

Nama : Muhamad Putra Nugraha

NIM : 224443014

Kelas : 1 AEC 1

LAPORAN HARIAN STRUKTUR DATA

MATERI :

- **VARIABEL**

Memori yang digunakan untuk menyimpan data yang dapat diubah

- **TIPE DATA**

Integer, String, Boolean, float, char, array, pointer

- **SISTEM BILANGAN**

BINER = Berbasis 2

OKTAN = Berbasis 8

DESIMAL = Berbasis 10

HEKSADESIMAL = Berbasis 16

- STRUKTUR KONTROL PERCABANGAN (IF ELSE)(SWITCH CASE)

- STRUKTUR KONTROL PERULANGAN (FOR)(WHILE)(DO WHILE)

- FUNGSI DAN PROSEDUR

□ Manfaat pointer bisa merubah data dari variable yang ditunjuknya

□ Array pemanggilan nama array tanpa indeks akan menampilkan Alamat

□ Suatu pointer yang menunjuk ke array maka pointer tersebut menjadi array

--- Hasil Percobaan Dari Modul ---

Menu Pilihan Program

```
=====
| 1. Contoh Program Pointer 1 |
=====
| 2. Contoh Program Pointer 2 |
=====
| 3. Contoh Program Pointer 3 |
=====
| 4. Contoh Program Pointer 4 |
=====
| 5. Contoh Program Pointer 5 |
=====
| 6. Contoh Program Pointer 6 |
=====
| 7. Contoh Program Pointer 7 |
=====
| 8. Contoh Program Pointer 8 |
=====
| 9. Contoh Latihan Soal No 1 |
=====
| 10. Contoh Latihan Soal No 2 |
=====
| 11. Contoh Selection Sort   |
=====
| 12. Contoh Insertion Sort   |
=====
| 13. Contoh Program Matrix   |
=====
Silahkan Pilih Program Yang Akan Dijalankan : |
```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> int main() { int program; char ulang; do { cout << "===== << endl; cout << " 1. Contoh Program Pointer 1 " << endl; cout << "-----" << endl; cout << " 2. Contoh Program Pointer 2 " << endl; cout << "-----" << endl; cout << " 3. Contoh Program Pointer 3 " << endl; cout << "-----" << endl; cout << " 4. Contoh Program Pointer 4 " << endl; cout << "-----" << endl; cout << " 5. Contoh Program Pointer 5 " << endl; cout << "-----" << endl; cout << " 6. Contoh Program Pointer 6 " << endl; cout << "-----" << endl; cout << " 7. Contoh Program Pointer 7 " << endl; cout << "-----" << endl; cout << " 8. Contoh Program Pointer 8 " << endl; cout << "-----" << endl; cout << " 9. Contoh Latihan Soal No 1 " << endl; cout << "-----" << endl; cout << " 10. Contoh Latihan Soal No 2 " << endl; cout << "-----" << endl; cout << " 11. Contoh Selection Sort " << endl; cout << "-----" << endl; cout << " 12. Contoh Insertion Sort " << endl; cout << "-----" << endl; } } </pre>	<p>Disini saya menggunakan Switch Case untuk berpindah antar 1 program ke program lainnya, agar tidak perlu membuat banyak file. Cara membuatnya adalah di Function Main, jadi function main ini kita akan meminta inputan dari user, dan inputan dari user itu akan diproses oleh Switch Case dan akan mencocokkannya ke function void yang sesuai.</p>

```
cout << "| 13. Contoh Program Matrix   |" <<
endl;
cout <<
"-----"
<< endl;
cout << "Silahkan Pilih Program Yang Akan
Dijalankan : ";
cin >> program;
```

```
switch (program) {
case 1:
    program1();
    break;
case 2:
    program2();
    break;
case 3:
    program3();
    break;
case 4:
    program4();
    break;
case 5:
    program5();
    break;
case 6:
    program6();
    break;
case 7:
    program7();
    break;
case 8:
    program8();
    break;
case 9:
    program9();
    break;
case 10:
    program10();
    break;
case 11:
    program11();
    break;
case 12:
    program12();
    break;
case 13:
    program13();
    break;
default:
```

<pre> cout << "Pilihan Tidak Ada" << endl; } cout << "\nApakah Anda ingin kembali ke menu? (y/n): "; cin >> ulang; } while (ulang == 'y' ulang == 'Y'); cout << "Program selesai." << endl; return 0; } </pre>	
--	--

Operator Pointer 1

Alamat Variable Nilai a Adalah : 5
 Nilai Variable a Adalah : 0x61fdd4

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program1() { int a = 5; //Memberikan Variable a Nilai 5 int *b; //Mendeklarasikan Variable b Sebagai Pointer b = &a; //Menyalin Alamat Variable a Kedalam Variable Pointer cout << "\nAlamat Variable Nilai a Adalah : "<< a << endl; //Memanggil Nilai Variable a cout << "Nilai Variable a Adalah : "<< &a << endl; //Memanggil Alamat Nilai Variable a } </pre>	<p>Program pertama ini mendemonstrasikan konsep dasar penggunaan pointer dalam C++. Pointer adalah variabel khusus yang digunakan untuk menyimpan alamat memori variabel lain. Dalam program tersebut:</p> <p>Sebuah variabel bernama a dideklarasikan dan diberi nilai 5.</p> <p>Sebuah pointer bernama b dideklarasikan sebagai penyimpan alamat memori.</p> <p>Alamat memori dari variabel a disalin ke pointer b.</p> <p>Program menampilkan alamat memori variabel a melalui pointer b.</p> <p>Selain itu, program juga menampilkan nilai dari variabel a.</p> <p>Output dari program menunjukkan bagaimana alamat memori dan nilai suatu variabel dapat diakses menggunakan pointer.</p>

