

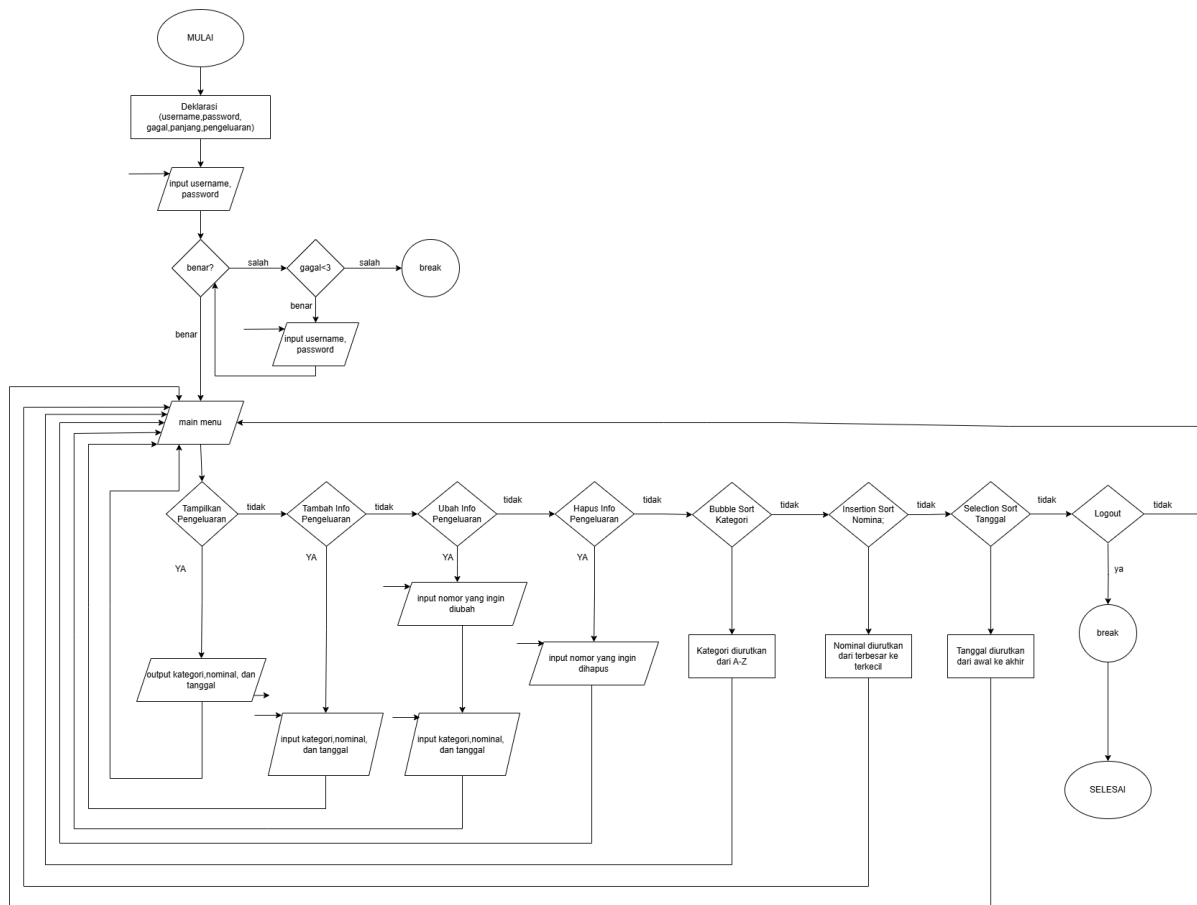
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Jovan Bagas Pangestu (2409106071)
Kelas (B2'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

Program ini bertujuan untuk membantu pengguna dalam mencatat dan mengelola pengeluaran mereka secara lebih terstruktur dan efisien. Dengan adanya fitur login dan registrasi, setiap pengguna dapat memiliki akun pribadi yang menyimpan data pengeluaran mereka dengan aman. Program menggunakan struktur data struct untuk mengorganisir informasi pengguna dan daftar pengeluaran, termasuk kategori, nominal, serta tanggal transaksi, sehingga memudahkan dalam pencatatan dan pemantauan keuangan.

