



LEMBAR ASISTENSI
PRAKTIKUM STRUKTUR DATA
LABORATORIUM TEKNIK KOMPUTER
JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG

Judul Praktikum : ARRAY

Praktikan : Alvin Reihansyah Makarim (2115061083)

Asisten : Aldo Wijaya (2015061012)

A. Gilang Aleyusta Savada (2015061017)

No.	Catatan	Tanggal	Paraf
1. 2. 3.	Perbaiki format space Perbaiki format halaman, dan judul subab. Format sesuaikan modul.	2 November 2022	
1. 2. 3.	Perbaiki pemformatan Perbaiki kesimpulan Perbaiki penomoran pada bab VI	4 November 2022	
1.	Perbaiki kesimpulan	9 November 2022	
1.	ACC	9 November 2022	

Bandar Lampung, 9 November 2022

Aldo Wijaya
NPM. 2015061012

I. JUDUL PERCOBAAN

ARRAY

II. TUJUAN PERCOBAAN

Adapun tujuan dari percobaan ini adalah sebagai berikut :

1. Mahasiswa mampu melakukan membuat array
2. Mahasiswa mampu memahami penggunaan array

III. TEORI DASAR

3.1 Pengertian Array

Array merupakan sebuah bentuk struktur data. Array terdiri dari kumpulan elemen yang bertipe sama dalam urutan tertentu. Tiap-tiap data disimpan pada alamat (address) memori yang berbeda dan disebut dengan elemen. Letak atau posisi dari elemen array ditunjukkan oleh indeks atau posisi. Indeks merupakan tipe data yang menyatakan kuberurutan, misalnya integer atau karakter. Bagian yang menyusun array disebut elemen array (isi), yang masing-masing elemen dapat diakses tersendiri melalui indeks array.

Dibandingkan variabel biasa, array memiliki beberapa kelebihan, yaitu:

- Merupakan struktur penyimpanan data yang paling sederhana
- Dapat menampung lebih dari satu data
- Waktu akses yang sama untuk setiap elemen
- Dapat diakses secara acak.

Walau begitu, array juga memiliki kelemahan jika dibandingkan dengan variabel

biasa, yaitu :

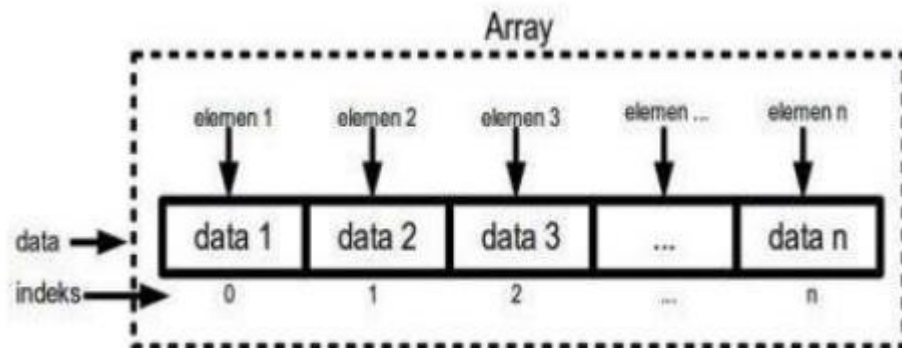
- Ukuran memori yang lebih besar
- Elemen kosong tetap memakan memori sehingga tidak efisien
- Harus bertipe homogen

Untuk mengakses elemen tertentu pada suatu array satu dimensi, dapat dilakukan dengan menyebutkan nama variabel array yang diikuti dengan indeks/posisi elemen yang diakses. Namun array juga memiliki beberapa kekurangan, di antaranya lebih boros memori jika banyak elemen yang tidak digunakan dan struktur penyimpanan datanya bersifat statis.

Pengaksesan elemen array dapat dilakukan berurutan atau secara acak berdasarkan indeks tertentu. Kaidah pengaksesan array adalah sebagai berikut :

`namaArray [indeks];`

Pengaksesan nilai pada indeks tertentu dapat dilakukan dengan mengeset nilai atau menampilkan nilai pada indeks yang dimaksud.



Gambar 3.1 Indeks array

Array bersifat statis. Maksudnya, ketika array sudah di deklarasikan dengan menggunakan tipe data dan panjang array tertentu, kita tidak dapat mengubah kembali tipe data dan Panjang array tersebut. Jika kita tetap memaksa untuk melakukannya, maka compiler akan mengalami eror.

3.2 Jenis-jenis Array

Berdasarkan dimensinya, array dapat dibedakan menjadi dua, yaitu array satu dimensi dan array multidimensi. Array berdimensi satu menyimpan data-data

dalam satu baris array, sehingga hanya dibutuhkan satu penomoran indeks. Elemen-elemen tersebut memiliki tipe data yang sama, tetapi isi dari elemen itu boleh berbeda. Pendeklarasian array diawali dengan penyebutan tipe data diikuti nama variabel array, diikuti jumlah elemen. Jika array hendak diberi nilai awal, nilai-nilai awal dituliskan dalam inisialisasi. Jika inisialisasi kurang dari jumlah elemen array, sisanya akan diinisialisasikan dengan nol. Elemen array diakses dengan menyebutkan indeks posisi elemen tersebut dalam array.

$$x = \begin{bmatrix} x[1] \\ x[2] \\ \dots \\ \dots \\ x[10] \end{bmatrix}$$

Gambar 3.2 Array satu dimensi

Sedangkan pada array multidimensi merujuk pada array yang mempunyai ukuran lebih dari satu subskrip. Array multi dimensi disebut juga dengan matriks. Array multidimensi dapat berupa array dua dimensi, tiga dimensi, empat dimensi, dan seterusnya. Pada array dua dimensi, data disimpan dalam array yang tiap indeksnya seperti memiliki arraynya masing-masing, sehingga dapat menyimpan data lebih banyak. Array dimensi n dideklarasikan dengan format sebagai berikut :

tipe_data nama_array [ukuran 1] [ukuran 2] ... [ukuran n];

Pada array dua dimensi, terdiri dari n buah baris dan m buah kolom, atau array dua dimensi juga biasa disebut sebagai array yang mempunyai dua subskrip, yaitu baris dan kolom. Bila divisualisasikan, maka akan berbentuk seperti tabel yang mana indeks pertama menunjukkan baris dan indeks kedua menunjukkan kolom

