

**LAPORAN  
PRAKTIKUM  
POSTTEST (3)  
ALGORITMA PEMROGRAMAN DASAR**

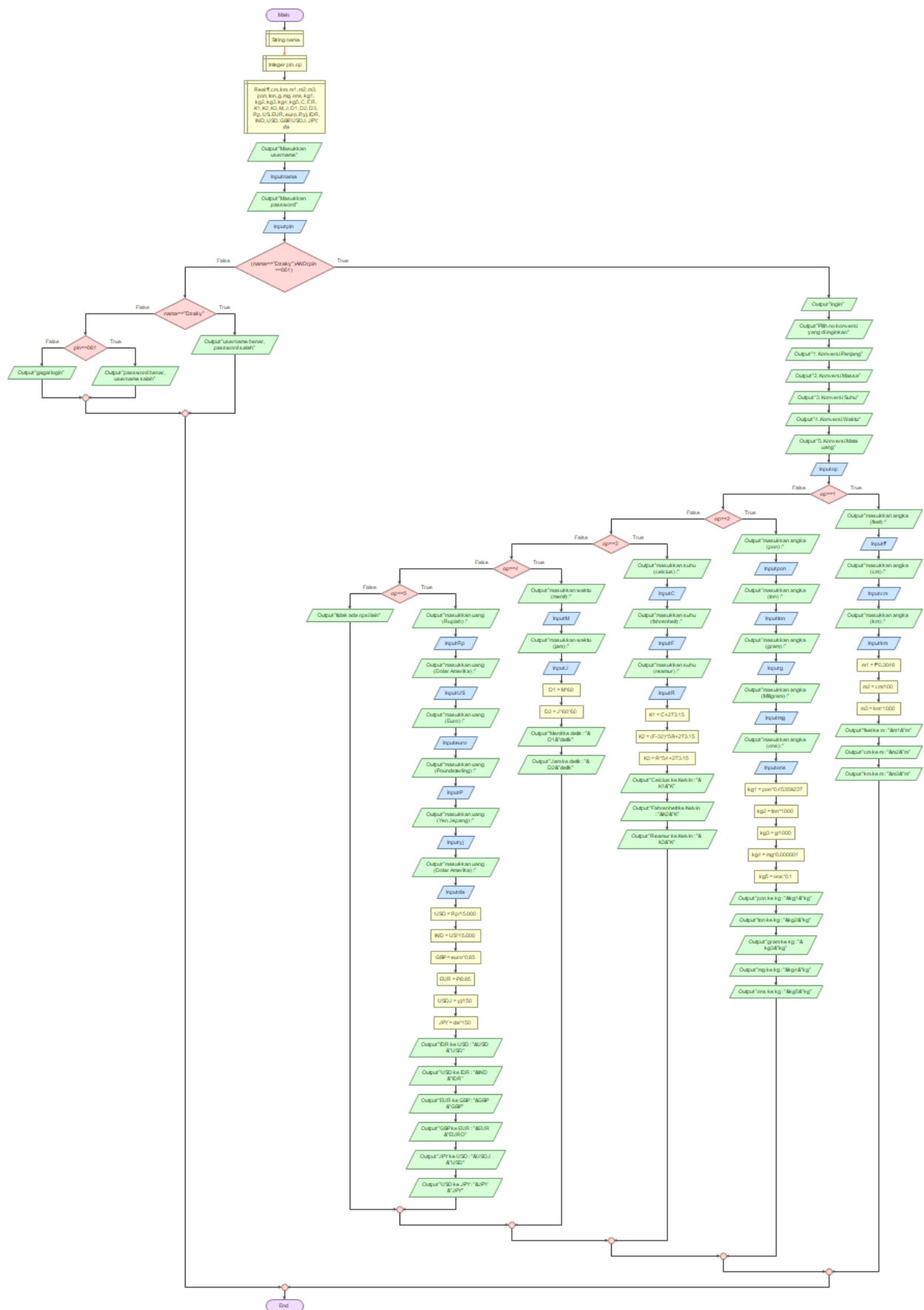


**Disusun oleh:  
Dzaky Ainur Rahman (2509106061)  
Kelas (B'25)**

**PROGRAM STUDI  
INFORMATIKA UNIVERSITAS  
MULAWARMAN SAMARINDA**

**2025**

## 1. Flowchart



## **2. Deskripsi Singkat Program**

*Program ini merupakan kalkulator multifungsi berbasis login, yang hanya dapat diakses jika pengguna memasukkan username dan password yang benar. Setelah login berhasil, pengguna akan diberi menu utama berisi lima jenis konversi, yaitu:*

