**LAPORAN PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**

Modul Praktikum 1



Disusun oleh:

|  |  |
| --- | --- |
| Nama | : Yusuf Budiawan Azri |
| NIM | : 2407135509 |
| Dosen | : T. Yudi Hadiwandra, S.Kom, M.Kom |
| Asisten | : 1. <<asdos 1>>  2. <<asdos 2>> |

**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK**

**UNIVERSITAS RIAU**

**GENAP 2021/2022**

# LEMBAR PENGESAHAN LAPORAN

Judul Praktikum : Array, Pointer, dan Struktur

Nama Instruktur :

Nama Asisten : 1. <<asdos 1>>

2. <<asdos 2>>

Nama Praktikan : ….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Bab** | **Tanggal pengesahaan** | **Keterangan** | **Paraf Asisten** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Menyetujui,

<<asdos 1>> <<asdos 2>>

…. …..

# LEMBAR ASISTENSI PRAKTIKUM

Nama Praktikum : …

Nama Asisten : 1. <<asdos 1>>

2. <<asdos 2>>

Nama Praktikan : …

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Hasil Pemeriksaan** | **Perbaikan** | **Paraf Asisten** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Menyetujui,

<<asdos 1>> <<asdos 2>>

…. …..

1. **TUGAS PENDAHULUAN**
2. Tuliskan algoritma pengurutan *....*.

|  |
| --- |
|  |

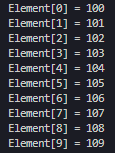
1. Tuliskan algoritma ......

|  |
| --- |
|  |

1. **PERCOBAAN**
2. Percobaan 1 : Cara memberikan nilai pada array dan mengakses array

|  |
| --- |
| #include <stdio.h>  *void* main () {  *int* n[ 10 ];  *int* i,j;      for ( i = 0; i < 10; i++ ) {          n[ i ] = i + 100;      }      for (j = 0; j < 10; j++ ) {          printf("Element[%d] = %d\n", j, n[j] );      }  } |

*Output* percobaan 1 :



1. Percobaan 2 : Menghitung total dari nilai yang terdapat pada array

|  |
| --- |
| #include <stdio.h>  #define SIZE 12  void main() {  int a[SIZE] = { 1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45 };  int i = 0;  int total = 0;  for (i = 0; i < SIZE; ++i) {  total += a[i];  }  printf("Total elemen yang terdapat di array : %d\n", total);  } |

*Output* percobaan 2 :



1. Percobaan 3 : Penggunaan Array pada Bilangan Fibonaci

|  |
| --- |
| #include <stdio.h>  #define MAX 20  int fibo[MAX];  void main() {  int i;  fibo[1] = 1;  fibo[2] = 1;  for (i = 3; i <= MAX; i++) {  fibo[i] = fibo[i - 2] + fibo[i - 1];  }  printf("%d Bilangan Fibonacci Pertama adalah:\n", MAX);    for (i = 1; i <= MAX; i++) {  printf("%d ", fibo[i]);  }  printf("\n");  } |

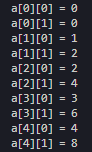
*Output* percobaan 3 :



1. Percobaan 4 : Cara mengakses array dua dimensi

|  |
| --- |
| #include <stdio.h>  void main() {  int a[5][2] = { {0, 0}, {1, 2}, {2, 4}, {3, 6}, {4, 8} };  int i, j;  for (i = 0; i < 5; i++) {  for (j = 0; j < 2; j++) {  printf("a[%d][%d] = %d\n", i, j, a[i][j]);  }  }  } |

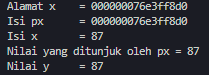
*Output* percobaan 4 :



1. Percobaan 5 : Program Merubah Isi Variabel melalui Pointer

|  |
| --- |
| #include <stdio.h>  void main() {  int y, x = 87; // x & y bertipe int  int \*px; // var pointer yang menunjuk ke data yang bertipe int  x = 87;  px = &x; // px diisi dengan alamat dari variabel x  y = \*px; // y diisi dengan nilai yang ditunjuk oleh px  printf("Alamat x = %p\n", &x);  printf("Isi px = %p\n", px);  printf("Isi x = %d\n", x);  printf("Nilai yang ditunjuk oleh px = %d\n", \*px);  printf("Nilai y = %d\n", y);  } |

*Output* percobaan 5 :



1. Percobaan 6 : Program Mengakses & Mengubah Isi Suatu Variabel Pointer

|  |
| --- |
| #include <stdio.h>  void main() {  float d = 54.5f, \*pd;  printf("Isi d mula-mula = %g\n", d);  pd = &d;  \*pd += 10;  printf("Isi d sekarang = %g\n", d);  } |

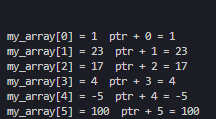
*Output* percobaan 6 :