1 →	3	6	8	1	9
2 →	12	13	67	16	90
	↑ 1	↑ 2	↑ 3	↑ 4	↑ 5

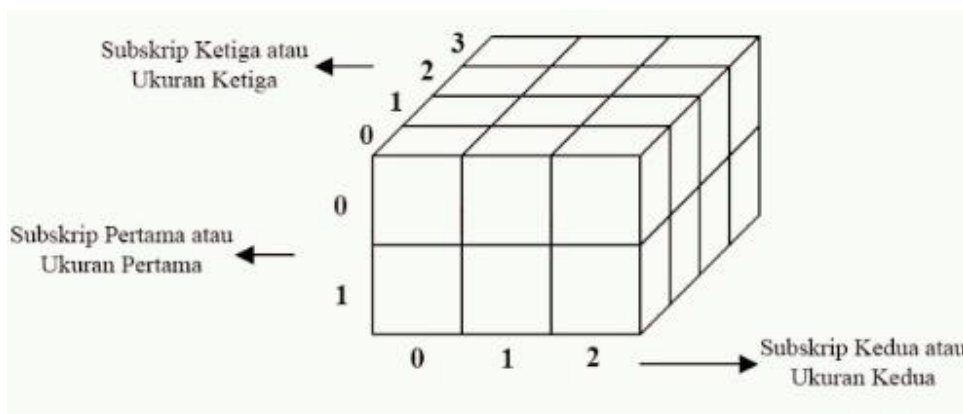
Kolom : 5

Gambar 3.3 Array dua dimensi

Contoh deklarasi array dua dimensi dengan ukuran 5×2 seperti pada gambar di atas adalah sebagai berikut :

```
Int array2d [5][2];
```

Sementara itu, pada array tiga dimensi, struktur yang dihasilkan akan lebih kompleks, karena tiap elemen seperti memiliki arraynya sendiri, dan tiap elemen dari array yang ada pada elemen tersebut juga memiliki elemennya sendiri. Jika divisualisasikan, array tiga dimensi akan berbentuk seperti tumpukan kotak yang membentuk balok seperti pada gambar di bawah ini.



Gambar 3.4 Array tiga dimensi

Contoh deklarasi array tiga dimensi dengan ukuran $2 \times 3 \times 4$ seperti pada gambar di atas adalah sebagai berikut :

```
Int array3d [2][3][4];
```

IV. PROSEDUR PERCOBAAN

Adapun source code percobaan ini adalah sebagai berikut :

4.1 Percobaan I-1 Array 1 Dimensi

4.1.1 Percobaan ArrayIntAddress

```
C++ ArrayIntAddress.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[8];
7      cout << &a << endl;
8      for (int i=0; i<8; i++) {
9          cout << &a[i] << endl;
10     }
11     return 0;
12 }
```

Gambar 4.1.1 Percobaan ArrayIntAddress

4.1.2 Percobaan ArrayCharAddress

```
C++ ArrayCharAddress.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char a[8];
7      cout << &a << endl;
8      for (int i=0; i<8; i++) {
9          cout << &a[i] << endl;
10     }
11     return 0;
12 }
```

Gambar 4.1.2 Percobaan 2 ArrayCharAddress

4.1.3 Percobaan ArrayInputOutput

```
C++ ArrayInputOutput.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[8];
7
8      for (int i=0; i<8; i++) {
9          cin >> a[i];
10     }
11
12     for (int i=0; i<8; i++) {
13         cout << a[i] << endl;
14     }
15
16     return 0;
17 }
```

Gambar 4.1.3 Percobaan ArrayInputOutput

4.1.4 Percobaan ArrayCharacter

```
C++ ArrayCharacter.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char a[15] = "informatika";
7      cout << a << endl;
8      a[9] = 'c';
9      a[10] = 's';
10     cout << a << endl;
11     cin >> a;
12     cout << a << endl;
13     return 0;
14 }
```

Gambar 4.1.4 Percobaan ArrayCharacter

4.2 Percobaan I-2 Array 2 Dimensi

4.2.1 Percobaan Array2DCharAddress

```
C++ Array2DCharAddress.cpp
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      char a[3][5];
6
7      for (int i=0; i<3; i++) {
8          cout << &a[i] << endl;
9
10         for (int j=0; j<5; j++) {
11             cout << &a[i][j] << endl;
12         }
13     }
14
15     cout << &a << endl;
16
17     return 0;
18 }
```

Gambar 4.2.1 Percobaan Array2DCharAddress

4.2.2 Percobaan Array2DIntAddress

```
C++ Array2DIntAddress.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[3][5];
7
8      for (int i=0; i<3; i++) {
9          cout << &a[i] << endl;
10
11         for (int j=0; j<5; j++) {
12             cout << &a[i][j] << endl;
13         }
14     }
15
16     cout << &a << endl;
17
18     return 0;
19 }
```

Gambar 4.2.2 Percobaan Array2DIntAddress

4.2.3 Percobaan Array2DInputOutput

```
C++ Array2DInputOutput.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[2][3];
7
8      for (int i=0; i<2; i++) {
9          for (int j=0; j<3; j++) {
10             cin >> a[i][j];
11         }
12     }
13
14     for (int i=0; i<2; i++) {
15         for (int j=0; j<3; j++) {
16             cout << a[i][j] << endl;
17         }
18     }
19
20     return 0;
21 }
```

Gambar 4.2.3 Percobaan Array2DInputOutput

4.2.4 Percobaan Array2DIndex

```
C++ Array2DIndex.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[2][3];
7      for (int i=0; i<2; i++) {
8          for (int j=0; j<3; j++) {
9              cin >> a[i][j];
10          }
11      }
12
13      a[0][1] = a[1][2] + a[0][0];
14      a[1][1] = a[0][0] * 20;
15
16      for (int i=0; i<2; i++) {
17          for (int j=0; j<3; j++) {
18              cout << a[i][j] << endl;
19          }
20      }
21
22      return 0;
23 }
```

Gambar 4.2.4 Percobaan Array2DIndex

V. PEMBAHASAN

Adapun pembahasan untuk percobaan ini adalah sebagai berikut :

5.1 Percobaan I-1 Array 1 Dimensi

5.1.1 Percobaan ArrayIntAddress

5.1.1.a Source Code

```
C++ ArrayIntAddress.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[8];
7      cout << &a << endl;
8      for (int i=0; i<8; i++) {
9          cout << &a[i] << endl;
10     }
11     return 0;
12 }
```

Gambar 5.1.1.a Percobaan ArrayIntAddress

Berdasarkan gambar 5.1.1.a yang merupakan source code dari percobaan ArrayIntAddress, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library `iostream` ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 4 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 5 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 6 terdapat pendeklarasian array `a` dengan Panjang array adalah 8 dan bertipe data integer. Pada baris ke 7 terdapat perintah untuk menampilkan address dari array `a`. Pada baris ke 8 terdapat instruksi untuk melakukan perulangan sebanyak 8 kali dimana perulangan tersebut akan menjalankan perintah untuk menampilkan address dari tiap-tiap elemen yang ada pada array `a` yang terdapat

pada baris ke 9. Kemudian pada baris ke 11 terdapat instruksi `return 0` yang menyatakan hasil keluaran dari fungsi `main()` bahwa program berakhir dengan normal.