Operator Pointer 2

```
Alamat Variable Nilai a Adalah : 5
Nilai Variable a Adalah : 0x61fdd4
Nilai Variable b Adalah : 0x61fdd4
Nilai Yang Tertampung Dalam Variable b Adalah : 5
```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre>void program2() { int a = 5; //Memberikan Variable a Nilai 5 int *b; //Mendeklarasikan Variable b Sebagai Pointer b = &a; //Menyalin Alamat Variable a Kedalam Variable Pointer cout << "\nAlamat Variable Nilai a Adalah : "<< a << endl; //Memanggil Nilai Variable a cout << "Nilai Variable a Adalah : "<< &a << endl; //Memanggil Alamat Nilai Variable a cout << "Nilai Variable b Adalah : "<< b << endl; //Memanggil Nilai Variable b cout << "Nilai Yang Tertampung Dalam Variable b Adalah : "<< *b << endl; //Memanggil Pointer b Dan Nilai Yang Tertampung Dalam Pointer b }</pre>	<p>Program ke 2 ini menunjukkan penggunaan pointer dalam C++. Variabel a diberi nilai 5, dan pointer b menyimpan alamat memori a menggunakan operator &. Program menampilkan alamat memori a, nilai a, nilai pointer b (alamat a), dan nilai yang ditunjuk oleh pointer b menggunakan operator dereferensi *.</p>

Mendeklarasikan Variable Pointer

OUTPUT

```
Nilai X : 89
Nilai Y : 89
Nilai Alamat X : 0x61fdcc
Nilai Alamat PX : 0x61fdcc
Nilai PX : 89

Nilai X : 108
Nilai Y : 89
Nilai Alamat X : 0x61fdcc
Nilai Alamat PX : 0x61fdcc
Nilai PX : 108
Nilai X : 123
Nilai Y : 89
Nilai Alamat X : 0x61fdcc
Nilai Alamat PX : 0x61fdcc
Nilai PX : 123
```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre>void program3() { int x, y; int *px; x = 89; y = x; px = &x; cout << "Nilai X : "<< x << endl; cout << "Nilai Y : "<< y << endl; cout << "Nilai Alamat X : "<< &x << endl; cout << "Nilai Alamat PX : "<< px << endl; cout << "Nilai PX : "<< *px << endl; x = 108; cout << "\nNilai X : "<< x << endl; cout << "Nilai Y : "<< y << endl; cout << "Nilai Alamat X : "<< &x << endl;</pre>	<p>Kode ke 3 ini mendemonstrasikan penggunaan pointer untuk menyimpan alamat variabel, membaca nilai dari alamat tersebut, dan memodifikasi nilai variabel melalui pointer. Berikut deskripsi singkatnya:</p> <ol style="list-style-type: none">1. Inisialisasi: x diberi nilai 89, y menyalin nilai x, dan pointer px menyimpan alamat x.2. Tampilan awal: Menampilkan nilai x, y, alamat x, dan nilai yang ditunjuk oleh px.3. Ubah langsung: Nilai x diubah menjadi 108, y tetap 89, px tetap menunjuk ke x.4. Ubah via pointer: Nilai x diubah menjadi 123 menggunakan pointer px.5. Kesimpulan: Pointer memungkinkan akses dan modifikasi nilai variabel melalui alamat memori.

<pre> cout << "Nilai Alamat PX : " << px << endl; cout << "Nilai PX : " << *px << endl; *px = 123; cout << "Nilai X : " << x << endl; cout << "Nilai Y : " << y << endl; cout << "Nilai Alamat X : " << &x << endl; cout << "Nilai Alamat PX : " << px << endl; cout << "Nilai PX : " << *px << endl; } </pre>	
--	--

Pointer Pada Pointer

OUTPUT

```

Nilai x = 175
Nilai px = 175
Nilai ppx = 175

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program4() { int x; int *px; //Pointer Ke Variable x int **ppx; //Pointer Ke Pointer px x = 175; px = &x; ppx = &px; cout << "Nilai x = " << x << endl; cout << "Nilai px = " << *px << endl; cout << "Nilai ppx = " << **ppx << endl; } </pre>	<p>Kode Ke 4 ini menggunakan Pointer ke pointer (ppx) memperluas fleksibilitas manipulasi nilai atau alamat memori dengan lebih dari satu tingkat dereferensi.</p> <ol style="list-style-type: none"> Deklarasi: <ul style="list-style-type: none"> x: Variabel integer. px: Pointer yang menunjuk alamat x. ppx: Pointer yang menunjuk ke pointer px. Inisialisasi: <ul style="list-style-type: none"> x diberi nilai 175. px menyimpan alamat dari x. ppx menyimpan alamat dari pointer px. Output: <ul style="list-style-type: none"> x: Menampilkan nilai langsung dari x (175).

	<ul style="list-style-type: none">○ *px: Menampilkan nilai di alamat yang ditunjuk px (nilai x, yaitu 175).○ **ppx: Menampilkan nilai di alamat yang ditunjuk oleh pointer ppx, yaitu nilai x (175).
--	---

Pointer Pada Array

OUTPUT

```

Masukan Nilai 1 : 5
Masukan Nilai 2 : 7
Masukan Nilai 3 : 2
Masukan Nilai 4 : 4
Masukan Nilai 5 : 3

Nilai a[0] = 5
Nilai a[1] = 7
Nilai a[2] = 2
Nilai a[3] = 4
Nilai a[4] = 3

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program5() { #define MAX 5 int a[MAX]; int *pa; pa = a; for (int i = 0; i < MAX; i++) { cout << "Masukan Nilai " << i+1 << " : "; cin >> a[i]; } cout << endl; for (int i = 0; i < MAX; i++) { cout << "Nilai a["<<i<<"] = " << *pa <<endl; pa++; } } </pre>	<p>Kode ke 5 ini mendemonstrasikan penggunaan pointer untuk mengakses elemen-elemen array secara dinamis. Berikut penjelasannya:</p> <p>Program mendefinisikan ukuran array menggunakan <code>#define MAX 5</code> dan mendeklarasikan sebuah array <code>a</code> dengan 5 elemen. Sebuah pointer <code>pa</code> digunakan untuk menunjuk ke elemen-elemen array tersebut.</p> <p>Pertama, pointer <code>pa</code> diinisialisasi untuk menunjuk ke elemen pertama array <code>a</code> dengan <code>pa = a</code>. Program kemudian meminta input dari pengguna untuk mengisi nilai setiap elemen array melalui loop <code>for</code>.</p> <p>Setelah input selesai, elemen-elemen array ditampilkan menggunakan pointer <code>pa</code>. Pointer <code>pa</code> tidak menggunakan indeks array tetapi melakukan increment (<code>pa++</code>) untuk berpindah dari satu elemen ke elemen berikutnya.</p> <p>Melalui pendekatan ini, pointer dimanfaatkan untuk mengakses elemen array tanpa menggunakan indeks secara eksplisit.</p>

