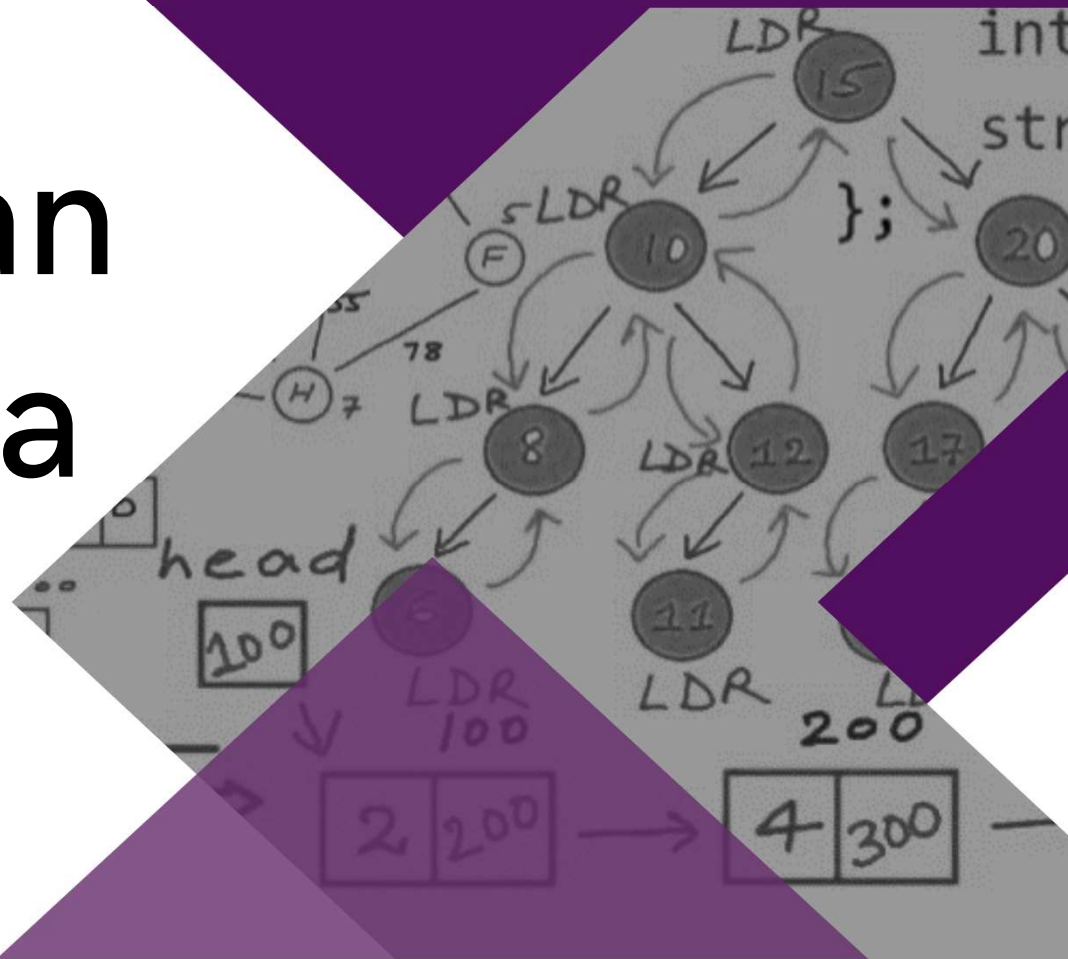


Algoritma dan Struktur Data



Pekan 1

**Array dan
Struct**

Alokasi
Memori

Varian
Linked List

Studi Kasus

Tree

Binary Tree

Advanced

Tree

Hash Table

Pointer

List

Stack dan
Queue

UTS

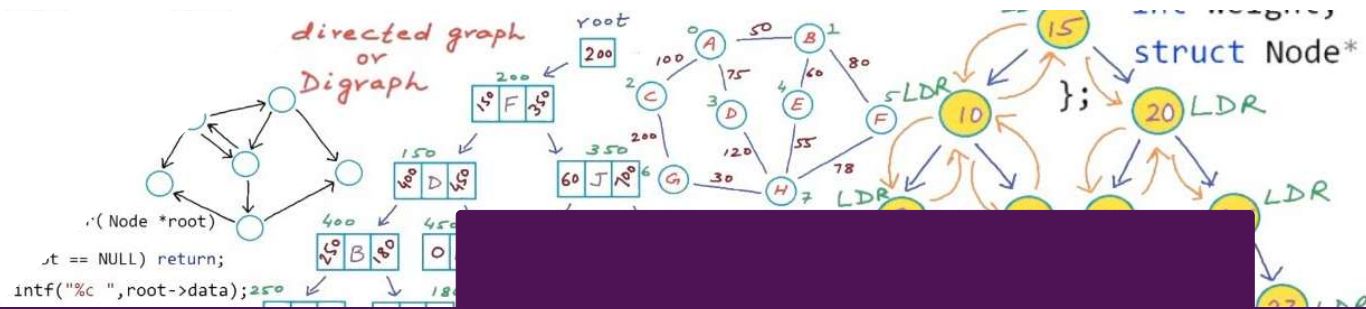
Non-Binary
Tree

Extended
Tree

Heap

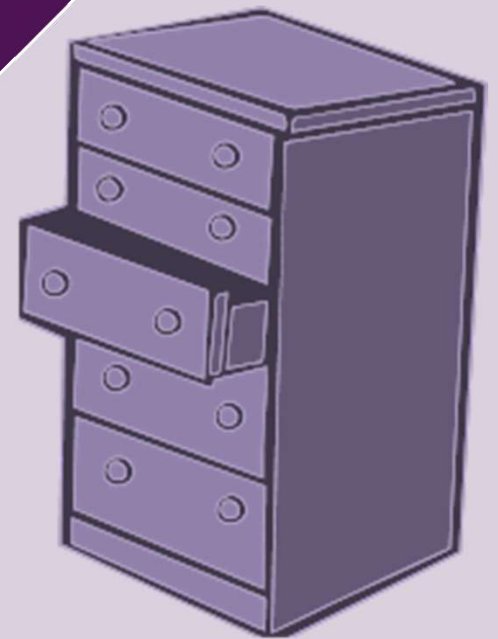
UAS

Tujuan



1	Mahasiswa mampu membedakan data komposit dan elementer
