

LAPORAN APLIKASI SISTEM GAJI PEGAWAI



Disusun oleh :

KELOMPOK 1

RAHMANDA AFEBRIO YURIS SOESATYO – 1103223024

AGUNG RIZKI PANGESTU - 1103220064

RIFI MUHAMMAD HARAHAP -11032223138

Untuk memenuhi Tugas Mata kuliah Pemrograman Berorientasi Objek

2023

DAFTAR ISI

Fitur-Fitur	2
Rancangan OOP	4
Listing program	5

FITUR-FITUR

- **Fitur login aplikasi**

Hal yang pertama kami buat adalah tampilan saat login aplikasi, yang dimana user akan menginput pilihan menu yang akan digunakan oleh user. Pilihan menu yang ditampilkan ada pilihan Admin, direktur SDM, direktur.

- **Fitur login Admin**

Pada menu admin user akan diminta untuk menginputkan username dan password. Username nya yaitu "admin" dan passwordnya adalah "123". Kemudian pada fitur admin user memilih menu yang akan dipilih yaitu ada menu edit direktur SDM dan direktur.

- **Fitur edit direktur SDM**

Dimenu edit direktur SDM user diminta untuk menuliskan username dan NIP yang akan disimpan di file yang ada di laptop user.

- **Fitur edit direktur**

Sedangkan pada fitur direktur hanya menampilkan data yang telah disimpan oleh dictionary yang telah di inputkan di fitur direktur SDM.

- **Tampilan menu direktur SDM**

Pada fitur ini user akan diminta untuk memilih menu yang sudah disediakan oleh method direktur SDM. Pilihan menu meliputi:

1. Menambah, mengedit data pegawai
2. menambah, mengedit data gaji pokok dan gaji perjam
3. menampilkan semua list pegawai beserta atribut data lainnya denganurut NIP.
4. menampilkan list pegawai dan gajinya pada bulan tertentu denganurut NIP atauurut gaji
5. menampilkan list divisi
6. menampilkan list divisi beserta standar gajinya
7. menampilkan list pegawai di divisi tertentu
8. menampilkan daftar pejabat dikantor
9. menampilkan total gaji yang harus dibayar setiap bulan

- **Fitur menu 1 pada direktur SDM**

1. Menu 1 digunakan untuk menginput data Pegawai dan disimpan pada dict Pegawai yang mempunyai key :

- NIP
- Nama
- Jabatan
- Divisi
- Gaji Pokok (di input di menu 2)
- Gaji per jam (di input di menu 2)

2. Menu 2 digunakan untuk menginput gaji pokok dan gaji per jam dari pegawai

3. Menu 3 digunakan untuk menampilkan list pegawai yang sudah di input pada Menu 1 dan 2 secara lengkap dengan tabel.

4. Menu 4 digunakan untuk menampilkan list pegawai dan gajinya pada bulan tertentu, di method ini user akan diminta untuk menginput bulan dan jam kerja pada bulan tersebut lalu akan dihitung pada method ke 4. Lalu akan ditampilkan sesuai urutan NIP yang di input pada bulan tersebut.

5. Menu 5 digunakan untuk menampilkan semua list divisi yang sudah di input di menu 1. Jadi menu 5 hanya bisa memunculkan output apabila divisi sudah di input di menu 1

6. Menu 6 digunakan untuk menampilkan 3 divisi beserta gaji standarnya.

7. Menu 7 digunakan untuk menampilkan list pegawai di divisi yang diinginkan oleh user.

8. Menu 8 digunakan untuk menampilkan list Pejabat pada list pegawai, jadi Jabatan Direktur dan Manager saja yang akan ditampilkan oleh method ini.