- 1. Konversi Panjang (feet, cm, km → meter)*
- 2. Konversi Massa (pound, ton, gram, mg, ons → kilogram)*
- 3. Konversi Suhu (Celsius, Fahrenheit, Reamur → Kelvin)*
- 4. Konversi Waktu (menit, jam → detik)*
- 5. Konversi Mata Uang (Rupiah, USD, Euro, Pound, Yen → silang-konversi)*

*Setiap opsi akan meminta input angka, lalu menghitung menggunakan rumus yang sudah ditentukan dan menampilkan hasil konversinya.*

*Jika user memilih opsi yang tidak valid, program akan menampilkan pesan "TIDAK ADA OPSI".*

*Jika login gagal, program memberi tahu apakah username atau password yang salah, atau keduanya salah.*

### 3. Source Code

```
print ("Kalkulator serba bisa")
username = (input("masukkan username anda : "))
password = (input("masukkan password anda : "))

if username == "Dzaky" and password == "061":
    print("""
        1. Konversi Panjang
        2. Konversi Massa
        3. Konversi Suhu
        4. Konversi Waktu
        5. Konversi Mata uang""")

    opsi = int(input("pilih no konversi yang di inginkan : "))

#konversi panjang
    if opsi == 1:
        print("\n<KONVERSI PANJANG>")

        print("""
            1. feet ke meter
            2. kilometer ke meter
            3. centimeter ke meter
            """)

        feet = int(input("masukkan angka (feet) :"))
        cm = int(input("masukkan angka (cm) :"))
        km = int(input("masukkan angka (km) :"))
        m1 = feet * 0.3048
        m2 = cm / 100
        m3 = km * 1000
        print("\n<Hasil Konversi>\n")
        print(feet, "feet", "ke", m1, "meter")
        print(cm, "cm", "ke", m2, "meter")
        print(km, "km", "ke", m3, "meter")

#konversi massa
    elif opsi == 2:
        print("\n<Konversi Massa>\n")
        massa =[
            int(input("Masukkan angka (pound)      : ")),
            int(input("Masukkan angka (ton)         : ")),
            int(input("Masukkan angka (gram)        : ")),
            int(input("Masukkan angka (mg)         : ")),
            int(input("Masukkan angka (ons)         : "))
        ]
        print("\n<Hasil Konversi>\n")
        pound    = massa[0] * 0.45
        ton       = massa[1] * 1000
        gram      = massa[2] / 1000
        mg        = massa[3] * 0.000001
```

```

        ons        = massa[4] / 0.1

    print(massa[0], "pound", "ke", pound, "kilogram")
    print(massa[1], "ton", "ke", ton, "kilogram")
    print(massa[2], "gram", "ke", gram, "kilogram")
    print(massa[3], "mg", "ke", mg, "kilogram")
    print(massa[4], "gram", "ke", ons, "kilogram")

#konversi suhu
elif opsi == 3:
    print("\n<Konversi Suhu>\n")
    suhu = [
        int(input("Masukkan suhu (Celcius)   ")),
        int(input("Masukkan suhu (Fahrenheit) ")),
        int(input("Masukkan suhu (Reamur)     "))
    ]

    print("\n<Hasil Konversi>\n")

    celcius    = suhu[0] + 273.15
    fahrenheit = (suhu[1] + 459.67) * 5/9
    reamur      = (suhu[2] / 0.8) + 273.15

    print(suhu[0], "Celcius", "ke", celcius, "Kelvin")
    print(suhu[0], "Fahrenheit", "ke", fahrenheit, "Kelvin")
    print(suhu[0], "Reamur", "ke", reamur, "Kelvin")

#konversi waktu
elif opsi == 4:
    print("\n<Konversi Waktu>\n")
    waktu = [
        int(input("Masukkan waktu(menit) ")),
        int(input("Masukkan waktu(jam)   "))
    ]

    print("\n<Hasil Konversi>\n")

    menit = waktu[0] * 60
    jam    = waktu[1] * 3600

    print(waktu[0], "Menit", "ke", menit, "Detik")
    print(waktu[1], "Jam", "ke", jam, "Detik")

#konversi mata uang
elif opsi == 5:
    print("\n<Konversi Mata uang>\n")
    uang = [
        int(input("Masukkan uang(Rupiah) ")),
        int(input("Masukkan uang(Dolar Amerika) ")),
        int(input("Masukkan uang(Euro) ")),
        int(input("Masukkan uang(Poundsterling) "))
    ]