Pointer Pada Array

OUTPUT

```

Diakses dengan pointer
Tanggal = 13
Bulan = 9
Tahun = 1982

Diakses dengan array biasa
Tanggal = 13
Bulan = 9
Tahun = 1982

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program6() { static int tgl_lahir[] = { 13,9,1982 }; int *ptgl; ptgl = tgl_lahir; /* ptgl berisi alamatarray */ cout<<"Diakses dengan pointer\n"; cout<<"Tanggal = "<< *ptgl<<"\n"; cout<<"Bulan = "<< *(ptgl + 1) <<"\n"; cout<<"Tahun = "<<*(ptgl + 2) <<"\n"; cout<<"\nDiakses dengan array biasa\n"; cout<<"Tanggal = "<< tgl_lahir[0] <<"\n"; cout<<"Bulan = "<< tgl_lahir[1] <<"\n"; cout<<"Tahun = "<< tgl_lahir[2] <<"\n"; getch(); } </pre>	<p>Kode ke 6 ini menunjukkan cara mengakses elemen array menggunakan pointer dan indeks array. Array <code>tgl_lahir</code> menyimpan tanggal, bulan, dan tahun. Pointer <code>ptgl</code> diinisialisasi untuk menunjuk ke elemen pertama array.</p> <p>Elemen array diakses dengan pointer menggunakan dereferensi (<code>*ptgl</code>, <code>*(ptgl + 1)</code>, <code>*(ptgl + 2)</code>) dan dibandingkan dengan akses langsung menggunakan indeks array (<code>tgl_lahir[0]</code>, <code>tgl_lahir[1]</code>, <code>tgl_lahir[2]</code>).</p>

Pointer Pada String 1

OUTPUT

```

Nama = Albert Einstein
pNama = Albert Einstein
Setelah pNama += 7
Nama = Albert Einstein
pNama = Einstein

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program7() { #define MAX 5 char nama[] = "Albert Einstein"; char *pNama = nama; cout << "Nama = " << nama << endl; cout << "pNama = " << pNama << endl; pNama += 7; cout << "Setelah pNama += 7" << endl; cout << "Nama = " << nama << endl; cout << "pNama = " << pNama << endl; } </pre>	<p>Program ke 7 ini menunjukkan penggunaan pointer untuk memanipulasi string. Array nama menyimpan string "Albert Einstein", dan pointer pNama menunjuk ke elemen pertama string tersebut. Setelah pointer pNama digeser 7 karakter ke depan (pNama += 7), pointer menunjuk ke substring "Einstein". String asli tetap tidak berubah.</p>

Pointer Pada String 2

OUTPUT

```

0 0 0x61fda0
1 1 0x61fda4
2 2 0x61fda8
3 3 0x61fdac
4 4 0x61fdb0
5 5 0x61fdb4
6 6 0x61fdb8
7 7 0x61fdbc
8 8 0x61fdc0
9 9 0x61fdc4

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program8() { #define MAX 5 int x[10] = {0,1,2,3,4,5,6,7,8,9}; int *px; int i; for (i = 0; i < 10; i++) { px = &x[i]; //membaca alamat dari x cout << x[i] << " " << *px << " " << px << endl; } } </pre>	<p>Program ke 8 ini mendemonstrasikan penggunaan pointer untuk mengakses elemen array. Array <code>x</code> menyimpan 10 elemen (0–9), dan pointer <code>px</code> menunjuk ke setiap elemen array dalam loop. Program mencetak nilai elemen (<code>x[i]</code>), nilai yang ditunjuk pointer (<code>*px</code>), dan alamat memori elemen (<code>px</code>).</p>

Pointer Pada String 2

OUTPUT

```
Masukkan teks: Halo Aku Putra, Umurku 20 Tahun.

Hasil Analisis:
Jumlah Vokal   : 11
Jumlah Konsonan: 12
Jumlah Angka   : 2
Jumlah Spasi   : 5
Jumlah Lainnya : 2
```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre>void program9() { const char angka[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '\0'}; const char h_vokal[] = {'A', 'I', 'U', 'E', 'O', '\0'}; string teks; int vokal = 0; int konsonan = 0; int bilangan = 0; int spasi = 0; int lainnya = 0; cout << "Masukkan teks: "; cin.ignore(); getline(cin, teks); for (char c : teks) { if (isalpha(c)) { char huruf = toupper(c); if (strchr(h_vokal, huruf)) { vokal++; } else { konsonan++; } } else if (isdigit(c)) { bilangan++; } else if (isspace(c)) { spasi++; } else { lainnya++; } } cout << "\nHasil Analisis:\n"; cout << "Jumlah Vokal : " << vokal << "\n"; cout << "Jumlah Konsonan: " << konsonan << "\n";</pre>	<p>Program ke 9 ini menganalisis teks yang dimasukkan pengguna untuk menghitung jumlah huruf vokal, konsonan, angka, spasi, dan karakter lainnya.</p> <p>Deklarasi:</p> <p>angka[] menyimpan karakter angka ('0'-'9'). h_vokal[] menyimpan huruf vokal (A, I, U, E, O). Variabel vokal, konsonan, bilangan, spasi, dan lainnya digunakan untuk menghitung jumlah masing-masing jenis karakter dalam teks.</p> <p>Input Teks:</p> <p>Pengguna diminta memasukkan teks, yang disimpan dalam variabel teks.</p> <p>Analisis Teks:</p> <p>Menggunakan loop for untuk memeriksa setiap karakter dalam teks: Jika karakter adalah huruf (isalpha()), huruf dikonversi ke huruf besar (toupper()), lalu diperiksa apakah vokal (strchr(h_vokal, huruf)). Jika karakter adalah angka (isdigit()), dihitung sebagai angka. Jika karakter adalah spasi (isspace()), dihitung sebagai spasi. Selain itu, dihitung sebagai karakter lain.</p> <p>Output Hasil:</p> <p>Program mencetak jumlah vokal, konsonan, angka, spasi, dan karakter lainnya.</p>

```
        cout << "Jumlah Angka   : " <<  
bilangan << "\n";  
        cout << "Jumlah Spasi   : " << spasi  
<< "\n";  
        cout << "Jumlah Lainnya : " <<  
lainnya << "\n";  
  
    }  
}
```

---LATIHAN DI PAPAN TULIS ---

OUTPUT

```

0x58c7e5ed1010
0x58c7e5ed1010
0x58c7e5ed1158
5000
5000
2500
4000

...Program finished with exit code 0
Press ENTER to exit console.

