

1. Fungsi dan tipe return

Di C, fungsi bisa mengembalikan berbagai tipe data:

Tipe Return	Contoh Fungsi	Catatan
int	int tambah(int a, int b)	Mengembalikan bilangan bulat
float	float rata(float x, float y)	Mengembalikan bilangan desimal
double	double luasLingkaran(double r)	Presisi lebih tinggi dari float
char	char getHuruf()	Mengembalikan satu karakter
void	void cetakHalo()	Tidak mengembalikan apa-apa
enum	enum Status evenCheck(int n)	Mengembalikan nilai enum (seperti int tapi dengan label)
struct	struct Mahasiswa buatMhs()	Mengembalikan struct, misal data mahasiswa
union	union Data buatData()	Mengembalikan union, mirip struct tapi hemat memori
Pointer	int* cari(int arr[], int n)	Mengembalikan alamat memori variabel/array

1. Intinya per tipe

void → fungsi tidak mengembalikan apa-apa

```
void halo() { printf("Halo\n"); }
```

int, float, double, char → fungsi mengembalikan nilai tunggal

```
int tambah(int a, int b) { return a + b; }
```

enum → seperti int, tapi ada label yang deskriptif

```
enum Status {FALSE, TRUE};  
enum Status cekGenap(int n) { return (n%2==0) ? TRUE : FALSE; }
```

struct / union → fungsi bisa mengembalikan data kompleks

```
struct Mahasiswa { int id; char nama[20]; };  
struct Mahasiswa buatMhs() { struct Mahasiswa m = {1, "Hans"}; return m; }
```

Pointer → mengembalikan alamat memori (sering dipakai untuk array atau dynamic memory)

```
int* getArray(int size) { return malloc(size * sizeof(int)); }
```

1. Kesimpulan sederhana
2. Semua tipe data di C bisa dijadikan return type fungsi, termasuk enum, struct, union, bahkan pointer.
3. Pilihan tipe tergantung apa yang mau dikembalikan fungsi:
4. Nilai sederhana → int, float, char
5. Status/opsi → enum
6. Data kompleks → struct/union
7. Tidak perlu mengembalikan apa-apa → void