9. Menu 9 digunakan untuk menampilkan total gaji yang harus dibayar pada bulan tertentu.

- **Tampilan menu direktur**

Pada fitur direktur user akan diminta memilih menu yang sudah disediakan oleh method direktur SDM. Pilihan menu meliputi:

1. menampilkan semua list pegawai beserta atribut data lainnya dengan urut NIP
2. menampilkan list pegawai dan pegawai dan gajinya pada bulan tertentu dengan urut NIP atau urut gaji
3. menampilkan list divisi
4. menampilkan list pegawai beserta standar gajinya
5. menampilkan list pegawai di divisi tertentu
6. menampilkan daftar pejabat di kantor
7. menampilkan total gaji yang harus dibayar setiap tahun

- **Evaluasi dan Catatan**

- Apabila typo terjadi program akan terpaksa keluar, Oleh karena itu, input harus Dituliskan dengan Berhati-hati.
- File digunakan hanya di kelas Admin hanya untuk menyimpan data akun Direktur dan Direktur SDM. Dan dikarenakan directory file nya menuju pada File manager salah satu laptop dari anggota kami, Maka, apabila program digunakan pada laptop lain program mungkin akan error dikarenakan perbedaan directory file yang ditulis pada kelas Admin
- Ada 2 fungsi diluar class, yaitu input data pegawai dan juga input jam kerja pada bulan tertentu. Input pegawai digunakan untuk memasukkan ke dalam dict kelas Direktur SDM

RANCANGAN OOP

Kelas yang dibangun terdiri atas 3 Kelas :

1. Kelas Admin

Kelas admin digunakan untuk menginput dan menyimpan data akun Direktur SDM dan Direktur pada file.

2. Kelas Direktur SDM

Kelas Direktur SDM digunakan untuk menginputkan semua method yang digunakan Direktur SDM dan menyimpan pada dict Pegawai, lalu otomatis di input ke list pegawai dan list divisi yang menyimpan data Pegawai dan Divisi pegawai.

3. Kelas Direktur

Kelas ini adalah anak kelas dari kelas Direktur SDM. Kelas Direktur hanya digunakan untuk menampilkan data data yang sudah di input pada kelas Direktur SDM.

Penggunaan inheritance digunakan pada Kelas Direktur SDM dan kelas Direktur, metode konstruktor kelas Direktur disamakan dengan kelas Direktur SDM.

Encapsulation digunakan pada kelas Admin, dimana atribut Password menggunakan access modifier Private. Sehingga hanya method dari kelas Admin saja yang bisa memanggil atribut Password.

LISTING PROGRAM

```
from tabulate import tabulate as tb

list_DirSDM = {}
list_Direktur = {}

class Admin:
    def __init__(self):
        self.id = None
        self.is_logged_in = False
        self.username = None
        self.__password = None

    def login(self, username, password):
        if username == "admin" and password == "123":
            self.is_logged_in = True
            self.username = username
            self.__password = password
            return "\nSelamat datang di Menu Admin."
        else:
            return "Invalid username or password."

    def edit(self):
        if self.is_logged_in:
            print("==== Menu Admin =====\n")
            x = int(input("Masukkan Menu yang ingin di edit\n1.Direktur
SDM\n2.Direktur\n"))
            if x == 1:
                list_DirSDM["Nama"] = input("Masukkan nama Direktur SDM : ")
                list_DirSDM["Password"] = input("Masukkan password Direktur
SDM : ")

                file = open('c:/college/PBO/List Direktur SDM.txt', 'w')
                file.write(f>Nama Direktur SDM : \n{list_DirSDM['Nama']}\n")
                file.write(f>Password : {list_DirSDM['Password']}\n")
                print("Data Direktur SDM Berhasil di input.")
            elif x == 2:
                list_Direktur["Nama"] = input("Masukkan nama Direktur : ")
                list_Direktur["Password"] = input("Masukkan password Direktur
: ")

                file = open('c:/college/PBO/List Direktur.txt', 'w')
                file.write(f>Nama Direktur : {list_Direktur['Nama']}\n")
                file.write(f>Password : {list_Direktur['Password']}\n")
                print("Data Direktur Berhasil di input.")
```

```

        else:
            return "Invalid"

class DirekturSDM:
    def __init__(self):
        self.is_logged_in = False
        self.pegawai_dict = {}
        self.divisi_dict = {}

    def login(self, username, password):
        file = open('c:/college/PBO/List Direktur SDM.txt', 'r')
        content = file.read()

        if username in content and password in content:
            self.is_logged_in = True
            return "\nSelamat datang di Menu Direktur SDM."
        else:
            return "Invalid username or password."

    def tambah_pegawai(self, nip, nama, jabatan, divisi):
        pegawai = {
            'nama': nama,
            'jabatan': jabatan,
            'divisi': divisi,
            'gaji_pokok': 0,
            'gaji_per_jam': 0
        }
        self.pegawai_dict[nip] = pegawai

        if divisi not in self.divisi_dict:
            self.divisi_dict[divisi] = {'standar_gaji': 0, 'jumlah_pegawai':
0}

    def edit_pegawai(self, nip, nama=None, jabatan=None, divisi=None):
        if nip in self.pegawai_dict:
            pegawai = self.pegawai_dict[nip]
            if nama is not None:
                pegawai['nama'] = nama
            if jabatan is not None:
                pegawai['jabatan'] = jabatan
            if divisi is not None:
                pegawai['divisi'] = divisi
        else:
            print(f"Pegawai dengan NIP {nip} tidak ditemukan.")

    def tambah_gaji(self, nip, gaji_pokok, gaji_per_jam):
        if nip in self.pegawai_dict:
            pegawai = self.pegawai_dict[nip]