2	Mahasiswa mampu membedakan array dan struct
3	Mahasiswa memahami kegunaan array dan struct
4	Mahasiswa mampu mengidentifikasi kebutuhan data untuk suatu masalah

Data Elementer dan Komposit



Data Elementer dan Komposit

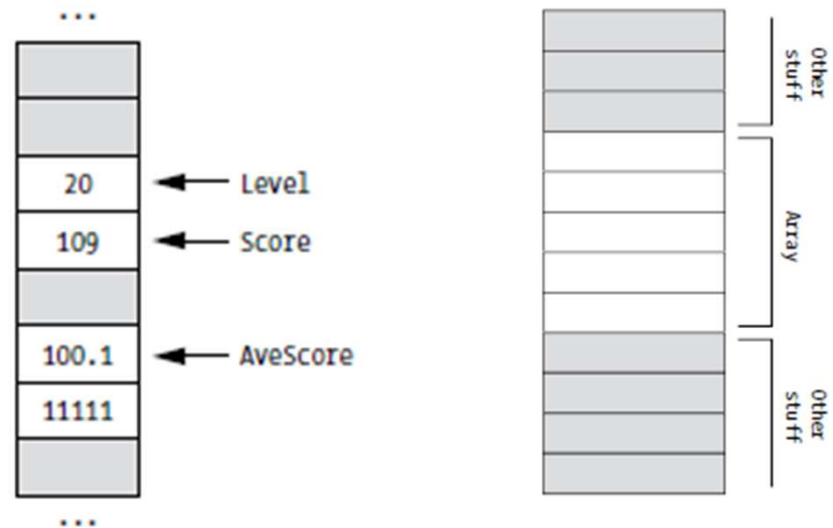
Data Elementer/Primitif

- Tipe data dasar dan fundamental yang mewakili nilai tunggal.
- Contoh: int, float, character, boolean, enumerasi

