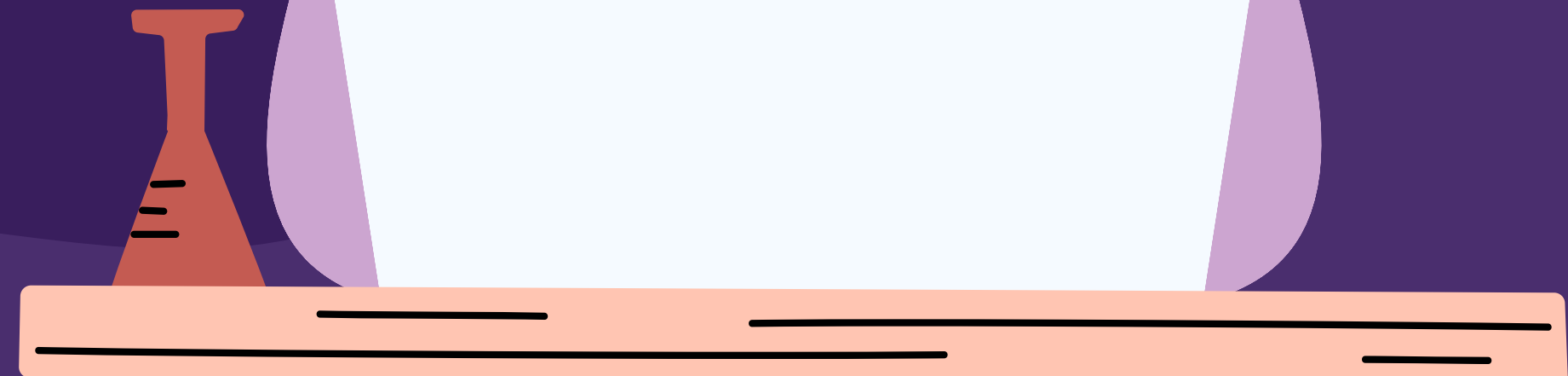
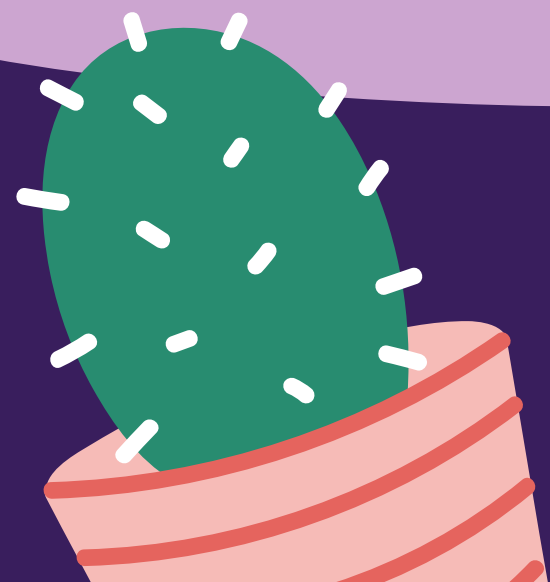





