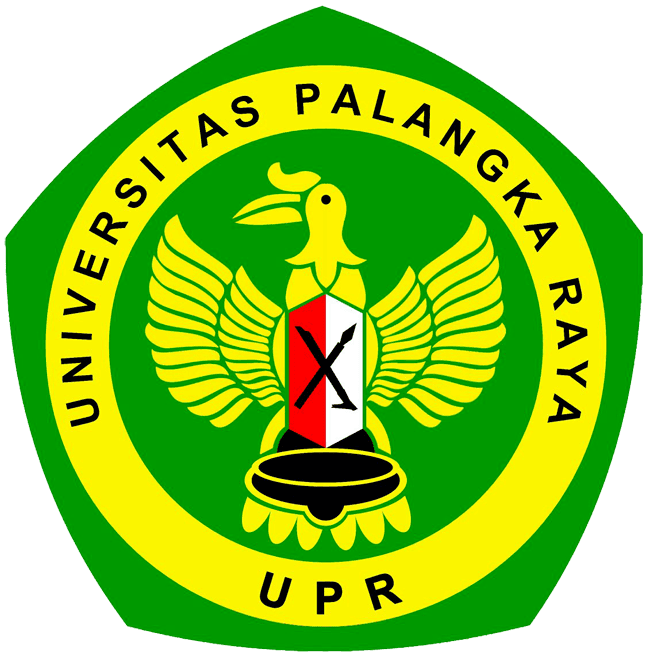
**LAPORAN RENCANA PRAKTIKUM**

**STRUKTUR DATA**



|  |  |  |
| --- | --- | --- |
| **NAMA** | **:** | **INDRA FIQI RIPANI** |
| **NIM** | **:** | **213010503002** |
| **KELAS** | **:** | **F** |
| **MODUL** | **:** | **1 (ARRAY, STRUCT, DAN POINTER)** |

**Program Studi S1 Teknik Informatika**

**Fakultas Teknik**

**Universitas Palangka Raya**

**Palangka Raya, Kalimantan Tengah**

**2022**

**MODUL 1**

**ARRAY, STRUCT, DAN POINTER**

1. **TUJUAN PRAKTIKUM**
2. Mahasiswa mampu mengimplementasikan Array, Struct, dan Pointer.
3. **Landasan Teori**
4. Array

Array adalah suatu tipe data terstruktur yang berupa sejumlah data sejenis (bertipe data sama) yang jumlahnya tetap dan diberi suatu nama tertentu. Array dapat berupa array 1 dimensi, 2 dimensi, bahkan n-dimensi.

1. Deklarasi

|  |
| --- |
| tipe\_data nama\_var [ukuran];  tipe\_data: tipe data elemen larik (int, char, float, dll)  nama\_var: menyatakan nama variabel yang dipakai.  ukuran: menunjukkan jumlah maksimal elemen larik. |

|  |
| --- |
| Contoh :  int nilai[6]; |

1. Inisialisasi

Menginisialisasi array sama dengan memberikan nilai awal array pada saat didefinisikan.

|  |
| --- |
| int nilai[5] = {8,7,5,6,4,3};  //atau  int a [2];  a [0] = 10;  a [1] = 3;  a [2] = 1; |

1. Pengaksesan

Pengisian dan pengambilan nilai pada indeks tertentu dapat dilakukan dengan mengeset nilai atau pun menampilkan nilai pada indeks yang dimaksud. Pengaksesan elemen array dapat dilakukan berurutan atau random berdasarkan indeks tertentu secara langsung.

1. Struct

Struct adalah bentuk struktur data yang dapat menyimpan variabel-variabel dalam 1 nama, namun memiliki tipe data yang berbeda ataupun sama. Variable-variabel tersebut memiliki kaitan satu sama yang lain.

1. Deklarasi

Ada 2 cara pendeklarasian struct, yaitu:

|  |
| --- |
| typedef struct Mahasiswa { //cara pertama  char NIM[8];  char nama[50];  float ipk;  }  //atau  struct { //cara kedua  char NIM[8];  char nama[50];  float ipk;  }; |

Contoh program:

|  |
| --- |
| #include <iostream>  int main(){  struct orang{  char nama[40];  short umur;  }saya;  cout<<”nama : “; cin.getline(saya.nama,40);  cout<<”umur : “; cin>>saya.umur;  cout<<saya.nama<<” berumur “<<saya.umur; |

1. Array of Struct

Apabila struct akan digunakan untuk beberapa kali, ada 2 cara:

|  |
| --- |
| #include <stdio.h>  typedef struct Mahasiswa {char NIM[8];  char NIM[8];  char nama[50]  float ipk;  }  int main() {  Mahasiswa a,b,c; //cara pertama  //atau  Mahasiswa mhs[3]; //cara kedua  .....  } |

1. Pointer

Pointer adalah penunjuk suatu variabel. Karena menunjuk suatu variabel, maka pointer wajib memiliki alamat dari variabel yang ditunjuknya itu. Dengan mekanisme pointer, suatu variabel di dalam suatu fungsi dapat diakses oleh fungsi yang lain.