5.1.1.b Output

```
if ($?) { g++ ArrayIntAddress.cpp -o ArrayIntAddress } ; if ($?) { .\ArrayIntAddress }  
0x61fdf0  
0x61fdf0  
0x61fdf4  
0x61fdf8  
0x61fdfc  
0x61fe00  
0x61fe04  
0x61fe08  
0x61fe0c
```

Gambar 5.1.1.b Output Percobaan ArrayIntAddress

Berdasarkan gambar 5.1.1.b yang merupakan output dari percobaan ArrayIntAddress, dapat dilihat bahwa output pada terminal menampilkan nilai address dari array a dan diikuti dengan address dari tiap-tiap elemennya, tepatnya sebanyak 8 alamat. Hal ini sesuai dengan perintah pada baris ke 7 pada source code yang menghasilkan tampilan address dari array a dan perintah pada baris 9 yang menghasilkan tampilan address dari tiap-tiap elemen pada array a.

5.1.2 Percobaan ArrayCharAddress

5.1.2.a Source Code

```
C++ ArrayCharAddress.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char a[8];
7      cout << &a << endl;
8      for (int i=0; i<8; i++) {
9          cout << &a[i] << endl;
10     }
11     return 0;
12 }
```

Gambar 5.1.2.a Source Code Percobaan ArrayCharAddress

Berdasarkan gambar 5.1.2.a yang merupakan source code dari percobaan ArrayCharAddress, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library `iostream` ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 4 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 5 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 6 terdapat pendeklarasian array `a` dengan Panjang array adalah 8 dan bertipe data `char`. Pada baris ke 7 terdapat perintah untuk menampilkan address dari array `a`. Pada baris ke 8 terdapat instruksi untuk melakukan perulangan sebanyak 8 kali dimana perulangan tersebut akan menjalankan perintah untuk menampilkan address dari tiap-tiap elemen yang ada pada array `a` yang terdapat pada baris ke 9. Kemudian pada baris ke 11 terdapat instruksi `return 0` yang menyatakan hasil keluaran dari fungsi `main()` bahwa program berakhir dengan normal.

5.1.2.b Output



```
ata\P1\" ; if ($?) { g++ Ar  
rayCharAddress.cpp -o Array  
CharAddress } ; if ($?) { .  
\ArrayCharAddress }  
0x61fe14
```

Gambar 5.1.2.b Output Percobaan ArrayCharAddress

Berdasarkan gambar 5.1.2.b yang merupakan output dari percobaan ArrayCharAddress, dapat dilihat bahwa output pada terminal menampilkan nilai address dari array a dan seharusnya diikuti dengan address dari tiap-tiap elemennya, tepatnya sebanyak 8 alamat. Namun, pada terminal justru terlihat kosong. Ini disebabkan karena array memiliki tipe data char, sehingga compiler tidak membaca address dari tiap indeks pada array. Hal ini sesuai dengan perintah pada baris ke 7 pada source code yang menghasilkan tampilan address dari array a dan perintah pada baris 9 yang menghasilkan tampilan address dari tiap-tiap elemen pada array.

5.1.3 Percobaan ArrayCharAddress

5.1.3.a Source Code

```
C++ ArrayInputOutput.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[8];
7
8      for (int i=0; i<8; i++) {
9          cin >> a[i];
10     }
11
12     for (int i=0; i<8; i++) {
13         cout << a[i] << endl;
14     }
15
16     return 0;
17 }
```

Gambar 5.1.3.a Source Code Percobaan ArrayInputOutput

Berdasarkan gambar 5.1.3.a yang merupakan source code dari percobaan ArrayInputOutput, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library iostream ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 4 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 5 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 6 terdapat pendeklarasian array a dengan Panjang array adalah 8 dan bertipe data integer. Berikutnya pada baris ke 8 terdapat perulangan sebanyak delapan kali yang tiap perulangannya memerintahkan kita untuk mengisi nilai dari tiap elemen pada array a secara berurutan mulai dari indeks 0 sampai indeks 7. Kemudian pada baris ke 12 terdapat perintah perulangan sebanyak delapan kali yang berfungsi untuk menampilkan tiap-tiap elemen elemen pada array a secara berurutan mulai dari indeks 0 sampai indeks 7. Kemudian pada baris ke 16 terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal.

5.1.3.b Output

```
● rayInputOutput.cpp -o Array  
InputOutput } ; if ($?) { .  
\\ArrayInputOutput }  
1  
2  
3  
4  
5  
6  
7  
8  
1  
2  
3  
4  
5  
6  
7  
8
```

Gambar 5.1.3.b Output Percobaan 3 ArrayInputOutput

Berdasarkan gambar 5.1.3.b yang merupakan output dari percobaan ArrayInputOutput, terlihat bahwa saya diminta untuk memasukkan nilai untuk tiap-tiap elemen pada array a. Pertama saya memasukkan angka 1, 2, 3, 4, 5, 6, 7, dan 8. Kemudian program menampilkan nilai dari tiap-tiap elemen array a, yaitu angka yang telah saya inputkan sebelumnya. Ini sesuai dengan perintah pada baris 9 dan 13 pada source code, dimana pada baris ke 9 terdapat perintah cin yang meminta kita untuk memasukkan nilai dan pada baris ke 13 terdapat perintah cout yang akan menampilkan nilainya.