Selain itu, program ini memiliki sistem keamanan yang membatasi jumlah percobaan login yang gagal guna mencegah akses tidak sah. Setelah berhasil login, pengguna dapat mengakses berbagai fitur, seperti melihat daftar pengeluaran, menambah pengeluaran baru, mengedit, dan menghapus data yang sudah ada. Dengan adanya fitur ini, pengguna dapat mengelola keuangan mereka dengan lebih baik, mengetahui pola pengeluaran, serta membuat keputusan finansial yang lebih bijak berdasarkan catatan yang tersimpan dalam program.

3. Source Code

A. BUBBLE SORT

```
void bubbleSortKategoriAscending(User *UserAktif) {
    for (int i = 0; i < UserAktif->panjangPengeluaran - 1; i++) {
        for (int j = 0; j < UserAktif->panjangPengeluaran - i - 1; j++) {
            if (UserAktif->pengeluaran[j].kategori >
                UserAktif->pengeluaran[j + 1].kategori) {
                swap(UserAktif->pengeluaran[j], UserAktif->pengeluaran[j +
1]);
            }
        }
    }
    cout << "Data berhasil diurutkan berdasarkan kategori (A-Z)." << endl;
}
```

B. INSERTION SORT

```
void insertionSortNominalDescending(User *UserAktif) {
    for (int i = 1; i < UserAktif->panjangPengeluaran; i++) {
        Pengeluaran key = UserAktif->pengeluaran[i];
        int j = i - 1;

        while (j >= 0 && stoi(UserAktif->pengeluaran[j].nominal) <
            stoi(key.nominal)) {
            UserAktif->pengeluaran[j + 1] = UserAktif->pengeluaran[j];
            j--;
        }
        UserAktif->pengeluaran[j + 1] = key;
    }
    cout << "Data berhasil diurutkan berdasarkan nominal (besar ke kecil)."
    << endl;
}
```

C. SELECTION SORT

```
void selectionSortTanggalAscending(User *UserAktif) {
    for (int i = 0; i < UserAktif->panjangPengeluaran - 1; i++) {
        int min_idx = i;
        for (int j = i + 1; j < UserAktif->panjangPengeluaran; j++) {
            if (UserAktif->pengeluaran[j].tanggal <
                UserAktif->pengeluaran[min_idx].tanggal) {
                min_idx = j;
            }
        }
        swap(UserAktif->pengeluaran[i], UserAktif->pengeluaran[min_idx]);
    }
    cout << "Data berhasil diurutkan berdasarkan tanggal (awal ke akhir)."
}
```

```
<< endl;
}
```

D. MENU UTAMA

```
void menuPengeluaran(User *UserAktif) {
    int pilihan;
    do {
        cout << "\nMenu Program" << endl;
        cout << "1. Tampilkan Info Pengeluaran" << endl;
        cout << "2. Tambah Info Pengeluaran" << endl;
        cout << "3. Ubah Info Pengeluaran" << endl;
        cout << "4. Hapus Info Pengeluaran" << endl;
        cout << "5. Keluar" << endl;
        cout << "6. Urutkan Kategori (A-Z)" << endl;
        cout << "7. Urutkan Nominal (besar ke kecil)" << endl;
        cout << "8. Urutkan Tanggal (awal ke akhir)" << endl;

        string input;
        cout << "Pilihan: ";
        getline(cin, input);

        if (input.length() == 1 && input[0] >= '1' && input[0] <= '8') {
            pilihan = input[0] - '0';
        } else {
            pilihan = -1;
        }

        switch (pilihan) {
            case 1:
                tampilkanPengeluaran(UserAktif);
                break;
            case 2:
                tambahPengeluaran(*UserAktif);
                break;
            case 3:
                ubahPengeluaran(UserAktif);
                break;
            case 4:
                hapusPengeluaran(UserAktif);
                break;
            case 5:
                cout << "Logout Berhasil!" << endl;
                break;
            case 6:
                bubbleSortKategoriAscending(UserAktif);
                break;
            case 7:
                break;
        }
    } while (pilihan != -1);
}
```

```
        insertionSortNominalDescending(UserAktif);
        break;
    case 8:
        selectionSortTanggalAscending(UserAktif);
        break;
    default:
        cout << "Pilihan tidak valid, silakan masukkan angka 1
sampai 8." << endl;
    }
} while (pilihan != 5);
}
```