1. Percobaan 7 : Mengakses elemen array dengan pointer.

|  |
| --- |
| #include <stdio.h>  void main() {  int my\_array[] = {1, 23, 17, 4, -5, 100};  int \*ptr;  int i;  ptr = &my\_array[0]; // Pointer ptr menunjuk ke elemen pertama dari my\_array  printf("\n\n");  for (i = 0; i < 6; i++) {  printf("my\_array[%d] = %d ", i, my\_array[i]);  printf("ptr + %d = %d\n", i, \*(ptr + i));  }  } |

*Output* percobaan 7 :



1. Percobaan 8 : Mengkopi string menggunakan pointer.

|  |
| --- |
| #include <stdio.h>  char strA[80] = "ABCDE";  char strB[80];  void main() {  char \*pA;  char \*pB;  puts(strA);  pA = strA;  puts(pA);  pB = strB;  putchar('\n');  while (\*pA != '\0') {  \*pB++ = \*pA++;  }  \*pB = '\0';  puts(strB);  } |

*Output* percobaan 8 :



1. Percobaan 9 : Penggunaan Pointer untuk Bilangan Fibonaci

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define MAX 20  void main() {  int \*fibo;  int i;  fibo = malloc(MAX \* sizeof(int));  \*(fibo + 0) = 0;  \*(fibo + 1) = 1;  for (i = 2; i < MAX; i++) {  \*(fibo + i) = \*(fibo + i - 2) + \*(fibo + i - 1);  }  printf("%d Bilangan Fibonacci Pertama adalah:\n", MAX);  for (i = 1; i < MAX; i++) {  printf("%d ", \*(fibo + i));  }  free(fibo);  } |

*Output* percobaan 9 :

****

1. Percobaan 10 : Penggunaan Struktur pada Konversi Koordinat Polar ke Koordinat Cartesian

|  |
| --- |
| #include <stdio.h>  #include <math.h>    struct polar {  double r;  double alpha;  };    struct kartesian {  double x;  double y;  };    void main() {  struct polar p1;  struct kartesian k1;  printf("Masukkan nilai r untuk koordinat polar : ");  scanf("%lf",&p1.r);    printf("Masukkan nilai alpha untuk koordinat polar : ");  scanf("%lf",&p1.alpha);    k1.x = p1.r \* cos(p1.alpha);  k1.y = p1.r \* sin(p1.alpha);    printf("Nilai koordinat kartesian untuk koordinat polar r= %2.2lf alpha= %2.2lf adalah:\n",p1.r,p1.alpha);  printf("x = %2.2lf y = %2.2lf",k1.x,k1.y);  } |

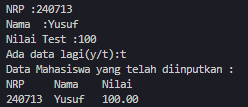
*Output* percobaan 10 :



1. Percobaan 11 : Program Struktur dalam Array

|  |
| --- |
| #include <stdio.h>  #include <string.h>    struct dtnilai {  char nrp[10];  char nama[20];  double nilai;  };    struct dtnilai data[10];  int j=0;    void tambah\_data() {  char jawab[2];  while(1) {  fflush(stdin);  printf("NRP :");scanf("%s",&data[j].nrp);  printf("Nama :");scanf("%s",&data[j].nama);  printf("Nilai Test :");scanf("%lf",&data[j].nilai);  printf("Ada data lagi(y/t):"); scanf("%s",&jawab);    if((strcmp(jawab,"Y")==0)||(strcmp(jawab,"y")==0)) {  j++;continue;  } else if ((strcmp(jawab,"T")==0)||(strcmp(jawab,"t")==0))  break;  }  }    void tampil() {  int i;  printf("Data Mahasiswa yang telah diinputkan :\n");  printf("NRP\tNama\tNilai\n");  for (i=0;i<=j;i++) {  printf("%s\t%s\t%6.2f\n",data[i].nrp,data[i].nama,  data[i].nilai);  }  }  void main() {  tambah\_data();  tampil();  } |

*Output* percobaan 11 :



1. Percobaan 12 : Mengakses Struktur dengan Pointer

|  |
| --- |
| #include <stdio.h>  #include <string.h>      struct dtnilai {  char nrp[10];  char nama[20];  double nilai;  };  void tampil(struct dtnilai \*p) ;    struct dtnilai my\_struct;    void main() {  struct dtnilai \*st\_ptr;  st\_ptr = &my\_struct;  strcpy(my\_struct.nrp,"01");  printf("\n%s ",my\_struct.nrp);    strcpy(my\_struct.nama,"Arini");  printf("\n%s ",my\_struct.nama);  my\_struct.nilai = 63.6;  tampil(st\_ptr);    }    void tampil(struct dtnilai \*p) {  printf("\n%s ", p->nrp);  printf("%s ", p->nama);  printf("%.1lf\n", p->nilai);  } |

*Output* percobaan 12 :