5.1.4 Percobaan ArrayCharacter

5.1.4.a Source Code

```
C++ ArrayCharacter.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char a[15] = "informatika";
7      cout << a << endl;
8      a[9] = 'c';
9      a[10] = 's';
10     cout << a << endl;
11     cin >> a;
12     cout << a << endl;
13     return 0;
14 }
```

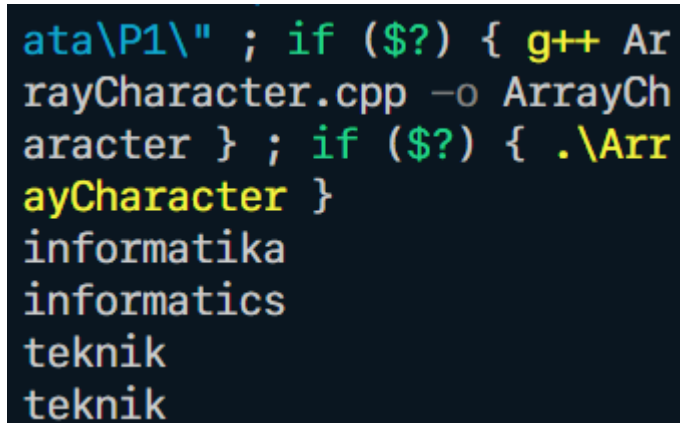
Gambar 5.1.4.a Source Code Percobaan ArrayCharacter

Berdasarkan gambar 5.1.4.a yang merupakan source code dari percobaan ArrayCharAddress, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library `iostream` ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 4 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 5 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 6 terdapat inisialisasi array `a` dengan Panjang array adalah 15, bertipe data `char` dan nilai pada indeks 0 hingga 9 pada array tersebut adalah “informatika” secara berurutan. Kemudian pada baris ke 7 terdapat perintah untuk menampilkan isi dari array `a`. Berikutnya pada baris ke 8 terdapat perintah untuk mengganti nilai dari array `a` indeks ke 9 dengan nilai `c` dan pada baris ke 9 terdapat perintah untuk mengganti nilai dari array `a` indeks ke 10 dengan nilai `s`. Pada baris ke 10, terdapat perintah untuk menampilkan kembali isi dari array `a`. Pada baris ke 11 terdapat perintah untuk mengganti seluruh nilai dari array `a`, kemudian pada baris 12 terdapat

perintah untuk menampilkan array a kembali. Kemudian pada baris ke 13 terdapat `return 0` yang menyatakan hasil keluaran dari fungsi `main()` bahwa program berakhir dengan normal.

5.1.4.b Output



```
ata\P1\" ; if ($?) { g++ Ar  
rayCharacter.cpp -o ArrayCh  
aracter } ; if ($?) { .\Arr  
ayCharacter }  
informatika  
informatics  
teknik  
teknik
```

Gambar 5.1.4.b Output Percobaan ArrayCharacter

Berdasarkan gambar 5.1.4.b yang merupakan output dari percobaan ArrayCharacter, pertama program akan menampilkan isi dari array a, yaitu “informatika”. Ini sesuai dengan isi dari inisialisasi array a pada baris ke 6 dan perintah cout pada baris ke 7. Kemudian, program menampilkan “informatics” setelah array a dipanggil kembali oleh perintah pada baris ke 10. Ini karena pada baris ke 8, nilai indeks ke 9 diubah menjadi c dan pada baris ke 9, nilai indeks 10 diubah menjadi s. Kemudian pada baris ke 11, program meminta kita untuk menginputkan nilai untuk menggantikan seluruh isi dari array a. Saya kemudian menuliskan “teknik”. Kemudian program menampilkan kembali isi dari array a yang sudah diganti, yaitu “teknik”. Hal ini sesuai dengan perintah pada baris ke 12, yaitu perintah cout untuk menampilkan isi dari seluruh array a.

5.2 Percobaan I-2 Array 2 Dimensi

5.2.1 Percobaan Array 2DCharAddress

5.2.1.a Source Code

```
C++ Array2DCharAddress.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char a[3][5];
7
8      for (int i=0; i<3; i++) {
9          cout << &a[i] << endl;
10
11         for (int j=0; j<5; j++) {
12             cout << &a[i][j] << endl;
13         }
14     }
15
16     cout << &a << endl;
17
18     return 0;
19 }
```

Gambar 5.2.1.a Source Code Percobaan Array2DCharAddress

Berdasarkan gambar 5.2.1.a yang merupakan source code dari percobaan Array2DCharAddress, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library `iostream` ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 4 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 5 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 6 terdapat pendeklarasian array `a` dengan Panjang array dimensi satu adalah 3 dan Panjang array dimensi dua adalah 5. Kemudian pada baris ke delapan terdapat perulangan sebanyak 3 kali yang akan menjalankan perintah yang ada di dalamnya, yaitu perintah untuk menampilkan address dari array `a` dimensi satu pada baris ke 9 dan melakukan nested loop pada baris 11. Nested loop tersebut memiliki perulangan sebanyak 5 kali dan memiliki perintah untuk menampilkan address dari array dimensi dua dari tiap array dimensi satu yang sedang dibaca, yaitu pada baris

ke 12. Kemudian setelah nested loop dan perulangan array dimensi satu selesai, terdapat perintah untuk menampilkan address dari array a pada baris ke 16. Kemudian pada baris ke 18 terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal.

5.2.1.b Output

```
; if ($?) { g++ Array2DCharAddress.cpp -o Array2DCharAddress } ; if ($?) { .\Array2DCharAddress }  
0x61fe09  
  
0x61fe0e  
  
Ém°  
m°  
°  
0x61fe13  
  
0x61fe09
```

Gambar 5.2.1.b Output Percobaan Array2DCharAddress

Berdasarkan gambar 5.2.1.b yang merupakan output dari Percobaan Array2DCharAddress, terlihat bahwa sistem pertama menampilkan address dari array dimensi satu. Hal ini sesuai dengan perintah pada baris ke 9 yaitu untuk menampilkan address dimensi satu dari array a. Kemudian program tidak menampilkan apa pun sebanyak 4 baris. Ini karena pada source code, program diminta untuk menampilkan address dari array dimensi kedua pada tiap array dimensi satu yang sedang dibaca. Namun karena array a bertipe data char, maka compiler tidak membaca address dari array tersebut. Compiler hanya membaca endl dari source code yang ada. Kedua hal tersebut terulang sebanyak 3 kali karena Panjang array dimensi satu pada array a adalah 3. Terakhir, program menampilkan address dari array a itu sendiri.