# PROJECT BASE LEARNING

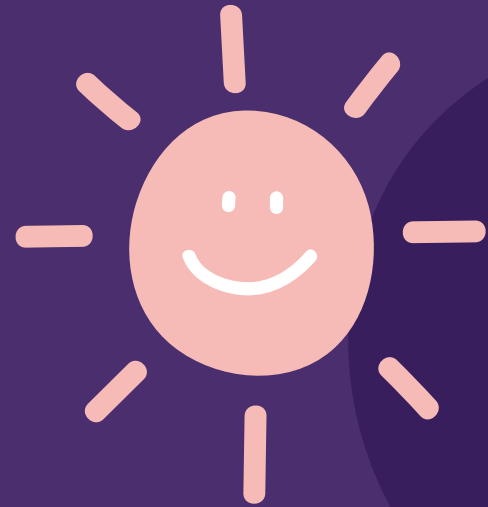
Menghitung Gaji Karyawan



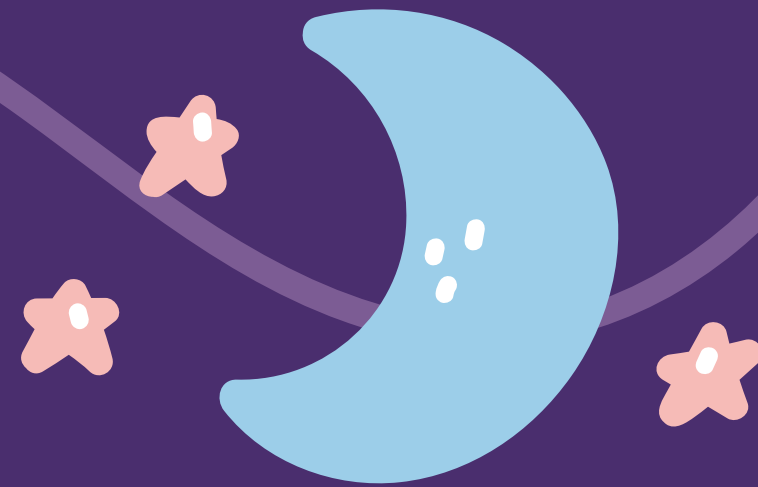


# PROJECT BASE LEARNING “MENGHITUNG GAJI KARYAWAN”

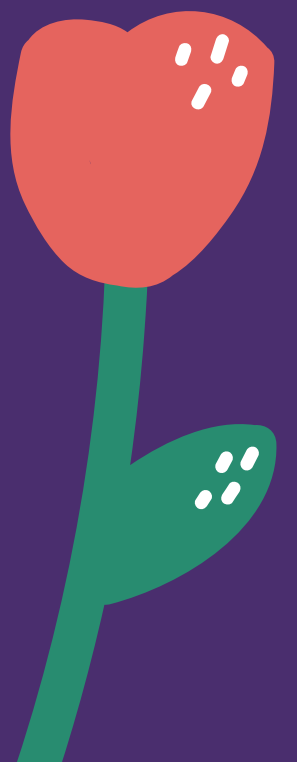
Mata Kuliah: Fundamental Pemrograman  
Dosen Pengampun: Vikri Aulia, S.pd, M.Kom



DIBUAT OLEH:



AUDITYA MEROSYA PUTRI  
24343001



# LATAR BELAKANG

Perusahaan dengan jumlah karyawan yang banyak membutuhkan sistem efisien untuk menghitung gaji setiap karyawan. Dengan perhitungan gaji karyawan yang efisien dan tepat membuat setiap pegawai dapat menerima imbalan yang sesuai dengan kontribusi yang dilakukan oleh pegawai. Selain itu, memberikan perhitungan rinci dan jelas sehingga memudahkan dalam analisis keuangan dan perencanaan anggaran perusahaan.



# TUJUAN CAPAIAN



Tujuan utama dari program menghitung gaji pegawai antara lain:

1. Mengurangi kesalahan ketika melakukan perhitungan secara manual.
2. Menghemat waktu dimana hanya dengan memasukkan data yang dibutuhkan.
3. Perhitungan yang didapatkan lebih akurat dengan memasukkan rumus perhitungan gaji yang ditetapkan berdasarkan kebutuhan.





# FITUR UTAMA

1. Fitur input jumlah pegawai dengan validasi untuk memastikan jumlah pegawai yang diinputkan dalam batas yang ditetapkan.
2. Fitur pengelolaan data pegawai untuk menyimpan informasi pegawai berupa nama, jabatan, jam kerja, upah perjam, bonus dan gaji total dengan menggunakan struct.
3. Fitur pilih jabatan sesuai dengan jabatan yang diembani oleh pegawai, serta sesuai dengan validasi pilihan yang akan dipilih.
4. Fitur inputkan data pegawai dimana memungkinkan untuk memasukkan nama, jabatan, jam kerja, upah perjam, dan bonus.