Data Komposit/Agregat

- Tipe data yang dapat menyimpan beberapa nilai atau elemen.
- Contoh: string, array, struct, list, dsb

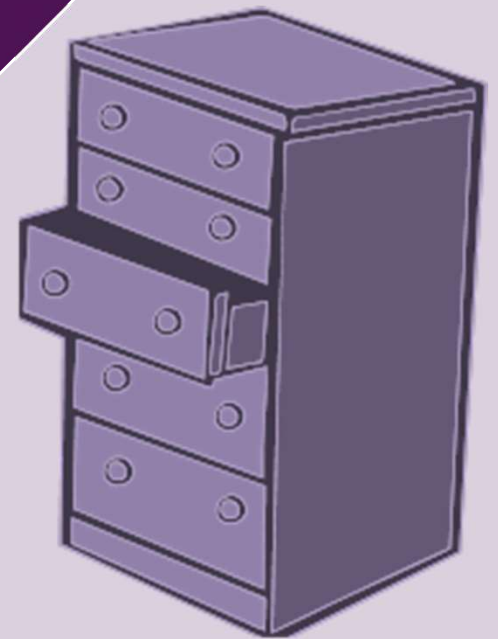
Data Elementer dan Komposit



(ki) memori computer digambarkan sebagai columns of bins

(ka) array sebagai bins pada memori komputer

Array



Array



Today

- ☐ Email the presentation to Jeff
- ☐ Buy a cake for dad's birthday

Tomorrow

- ☐ Research hotels in Paris

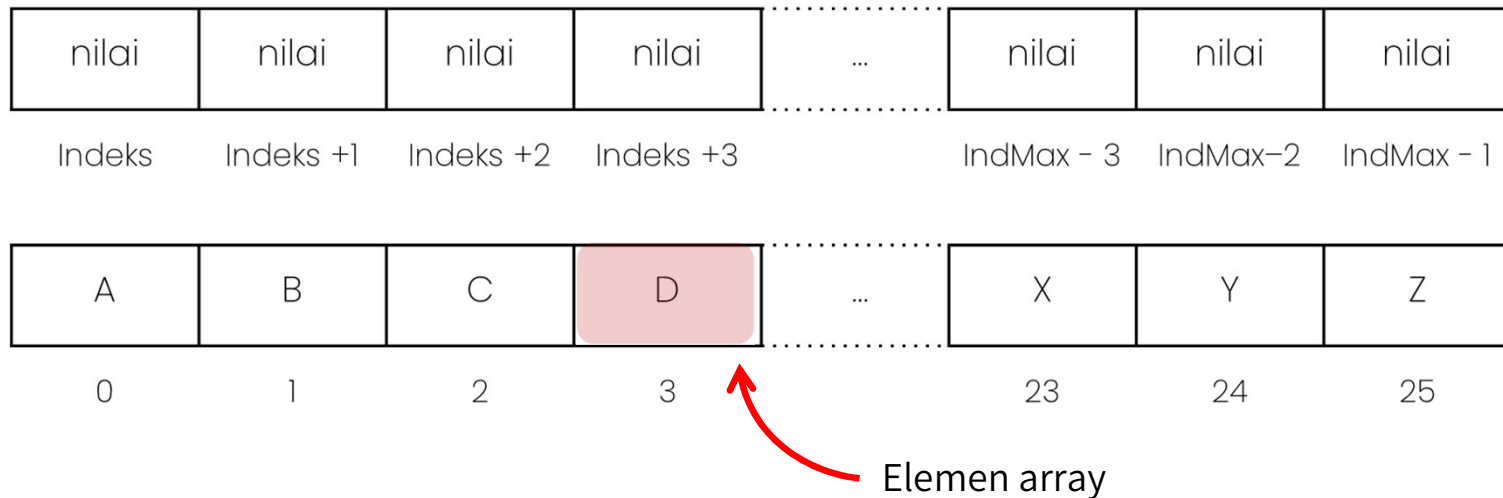
Upcoming

- ☐ Buy flight tickets to Paris

Kindergarten		
Baker	Otey	Rich
Arnold, Alexis	Baker, Bradley	Boyer, Abbot "Bo"
Carter, Keestyn	Bandy, Aiden	Bratcher, Noah
Childres, Raven	Carpenter, Jacob	Burden, Caleb
Darity, Dalyn	Cockereil, Noah	Burdette, Carter
Dela-Rosa, Nataly	Costello, Carly	Castro, Ziva
Dockery, Abrey	Cox, Wyatt	Collins, Brayden
Dubin, Xavier	Crafton, Brady	Cox, Alivia
Evans, Gage	Daugherty, Levi	Foote, Landyn
Fiener, Weston	Dodson, Addison	Gomez-Martinez, Osvaldo
Fuller, Hudson	Dunn, Kaitlyn	Green, Audrey