```

```

        pegawai['gaji_pokok'] = gaji_pokok
        pegawai['gaji_per_jam'] = gaji_per_jam
    else:
        print(f"Pegawai dengan NIP {nip} tidak ditemukan.")

def edit_gaji(self, nip, gaji_pokok=None, gaji_per_jam=None):
    if nip in self.pegawai_dict:
        pegawai = self.pegawai_dict[nip]
        if gaji_pokok is not None:
            pegawai['gaji_pokok'] = gaji_pokok
        if gaji_per_jam is not None:
            pegawai['gaji_per_jam'] = gaji_per_jam
    else:
        print(f"Pegawai dengan NIP {nip} tidak ditemukan.")

def tampilkan_semua_pegawai(self,urut='nip'):
    sorted_pegawai = sorted(self.pegawai_dict.items(), key=lambda x:
x[1].get(urut, 0))
    headers = ["NIP", "Nama", "Jabatan", "Divisi", "Gaji Pokok", "Gaji per
Jam"]
    data = [[nip, pegawai['nama'], pegawai['jabatan'], pegawai['divisi'],
pegawai['gaji_pokok'], pegawai.get('gaji_per_jam', 0)] for nip, pegawai in
sorted_pegawai]
    print(tb(data, headers=headers, stralign="right"))

def tampilkan_gaji_bulan(self, bulan,urut='nip'):
    sorted_pegawai = sorted(self.pegawai_dict.values(), key=lambda x:
x['gaji_pokok'] + x.get('gaji_per_jam', 0))
    for nip, pegawai in self.pegawai_dict.items():
        gaji_total = pegawai['gaji_pokok']
        if 'gaji_per_jam' in pegawai:
            # Fungsi jam_kerja_bulan harus diimplementasikan sesuai
kebutuhan
            gaji_total += pegawai['gaji_per_jam'] * jam_kerja_bulan(nip,
bulan)
        print(f"NIP: {nip}, Nama: {pegawai['nama']}, Gaji Bulan {bulan}:
{gaji_total}")

def tampilkan_list_divisi(self):
    print("List Divisi:")
    for divisi in self.divisi_dict.keys():
        print(divisi)

def tampilkan_list_divisi_gaji(self):
    print("List Divisi beserta Standar Gajinya:")
    print("HR : 1.000.000")
    print("IT : 2.000.000")
    print("RND : 3.000.000")

