

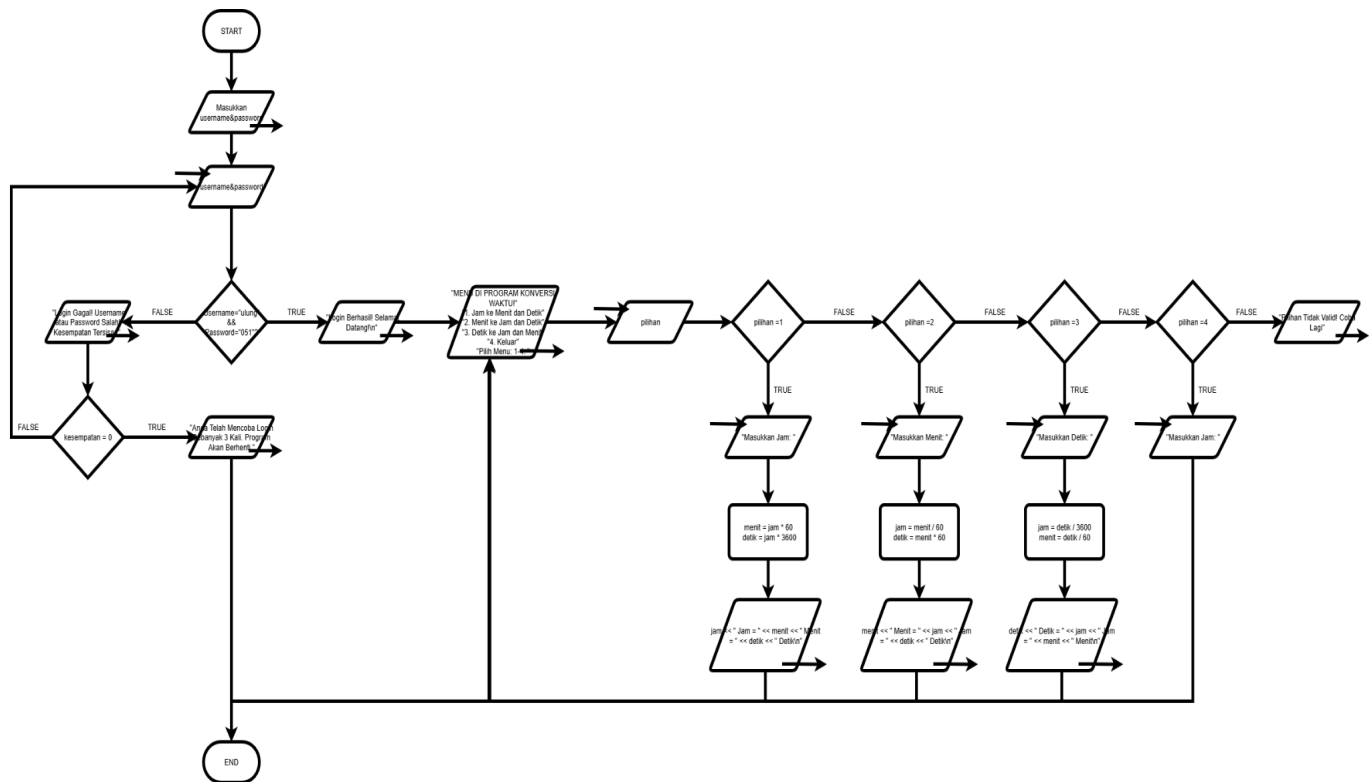
**LAPORAN PRAKTIKUM**  
**POSTTEST 1**  
**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**  
**Ulung Priskeaji Bintang Ramadhan (2509106051)**  
**Kelas (B1 '25)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

# 1. Flowchart



Program dimulai dengan meminta username dan password pengguna. Jika benar, pengguna masuk ke menu utama dan jika salah username atau password hingga 3 kali, program berhenti. Setelah login berhasil, pengguna memilih jenis konversi waktu (jam, menit, detik). Program kemudian meminta input angka sesuai pilihan, menghitung hasil konversi, lalu menampilkan hasil konversinya. Program akan berhenti ketika pengguna memilih opsi keluar.

## 2. Deskripsi Singkat Program

*Program ini dibuat untuk pengguna melakukan konversi waktu antara jam, menit, dan detik dengan melakukan proses login terlebih dahulu.*

*fungsi/manfaat program:*

- *program ini menggunakan keamanan dasar melalui login dengan batas 3 kali percobaan.*
- *Menyediakan fitur konversi waktu untuk pengguna.*
- *Memudahkan pengguna untuk mengonversi waktu secara cepat dan otomatis.*

## 3. Source Code

```
#include <iostream>

using namespace std;

string username;

string password;

int kesempatan;

int pilihan;

float jam;

float menit;

float detik;

int main() {
```

```

kesempatan = 3;

while (kesempatan > 0) {

    cout << "Masukkan username: ";

    cin >> username;

    cout << "Masukkan password: ";

    cin >> password;

    if (username == "uLung" && password == "051") {

        cout << "Login Berhasil! Selamat Datang!\n" << endl;

        break;

    } else {

        kesempatan--;

        cout << "Login Gagal! Username atau Password Salah! Kesempatan Tersisa: " << kesempatan << endl;

    }

}

if (kesempatan == 0) {

    cout << "Anda Telah Mencoba Login Sebanyak 3 Kali. Program Akan Berhenti." << endl;

    return 0;

}

while (true) {

    cout << "MENU KONVERSI WAKTU!" << endl;