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> #include <iostream> using namespace std; int harga = 5000; int *p_harga; int main() { p_harga = &harga; cout<<&harga<<endl; //0x404004 cout<<p_harga<<endl; //0x404004 cout<<&p_harga<<endl; //0x407020 cout<<*p_harga<<endl; //5000 cout<<harga<<endl; //5000 *p_harga = 2500; cout<<harga<<endl; //2500 cout<<*p_harga + 1500<<endl; //4000 return 0; } </pre>	<ul style="list-style-type: none"> • <code>#include <iostream></code>: Mengimpor header untuk operasi input dan output. • <code>using namespace std</code>: Menggunakan ruang lingkup <code>std</code> agar lebih praktis dalam penulisan. • <code>int harga = 5000</code>: Mendeklarasikan variabel <code>harga</code> bertipe integer dengan nilai awal 5000. • <code>int *p_harga</code>: Membuat pointer bertipe integer bernama <code>p_harga</code>. • <code>int main() {}</code>: Memulai blok utama dari program. • <code>p_harga = &harga</code>: Menyimpan alamat memori variabel <code>harga</code> ke dalam pointer <code>p_harga</code>. • <code>cout<<&harga<<endl</code>: Menampilkan alamat memori tempat variabel <code>harga</code> berada. • <code>cout<<p_harga<<endl</code>: Menampilkan isi dari pointer <code>p_harga</code>, yaitu alamat <code>harga</code>. • <code>cout<<&p_harga<<endl</code>: Menampilkan lokasi memori tempat pointer <code>p_harga</code> disimpan. • <code>cout<<*p_harga<<endl</code>: Menampilkan nilai yang tersimpan di alamat yang ditunjuk oleh <code>p_harga</code>. • <code>cout<<harga<<endl</code>: Menampilkan nilai variabel <code>harga</code>. • <code>*p_harga = 2500</code>: Mengubah nilai di memori yang ditunjuk oleh <code>p_harga</code> menjadi 2500.

	<ul style="list-style-type: none"> • <code>cout<<harga<<endl;</code>: Menampilkan nilai harga setelah perubahan melalui pointer. • <code>cout<<*p_harga + 1500<<endl;</code>: Menampilkan nilai yang ditunjuk oleh <code>p_harga</code> ditambah 1500.
--	--

OUTPUT

```
Indonesia Jepang
Indonesia

...Program finished with exit code 0
Press ENTER to exit console.
```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre>#include <iostream> using namespace std; string negara[5] = {"Indonesia", "Malaysia", "Singapore", "Filliphina", "Laos"}; string *p_negara; int main() { negara[1] = "Jepang"; cout<<negara[0] + " " + negara[1]<<endl; p_negara = negara; cout<<*p_negara<<endl; //Indonesia --> Alamat dari indeks awal return 0; }</pre>	<ol style="list-style-type: none"> 1. <code>string negara[5]</code>: Deklarasi array negara dengan 5 elemen bertipe string. 2. <code>negara[1] = "Jepang";</code>: Mengubah elemen kedua array (Malaysia) menjadi Jepang. 3. <code>p_negara = negara;</code>: Pointer <code>p_negara</code> menunjuk ke elemen pertama array negara. 4. <code>cout<<*p_negara<<endl;</code>: Menampilkan nilai elemen pertama array (Indonesia) melalui pointer. 5. Program menampilkan kombinasi elemen array yang dimanipulasi dan elemen pertama yang diakses menggunakan pointer.

OUTPUT

```
0x5f9e41edd160
Indonesia

...Program finished with exit code 0
Press ENTER to exit console.█
```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre>#include <iostream> using namespace std; int main() { string negara[4] = {"Indonesia", "Malaysia", "Singapura", "Filipina"}; string *p_negara = negara; cout << p_negara << endl; cout << *p_negara << endl; return 0; }</pre>	<ol style="list-style-type: none">1. string negara[4]: Deklarasi array negara dengan 4 elemen bertipe string.2. string *p_negara = negara;: Pointer p_negara menunjuk ke elemen pertama array negara.3. cout << p_negara: Menampilkan alamat elemen pertama array.4. cout << *p_negara: Menampilkan nilai elemen pertama array (Indonesia).

OUTPUT

```
0x7ffdd0511160
3000
3002
1500
9500

...Program finished with exit code 0
Press ENTER to exit console.█
```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre>#include <iostream> using namespace std; int main() { int harga[5] = {3000, 4500, 1500, 7450, 9500}; int *p_harga = harga; cout << p_harga << endl; cout << *p_harga << endl; cout << *p_harga + 2 << endl; cout << *(p_harga + 2) << endl; cout << *(p_harga + 4) << endl; return 0; }</pre>	<ol style="list-style-type: none">1. int harga[5]: Deklarasi array harga dengan 5 elemen bertipe integer.2. int *p_harga = harga: Pointer p_harga menunjuk ke elemen pertama array harga.3. cout << p_harga: Menampilkan alamat elemen pertama array harga.4. cout << *p_harga: Menampilkan nilai elemen pertama array (3000).5. cout << *p_harga + 2: Menampilkan hasil penjumlahan elemen pertama array dengan 2 (3000 + 2 = 3002).6. cout << *(p_harga + 2): Menampilkan nilai elemen ketiga array (1500).7. cout << *(p_harga + 4): Menampilkan nilai elemen kelima array (9500).

---SOAL---

1. Buatlah fungsi dan prosedur yang telah dideklarasikan diatas untuk menentukan jumlah vokal, konsonan bilangan, karakter spasi, dan karakter lain yang di input pada sebuah baris teks.
2. Fungsi urutkan di bawah ini mengurutkan serangkaian bilangan dari kecil ke besar dengan menggunakan algoritma bubble sort. Tulis kembali fungsi di atas dengan menggunakan pointer untuk mengambil array. Buatlah fungsi main untuk membuat program pengurutan, di mana jumlah bilangan yang di input ditentukan oleh user dan memori yang dialokasikan sesuai dengan jumlah bilangan yang dipesan.