# FITUR UTAMA

5. Fitur perhitungan gaji pegawai berdasarkan formula:

$$\text{gaji total} = \text{upah perjam} \times \text{jam kerja} + \text{bonus}$$

Dimana menggunakan fungsi hitung gaji untuk perhitungannya.

6. Fitur menampilkan output data gaji pegawai, dengan menampilkan ringkasan setiap data pegawai yang sudah diinputkan.



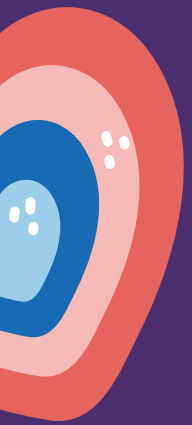
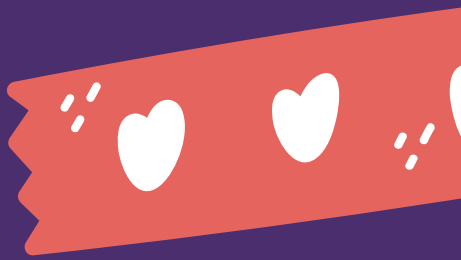
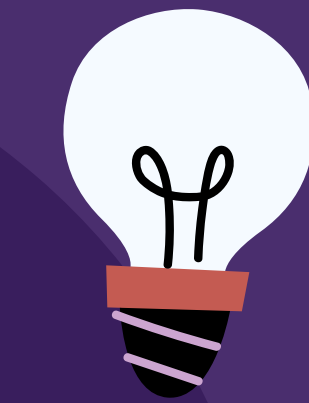
# INPUT

Input yang diminta oleh program, yaitu:

1. Jumlah pegawai yang datanya akan diinputkan.

2. Data setiap pegawai:

- nama lengkap dari pegawai
- jabatan dari pilihan yang tersedia berupa: manager, supervisor, dan staff.
- jam kerja pegawai
- upah perjam yang diterima
- bonus yang diterima pegawai



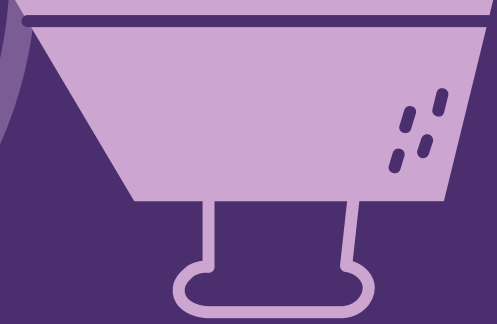
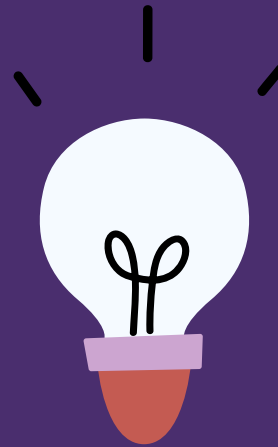


# OUTPUT

Output yang ditampilkan oleh program, yaitu:

Setelah semua data diinputkan, program akan menampilkan ringkasan setiap pegawai.

- Nama Pegawai
- Jabatan
- Upah Per Jam
- Jam Kerja
- Bonus
- Gaji Total: Menghitung total gaji berdasarkan rumus:  
$$\text{GajiTotal} = (\text{Jam Kerja} \times \text{Upah Per Jam}) + \text{BonusGaji}$$



# PROSES

1. Inisialisasikan konstanta jumlah pegawai untuk menyimpan banyaknya jumlah pegawai.
2. Deklarasikan struktur data untuk menyimpan informasi pegawai berupa: nama, jabatan, jam kerja, upah perjam, bonus, dan gaji total.
3. Fungsi hitung gaji untuk menginputkan total gaji.
4. Program meminta pengguna untuk menginputkan jumlah pegawai yang akan dimasukkan.
5. Kemudian akan divalidasi dimana jika jumlah pegawai tidak valid (kurang dari 1 atau lebih dari konstanta jumlah pegawai yang dapat diinputkan, maka akan menampilkan pesan kesalahan atau keluar dari program.