```



```

def tampilkan_pegawai_divisi(self, divisi):
    print(f"List Pegawai di Divisi {divisi}:")
    for nip, pegawai in self.pegawai_dict.items():
        if pegawai['divisi'] == divisi:
            print(f"NIP: {nip}, Nama: {pegawai['nama']], Jabatan: {pegawai['jabatan']}")

def tampilkan_pejabat_kantor(self):
    for nip, pegawai in self.pegawai_dict.items():
        if pegawai['jabatan'] in ['Direktur', 'Manager']:
            print(f"NIP: {nip}, Nama: {pegawai['nama']], Jabatan: {pegawai['jabatan']}")

def tampilkan_total_gaji_bulan(self, bulan):
    total_gaji = 0
    for nip, pegawai in self.pegawai_dict.items():
        gaji_total = pegawai['gaji_pokok']
        if 'gaji_per_jam' in pegawai:
            # Fungsi jam_kerja_bulan harus diimplementasikan sesuai
            # kebutuhan
            gaji_total += pegawai['gaji_per_jam'] * jam_kerja_bulan(nip,
bulan)
        total_gaji += gaji_total
    print(f"Total Gaji yang Harus Dibayar pada Bulan {bulan}: {total_gaji}")

# Fungsi untuk mendapatkan jam kerja pegawai pada bulan tertentu
def jam_kerja_bulan(nip, bulan):
    if bulan == "januari" :
        x = int(input(f"Masukkan jam kerja anda di bulan {bulan} : "))
        return x
    elif bulan == "februari":
        x = int(input("Masukkan jam kerja anda di bulan februari : "))
        return x
    elif bulan == "maret" :
        x = int(input("Masukkan jam kerja anda di bulan maret : "))
        return x
    elif bulan == "april" :
        x = int(input("Masukkan jam kerja anda di bulan april : "))
        return x
    elif bulan == "mei" :
        x = int(input("Masukkan jam kerja anda di bulan mei : "))
        return x
    elif bulan == "juni":
        x = int(input("Masukkan jam kerja anda di bulan juni : "))
        return x
    elif bulan == "juli" :

```

```

        x = int(input("Masukkan jam kerja anda di bulan juli : "))
        return x
    elif bulan == "agustus" :
        x = int(input("Masukkan jam kerja anda di bulan agustus : "))
        return x
    elif bulan == "september" :
        x = int(input("Masukkan jam kerja anda di bulan september : "))
        return x
    elif bulan == "oktober" :
        x = int(input("Masukkan jam kerja anda di bulan oktober : "))
        return x
    elif bulan == "november" :
        x = int(input("Masukkan jam kerja anda di bulan november : "))
        return x
    elif bulan == "desember" :
        x = int(input("Masukkan jam kerja anda di bulan desember : "))
        return x
    return x

class Direktur(DirekturSDM):
    def __init__(self):
        super().__init__()

    def login(self, username, password):
        file = open('c:/college/PBO/List Direktur.txt', 'r')
        content = file.read()

        if username in content and password in content:
            self.is_logged_in = True
            return "\nSelamat datang di Menu Direktur."
        else:
            return "Invalid username or password."

    def tampilkan_semua_pegawai(self,urut='nip'):
        sorted_pegawai = sorted(self.pegawai_list, key=lambda x: x[urut])
        headers = ["NIP", "Nama", "Jabatan", "Divisi", "Gaji Pokok", "Gaji per
Jam"]
        data = [[pegawai['nip'], pegawai['nama'], pegawai['jabatan'],
pegawai['divisi'], pegawai['gaji_pokok'], pegawai.get('gaji_per_jam', 0)] for
pegawai in sorted_pegawai]
        print(tb(data, headers=headers, stralign="right"))

    def tampilkan_gaji_bulan(self, bulan,urut='nip'):
        sorted_pegawai = sorted(self.pegawai_list, key=lambda x:
x['gaji_pokok'] + x.get('gaji_per_jam', 0))
        for pegawai in sorted_pegawai:
            gaji_total = pegawai['gaji_pokok']
            if 'gaji_per_jam' in pegawai:

```

```

        # Fungsi jam_kerja_bulan harus diimplementasikan sesuai
kebutuhan
        gaji_total += pegawai['gaji_per_jam'] *
jam_kerja_bulan(pegawai['nip'], bulan)
        print(f"NIP: {pegawai['nip']}, Nama: {pegawai['nama']}, Gaji Bulan
{bulan}: {gaji_total}")

    def tampilkan_list_divisi(self):
        print("List Divisi:")
        for divisi in self.divisi_list:
            print(divisi)

    def tampilkan_list_divisi_gaji(self):
        print("List Divisi beserta Standar Gajinya:")
        print("List Divisi beserta Standar Gajinya:")
        print("HR : 1.000.000")
        print("IT : 2.000.000")
        print("RND : 3.000.000")

    def tampilkan_pegawai_divisi(self, divisi):
        print(f"List Pegawai di Divisi {divisi}:")
        for pegawai in self.pegawai_list:
            if pegawai['divisi'] == divisi:
                print(f"NIP: {pegawai['nip']}, Nama: {pegawai['nama']},
Jabatan: {pegawai['jabatan']}")

    def tampilkan_pejabat_kantor(self):
        for pegawai in self.pegawai_list:
            if pegawai['jabatan'] in ['Direktur', 'Manager']:
                print(f"NIP: {pegawai['nip']}, Nama: {pegawai['nama']},
Jabatan: {pegawai['jabatan']}")

    def tampilkan_total_gaji_bulan(self, bulan):
        total_gaji = 0
        for pegawai in self.pegawai_list:
            gaji_total = pegawai['gaji_pokok']
            if 'gaji_per_jam' in pegawai:
                # Fungsi jam_kerja_bulan harus diimplementasikan sesuai
kebutuhan
                gaji_total += pegawai['gaji_per_jam'] *
jam_kerja_bulan(pegawai['nip'], bulan)
            total_gaji += gaji_total
        print(f"Total Gaji yang Harus Dibayar pada Bulan {bulan}:
{total_gaji}")

    def input_data_pegawai(direktur_sdm, jumlah_pegawai, menu):
        for _ in range(jumlah_pegawai):
            nip = input("\nMasukkan NIP Pegawai: ")
            nama = input("Masukkan Nama Pegawai: ")

```

```

jabatan = input("Masukkan Jabatan Pegawai: ")
divisi = input("Masukkan Divisi Pegawai: ")

if menu == 1:
    direktur_sdm.tambah_pegawai(nip, nama, jabatan, divisi)
elif menu == 2:
    gaji_pokok = int(input("Masukkan Gaji Pokok Pegawai: "))
    gaji_per_jam = int(input("Masukkan Gaji per Jam Pegawai: "))
    direktur_sdm.tambah_gaji(nip, gaji_pokok, gaji_per_jam)

return direktur_sdm

def main(direktur_sdm):
    jawab = "ya"
    while jawab.lower() == "ya":
        print("Selamat datang di aplikasi Gaji Pegawai !")
        print("1. Admin")
        print("2. Direktur SDM")
        print("3. Direktur")
        x = int(input("Masukkan pilihan menu anda : "))

        if x == 1:
            admin = Admin()
            while not admin.is_logged_in:
                user = input("Masukkan username : ")
                pw = input("Masukkan password : ")
                hasil = admin.login(user, pw)
                print(hasil)
                if admin.is_logged_in:
                    admin.edit()

        elif x == 2:
            direktur_sdm = DirekturSDM()
            while not direktur_sdm.is_logged_in:
                user = input("Masukkan username : ")
                pw = input("Masukkan password : ")
                hasil = direktur_sdm.login(user, pw)
                print(hasil)

            if direktur_sdm.is_logged_in:
                jawab = "ya"
                while jawab.lower() == "ya":
                    print("\n1. Menambah, mengedit data pegawai")
                    print("2. Menambah, mengedit data gaji pokok dan gaji per
jam")
                    print("3. Menampilkan semua list pegawai beserta atribut
data lainnya denganurut NIP")
                    print("4. Menampilkan list pegawai dan gajinya pada bulan
tertentu denganurut NIP atauurut gaji")