---JAWAB---

Soal No 1

OUTPUT

```
Masukkan teks: Halo Aku Putra, Umurku 20 Tahun.

Hasil Analisis:
Jumlah Vokal   : 11
Jumlah Konsonan: 12
Jumlah Angka   : 2
Jumlah Spasi   : 5
Jumlah Lainnya : 2
```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre>void program9() { const char angka[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '\0'}; const char h_vokal[] = {'A', 'I', 'U', 'E', 'O', '\0'}; string teks; int vokal = 0; int konsonan = 0; int bilangan = 0; int spasi = 0; int lainnya = 0; cout << "Masukkan teks: "; cin.ignore(); getline(cin, teks); for (char c : teks) { if (isalpha(c)) { char huruf = toupper(c); if (strchr(h_vokal, huruf)) { vokal++; } else { konsonan++; } } else if (isdigit(c)) { bilangan++; } else if (isspace(c)) { spasi++; } else { lainnya++; } } }</pre>	<p>Soal program ke 1 ini menganalisis teks input dari pengguna untuk menghitung jumlah vokal, konsonan, angka, spasi, dan karakter lainnya.</p> <p>Deklarasi Konstanta:</p> <p>angka[]: Berisi karakter angka dari '0' hingga '9'. H_vokal[]: Berisi huruf vokal besar ('A', 'I', 'U', 'E', 'O'). Input Teks:</p> <p>Mengambil input teks dari pengguna menggunakan getline() untuk mendukung input yang berisi spasi. Perulangan untuk Analisis:</p> <p>Huruf: Menggunakan isalpha() untuk mengecek apakah karakter adalah huruf. Mengubah huruf menjadi huruf kapital dengan toupper() untuk dibandingkan dengan h_vokal. Jika huruf termasuk vokal, tambahkan ke vokal. Selain itu, tambahkan ke konsonan. Angka: Menggunakan isdigit() untuk mengecek apakah karakter adalah angka, lalu tambahkan ke bilangan. Spasi: Menggunakan isspace() untuk mendeteksi spasi, tambahkan ke spasi. Karakter Lain: Selain huruf, angka, atau spasi, dihitung sebagai lainnya.</p> <p>Output:</p>

<pre> } } cout << "\nHasil Analisis:\n"; cout << "Jumlah Vokal : " << vokal << "\n"; cout << "Jumlah Konsonan: " << konsonan << "\n"; cout << "Jumlah Angka : " << bilangan << "\n"; cout << "Jumlah Spasi : " << spasi << "\n"; cout << "Jumlah Lainnya : " << lainnya << "\n"; }</pre>	Menampilkan jumlah masing-masing kategori (vokal, konsonan, angka, spasi, dan lainnya).
--	---

Soal No 2

OUTPUT

```

Masukkan jumlah bilangan yang ingin diurutkan: 6
Masukkan 6 bilangan:
Bilangan ke-1: 4
Bilangan ke-2: 7
Bilangan ke-3: 9
Bilangan ke-4: 2
Bilangan ke-5: 3
Bilangan ke-6: 1

Bilangan setelah diurutkan:
1 2 3 4 7 9

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program10() { int n; cout << "Masukkan jumlah bilangan yang ingin diurutkan: "; cin >> n; int *arr = new int[n]; if (arr == nullptr) { cout << "Gagal mengalokasikan memori." << endl; return; } cout << "Masukkan " << n << " bilangan:" << endl; for (int i = 0; i < n; i++) { cout << "Bilangan ke-" << i + 1 << ": "; cin >> *(arr + i); } for (int i = 0; i < n - 1; i++) { for (int j = 0; j < n - i - 1; j++) { if (*(arr + j) > *(arr + j + 1)) { int temp = *(arr + j); *(arr + j) = *(arr + j + 1); *(arr + j + 1) = temp; } } } cout << "\nBilangan setelah diurutkan:" << endl; </pre>	<p>Soal program ke 2 ini mengurutkan sekumpulan bilangan yang dimasukkan oleh pengguna menggunakan algoritma Bubble Sort. Program memanfaatkan pointer untuk mengelola array dinamis yang dialokasikan secara manual di memori.</p> <p>Langkah-Langkah Program: Input Jumlah Bilangan:</p> <p>Program meminta pengguna memasukkan jumlah bilangan yang ingin diurutkan (n). Alokasi Memori:</p> <p>Program menggunakan new untuk mengalokasikan array secara dinamis dengan ukuran n. Jika alokasi memori gagal, program akan memberikan pesan error dan keluar. Input Nilai:</p> <p>Pengguna diminta memasukkan n bilangan, satu per satu. Nilai-nilai ini disimpan dalam array dinamis melalui pointer. Proses Pengurutan:</p> <p>Program menggunakan algoritma Bubble Sort untuk mengurutkan array. Dua loop digunakan: Outer loop untuk setiap elemen array. Inner loop untuk membandingkan dan menukar elemen jika urutannya salah. Output Hasil:</p> <p>Setelah selesai diurutkan, program mencetak bilangan dalam urutan yang telah diatur.</p>

```
for (int i = 0; i < n; i++) {  
    cout << *(arr + i) << " ";  
}  
cout << endl;  
  
delete[] arr;  
}
```

Dealokasi Memori:

Array dinamis dihapus menggunakan delete[] untuk mencegah kebocoran memori.

---TUGAS TAMBAHAN---

1. Buat Selection Sort.
2. Buat Insertion Sort.
3. Buat Matrix Dinamis Menggunakan Pointer

---JAWAB---

Soal No 1

OUTPUT

```

Masukkan jumlah elemen array: 4
Masukkan 4 bilangan:
Bilangan ke-1: 6
Bilangan ke-2: 2
Bilangan ke-3: 1
Bilangan ke-4: 4