5.2.2 Percobaan Array2DIntAddress

5.2.2.a Source Code

```
C++ Array2DIntAddress.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[3][5];
7
8      for (int i=0; i<3; i++) {
9          cout << &a[i] << endl;
10
11         for (int j=0; j<5; j++) {
12             cout << &a[i][j] << endl;
13         }
14     }
15
16     cout << &a << endl;
17
18     return 0;
19 }
```

Gambar 5.2.2.a Source Code Percobaan Array2DIntAddress

Berdasarkan gambar 5.2.2.a yang merupakan source code dari percobaan Array2DIntAddress, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library `iostream` ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 4 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 5 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 6 terdapat array multidimensi `a` dengan panjang array dimensi satu sebesar 3 dan panjang array dimensi dua sebesar 5. Kemudian pada baris ke 8, terdapat perulangan sebanyak 3 kali. Pada perulangan tersebut, terdapat perintah untuk menampilkan address dari dimensi satu array `a` pada baris ke 9 dan perulangan sebanyak 5 kali pada baris ke 11. Di dalam nested loop tersebut terdapat perintah untuk melakukan menampilkan address dari dimensi dua dari indeks dimensi satu yang sedang dibaca. Terakhir pada baris ke 18 terdapat `return 0` yang menyatakan hasil keluaran dari fungsi `main()` bahwa program berakhir dengan normal.

5.2.2.b Output

```
; if ($?) { g++ Array2DIntAddress.cpp -o Array2D
IntAddress } ; if ($?) { .\Array2DIntAddress }
0x61fdd0
0x61fdd0
0x61fdd4
0x61fdd8
0x61fddc
0x61fde0
0x61fde4
0x61fde4
0x61fde8
0x61fdec
0x61fdf0
0x61fdf4
0x61fdf8
0x61fdf8
0x61fdfc
0x61fe00
0x61fe04
0x61fe08
0x61fdd0
```

Gambar 5.2.2.b Output Percobaan Array2DIntAddress

Berdasarkan gambar 5.2.2.b yang merupakan output dari percobaan Array2DIntAddress, terlihat bahwa sistem pertama menampilkan address dari array dimensi satu. Hal ini sesuai dengan perintah pada baris ke 9 yaitu untuk menampilkan address dimensi satu dari array a dengan melakukan perulangan. Kemudian terdapat perulangan sebanyak 4 kali yang mana perulangan itu menampilkan address dari tiap-tiap elemen dimensi dua dari dimensi satu yang sedang dibaca pada array a. Kemudian sistem menampilkan address array dimensi satu kembali dan diikuti dengan menampilkan 4 address dari tiap-tiap elemen dimensi dua dari dimensi satu yang sedang dibaca pada array a. Hal ini akan terulang sekali lagi sebanyak panjang array dimensi satu. Kemudian terakhir program menampilkan address dari array a itu sendiri.

Tidak seperti pada percobaan 5, pada program ini address dari dimensi dua array a dapat terbaca karena array a bertipe data integer.

5.2.3 Percobaan Array2DInputOutput

5.2.3.a Source Code

```
C++ Array2DInputOutput.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[2][3];
7
8      for (int i=0; i<2; i++) {
9          for (int j=0; j<3; j++) {
10             cin >> a[i][j];
11         }
12     }
13
14     for (int i=0; i<2; i++) {
15         for (int j=0; j<3; j++) {
16             cout << a[i][j] << endl;
17         }
18     }
19
20     return 0;
21 }
```

Gambar 5.2.3.a Source Code Percobaan Array2DInputOutput

Berdasarkan gambar 5.2.3.a yang merupakan source code dari percobaan Array2DInputOutput, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library `iostream` ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 4 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 5 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 6 terdapat array multidimensi `a` dengan panjang array dimensi satu sebesar 2 dan panjang array dimensi dua sebesar 3. Kemudian pada baris ke 8 terdapat perulangan sebanyak 2 perulangan yang di dalamnya terdapat perulangan pula (nested loop) dengan banyak perulangan sebanyak 3 perulangan. Di dalam nested loop tersebut, tepatnya pada baris ke 10, terdapat perintah untuk memasukkan nilai yang nantinya akan mengisi keseluruhan elemen dari array `a` secara satu per satu.

Berikutnya pada baris ke 14 terdapat perulangan yang hampir serupa dengan perulangan sebelumnya, yaitu terdapat perulangan sebanyak 2 perulangan yang di dalamnya terdapat perulangan pula (nested loop) dengan banyak perulangan sebanyak 3 perulangan. Di dalam nested loop tersebut, tepatnya pada baris ke 16, terdapat perintah untuk menampilkan nilai dari array a secara satu per satu. Terakhir pada baris ke 20 terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal.

5.2.3.b Output

```
; if ($?) { g++ Array2DInputOutput.cpp -o Array2DInputOutput } ; if ($?) { .\Array2DInputOutput }  
1  
2  
3  
4  
5  
6  
1  
2  
3  
4  
5  
6
```

Gambar 5.2.3.b Output Percobaan Array2DInputOutput

Berdasarkan gambar 5.2.2.b yang merupakan output dari percobaan Array2DInputOutput, dapat dilihat bahwa pertama kita diminta untuk memasukkan nilai sebanyak 6 kali, sesuai dengan banyak indeks dari array a. Hal ini sesuai dengan perintah pada baris ke 10 yang berada di dalam nested loop baris ke 8 dan 9. Setelah memasukkan nilai pada tiap elemen array multidimensi a, program akan menampilkan nilai dari tiap elemen tersebut. Hal ini sesuai dengan perintah yang ada pada baris ke 16 source code yang berada di dalam nested loop baris ke 14 dan 15.

5.2.4 Percobaan Array2DIndex

5.2.4.a Source Code

```
Array2DIndex.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a[2][3];
7      for (int i=0; i<2; i++) {
8          for (int j=0; j<3; j++) {
9              cin >> a[i][j];
10         }
11     }
12
13     a[0][1] = a[1][2] + a[0][0];
14     a[1][1] = a[0][0] * 20;
15
16     for (int i=0; i<2; i++) {
17         for (int j=0; j<3; j++) {
18             cout << a[i][j] << endl;
19         }
20     }
21
22     return 0;
23 }
```

Gambar 5.2.4.a Source Code percobaan Array2DIndex

Berdasarkan gambar 5.2.4.a yang merupakan source code dari percobaan Array2DIndex, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library `iostream` ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 4 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 5 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 6 terdapat array multidimensi `a` dengan panjang array dimensi satu sebesar 2 dan panjang array dimensi dua sebesar 3. Kemudian pada baris ke 8 terdapat perulangan sebanyak 2 perulangan yang di dalamnya terdapat perulangan pula (nested loop) dengan banyak perulangan sebanyak 3 perulangan. Di dalam nested loop tersebut, tepatnya pada baris ke 10, terdapat perintah untuk memasukkan nilai yang nantinya akan mengisi keseluruhan elemen dari array `a`

secara satu per satu.

