**Flowchart**

ダイアグラム

自動的に生成された説明

**Program**

#include <stdio.h>

#include <string.h>

// Inisialisasi Struct & Union

union media

{

char buku[30];

char majalah[30];

};

struct pustaka

{

char judul[100];

int tahun;

union media jenis;

} item;

void InputData (struct pustaka \*item);

void OutputData (struct pustaka \*item);

int main()

{

int tipe;

// Start Page

printf ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf ("| SISTEM INFORMASI |\n");

printf ("| PERPUSTAKAAN SEDERHANA |\n");

printf ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

// Input Data

InputData (&item);

getchar ();

printf ("\nInformasi item sudah tersimpan. Tekan ENTER untuk menampilkan hasil . . .");

getchar ();

// Output Data

OutputData (&item);

printf ("TERIMA KASIH\n");

return 0;

}

void InputData (struct pustaka \*item)

{

int tipe;

// Input Data

printf ("\n\* Masukkan Informasi Item \*\n");

printf ("---------------------------\n\n");

printf ("Judul Item : ");

scanf (" *%*[^\n]", item->judul);

printf ("Tahun Terbit : ");

scanf ("%d", &item->tahun);

printf ("\nJenis Item \n< 1 > Buku\n< 2 > Majalah\nPilih Jenis Item : ");

scanf ("%d", &tipe);

// Set Union

if (tipe == 1)

{

strcpy (item->jenis.buku, "Buku");

}

else if (tipe == 2)

{

strcpy (item->jenis.majalah, "Majalah");

}

else

{

strcpy (item->jenis.buku, "Maaf, pilihan anda tidak tersedia.");

}

}

void OutputData (struct pustaka \*item)

{

// Output Data

printf ("\n\n>> Informasi Item <<\n");

printf ("--------------------\n\n");

printf ("Judul Item : %s\n", item->judul);

printf ("Tahun Terbit : %d\n", item->tahun);

printf ("Jenis Item : %s\n\n", item->jenis.buku);

}

**Kelebihan Menggunakan Union dalam Kasus Ini**

Union adalah tipe data khusus dalam bahasa C yang memungkinkan beberapa anggota berbagi ruang memori yang sama. Dalam kasus ini, union digunakan untuk menyimpan jenis media (buku atau majalah) di dalam struktur pustaka. Berikut adalah penjelasan rinci mengenai kelebihan menggunakan union, termasuk efisiensi penyimpanan memori:

**Efisiensi Penyimpanan Memori**

* **Berbagi Memori** : Union hanya menggunakan satu blok memori yang cukup besar untuk menampung anggota terbesarnya. Dalam kode Anda, union media memiliki dua anggota:
  + char buku[30]
  + char majalah[30]

Karena kedua anggota memiliki ukuran yang sama (30 byte), union hanya akan menggunakan **30 byte** untuk menyimpan salah satu dari kedua nilai tersebut. Jika menggunakan struct, kedua anggota akan dialokasikan memori secara terpisah, sehingga total penggunaan memori menjadi **60 byte** (30 byte untuk buku + 30 byte untuk majalah).

* **Penghematan Memori** : Dengan union, hanya satu jenis media (buku atau majalah) yang disimpan pada satu waktu, sehingga tidak ada pemborosan memori untuk menyimpan data yang tidak digunakan.

**Fleksibilitas dalam Representasi Data**

* **Satu Wadah untuk Banyak Tipe** : Union memungkinkan Anda menyimpan data dengan tipe yang berbeda dalam satu wadah, tetapi hanya satu tipe yang aktif pada satu waktu. Misalnya:
  + Jika item adalah **buku** , maka union hanya menyimpan informasi tentang buku.
  + Jika item adalah **majalah** , maka union hanya menyimpan informasi tentang majalah.

Ini membuat representasi data lebih fleksibel tanpa perlu mendefinisikan variabel tambahan.

**Output**

テキスト

自動的に生成された説明