Array setelah diurutkan dengan Selection Sort:
1 2 4 6

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program11() { int n; cout << "Masukkan jumlah elemen array: "; cin >> n; int *arr = new int[n]; if (arr == nullptr) { cout << "Gagal mengalokasikan memori." << endl; return; } cout << "Masukkan " << n << " bilangan:" << endl; for (int i = 0; i < n; i++) { cout << "Bilangan ke-" << i + 1 << ": "; cin >> *(arr + i); } for (int i = 0; i < n - 1; i++) { int minIndex = i; for (int j = i + 1; j < n; j++) { if (*(arr + j) < *(arr + minIndex)) { minIndex = j; } } if (minIndex != i) { int temp = *(arr + i); *(arr + i) = *(arr + minIndex); *(arr + minIndex) = temp; } } } </pre>	<p>Program soal ke 1 ini mengurutkan elemen-elemen array menggunakan algoritma Selection Sort. Array dialokasikan secara dinamis dengan memanfaatkan pointer, dan program menyusun elemen dalam urutan menaik.</p> <p>Langkah-Langkah Program: Input Jumlah Elemen:</p> <p>Program meminta pengguna untuk memasukkan jumlah elemen array (n). Alokasi Memori:</p> <p>Array dialokasikan secara dinamis menggunakan new. Jika alokasi memori gagal, program akan menampilkan pesan error dan keluar. Input Nilai Array:</p> <p>Pengguna diminta untuk memasukkan n bilangan satu per satu, yang disimpan ke dalam array menggunakan pointer. Proses Pengurutan (Selection Sort):</p> <p>Untuk setiap elemen, program mencari nilai minimum di sisa elemen array dan menukarnya dengan elemen saat ini, jika perlu. Pointer digunakan untuk mengakses elemen array secara langsung. Output Hasil:</p> <p>Setelah pengurutan selesai, program menampilkan array yang telah diurutkan. Dealokasi Memori:</p>

```
cout << "\nArray setelah diurutkan  
dengan Selection Sort:" << endl;  
for (int i = 0; i < n; i++) {  
    cout << *(arr + i) << " ";  
}  
cout << endl;  
  
delete[] arr;  
}
```

Program menghapus array yang dialokasikan secara dinamis menggunakan delete[] untuk menghindari kebocoran memori.

Soal No 2

OUTPUT

```

Masukkan jumlah elemen array: 8
Masukkan 8 bilangan:
Bilangan ke-1: 5
Bilangan ke-2: 2
Bilangan ke-3: 4
Bilangan ke-4: 5
Bilangan ke-5: 1
Bilangan ke-6: 7
Bilangan ke-7: 9
Bilangan ke-8: 12

Array setelah diurutkan dengan Insertion Sort:
1 2 4 5 5 7 9 12

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program12() { int n; cout << "Masukkan jumlah elemen array: "; cin >> n; int *arr = new int[n]; if (arr == nullptr) { cout << "Gagal mengalokasikan memori." << endl; return; } cout << "Masukkan " << n << " bilangan:" << endl; for (int i = 0; i < n; i++) { cout << "Bilangan ke-" << i + 1 << ": "; cin >> *(arr + i); } for (int i = 1; i < n; i++) { int key = *(arr + i); int j = i - 1; while (j >= 0 && *(arr + j) > key) { *(arr + j + 1) = *(arr + j); j--; } *(arr + j + 1) = key; } } </pre>	<p>Program soal ke 2 ini mengurutkan elemen-elemen array menggunakan algoritma Insertion Sort. Elemen-elemen array disimpan secara dinamis dengan bantuan pointer.</p> <p>Langkah-Langkah Program: Input Jumlah Elemen:</p> <p>Pengguna diminta untuk memasukkan jumlah elemen array (n). Alokasi Memori Dinamis:</p> <p>Array dialokasikan secara dinamis menggunakan new. Jika alokasi gagal, program menampilkan pesan error dan keluar. Input Nilai Array:</p> <p>Pengguna memasukkan n bilangan, yang disimpan di array melalui pointer. Proses Pengurutan (Insertion Sort):</p> <p>Setiap elemen array diperiksa mulai dari indeks ke-1. Elemen saat ini (key) dibandingkan dengan elemen sebelumnya. Jika elemen sebelumnya lebih besar, elemen sebelumnya digeser ke depan sampai posisi yang sesuai ditemukan untuk key. Output Hasil:</p> <p>Setelah pengurutan selesai, program menampilkan array yang telah diurutkan.</p>

```
cout << "\nArray setelah diurutkan  
dengan Insertion Sort:" << endl;  
for (int i = 0; i < n; i++) {  
    cout << *(arr + i) << " ";  
}  
cout << endl;  
  
delete[] arr;  
}
```

Dealokasi Memori:

Memori yang digunakan oleh array dihapus menggunakan delete[] untuk mencegah kebocoran memori.

Fungsi Program:

Program ini menunjukkan implementasi Insertion Sort dengan pointer, menjadikannya berguna untuk memahami cara kerja algoritma dan penerapan pengelolaan memori dinamis.

Soal No 3

OUTPUT

```

Masukkan jumlah elemen array: 8
Masukkan 8 bilangan:
Bilangan ke-1: 5
Bilangan ke-2: 2
Bilangan ke-3: 4
Bilangan ke-4: 5
Bilangan ke-5: 1
Bilangan ke-6: 7
Bilangan ke-7: 9
Bilangan ke-8: 12

Array setelah diurutkan dengan Insertion Sort:
1 2 4 5 5 7 9 12

```

KODE DAN PENJELASAN

Source Code	Penjelasan
<pre> void program13() { int rows, cols; cout << "Masukkan jumlah baris: "; cin >> rows; cout << "Masukkan jumlah kolom: "; cin >> cols; int **matrix = new int*[rows]; for (int i = 0; i < rows; i++) { matrix[i] = new int[cols]; } cout << "Masukkan elemen matriks:\n"; for (int i = 0; i < rows; i++) { for (int j = 0; j < cols; j++) { cout << "Elemen matriks[" << i << "][" << j << "]: "; cin >> matrix[i][j]; } } cout << "\nMatriks:\n"; for (int i = 0; i < rows; i++) { for (int j = 0; j < cols; j++) { cout << matrix[i][j] << " "; } cout << endl; } for (int i = 0; i < rows; i++) { delete[] matrix[i]; } </pre>	<p>Program soal ke 3 ini digunakan untuk membuat, mengisi, dan menampilkan elemen sebuah matriks 2 dimensi yang dialokasikan secara dinamis menggunakan pointer. Setelah selesai, memori matriks dibebaskan untuk menghindari kebocoran memori.</p> <p>Langkah-Langkah Program: Input Dimensi Matriks:</p> <p>Pengguna diminta memasukkan jumlah baris (rows) dan kolom (cols) untuk matriks. Alokasi Memori Dinamis:</p> <p>Matriks dialokasikan secara dinamis sebagai array pointer (int**) untuk baris, dan setiap baris dialokasikan sebagai array elemen (int*).</p> <p>Input Elemen Matriks:</p> <p>Pengguna diminta memasukkan elemen matriks sesuai jumlah baris dan kolom. Elemen disimpan pada posisi matrix[i][j]. Menampilkan Matriks:</p> <p>Elemen matriks yang telah dimasukkan ditampilkan dalam bentuk baris dan kolom. Dealokasi Memori:</p> <p>Setiap baris (array elemen) dibebaskan menggunakan delete[]. Pointer ke baris (array pointer) juga dibebaskan menggunakan delete[].</p>