1. **PRAKTIKUM**
2. TUJUAN PRAKTIKUM

|  |
| --- |
| 1. Memahami konsep dan implementasi array dalam bahasa C. 2. Menguasai penggunaan pointer untuk mengakses dan mengelola memori. 3. Memahami struktur (struct) untuk mengelompokkan data terkait. 4. Mengerti perbedaan dan hubungan antara array, pointer, dan struktur. 5. Mengintegrasikan ketiga konsep tersebut dalam program yang lebih kompleks. 6. Melatih kemampuan menyelesaikan masalah menggunakan struktur data dasar. 7. Mengembangkan logika pemrograman untuk mengoptimalkan penggunaan memori dan performa program. |

1. DASAR TEORI

|  |
| --- |
| 1. Array   Array adalah struktur data yang menyimpan sekumpulan elemen dengan tipe data yang sama dalam satu variabel. Setiap elemen array diakses menggunakan indeks yang dimulai dari nol. Array memungkinkan penyimpanan data secara berurutan di memori, yang mempermudah pengelolaan dan manipulasi data yang bersifat homogen.  **Contoh deklarasi array dalam C:**  int angka[5] = {1, 2, 3, 4, 5};  Dalam contoh ini, array angka menyimpan lima bilangan bulat. Elemen pertama dapat diakses dengan angka[0].   1. Pointer   Pointer adalah variabel yang menyimpan alamat memori dari variabel lain. Pointer memungkinkan manipulasi langsung terhadap lokasi memori, yang memberikan fleksibilitas lebih dalam pemrograman.  **Contoh deklarasi pointer dalam C:**  int a = 10;  int \*ptr = &a;  Variabel ptr menyimpan alamat variabel a. Nilai a dapat diakses melalui pointer dengan \*ptr (dereferensi).   1. Struktur (Struct)   Struktur adalah kumpulan elemen data (field) yang dapat memiliki tipe data berbeda, dikelompokkan dalam satu kesatuan. Struktur berguna untuk merepresentasikan objek yang memiliki banyak atribut.  **Contoh deklarasi struktur dalam C:**  struct Mahasiswa {  char nama[50];  int nim;  float ipk;  };  Struktur Mahasiswa memiliki tiga field: nama, nim, dan ipk. Struktur mempermudah pengelolaan data yang kompleks dalam program.  Ketiga konsep ini — array, pointer, dan struktur — merupakan dasar penting dalam pemrograman yang mendukung pengelolaan data secara efisien dan terstruktur. |

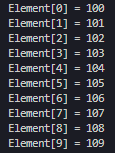
1. PEMBAHASAN

|  |
| --- |
|  |

1. HASIL
2. Latihan 1:

Dari percobaan ….

*Output* Latihan 1:



|  |
| --- |
| Penjelasan :  …….. |

1. Latihan 2:

Dari percobaan ….

*Output* Latihan 2:

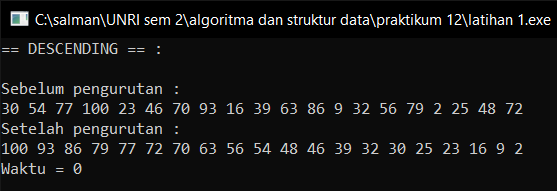


|  |
| --- |
| Penjelasan :  …….. |

1. Latihan 3:

Dari percobaan ….

*Output* Latihan 3:

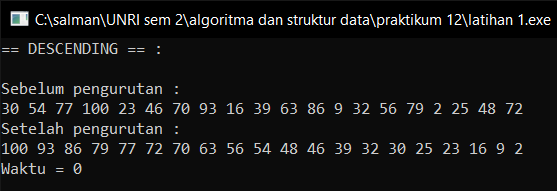


|  |
| --- |
| Penjelasan :  …….. |

1. Latihan 4:

Dari percobaan ….

*Output* Latihan 4:

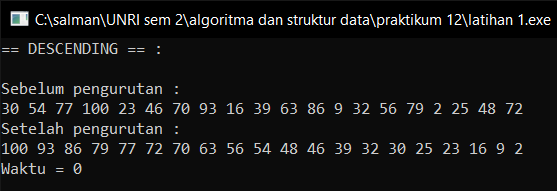


|  |
| --- |
| Penjelasan :  …….. |

1. Latihan 5:

Dari percobaan ….

*Output* Latihan 5:



|  |
| --- |
| Penjelasan :  …….. |

1. LISTING PROGRAM
2. Latihan 1 :

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  char arr[] = {'p', 'e', 'n', 's', 'i', 't'};  int panjang = sizeof(arr) / sizeof(arr[0]);  // Membalik array  for (int i = 0; i < panjang / 2; i++) {  char temp = arr[i];  arr[i] = arr[panjang - i - 1];  arr[panjang - i - 1] = temp;  }  // Cetak hasil  for (int i = 0; i < panjang; i++) {  cout << arr[i];  }  cout << endl;  return 0;  } |

1. Latihan 2 :

|  |
| --- |
|  |

1. Latihan 3 :

|  |
| --- |
|  |

1. Latihan 4 :

|  |
| --- |
|  |

1. Latihan 5 :

|  |
| --- |
|  |