```

```

        int(input("Masukkan uang(Yen Jepang) ")),
        int(input("Masukkan uang(Dolar Amerika) "))
    ]

    print("\n<Hasil Konversi>\n")

    USD = uang[0]/15000
    IDR = uang[1]*15000
    GBP = uang[2]*0.85
    EUR = uang[3]/0.85
    USDJ= uang[4]/150
    JPY = uang[5]*150

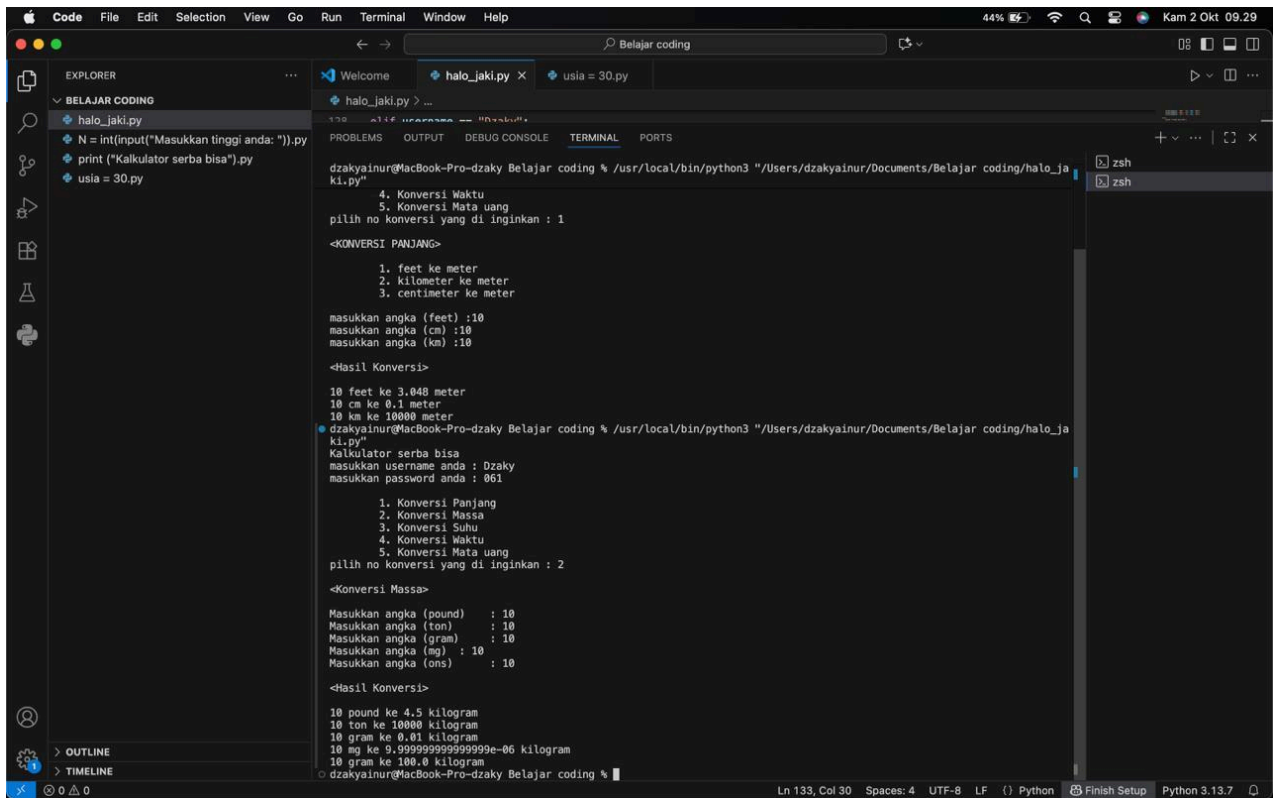
    print(uang[0], "Rp", "ke", USD, "USD")
    print(uang[1], "USD", "ke", IDR, "Rp")
    print(uang[2], "EUR", "ke", GBP, "GBP")
    print(uang[3], "GBP", "ke", EUR, "EUR")
    print(uang[4], "USD", "ke", JPY, "JPY")
    print(uang[5], "JPY", "ke", USD, "USD")

else:
    print ("TIDAK ADA OPSI")

elif username == "Dzaky":
    print ("Username benar, password salah")
elif password == "061":
    print ("Password benar, username salah")
else:
    print("TIDAK BISA LOGIN")