Fulton, Kayden	Glass, Gracyn	Hankins, Ace
Hale, Tyler	Gomez-Pablo, Lucas	Hudnall, Kinze
Hudnall, Joshua	Hauan, Darian	Jacinto-Alonzo, Matthew
Kimble, Dakota	Hooten, Emily	Lee, Piper
Logsdon, Xavier	House, Henry	Lenz, Gavn
Lopez-Gomez, Celso	Hunt, Clayton	Lopez-Hernandez, Natali
Lykins, Kaydence	Inglis, Lawson	Nicewinter, Isaiah
Lynch, Eliza	Jones, Brooklyn	Phelps, Mallory
Paille, Remy	McKinzie, Tyler	Rife, Ariannah
Rivera Huertas, Daniela	Nebolina, Isaiah	Smith, Cailyn
Robbins, Jace	Richards, Sawyer	Smith, Jayden
Stevens, Kaley	Smith, Gabriel	Sykes, Jerzeigh
Tyree, Alexandria	Taylor, Dauphina	Tomas, Mateo
Vazquez-Nebolina, Yasmin	Turner, Halo	Trejo-Alonzo, Justin
Williams, Kaulin	Vazquez-Nebolina, Yesinia	Wemyss, Carter
Woodcock, Jayden	Warren, Kaylen	Wormack, Zoey

Array

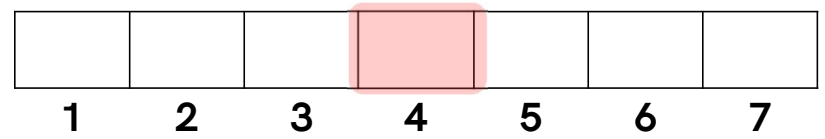
Contiguous area of memory consisting of equal-size elements indexed by contiguous integers.



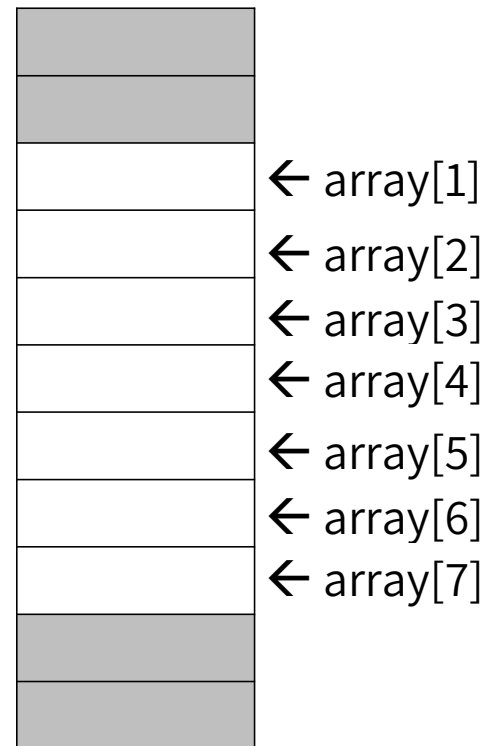
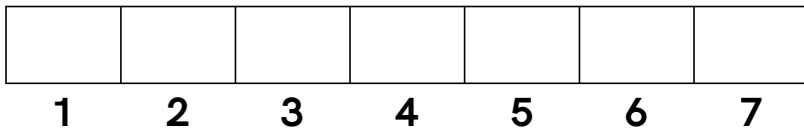
Karakteristik Array