Kemudian pada baris ke 13 terdapat pengisian nilai indeks ke [0][1] dari array a, dengan hasil operasi dari nilai indeks ke [1][2] ditambah dengan nilai dari indeks ke [0][0]. Lalu pada baris ke 14 terdapat pengisian nilai indeks ke [1][1] dari array a, dengan hasil operasi dari nilai indeks ke [0][0] dikali dengan 20.

Berikutnya pada baris ke 16 terdapat perulangan yang hampir serupa dengan perulangan sebelumnya, yaitu terdapat perulangan sebanyak 2 perulangan yang di dalamnya terdapat perulangan pula (nested loop) dengan banyak perulangan sebanyak 3 perulangan. Di dalam nested loop tersebut, tepatnya pada baris ke 18, terdapat perintah untuk menampilkan nilai dari array a secara satu per satu. Terakhir pada baris ke 20 terdapat return 0 yang menyatakan hasil keluaran dari fungsi main() bahwa program berakhir dengan normal.

5.2.4.b Output

```
p -o Array2DIndex } ; if ($?) { .\A  
rray2DIndex }  
1  
2  
3  
4  
5  
6  
1  
7  
3  
4  
20  
6
```

Gambar 5.2.4.b Output percobaan 8 Array2DIndex

Berdasarkan gambar 5.2.2.b yang merupakan output dari percobaan 8 Array2DIndex, dapat dilihat bahwa pertama kita diminta untuk memasukkan nilai sebanyak 6 kali, sesuai dengan banyak indeks dari array a. Hal ini sesuai dengan perintah pada baris ke 9 yang berada di dalam nested loop baris ke 7 dan 8. Setelah memasukkan nilai pada tiap elemen array multidimensi a, program akan menampilkan nilai dari tiap elemen tersebut. Namun terdapat perbedaan pada saat menampilkan elemen array dengan indeks [0][1] dan [1][1]. Ini karena pada indeks [0][1], terdapat pengisian nilai indeks dari hasil operasi dari nilai indeks ke [1][2], yaitu 6, ditambah dengan nilai dari indeks ke [0][0], yaitu 1 sehingga menghasilkan nilai 7. Sementara itu pada indeks [1][1], terdapat pengisian nilai indeks dari hasil operasi dari nilai indeks ke [0][0], yaitu 1, dikali dengan 20 sehingga menghasilkan nilai 20. Kedua operasi tersebut sesuai dengan perintah yang ada pada baris ke 13 dan 14, sedangkan penampilan nilai dari masing-masing elemen array a sesuai dengan perintah yang ada pada baris ke 16 source code yang berada di dalam nested loop baris ke 14 dan 15.

VI. KESIMPULAN

Adapun kesimpulan dari percobaan ini adalah sebagai berikut :

1. Berdasarkan percobaan 5.1.1 ArrayIntAddress, dapat disimpulkan bahwa array satu dimensi hanya menggunakan satu perulangan pada proses input dan outputnya, sedangkan array dua dimensi menggunakan dua perulangan (nested loop) untuk proses input dan outputnya.
2. Berdasarkan percobaan 5.1.2 ArrayCharAddress, dapat disimpulkan bahwa pada array satu dimensi, deklarasi dapat dilakukan hanya dengan menuliskan variabel dan satu indeks saja, contohnya `a[8]` yang artinya array variabel `a` memiliki panjang elemen sebanyak 4. Sedangkan untuk array dua dimensi deklarasi menuliskan variabel dan dua indeks, contohnya `a[2][3]` yang artinya array variabel `a` memiliki panjang elemen 2 baris dan 3 kolom.
3. Berdasarkan percobaan 5.1.2 ArrayCharAddress, dapat disimpulkan bahwa compiler tidak membaca address dari tiap elemen pada array jika array tersebut bertipe data `char`. Compiler hanya membaca address dari array tersebut.
4. Berdasarkan percobaan 5.1.4 ArrayCharacter, dapat disimpulkan bahwa kita dapat mengubah nilai dari suatu elemen dari array tanpa mengubah nilai dari elemen lainnya dengan cara mengakses elemen tersebut melalui indeks.
5. Berdasarkan percobaan 5.2.1 Array2DCharAddress, dapat disimpulkan bahwa compiler tidak membaca address dari elemen dimensi kedua dan seterusnya dari array pada array multidimensi, jika array tersebut bertipe data `char`. Compiler hanya membaca address dari array tersebut dan address pada array dimensi pertamanya.

DAFTAR PUSTAKA

Erliansyah Nasution dan Indra Yatini B.2005. Algoritma dan Struktur Data. Graha Ilmu. Yogyakarta.

Sjukani, Moh. 2010. (Algoritma Dan Struktur Data I) Dengan C, C++ dan Java. Jakarta: Mitra Wacana Media

Muhammad, M.A. (018. Struktur Data. Modul Struktur Data Unila PSTI. 2 (2) : 1 - 2.

TUGAS AKHIR

Membuat program analisis penjualan yang dapat menghitung penjualan terendah, tertinggi, total penjualan, dan rata-rata penjualan dari suatu toko.