# PROSES

6. Inisialisasikan array dari pointer pilihan jabatan.
7. Inputkan data pegawai dengan menginisiasikannya dengan array.
8. Setiap data akan diinputkan satu per satu melalui loop.
9. Inputkan nama.
10. Inputkan jabatan. Program nantinya akan menampilkan pilihan jabatan yang telah disediakan (Direktur, Manager, Supervisor, dan Staff).
11. Program akan meminta pengguna untuk menginputkan angka sesuai dengan pilihan jabatan yang ditampilkan.
12. Pilihan jabatan ini menggunakan perulangan. Jika angka yang diinputkan tidak valid, maka program akan meminta pengguna untuk meninputkan lagi.

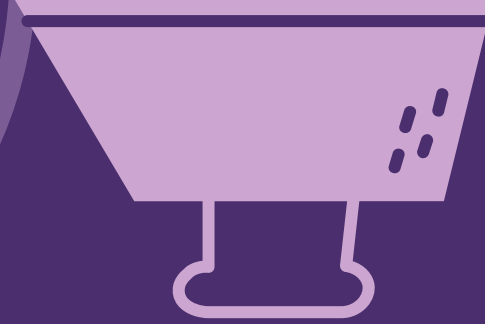


# PROSES

13. Inputkan jam kerja dan upah perjam dan bonus untuk pegawai.
14. Menghitung gaji total, fungsi hitung gaji dipanggil untuk menghitung total gaji pegawai berdasarkan jam kerja, upah per jam, dan bonus.
15. Setelah semua data diinputkan dan gaji dihitung maka program akan menampilkan data dan gaji total untuk setiap pegawai.
16. Setelah semua data ditampilkan maka program akan berakhir dan mengembalikan nilai 0 yang menandakan program sudah selesai atau sukses dilakukan.



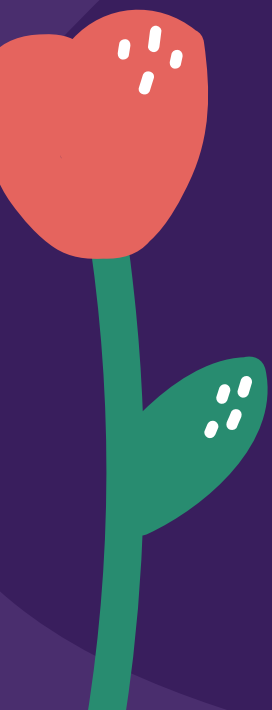
# STUKTUR DATA



A. Menggunakan struct pegawai didefenisikan kumpulan untuk menyimpan beberapa variabel yang beragam dari informasi setiap data pegawai yang dikumpulkan dalam satu variabel berupa struct pegawai

B. Struct pegawai berisi beberapa variabel dengan tipe data seperti berikut:

- Char nama pegawai : tipe data array yang berisi kumpulan karakter yang digunakan untuk menyimpan nama pegawai.
- Char jabatan : tipe data array yang digunakan untuk menyimpan jabatan berupa Manager, Supervisor, dan Staff.
- Int jam kerja : tipe data untuk menyimpan bilangan berupa jam kerja dalam periode.







- Float upah perjam : tipe data untuk menyimpan nilai dari jumlah upah perjam.
- Float bonus : tipe data untuk menyimpan jumlah bonus yang didapatkan.
- Float gaji total : tipe data untuk menyimpan jumlah gaji yang telah dilakukan dengan perhitungan gaji.