Constant- time access

$\text{array_addr} + \text{elem_size} \times (i - \text{first index})$



Ilustrasi Penyimpanan Array di Memori



Karakteristik Array

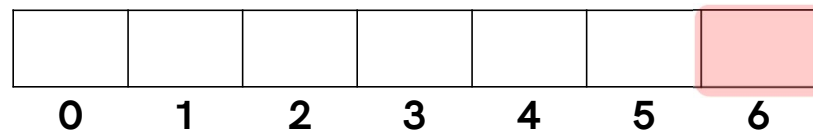
Constant- time access

Jika diketahui:

Address adalah 1000

Ukuran elemen adalah 8

Indeks pertama adalah 0



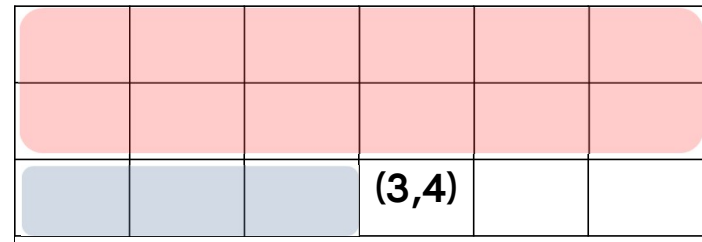
Berapa Alamat elemen yang berada pada indeks 6?

POP QUIZ

Karakteristik Array (Multi-Dimensi)

Constant- time access

$\text{array_addr} + \text{elem_size} \times ((3-1) \times 6 + (4-1))$



			(3,4)		

Ilustrasi Penyimpanan Array di Memori

Row-major

(1,1)
(1,2)
(1,3)
(1,4)
(1,5)
(1,6)
(2,1)
(2,2)
...
...
(3,6)

Column-major

(1,1)
(2,1)
(3,1)
(1,2)
(2,2)
(3,2)
(1,3)
(2,3)
...
...
(3,6)

Ilustrasi Penyimpanan Array di Memori

Asumsikan Anda memiliki array tiga dimensi yang disusun dalam urutan column-major dengan elemen pertama di indeks (1, 1, 1). Berapa indeks elemen berikutnya yang disimpan dalam memori?



POP QUIZ

(1,1,1)
...
...
...
...

Deklarasi Array

```
tipe_data namaArray [ jumlah elemen ]
```

```
string nama[5];  
char nilaiHuruf[6];  
bool statusLulus[3];  
float ips[10];  
int nilai[3]
```


Mengisi Nilai Array

```
string nama[5] = {"John", "Jane", "Mark", "Anne", "Alex"};
```

```
nama[0] = "John";
```

Mengakses Array

nama_array [indeks_array]

```
#include <iostream>
using namespace std;
int main()
{
    string nama[5]= {"John","Jane","Mark","Anne","Alex"};
    cout << nama[0] << endl;
    string fullname = nama[1] + " Doe" ;
    cout << fullname;
    return 0;
}
```

Matriks/Array 2 Dimensi

Struktur data dua dimensi yang terdiri dari baris dan kolom

	Kolom 1	Kolom 2	Kolom 3
Baris 1	array [0] [0]	array [0] [1]	array [0] [3]
Baris 2	array [1] [0]	array [1] [1]	array [1] [3]
Baris 3	array [2] [0]	array [2] [1]	array [2] [3]

Matriks/Array 2 Dimensi

Karena matriks sebenarnya adalah larik, maka konsep umum dari larik juga berlaku, seperti:

- Kumpulan elemen yang bertipe sama.
- Setiap elemen matriks dapat diakses secara langsung jika indeksnya (baris dan kolom) diketahui.
- Merupakan struktur data yang statis.