4. Uji Coba dan Hasil Output

Menu Program

1. Tampilkan Info Pengeluaran
2. Tambah Info Pengeluaran
3. Ubah Info Pengeluaran
4. Hapus Info Pengeluaran
5. Keluar
6. Urutkan Kategori (A-Z)
7. Urutkan Nominal (besar ke kecil)
8. Urutkan Tanggal (awal ke akhir)

Pilihan: 1

Daftar Pengeluaran Bulanan:

| No. | Kategori | Nominal | Tanggal |
|-----|----------|---------|---------|
|-----|----------|---------|---------|

| | | | |
|----|------------|--------|------------|
| 1 | AIR | 30000 | 05-09-2009 |
| 2 | CABE | 20000 | 12-12-2012 |
| 3 | JAJAN | 230999 | 12-03-2021 |
| 4 | API | 25000 | 01-09-2022 |
| 5 | BAJU | 400000 | 23-12-2024 |
| 6 | CELANA | 500000 | 27-03-2025 |
| 7 | KULIAH | 733999 | 28-08-2024 |
| 8 | KUOTA | 100000 | 08-09-2020 |
| 9 | MEMBERSHIP | 50000 | 01-01-2025 |
| 10 | KOMFOR | 98000 | 09-09-2023 |

Menu Program

1. Tampilkan Info Pengeluaran
2. Tambah Info Pengeluaran
3. Ubah Info Pengeluaran
4. Hapus Info Pengeluaran
5. Keluar
6. Urutkan Kategori (A-Z)
7. Urutkan Nominal (besar ke kecil)
8. Urutkan Tanggal (awal ke akhir)

Pilihan: 6

Data berhasil diurutkan berdasarkan kategori (A-Z).

Menu Program

1. Tampilkan Info Pengeluaran
2. Tambah Info Pengeluaran
3. Ubah Info Pengeluaran
4. Hapus Info Pengeluaran
5. Keluar
6. Urutkan Kategori (A-Z)
7. Urutkan Nominal (besar ke kecil)
8. Urutkan Tanggal (awal ke akhir)

Pilihan: 1

Daftar Pengeluaran Bulanan:

| No. | Kategori | Nominal | Tanggal |
|-----|----------|---------|---------|
|-----|----------|---------|---------|

| | | | |
|----|------------|--------|------------|
| 1 | AIR | 30000 | 05-09-2009 |
| 2 | API | 25000 | 01-09-2022 |
| 3 | BAJU | 400000 | 23-12-2024 |
| 4 | CABE | 20000 | 12-12-2012 |
| 5 | CELANA | 500000 | 27-03-2025 |
| 6 | JAJAN | 230999 | 12-03-2021 |
| 7 | KOMFOR | 98000 | 09-09-2023 |
| 8 | KULIAH | 733999 | 28-08-2024 |
| 9 | KUOTA | 100000 | 08-09-2020 |
| 10 | MEMBERSHIP | 50000 | 01-01-2025 |

Menu Program

1. Tampilkan Info Pengeluaran
2. Tambah Info Pengeluaran
3. Ubah Info Pengeluaran
4. Hapus Info Pengeluaran
5. Keluar
6. Urutkan Kategori (A-Z)
7. Urutkan Nominal (besar ke kecil)
8. Urutkan Tanggal (awal ke akhir)

Pilihan: 7

Data berhasil diurutkan berdasarkan nominal (besar ke kecil).