C. Fungsi dan pointer hitung gaji: Di fungsi hitung gaji digunakan konsep pointer ke variabel berupa parameter upah perjam, jam kerja, dan bonus yang dapat diakses dengan mudah. Kemudian menghitung gajiTotal, dimana fungsi ini menggunakan parameter pointer untuk mengubah nilai gajiTotal secara langsung tanpa perlu mengembalikan nilai. Dimana hasil dari perhitungan gaji total disimpan di alamat memori gaji total.



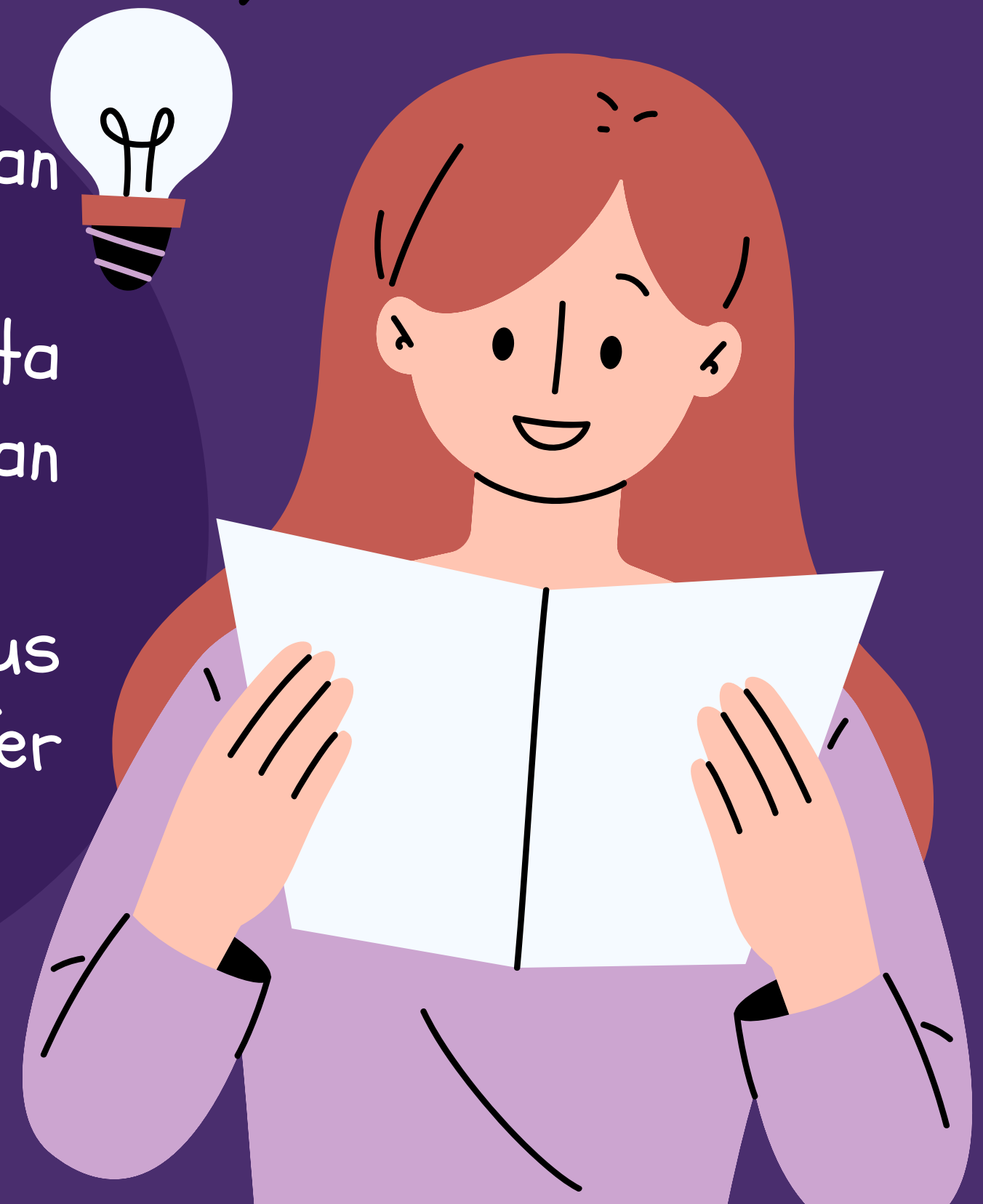
D. Pointer dan array pilihan jabatan : array ini digunakan untuk menyimpan tampilan pilihan jabatan pegawai. Pointer disini berarti setiap data array merupakan pointer yang artinya akan menyimpan masing-masing alamat memori dari array pilihan jabatan.