Sebagai analogi, sebuah nomor halaman dalam buku dapat dianggap sebagai pointer ke halaman yang sesuai; *dereferencing* seperti pointer akan dilakukan dengan membalik ke halaman dengan nomor halaman yang diberikan.

|  |  |
| --- | --- |
| P  Pointer | Alamat ke-1 Alamat |
| Alamat ke-2 Alamat |
| Alamat ke-3 Alamat |
| Alamat ke-4 Alamat |
| Alamat ke-N Alamat |

Gambar 1.1 Ilustrasi Pointer

Ada dua operator pada pointer yaitu operator *deference* dan *reference*.

1. Deference (&)

Deference dengan simbol & (ampersand) merupakan suatu operator yang berfungsi menanyakan alamat dari suatu variabel. Memberikan simbol & pada awal variabel dan mencetak hasilnya pada jendela CLI, maka yang akan tercetak adalah alamat dari variabel tersebut bukan nilai yang ditampung oleh variabel tersebut. Berikut merupakan contoh programnya:

|  |
| --- |
| #include <iostream>  int main(){  int a = 5;  cout<<"Alamat Variabel a adalah :"<<&a<<endl;  cout<<"Nilai Variabel a adalah :"<<a<<endl;  return 0;  } |

1. Reference (\*)

Reference dengan simbol \* (asterisk) merupakan suatu operator yang berfungsi menyatakan suatu variabel adalah variabel pointer. Sama halnya dengan operator deference, peletakan simbol operator reference diletakan diawal variabel. Operator reference ini akan membuat suatu variabel pointer untuk menampung alamat. Perhatikan contoh berikut:

|  |
| --- |
| #include <iostream>  int main(){  int a=5; //Memberikan nilai 5 pada variabel a  int \*b; //Mendeklarasikan variabel b sebagai pointer  b = &a; //Mengkopikan alamat a kedalam pointer b  cout<<"Nilai variabel a adalah "<<a<<endl;  cout<<"Alamat variabel a adalah "<<&a<<endl;  cout<<"Isi dari variabel b adalah "<<b<<endl;  cout<<"Nilai yang tertampung dalam variabel b adalah  "<<\*b<<endl;  return 0;  } |

Ada beberapa cara atau pengoperasian pointer yang dapat dilakukan, diantaranya:

1. Pointer dan Array

Pointer dapat digunakan sebagai penunjuk dari nilai index yang akan digunakan pada sebuah array, bahkan pointer itu sendiri pun dapat di berikan index sama seperti array.

|  |
| --- |
| #include <iostream>  int main() {  int nilaiArray[3] = {1,2,3};  int \*pointerArray;  pointerArray = nilaiArray;  for (int i = 0; i<3; i++){  cout<<”index [“ << i << ”] = “;  cout<<\*(pointerArray+i)<<endl;  }  } |

1. Pointer dan Struct

|  |
| --- |
| #include <iostream>  struct mhs{  int nim;  string nama;  };  int main() {  mhs m;  mhs \*p = &m;  //struct biasa  m.nama = "Anne";  cout<<m.nama<<endl;  //struct pointer  p->nim = 123;  cout<<p->nim; |

1. Pointer sebagai paramater suatu fungsi

Pointer dapat juga digunakan sebagai parameter pada sebuah fungsi. Parameter ini akan bersifat by reference yang mana artinya apa yang dikirimkan sebagai parameter adalah alamat variabelnya, yaitu bukan nilainya. Jadi perubahan nilai yang terjadi di dalam fungsinya akan mengubah juga dari variabel alamat aslinya.

|  |
| --- |
| #include <iostream>  void tambah(int \*angka){  \*angka +=20;  }  int main() {  int nilai = 10;  cout<<”Nilai = “<<nilai<<endl;  tambah(&nilai);  cout<<”Nilai = “<<nilai<<endl;  return 0;  } |

1. **TUGAS**
2. Buatlah sebuah program untuk memasukan NIM, nama, email mahasiswa menggunakan Array Of Struct sebanyak N inputan. Selain itu, buatlah juga program untuk menampilkan data-data yang telah dimasukkan tadi dan gunakan looping untuk menambahkan dan memasukkan datanya!
3. Cobalah source code dibawah ini! Perbaikilah jika ada kesalahan, sehingga program mampumenampilkan output deretan bilangan fibonacci dengan benar. Jelaskan kedalam laporan anda.

|  |
| --- |
| #include <iostream>  #define max 50  using namespace std;  int\* fibo(int n){  int arr[max];  arr[0] = 1;  arr[1] = 1;  for (int i = 2; i < n; i++){  arr[i] = arr[i - 1] + arr[i - 2];  }  return arr;  }  int main(){  int num, \*ptr;  cout << "jumlah bilangan fibonacci : ";  cin >> num;  ptr = fibo(num);  for (int i = 0; i < num; i++){  cout << (ptr + i) << " ";  }  } |