Penggunaan Matriks dalam Program

```
int main() {  
    // Contoh matriks (array 2D) dalam C++  
    int matrix[3][3] = {  
        {1, 2, 3},  
        {4, 5, 6},  
        {7, 8, 9}  
    };  
  
    // Mengakses dan mencetak elemen-elemen matriks  
    std::cout << "Elemen-elemen dalam matriks (array 2D):" << std::endl;  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 3; j++) {  
            std::cout << matrix[i][j] << " ";  
        }  
        std::cout << std::endl;  
    }  
  
    return 0;  
}
```

Struct

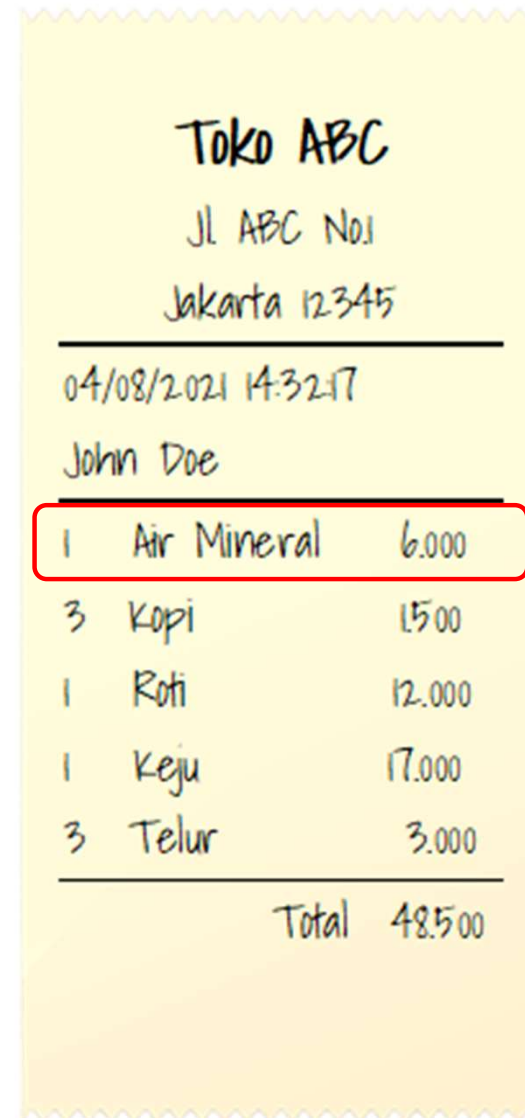
A diagram representing a struct instance. It consists of three adjacent rounded rectangular boxes. The first box is yellow and contains the number '1'. The second box is light blue and contains the text 'Roy'. The third box is light purple and contains the letter 'A'.

1	Roy	A
---	-----	---

```
typedef struct employee
{
    int roll_no;
    string name;
    char div;
};
```

Struct

Tipe data yang memungkinkan pengguna menggabungkan beberapa tipe data dalam satu unit data.

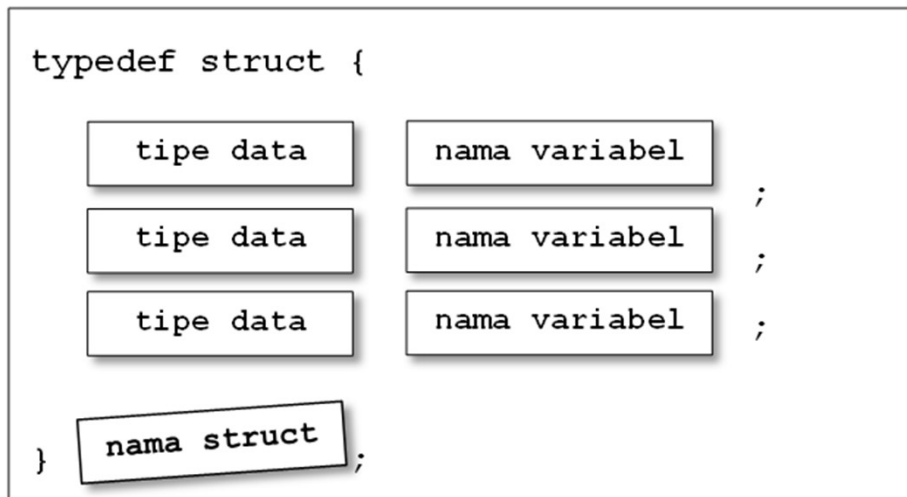


A handwritten receipt on a yellow, torn-edge paper. The receipt is for 'Toko ABC' located at 'Jl. ABC No.1 Jakarta 12345'. The date and time are '04/08/2021 14:32:17' and the customer is 'John Doe'. The items purchased are listed in a table with a red border around the first row. The items are: 1 Air Mineral (6.000), 3 Kopi (15.00), 1 Roti (12.000), 1 Keju (17.000), and 3 Telur (3.000). The total amount is 48.500.