E. Array struct pegawai: sebuah array dari tipe data struct Pegawai digunakan untuk menyimpan data pegawai yang jumlahnya bisa bervariasi sesuai dengan input dari pengguna dengan konstanta yang ditetapkan untuk jumlah pegawai. Setiap elemen array adalah sebuah struktur Pegawai yang menyimpan informasi tentang satu pegawai.

F. Variabel yang digunakan bertipe data integer:

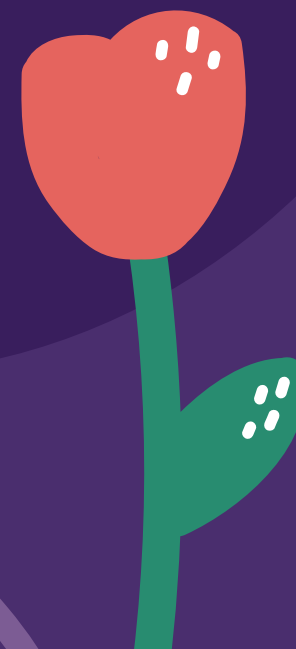
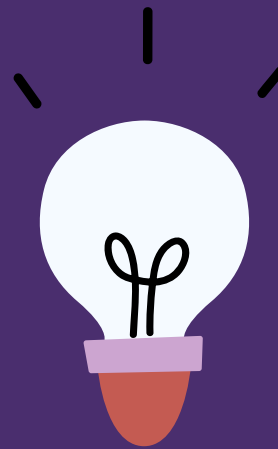
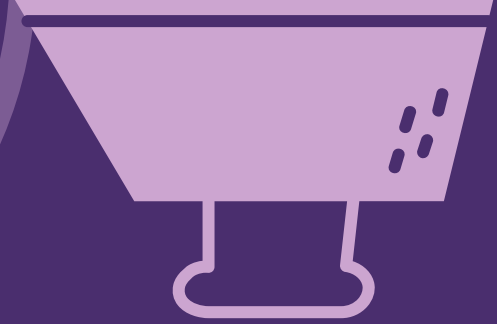
- int jumlah pegawai untuk menyimpan jumlah pegawai yang akan diinputkan.
- int pilihan untuk menyimpan pilihan jabatan yang akan diinputkan.
- int valid jabatan untuk menyimpan apakah data yang diinputkan oleh pengguna dari pilihan jabatan (1-3) adalah valid.