```

```

    cout << "1. Jam ke Menit dan Detik" << endl;

    cout << "2. Menit ke Jam dan Detik" << endl;

    cout << "3. Detik ke Jam dan Menit" << endl;

    cout << "4. Keluar" << endl;

    cout << "Pilih Menu: 1-4: ";

    cin >> pilihan;

    if (pilihan == 1) {

        cout << "Masukkan Jam: ";

        cin >> jam;

        menit = jam * 60;

        detik = jam * 3600;

        cout << jam << " Jam = " << menit << " Menit = " << detik << "
Detik\n" << endl;

    } else if (pilihan == 2) {

        cout << "Masukkan Menit: ";

        cin >> menit;

        jam = menit / 60;

        detik = menit * 60;

        cout << menit << " Menit = " << jam << " Jam = " << detik << "
Detik\n" << endl;

    } else if (pilihan == 3) {

        cout << "Masukkan Detik: ";

        cin >> detik;

        jam = detik / 3600;

```

```

        menit = detik / 60;

        cout << detik << " Detik = " << jam << " Jam = " << menit << "
Menit\n" << endl;

    } else if (pilihan == 4) {

        cout << "Terima Kasih Sudah Menggunakan Program Ini!" << endl;

        break;

    } else {

        cout << "Pilihan Tidak Valid! Coba Lagi" << endl;

    }

}

}

```

#### 4. Hasil Output

```
Masukkan username: ulung
Masukkan password: 051
Login Berhasil! Selamat Datang!
```

```
MENU KONVERSI WAKTU!
1. Jam ke Menit dan Detik
2. Menit ke Jam dan Detik
3. Detik ke Jam dan Menit
4. Keluar
Pilih Menu: 1-4: 1
Masukkan Jam: 1
1 Jam = 60 Menit = 3600 Detik
```

Gambar 4.1 Output  
<login-dan-menu-1>

```
MENU KONVERSI WAKTU!
1. Jam ke Menit dan Detik
2. Menit ke Jam dan Detik
3. Detik ke Jam dan Menit
4. Keluar
Pilih Menu: 1-4: 2
Masukkan Menit: 55
55 Menit = 0.916667 Jam = 3300 Detik
```

Gambar 4.2 Output  
<menu-2>

MENU KONVERSI WAKTU!

1. Jam ke Menit dan Detik
2. Menit ke Jam dan Detik
3. Detik ke Jam dan Menit
4. Keluar

Pilih Menu: 1-4: 3

Masukkan Detik: 90

90 Detik = 0.025 Jam = 1.5 Menit

MENU KONVERSI WAKTU!

1. Jam ke Menit dan Detik
2. Menit ke Jam dan Detik
3. Detik ke Jam dan Menit
4. Keluar

Pilih Menu: 1-4: 4

Terima Kasih Sudah Menggunakan Program Ini!

PS C:\praktikum-apl\post-test\post-test-apl-1>

Gambar 4.3 Output  
<menu-3-dan-menu-4>



## 5. Langkah-langkah GIT

```
PS C:\praktikum-apl\post-test\post-test-apl-1> git init
Initialized empty Git repository in C:/praktikum-apl/post-test/post-test-apl-1/.git/
PS C:\praktikum-apl\post-test\post-test-apl-1> git add .
PS C:\praktikum-apl\post-test\post-test-apl-1> git commit -m "first commit"
[main (root-commit) ab5cfa6] first commit
 2 files changed, 69 insertions(+)
 create mode 100644 2509106051-UlungPriskeajiBintangRamadhan-PT-1.cpp
 create mode 100644 2509106051-UlungPriskeajiBintangRamadhan-PT-1.exe
PS C:\praktikum-apl\post-test\post-test-apl-1> git branch -M main
PS C:\praktikum-apl\post-test\post-test-apl-1> git remote add origin https://github.com/UlungPriskeaji/praktikum-apl.git
PS C:\praktikum-apl\post-test\post-test-apl-1> git push -u origin main
Enumerating objects: 4, done.
```

### 5.1 GIT Init

GIT Init adalah perintah yang digunakan untuk membuat repository Git baru pada sebuah folder. Perintah ini akan membuat folder tersembunyi bernama .git yang berfungsi untuk menyimpan seluruh riwayat perubahan dan konfigurasi repository.

### 5.2 GIT Add

GIT add merupakan perintah yang digunakan untuk menambahkan perubahan pada folder/file ke area staging sebelum melakukan commit. Area staging adalah area perantara yang menyimpan informasi tentang perubahan yang akan dimasukkan dalam commit berikutnya.

### 5.3 GIT Commit

GIT Commit adalah instruksi yang digunakan untuk menyimpan perubahan yang telah dimasukkan ke dalam staging area ke dalam repository Git lokal secara tetap. Setiap commit menyimpan cuplikan proyek pada saat itu dan disertai pesan deskriptif yang menjelaskan perubahan yang telah dilakukan.

### 5.4 GIT Branch

GIT Branch digunakan untuk membuat dan mengelola cabang dalam repository. Branch memungkinkan pengembangan dilakukan tanpa mengganggu branch utama.

### 5.5 GIT Remote

GIT Remote digunakan untuk menghubungkan repository lokal dengan repository online (seperti GitHub) sebagai tujuan push dan pull.

### 5.6 GIT Push

GIT Push digunakan untuk mengirim commit dari repository lokal ke repository remote (misalnya GitHub).