```
delete[] matrix;  
  
cout << "Memori telah dibebaskan."  
<< endl;  
}
```

Output Akhir:

Program menampilkan pesan bahwa memori telah dibebaskan.

Fungsi Program:

Program ini menunjukkan implementasi alokasi dan dealokasi memori dinamis untuk matriks. Hal ini berguna untuk memahami pengelolaan memori dalam struktur data 2 dimensi.

---- Source Code ---

```
#include <iostream>  
#include <conio.h>  
#include <cctype>  
#include <cstring>
```

```

#include <string>

using namespace std;

void program1()
{
    int a = 5; //Memberikan Variable a Nilai 5
    int *b; //Mendeklarasikan Variable b Sebagai Pointer
    b = &a; //Menyalin Alamat Variable a Kedalam Variable Pointer
    cout << "\nAlamat Variable Nilai a Adalah : "<< a << endl; //Memanggil Nilai
Variable a
    cout << "Nilai Variable a Adalah : "<< &a << endl; //Memanggil Alamat Nilai
Variable a
}

void program2()
{
    int a = 5; //Memberikan Variable a Nilai 5
    int *b; //Mendeklarasikan Variable b Sebagai Pointer
    b = &a; //Menyalin Alamat Variable a Kedalam Variable Pointer
    cout << "\nAlamat Variable Nilai a Adalah : "<< a << endl; //Memanggil Nilai
Variable a
    cout << "Nilai Variable a Adalah : "<< &a << endl; //Memanggil Alamat Nilai
Variable a
    cout << "Nilai Variable b Adalah : "<< b << endl; //Memanggil Nilai Variable b
    cout << "Nilai Yang Tertampung Dalam Variable b Adalah : "<< *b << endl;
//Memanggil Pointer b Dan Nilai Yang Tertampung Dalam Pointer b
}

void program3()
{
    int x, y;
    int *px;

    x = 89;
    y = x;
    px = &x;

    cout << "Nilai X : "<< x << endl;
    cout << "Nilai Y : "<< y << endl;
    cout << "Nilai Alamat X : "<< &x << endl;
    cout << "Nilai Alamat PX : "<< px << endl;
    cout << "Nilai PX : "<< *px << endl;

    x = 108;

    cout << "\nNilai X : "<< x << endl;
    cout << "Nilai Y : "<< y << endl;
    cout << "Nilai Alamat X : "<< &x << endl;
    cout << "Nilai Alamat PX : "<< px << endl;
    cout << "Nilai PX : "<< *px << endl;
}

```



```

    *px = 123;

    cout << "Nilai X : "<< x << endl;
    cout << "Nilai Y : "<< y << endl;
    cout << "Nilai Alamat X : "<< &x << endl;
    cout << "Nilai Alamat PX : "<< px << endl;
    cout << "Nilai PX : "<< *px << endl;
}

void program4()
{
    int x;
    int *px; //Pointer Ke Variable x
    int **ppx; //Pointer Ke Pointer px

    x = 175;
    px = &x;
    ppx = &px;

    cout << "Nilai x = " << x << endl;
    cout << "Nilai px = " << *px << endl;
    cout << "Nilai ppx = " << **ppx << endl;
}

void program5()
{
#define MAX 5

    int a[MAX];
    int *pa;
    pa = a;

    for (int i = 0; i < MAX; i++)
    {
        cout << "Masukan Nilai " << i+1 << " : ";
        cin >> a[i];
    }

    cout << endl;
    for (int i = 0; i < MAX; i++)
    {
        cout << "Nilai a["<<i<<"] = " << *pa <<endl;
        pa++;
    }
}

void program6()
{
    static int tgl_lahir[] = { 13,9,1982 };
    int *ptgl;
    ptgl = tgl_lahir; /* ptgl berisi alamatarray */
    cout<<"Diakses dengan pointer\n";
}

```

```

    cout<<"Tanggal = "<< *ptgl<<"\n";
    cout<<"Bulan = "<< *(ptgl + 1) <<"\n";
    cout<<"Tahun = "<<*(ptgl + 2) <<"\n";
    cout<<"\nDiakses dengan array biasa\n";
    cout<<"Tanggal = "<< tgl_lahir[0] <<"\n";
    cout<<"Bulan = "<< tgl_lahir[1] <<"\n";
    cout<<"Tahun = "<< tgl_lahir[2] <<"\n";
    getch();
}

void program7()
{
#define MAX 5

    char nama[] = "Albert Einstein";
    char *pNama = nama;

    cout << "Nama = " << nama << endl;
    cout << "pNama = " << pNama << endl;

    pNama += 7;

    cout << "Setelah pNama += 7" << endl;
    cout << "Nama = " << nama << endl;
    cout << "pNama = " << pNama << endl;
}

void program8()
{
#define MAX 5
    int x[10] = {0,1,2,3,4,5,6,7,8,9};
    int *px;
    int i;
    for (i = 0; i < 10; i++)
    {
        px = &x[i]; //membaca alamat dari x
        cout << x[i] << " " << *px << " " << px << endl;
    }
}

void program9()
{
    const char angka[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '\0'};
    const char h_vokal[] = {'A', 'I', 'U', 'E', 'O', '\0'};
    string teks;
    int vokal = 0;
    int konsonan = 0;
    int bilangan = 0;
    int spasi = 0;
    int lainnya = 0;

    cout << "Masukkan teks: ";

```

```

cin.ignore();
getline(cin, teks);

for (char c : teks) {
    if (isalpha(c)) {
        char huruf = toupper(c);
        if (strchr(h_vokal, huruf)) {
            vokal++;
        } else {
            konsonan++;
        }
    } else if (isdigit(c)) {
        bilangan++;
    } else if (isspace(c)) {
        spasi++;
    } else {
        lainnya++;
    }
}

cout << "\nHasil Analisis:\n";
cout << "Jumlah Vokal   : " << vokal << "\n";
cout << "Jumlah Konsonan: " << konsonan << "\n";
cout << "Jumlah Angka   : " << bilangan << "\n";
cout << "Jumlah Spasi   : " << spasi << "\n";
cout << "Jumlah Lainnya : " << lainnya << "\n";
}