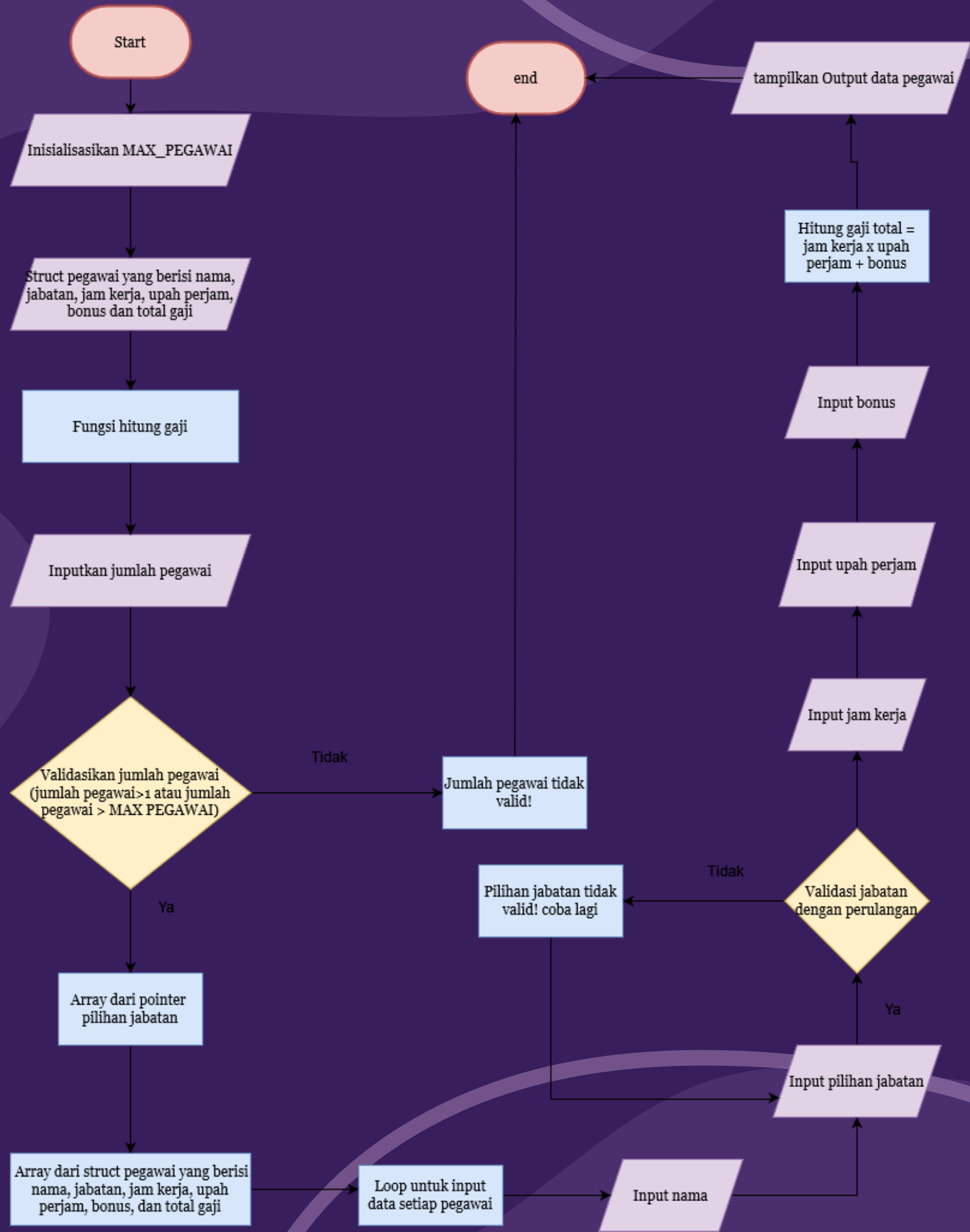
I. Getchar: Fungsi ini digunakan untuk menghapus karakter newline yang mungkin tertinggal di buffer setelah pemrograman input lainnya, seperti scanf.



G. Fungsi strcpy : digunakan untuk meng-copy string dari sebuah variabel ke variabel seperti nama pegawai dengan batas panjang tertentu. Fungsi ini lebih aman dibandingkan dengan scanf, karena dapat menangani spasi dan buffer overflow.

H. Fungsi strcspn : digunakan untuk menghapus karakter newline (\n) yang mungkin tertinggal dalam string setelah input menggunakan fgets().





# Flowchart