```

```

        print("5. Menampilkan list divisi")
        print("6. Menampilkan list divisi beserta standar
gajinya")

        print("7. Menampilkan list pegawai di divisi tertentu")
        print("8. Menampilkan daftar pejabat di kantor")
        print("9. Menampilkan total gaji yang harus dibayar setiap
bulan")

        y = int(input("Masukkan pilihan menu anda : "))

        if y in [1, 2]:
            a = int(input("Masukkan jumlah pegawai yang ingin di
input : "))

            direktur_sdm = input_data_pegawai(direktur_sdm, a, y)
        elif y == 3:
            direktur_sdm.tampilkan_semua_pegawai()
        elif y == 4:
            bulan = input("\nMasukkan Bulan untuk Menampilkan
Gaji: ")

            print(f"\nGaji Pegawai pada Bulan {bulan}: ")
            direktur_sdm.tampilkan_gaji_bulan(bulan)
        elif y == 5:
            direktur_sdm.tampilkan_list_divisi()
        elif y == 6:
            direktur_sdm.tampilkan_list_divisi_gaji()
        elif y == 7:
            divisi = input("\nMasukkan Divisi untuk Menampilkan
Pegawai: ")

            print(f"\nPegawai di Divisi {divisi}:")
            direktur_sdm.tampilkan_pegawai_divisi(divisi)
        elif y == 8:
            print("\nDaftar Pejabat di Kantor:")
            direktur_sdm.tampilkan_pejabat_kantor()
        elif y == 9:
            bulan_total_gaji = input("\nMasukkan Bulan untuk
Menampilkan Total Gaji: ")
            print(f"\nTotal Gaji yang Harus Dibayar pada Bulan
{bulan_total_gaji}:")
            direktur_sdm.tampilkan_total_gaji_bulan(bulan_total_ga
ji)

        jawab = input("Apakah ingin kembali ke menu Direktur SDM?
(ya/tidak): ")

        if jawab.lower() != "ya":
            main(direktur_sdm)

    elif x == 3 :
        direktur = Direktur()
        while not direktur.is_logged_in:
            user = input("Masukkan username : ")

```

```

        pw = input("Masukkan password : ")
        hasil = direktur.login(user, pw)
        print(hasil)

    if direktur.is_logged_in:
        jawab = "ya"
        while jawab == "ya":
            print("1. Menampilkan semua list pegawai beserta atribut
data lainnya denganurut NIP")
            print("2. Menampilkan list pegawai dan gajinya pada bulan
tertentu denganurut NIP atauurut gaji")
            print("3. Menampilkan list divisi")
            print("4. Menampilkan list divisi beserta standar
gajinya")

            print("5. Menampilkan list pegawai di divisi tertentu")
            print("6. Menampilan daftar pejabat di kantor")
            print("7. Menampilkan total gaji yang harus dibayar setiap
bulan")

            y = int(input("Masukkan pilihan menu anda : "))
            if y == 1 :
                direktur_sdm.tampilkan_semua_pegawai()
            elif y == 2 :
                bulan = input("\nMasukkan Bulan untuk Menampilkan
Gaji: ")

                print(f"\nGaji Pegawai pada Bulan {bulan}: ")
                direktur_sdm.tampilkan_gaji_bulan(bulan)
            elif y == 3 :
                direktur_sdm.tampilkan_list_divisi()
            elif y == 4 :
                direktur_sdm.tampilkan_list_divisi_gaji()
            elif y == 5 :
                divisi = input("\nMasukkan Divisi untuk Menampilkan
Pegawai: ")

                print(f"\nPegawai di Divisi {divisi}:")
                direktur_sdm.tampilkan_pegawai_divisi(divisi)
            elif y == 6 :
                print("\nDaftar Pejabat di Kantor:")
                direktur_sdm.tampilkan_pejabat_kantor()
            elif y == 7 :
                bulan_total_gaji = input("\nMasukkan Bulan untuk
Menampilkan Total Gaji: ")
                print(f"\nTotal Gaji yang Harus Dibayar pada Bulan
{bulan_total_gaji}:")
                direktur_sdm.tampilkan_total_gaji_bulan(bulan_total_ga
ji)

                jawab = input("Apakah masih ingin menggunakan?(ya/tidak):
")

                print("Thankyou for using our Services, Logged out.")
        else :

```

```
        return f"Invalid submit."

if __name__ == "__main__":
    main(DirekturSDM)
```