Menu Program

1. Tampilkan Info Pengeluaran
2. Tambah Info Pengeluaran
3. Ubah Info Pengeluaran
4. Hapus Info Pengeluaran
5. Keluar
6. Urutkan Kategori (A-Z)
7. Urutkan Nominal (besar ke kecil)
8. Urutkan Tanggal (awal ke akhir)

Pilihan: 1

Daftar Pengeluaran Bulanan:

| No. | Kategori | Nominal | Tanggal |
|-------|------------|---------|------------|
| ----- | | | |
| 1 | KULIAH | 733999 | 28-08-2024 |
| 2 | CELANA | 500000 | 27-03-2025 |
| 3 | BAJU | 400000 | 23-12-2024 |
| 4 | JAJAN | 230999 | 12-03-2021 |
| 5 | KUOTA | 100000 | 08-09-2020 |
| 6 | KOMFOR | 98000 | 09-09-2023 |
| 7 | MEMBERSHIP | 50000 | 01-01-2025 |
| 8 | AIR | 30000 | 05-09-2009 |
| 9 | API | 25000 | 01-09-2022 |
| 10 | CABE | 20000 | 12-12-2012 |

Menu Program

1. Tampilkan Info Pengeluaran
2. Tambah Info Pengeluaran
3. Ubah Info Pengeluaran
4. Hapus Info Pengeluaran
5. Keluar
6. Urutkan Kategori (A-Z)
7. Urutkan Nominal (besar ke kecil)
8. Urutkan Tanggal (awal ke akhir)

Pilihan: 8

Data berhasil diurutkan berdasarkan tanggal (awal ke akhir).

Menu Program

1. Tampilkan Info Pengeluaran
2. Tambah Info Pengeluaran
3. Ubah Info Pengeluaran
4. Hapus Info Pengeluaran
5. Keluar
6. Urutkan Kategori (A-Z)
7. Urutkan Nominal (besar ke kecil)
8. Urutkan Tanggal (awal ke akhir)

Pilihan: 1

Daftar Pengeluaran Bulanan:

| No. | Kategori | Nominal | Tanggal |
|-------|------------|---------|------------|
| ----- | | | |
| 1 | MEMBERSHIP | 50000 | 01-01-2025 |
| 2 | API | 25000 | 01-09-2022 |
| 3 | AIR | 30000 | 05-09-2009 |
| 4 | KUOTA | 100000 | 08-09-2020 |
| 5 | KOMFOR | 98000 | 09-09-2023 |
| 6 | JAJAN | 230999 | 12-03-2021 |
| 7 | CABE | 20000 | 12-12-2012 |
| 8 | BAJU | 400000 | 23-12-2024 |
| 9 | CELANA | 500000 | 27-03-2025 |
| 10 | KULIAH | 733999 | 28-08-2024 |

5. Langkah-Langkah Git pada VSCode

A. GIT ADD .

```
pjova@JovanBagas MINGW64 ~/OneDrive/Dokumen/post-test/post-test-apl-6 (main)
$ git add .
```

Menambahkan file ke staging area untuk di commit.

B. GIT COMMIT -M

```
pjova@JovanBagas MINGW64 ~/OneDrive/Dokumen/post-test/post-test-apl-6 (main)
$ git commit -m "uplod cpp dan exe"
[main 9783b57] uplod cpp dan exe
2 files changed, 265 insertions(+)
create mode 100644 OneDrive/Dokumen/post-test/post-test-apl-6/2409106071-JovanBagasPangestu-PT-6.cpp
create mode 100644 OneDrive/Dokumen/post-test/post-test-apl-6/2409106071-JovanBagasPangestu-PT-6.exe
```

Menyimpan file dari staging area ke repository disertai pesan singkat deskriptif.

C. GIT PUSH ORIGIN MAIN

```
pjova@JovanBagas MINGW64 ~/OneDrive/Dokumen/post-test/post-test-apl-6 (main)
$ git push --force origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 680.19 KiB | 7.31 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/JovanBagasPangestu/praktikum-apl.git
+ 4b90f6e...9783b57 main -> main (forced update)
```

Mengunggah file dari branch main di repo lokal ke repo remote.