```

## 4. Hasil Output



The screenshot shows a VS Code editor with a Python script named `halo_jaki.py` in the Explorer view. The script contains the following code:

```
N = int(input("Masukkan tinggi anda: "))
print("Kalkulator serba bisa")
usia = 30
```

The Terminal view shows the output of the script, which is a menu for a "Kalkulator serba bisa" (All-purpose calculator). The user has selected option 1, "Konversi Panjang" (Length Conversion). The program prompts the user to enter a value for "feet", "cm", or "km". The user has entered "10" for "feet". The program then displays the conversion results:

```
10 feet ke 3.048 meter
10 cm ke 0.1 meter
10 km ke 10000 meter
```

The user has then selected option 2, "Konversi Massa" (Mass Conversion). The program prompts the user to enter a value for "pound", "ton", "gram", "mg", or "ons". The user has entered "10" for "pound". The program then displays the conversion results:

```
10 pound ke 4.5 kilogram
10 ton ke 10000 kilogram
10 gram ke 0.01 kilogram
10 mg ke 9.999999999999999e-06 kilogram
10 gram ke 100.0 kilogram
```

## 5. Langkah-langkah Git

### 5.1 Git init

**Git init** digunakan untuk membuat repository Git baru di folder lokal, sehingga folder tersebut dapat dilacak perubahannya oleh Git. Setelah perintah ini dijalankan, akan muncul folder tersembunyi bernama `.git` yang menyimpan semua informasi version control.

```
dzakyainur@MacBook-Pro-dzaky Pratikum-apd % git init
Reinitialized existing Git repository in /Users/dzakyainur/Documents/Pratikum-apd/.git/
```

### 5.2 Git add

**Git add** berfungsi menambahkan file atau perubahan file ke staging area sebelum benar- benar disimpan. Dengan ini, Git tahu perubahan mana yang siap untuk dicatat ke dalam commit berikutnya.

```
dzakyainur@MacBook-Pro-dzaky Pratikum-apd % git add .
```

### 5.3 Git commit

**Git commit** menyimpan snapshot dari perubahan yang sudah ada di staging area ke repository lokal. Commit biasanya dilengkapi pesan singkat yang menjelaskan perubahan yang dilakukan agar riwayat mudah dipahami.

```
dzakyainur@MacBook-Pro-dzaky Pratikum-apd % git commit -m "2509106061_DzakyAinurRahman-PT-2"
On branch master
nothing to commit, working tree clean
```

### 5.4 Git remote

**Git remote** dipakai untuk menghubungkan repository lokal dengan repository remote (seperti GitHub atau GitLab). Dengan koneksi ini, pengguna bisa mengirim atau mengambil perubahan dari repository jarak jauh.

```
dzakyainur@MacBook-Pro-dzaky Pratikum-apd % git remote add origin https://github.com/dzakyainur-web/Pratikum-apd.git
error: remote origin already exists.
```

### 5.5 Git push

**Git push** berfungsi mengirim commit dari repository lokal ke repository remote. Dengan begitu, perubahan yang sudah disimpan di lokal akan tersedia juga di server atau platform hosting Git

```
dzakyainur@MacBook-Pro-dzaky Pratikum-apd % git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.37 KiB | 1.37 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/dzakyainur-web/Pratikum-apd.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
dzakyainur@MacBook-Pro-dzaky Pratikum-apd %
```