1. Source Code

Adapun source code dari program adalah sebagai berikut :

```
C++ TugasAkhir_AlvinReihansyahMakarim.cpp
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7      string day[7] = {"Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"};
8      int sales[7][1];
9      int counter, lowest, highest, totalSales = 0;
10     string lowestDay, highestDay;
11
12     cout << "===== " << endl
13         << "===== Selamat datang di program analisis penjualan ===== " << endl
14         << "===== " << endl
15         << endl;
16
17     cout << "Silahkan masukkan jumlah penjualan untuk setiap hari berikut : " << endl;
18
19     for (counter = 0; counter < 7; counter++){
20         cout << endl;
21         cout << day[counter] << endl;
22         cout << "Jumlah penjualan : ";
23         cin >> sales[counter][0];
24         if (sales[counter][0] >= highest) {
25             highest = sales[counter][0];
26             highestDay = day[counter];
27         }
28         if (sales[counter][0] <= lowest) {
29             lowest = sales[counter][0];
30             lowestDay = day[counter];
31         }
32         totalSales += sales[counter][0];
33     }
34     cout << endl;
35
36     cout << "===== " << endl
37         << "===== Analisis penjualan pekan ini ===== " << endl
38         << "===== " << endl;
39     cout << left << setw(10) << "Hari" << left << setw(5) << "Jumlah Penjualan" << endl;
40
41     for (counter = 0; counter < 7; counter++) {
42         cout << left << setw(10) << day[counter]
43             << left << setw(5) << sales[counter][0]
44             << endl << endl;
45     }
46
47     cout << "Penjualan tertinggi ada pada hari "
48         << highestDay << ", yaitu sebesar "
49         << highest << " pcs" << endl << endl;
50     cout << "Penjualan terendah ada pada hari "
51         << lowestDay << ", yaitu sebesar "
52         << lowest << " pcs" << endl << endl;
53     cout << "Total penjualan pekan ini adalah sebanyak "
54         << totalSales << " pcs" << endl << endl;
55     cout << "Rata-rata penjualan per hari pada pekan ini adalah sebanyak "
56         << totalSales / 7 << " pcs" << endl << endl;
57
58     return 0;
59 }
```

Gambar 1 Source code program analisis penjualan

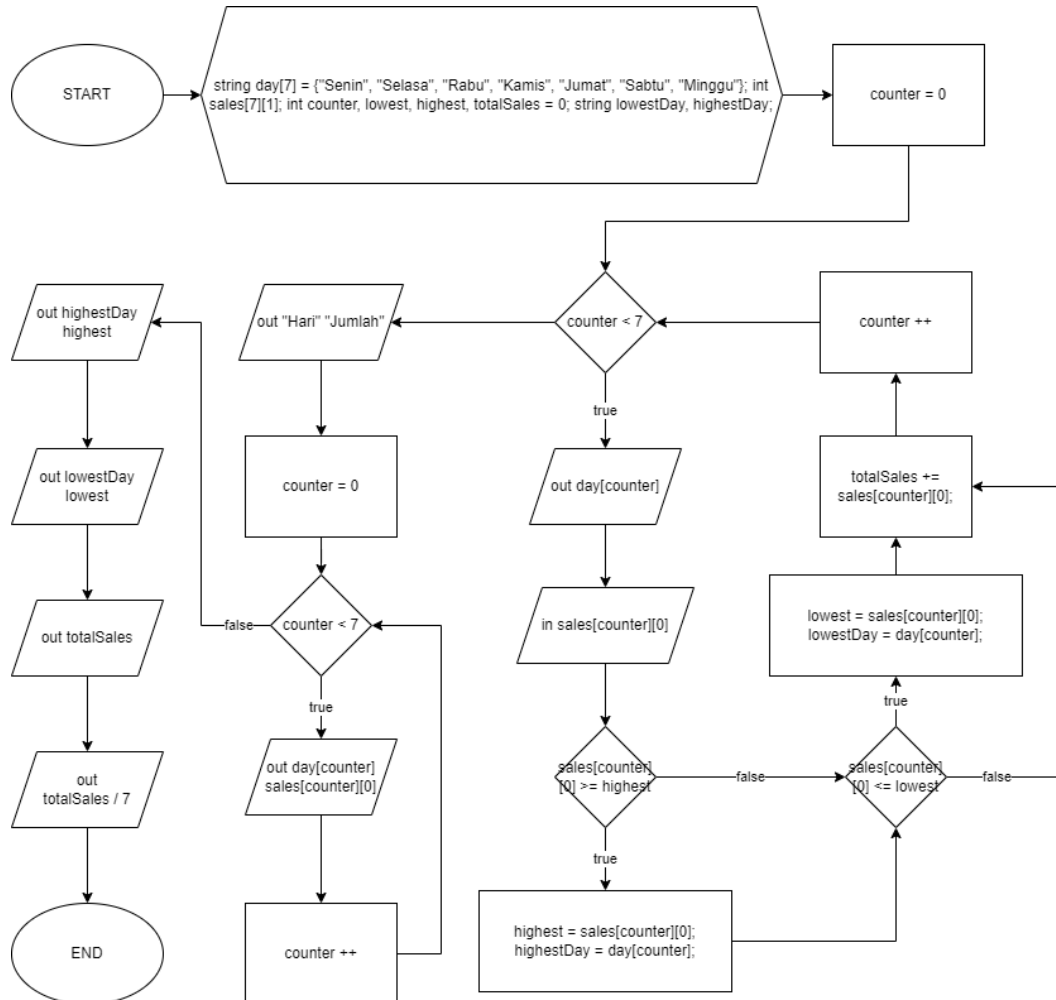
Berdasarkan gambar 1 yang merupakan source code dari program analisis penjualan, dapat dilihat bahwa pada baris ke 1 terdapat penggunaan library iostream ke dalam program yang berfungsi untuk menambahkan perintah input dan output. Kemudian pada baris ke 2 penggunaan library iomanip ke dalam program yang berfungsi untuk memanipulasi output. Kemudian pada baris ke 3 terdapat instruksi penggunaan penamaan standar untuk compiler.

Kemudian pada baris ke 5 terdapat deklarasi fungsi utama dari source code tersebut. Berikutnya pada baris ke 6 terdapat bracket yang memuat isi dari fungsi utama. Pada baris ke 7 terdapat inisialisasi array day bertipe data string dengan panjang 7. Array ini memiliki nilai "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", dan "Minggu". Pada baris ke 8 terdapat deklarasi array dua dimensi sales dengan panjang dimensi satu 7 dan panjang dimensi dua 1. Pada baris 9 terdapat deklarasi variabel counter, lowest, dan highest yang , serta inisialisasi variabel totalSales dengan nilai 0 yang semuanya bertipe data integer. Berikutnya pada baris 12 terdapat perintah untuk menampilkan string yang berbentuk seperti banner dan bertuliskan "Selamat datang di program analisis penjualan". Pada baris ke 17 terdapat perintah untuk menampilkan kalimat "Silahkan masukkan jumlah penjualan untuk setiap hari berikut : ". Pada baris ke 19 terdapat perulangan sebanyak 7 kali. Di dalam perulangan tersebut, terdapat perintah untuk menampilkan nilai dari array hari sesuai dengan indeks, tepatnya pada baris ke 21. Kemudian pada baris berikutnya, yaitu baris ke 22 terdapat perintah untuk menampilkan pesan "Jumlah penjualan : ". Pada baris ke 23, terdapat perintah untuk input user pada array sales dengan indeks dimensi satu sesuai dengan nilai counter dan indeks dimensi dua adalah 0. Kemudian pada baris ke 24, terdapat percabangan if dengan kondisi jika nilai dari sales[counter][0] lebih besar atau sama dengan nilai variable highest, maka lakukan inisialisasi nilai sales[counter][0] ke variabel highest dan inisialisasikan nilai string day[counter] ke dalam variabel highestDay. Kemudian pada baris ke 28, terdapat percabangan if dengan kondisi jika nilai dari sales[counter][0] lebih kecil atau sama dengan nilai variable lowest, maka lakukan inisialisasi nilai sales[counter][0] ke variabel lowest dan inisialisasikan nilai string day[counter] ke dalam variabel lowestDay. Berikutnya pada baris ke 36 terdapat perintah untuk menampilkan string yang berbentuk seperti banner dan bertuliskan

