeluaran
	Pengeluaran	Stok makan minggu 2	200000	Pengeluaran
	Pengeluaran	Top Up Gopay	400000	Pengeluaran
	Pengeluaran	Top Up ShopeePay	120000	Pengeluaran
	Pengeluaran	Top Up DANA	110000	Pengeluaran
	Pengeluaran	Restock Skincare	778000	Pengeluaran
	Pengeluaran	Makan	20000	Pengeluaran
	Pengeluaran	Jajan	150000	Pengeluaran
	Pengeluaran	Jajan	50000	Pengeluaran
	Pengeluaran	Stok makan minggu 3	200000	Pengeluaran
	Pengeluaran	Hangout	100000	Pengeluaran
	Pengeluaran	Bayar Kunjin	75000	Pengeluaran
	Pengeluaran	Jajan	50000	Pengeluaran
	Pengeluaran	Jajan	30000	Pengeluaran
	Pengeluaran	Grill	200000	Pengeluaran
	Pengeluaran	Beli kado	175000	Pengeluaran
	Pengeluaran	Stok makan minggu 4	200000	Pengeluaran
	Pengeluaran	Grocery	200000	Pengeluaran
	Pengeluaran	Beli lunch box	100000	Pengeluaran
	Pengeluaran	Grocery	68000	Pengeluaran
	Pengeluaran	Makan	12000	Pengeluaran
	Pengeluaran	Jajan	20000	Pengeluaran
	Pengeluaran	Street food	120000	Pengeluaran
	Pengeluaran	Bayar kas	5000	Pengeluaran
	Pengeluaran	Shopping	500000	Pengeluaran
	Pengeluaran	Hangout	250000	Pengeluaran
		Total Pendapatan:	5e+06	
		Total Pengeluaran:	4.793e+06	
		Saldo Akhir:	207000	

Gambar 4.2 Menampilkan informasi saldo keuangan bulanan Putri

Menu:

1. Tambah Pendapatan
2. Tambah Pengeluaran
3. Informasi Saldo
4. Simpan Data
5. Exit

Pilih: 4

Data berhasil disimpan ke dalam file 'financial_data.txt'.

Gambar 4.3 Menampilkan data bulanan Putri telah disimpan

```

Menu:
1. Tambah Pendapatan
2. Tambah Pengeluaran
3. Informasi Saldo
4. Simpan Data
5. Exit
Pilih: 5
Keluar dari program.

```

Gambar 4.4 Menampilkan pengguna Putri telah keluar dari program

Implementasi Kasus 2:

```

Menu:
1. Tambah Pendapatan
2. Tambah Pengeluaran
3. Informasi Saldo
4. Simpan Data
5. Exit
Pilih: 3

```

	Jenis	Nama	Nominal	Kategori
Pendapatan	SaldoAwal		2e+07	SaldoAwal
Pendapatan	PenjualanCat2,5L(2Buah)		380000	Penjualan
Pendapatan	PenjualanSemen40Kg(2buah)		120000	
Pendapatan	PenjualanPipaPVC3/4Inch(100Buah)		3.5e+06	Penjualan
Pendapatan	PenjualanPintuGeserAluminium		1.2e+06	Penjualan
Pengeluaran	PembayaranStokTanki600L(2buah)		4.5e+06	Pembelian
Pendapatan	PenjualanLantaiKayu(5m)		1e+06	Penjualan
Pendapatan	PenjualanSekrup(2dus)		280000	Penjualan
Pendapatan	PenjualanKusenJendelaAluminium(1setUk120x120)		1.85e+06	Penjualan
Pengeluaran	PembelianATK		150000	Pembelian
Pendapatan	PenjualanKuasCat5inch(6buah)		36000	Penjualan
Pendapatan	PenjualanKeranAirPlastik(2buah)		15000	Penjualan
Pendapatan	PenjualanBaut		70000	Penjualan
Pengeluaran	PembelianStokPipa4/3Inch(50Buah)		1.35e+06	Pembelian
Pendapatan	PenjualanPipaPVC2Inch(25Buah)		2e+06	Penjualan
Pendapatan	PenjualanStopKontak		140000	Penjualan
Pendapatan	PenjualanKeramik40x40(4dus)		375000	Penjualan
Pendapatan	PenjualanCat5kg(5buah)		675000	Penjualan
Pengeluaran	PembelianMejaKantor		700000	Pembelian
Pengeluaran	PembayaranOngkosKirim		125000	Pembayaran
Pendapatan	PenjualanLampuLED40watt		188000	Penjualan
Pengeluaran	PembayaranCicilanTruk		2.5e+06	Pembayaran
Pendapatan	PenjualanKanopiBajaRingan(12m)		1.2e+06	Penjualan
Pendapatan	PenjualanKeramik40x40(6Dus)		480000	Penjualan
Pendapatan	PenjualanTanggaLipat2Meter(2buah)		1.37e+06	Penjualan
Pendapatan	PenjualanPaku10cm(1kg)		16000	Penjualan
Pendapatan	PenjualanJendelaKayu		500000	Penjualan
Pendapatan	PenjualanLemPipaPVCDeftone(1kaleng)		40000	Penjualan
Pengeluaran	PembayaranStokCat5kg(50Buah)		2.5e+06	Pembelian
Pengeluaran	PembayaranKeramik40x40(10Dus)		800000	Pembelian
Pendapatan	PenjualanCat5kg(5buah)		405000	Penjualan
Pendapatan	PenjualanPasir(1Colt)		560000	Penjualan
Pendapatan	PenjualanSaklarListrik(3buah)		45000	Penjualan
Pendapatan	PenjualanRollerCat(1buah)		46000	Penjualan
Pendapatan	PenjualanCat2,5L(2buah)		380000	Penjualan
Pendapatan	PenjualanKeranAirPlastik(3buah)		22500	Penjualan
Pengeluaran	PembayaranSewaGedung		5e+06	Pembayaran
Pengeluaran	PembayaranListrik,Air,Internet		1.25e+06	Pembayaran
Pengeluaran	PembayaranGajiKaryawan		4.5e+06	Pembayaran
Total Pendapatan:			3.68935e+07	
Total Pengeluaran:			2.3375e+07	
Saldo Akhir:			1.35185e+07	