// Bubble Sort
void program10()
{
    int n;

    cout << "Masukkan jumlah bilangan yang ingin diurutkan: ";
    cin >> n;

    int *arr = new int[n];
    if (arr == nullptr) {
        cout << "Gagal mengalokasikan memori." << endl;
        return;
    }

    cout << "Masukkan " << n << " bilangan:" << endl;
    for (int i = 0; i < n; i++) {
        cout << "Bilangan ke-" << i + 1 << ": ";
        cin >> *(arr + i);
    }

    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (*(arr + j) > *(arr + j + 1)) {

```

```

        int temp = *(arr + j);
        *(arr + j) = *(arr + j + 1);
        *(arr + j + 1) = temp;
    }
}

cout << "\nBilangan setelah diurutkan:" << endl;
for (int i = 0; i < n; i++) {
    cout << *(arr + i) << " ";
}
cout << endl;

delete[] arr;
}

// Selection Sort
void program11()
{
    int n;
    cout << "Masukkan jumlah elemen array: ";
    cin >> n;

    int *arr = new int[n];
    if (arr == nullptr) {
        cout << "Gagal mengalokasikan memori." << endl;
        return;
    }

    cout << "Masukkan " << n << " bilangan:" << endl;
    for (int i = 0; i < n; i++) {
        cout << "Bilangan ke-" << i + 1 << ": ";
        cin >> *(arr + i);
    }

    for (int i = 0; i < n - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (*(arr + j) < *(arr + minIndex)) {
                minIndex = j;
            }
        }
        if (minIndex != i) {
            int temp = *(arr + i);
            *(arr + i) = *(arr + minIndex);
            *(arr + minIndex) = temp;
        }
    }

    cout << "\nArray setelah diurutkan dengan Selection Sort:" << endl;
    for (int i = 0; i < n; i++) {
        cout << *(arr + i) << " ";
    }
}

```

```

    }
    cout << endl;

    delete[] arr;
}

// Insertion Sort
void program12()
{
    int n;
    cout << "Masukkan jumlah elemen array: ";
    cin >> n;

    int *arr = new int[n];
    if (arr == nullptr) {
        cout << "Gagal mengalokasikan memori." << endl;
        return;
    }

    cout << "Masukkan " << n << " bilangan:" << endl;
    for (int i = 0; i < n; i++) {
        cout << "Bilangan ke-" << i + 1 << ": ";
        cin >> *(arr + i);
    }

    for (int i = 1; i < n; i++) {
        int key = *(arr + i);
        int j = i - 1;

        while (j >= 0 && *(arr + j) > key) {
            *(arr + j + 1) = *(arr + j);
            j--;
        }
        *(arr + j + 1) = key;
    }

    cout << "\nArray setelah diurutkan dengan Insertion Sort:" << endl;
    for (int i = 0; i < n; i++) {
        cout << *(arr + i) << " ";
    }
    cout << endl;

    delete[] arr;
}

// Matrix Pointer
void program13()
{
    int rows, cols;

    cout << "Masukkan jumlah baris: ";
    cin >> rows;

```

```

    cout << "Masukkan jumlah kolom: ";
    cin >> cols;

    int **matrix = new int*[rows];
    for (int i = 0; i < rows; i++) {
        matrix[i] = new int[cols];
    }

    cout << "Masukkan elemen matriks:\n";
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cout << "Elemen matriks[" << i << "][" << j << "]: ";
            cin >> matrix[i][j];
        }
    }

    cout << "\nMatriks:\n";
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }

    for (int i = 0; i < rows; i++) {
        delete[] matrix[i];
    }
    delete[] matrix;

    cout << "Memori telah dibebaskan." << endl;
}

int main()
{
    int program;
    char ulang;

    do {
        cout << "=====" << endl;
        cout << "| 1. Contoh Program Pointer 1 |" << endl;
        cout << "-----" << endl;
        cout << "| 2. Contoh Program Pointer 2 |" << endl;
        cout << "-----" << endl;
        cout << "| 3. Contoh Program Pointer 3 |" << endl;
        cout << "-----" << endl;
        cout << "| 4. Contoh Program Pointer 4 |" << endl;
        cout << "-----" << endl;
        cout << "| 5. Contoh Program Pointer 5 |" << endl;
        cout << "-----" << endl;
        cout << "| 6. Contoh Program Pointer 6 |" << endl;
        cout << "-----" << endl;
    } while (program < 6);
    cout << "=====" << endl;
    cout << "Apakah ingin melanjutkan? (y/n): ";
    char jawab;
    while (jawab != 'y' && jawab != 'n') {
        jawab = getch();
    }
    if (jawab == 'y') {
        program++;
    } else {
        program = 6;
    }
    cout << "Program selesai. Terima kasih." << endl;
}

```

```

cout << "| 7. Contoh Program Pointer 7   |" << endl;
cout << "-----" << endl;
cout << "| 8. Contoh Program Pointer 8   |" << endl;
cout << "-----" << endl;
cout << "| 9. Contoh Latihan Soal No 1   |" << endl;
cout << "-----" << endl;
cout << "| 10. Contoh Latihan Soal No 2  |" << endl;
cout << "-----" << endl;
cout << "| 11. Contoh Selection Sort     |" << endl;
cout << "-----" << endl;
cout << "| 12. Contoh Insertion Sort     |" << endl;
cout << "-----" << endl;
cout << "| 13. Contoh Program Matrix     |" << endl;
cout << "===== " << endl;
cout << "Silahkan Pilih Program Yang Akan Dijalankan : ";
cin >> program;

```

```

switch (program) {
    case 1:
        program1();
        break;
    case 2:
        program2();
        break;
    case 3:
        program3();
        break;
    case 4:
        program4();
        break;
    case 5:
        program5();
        break;
    case 6:
        program6();
        break;
    case 7:
        program7();
        break;
    case 8:
        program8();
        break;
    case 9:
        program9();
        break;
    case 10:
        program10();
        break;
    case 11:
        program11();
        break;
}

```

```
        case 12:
            program12();
            break;
        case 13:
            program13();
            break;
        default:
            cout << "Pilihan Tidak Ada" << endl;
    }

    cout << "\nApakah Anda ingin kembali ke menu? (y/n): ";
    cin >> ulang;

} while (ulang == 'y' || ulang == 'Y');

cout << "Program selesai." << endl;
return 0;
}
```