“Analisis penjualan pekan ini”. Pada baris ke 39, terdapat perintah untuk menampilkan kata “Hari” yang sudah dimodifikasi agar rata kiri dan memiliki lebar 10 dengan perintah `setw` dan “Jumlah Penjualan” yang sudah dimodifikasi agar rata kiri dan memiliki lebar 5 dengan perintah `setw`. Kemudian pada baris ke 41 terdapat perulangan sebanyak 7 kali yang berisi perintah untuk menampilkan nilai dari array `day` dengan indeks sesuai dengan nilai counter yang sudah dimodifikasi agar rata kiri dan memiliki lebar 10 dengan perintah `setw` serta perintah untuk menampilkan nilai dari array `sales` dengan indeks dimensi satu sesuai dengan nilai counter dan indeks dimensi dua adalah 0. Kemudian pada baris ke 47 terdapat perintah untuk menampilkan kalimat "Penjualan tertinggi ada pada hari ", nilai dari variable `highestDay`, ", yaitu sebesar ", nilai dari variable `highest`, dan “ pcs”. Kemudian pada baris ke 50 terdapat perintah untuk menampilkan kalimat "Penjualan terendah ada pada hari ", nilai dari variable `lowestDay`, ", yaitu sebesar ", nilai dari variable `lowest`, dan “ pcs”. Kemudian pada baris ke 53 terdapat perintah untuk menampilkan kalimat "Total penjualan pekan ini adalah sebanyak ", nilai dari variable `totalSales`, dan “ pcs”. Kemudian pada baris ke 55 terdapat perintah untuk menampilkan kalimat " Rata-rata penjualan per hari pada pekan ini adalah sebanyak ", nilai dari variable `totalSales` dibagi 7, dan “ pcs”.

2. Flowchart

Adapun flowchart dari program adalah sebagai berikut :



Gambar 2 Flowchart

3. Output

Adapun output dari program adalah sebagai berikut :

```
-o TugasAkhir_AlvinReihansyahMakarim } ; if ($?) { .\TugasAkhir_AlvinReihansyahMakarim }  
===== Selamat datang di program analisis penjualan =====  
=====  
Silahkan masukkan jumlah penjualan untuk setiap hari berikut :  
  
Senin  
Jumlah penjualan : 1  
  
Selasa  
Jumlah penjualan : 2  
  
Rabu  
Jumlah penjualan : 3  
  
Kamis  
Jumlah penjualan : 4  
  
Jumat  
Jumlah penjualan : 5  
  
Sabtu  
Jumlah penjualan : 6  
  
Minggu  
Jumlah penjualan : 7  
  
===== Analisis penjualan pekan ini =====  
=====  
Hari      Jumlah Penjualan  
Senin      1  
  
Selasa     2  
  
Rabu       3  
  
Kamis      4  
  
Jumat      5  
  
Sabtu      6  
  
Minggu     7  
  
Penjualan tertinggi ada pada hari Minggu, yaitu sebesar 7 pcs  
Penjualan terendah ada pada hari Senin, yaitu sebesar 1 pcs  
Total penjualan pekan ini adalah sebanyak 28 pcs  
Rata-rata penjualan per hari pada pekan ini adalah sebanyak 4 pcs
```

Gambar 3 Output program analisis penjualan

Berdasarkan gambar 3 yang merupakan output dari program analisis penjualan, dapat dilihat bahwa pertama program menampilkan string yang berbentuk seperti banner dan bertuliskan "Selamat datang di program analisis penjualan". Ini sesuai dengan perintah pada baris ke 12. Kemudian program menampilkan pesan "Silahkan masukkan jumlah penjualan untuk setiap hari berikut : " yang sesuai dengan perintah pada baris ke 17. Kemudian program menampilkan kata hari dan kalimat jumlah penjualan, diikuti dengan input user berupa angka sebanyak 7 kali.

Ini sesuai dengan perintah pada baris ke 20-23 yang ada pada perulangan sebanyak 7 kali. Kemudian, program menampilkan string yang berbentuk seperti banner dan bertuliskan “Analisis penjualan pekan ini” yang sesuai dengan perintah pada baris ke 36. Kemudian program menampilkan tabel yang berisi hari dan jumlah penjualan. Hal ini sesuai dengan perintah yang ada pada baris ke 39-45. Kemudian program menampilkan string "Penjualan tertinggi ada pada hari "; nilai dari variable highestDay, yaitu “Minggu”; string ", yaitu sebesar "; nilai dari variable highest, yaitu 7; dan string “ pcs” yang sesuai dengan perintah pada baris ke 4. Kemudian program menampilkan string "Penjualan terendah ada pada hari "; nilai dari variable lowestDay, yaitu “Senin”; string ", yaitu sebesar "; nilai dari variable lowest, yaitu 1; dan string “ pcs” yang sesuai dengan perintah pada baris ke 50. Kemudian program menampilkan string "Total penjualan pekan ini adalah sebanyak "; nilai dari variable totalSales, yaitu 28; dan string “ pcs” yang sesuai dengan perintah pada baris ke 5. Kemudian program menampilkan string " Rata-rata penjualan per hari pada pekan ini adalah sebanyak "; nilai dari variable totalSales dibagi 7, yaitu $28/7 = 4$; dan string “ pcs” yang sesuai dengan perintah pada baris ke 55.