Gambar 4.5 Menampilkan informasi saldo keuangan Toko Jaya Bakti

```

Menu:
1. Tambah Pendapatan
2. Tambah Pengeluaran
3. Informasi Saldo
4. Simpan Data
5. Exit
Pilih: 4
Data berhasil disimpan ke dalam file 'financial_data.txt'.

```

Gambar 4.6 Menampilkan data Toko Jaya Bakti telah disimpan

```
Menu:  
1. Tambah Pendapatan  
2. Tambah Pengeluaran  
3. Informasi Saldo  
4. Simpan Data  
5. Exit  
Pilih: 5  
Keluar dari program.
```

Gambar 4.7 Menampilkan pegawai Toko Jaya Bakti telah keluar dari program

Pengujian program terhadap dua contoh kasus diatas berhasil dilakukan. Program dapat menginput data-data yang dapat dimodifikasi sesuai dengan kebutuhan pengguna. Data perhitungan terhadap total pemasukan ,pengeluaran, dan juga saldo akhir hasilnya valid dan sesuai.

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan hasil pengujian yang telah dilakukan, maka dapat diambil kesimpulan bahwa penulisan bahasa pemrograman aplikasi Dana Cerdas dengan menggunakan tools C++ berhasil. Hal itu dibuktikan dengan uji coba terhadap dua contoh kasus, segala input yang dimasukkan pengguna berhasil menampilkan perhitungan pemasukan pengeluaran dan saldo yang tepat berdasarkan pada proses rumus yang ada pada program. Datanya dapat terus diupdate sesuai kebutuhan pengguna dalam waktu yang tidak terbatas.

5.2 Saran

Berdasarkan kesimpulan diatas, terdapat saran yang akan menjadi evaluasi dalam pengembangan aplikasi Dana Cerdas. Program aplikas dapat dikembangkan lebih baik lagi supaya lebih efisien, efektif, dan dapat menampilkan banyak pilihan pengolahan hasil data keuangan menjadi lebih rinci. Serta penulisan program harus lebih rapih dan terstruktur.

DAFTAR PUSTAKA

- Enterprise, Jubilee. (2015). *Mengenal Pemrograman Dasar*. Jakarta: Alex Media Komputindo.
- Horwits, E. & Sahni S. (1993). *Fundamental of Data Structure in C++*, source DBLP.
- Jogiyanto. (2009). *Sistem Teknologi Informasi*. Edisi Ketiga. Yogyakarta: ANDI.
- Kabar Harian. (2017). *Bahasa Pemrograman C++: Pengertian, Konsep Dasar, dan Keunggulannya*. Kumparan.com. 29 Novembr 2017.
- Kani. (2020). *Pengantar Algoritma dan Pemrograman*. Penerbit Pustaka UT.
- Rinaldi, M. (1999). *Algoritma dan Pemrograman Jilid1*. Penerbit IPB.
- Sismoro. (2005). *Pengantar Logika Informatika Algoritma dan Pemrograman*. Penerbit Perpustakaan UBSI.
- Suparta, Wayan. (2019). *What Is Data: Konsep Dasar Pemrograman dengan Dev C++*. Struktur Data, Universitas Pembangunan Jaya.
- Suratna, Agus. (2021). *Macam-macam File Header pada C++ dan Fungsinya*. Agussuratna.net. 11 Februari 2021.
- Wardana. (2010). *Menjadi Master PHP dengan Framework Codeigniter*. Jakarta: PT Alex Media Komputindo.