Toko ABC		
Jl. ABC No.1		
Jakarta 12345		
<hr/>		
04/08/2021 14:32:17		
John Doe		
<hr/>		
1	Air Mineral	6.000
3	Kopi	15.00
1	Roti	12.000
1	Keju	17.000
3	Telur	3.000
		<hr/>
Total		48.500

Struct

Tipe data yang didefinisikan sendiri oleh pembuat program (*user defined variable*).



```
typedef struct {  
    int jumlah;  
    string barang;  
    int harga;  
} nota;
```


Declaration of Pre-processor Statements

Berisi file header atau library function

Global Declaration

Semua nama function dan variable yang ingin dikenali disemua bagian program (akan dibahas pada Bab 8)

Main function

Bagian yang pertama kali akan dijalankan program

User Defined Function

Bagian yang berisi realisasi function (akan dibahas pada Bab 8)


Deklarasi Struct

```
tipe_dataStruct namaVariabel;
```

```
nota belanja;
```

Mengisi Nilai Struct

```
belanja.jumlah = 1;  
belanja.barang = "air mineral";  
belanja.harga = 6000;
```



Gunakan titik untuk memisahkan
nama variable dan elemen struct
yang akan diisi

Mengakses Struct

```
namaVariabel.namaField
```

```
cout << belanja.barang;
```

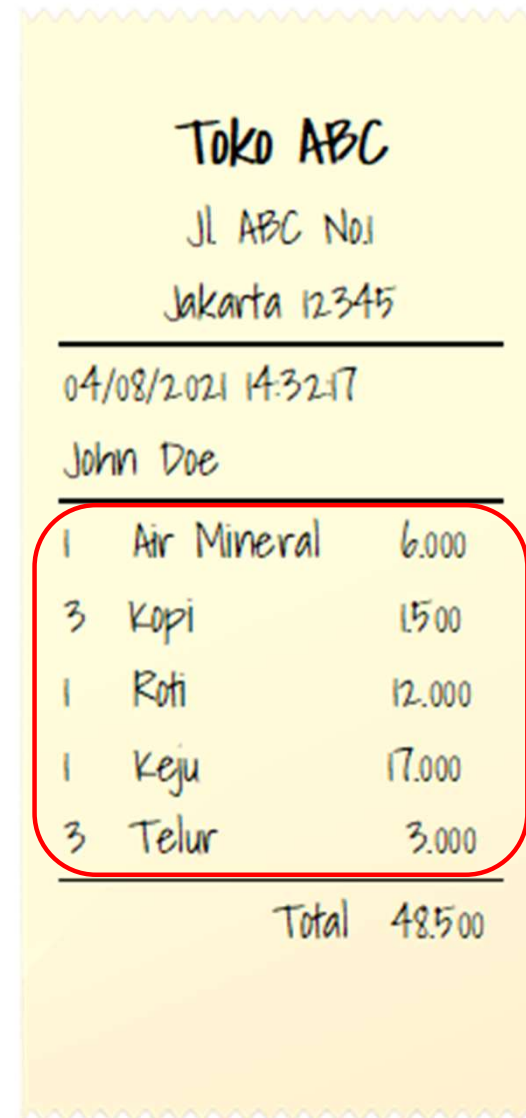
Array of Struct

 - Roll no
 - Name
 - Div

1	Roy	A	2	Jason	B	3	Tom	A
---	-----	---	---	-------	---	---	-----	---

Array of Struct

Kumpulan struktur yang memiliki tipe data yang sama.



Deklarasi Array of Struct

```
tipe_dataStruct namaVariabel [ jumlah elemen ] ;
```

```
nota daftarBelanja[5];
```

Mengisi Nilai Array of Struct

```
namaArray[indeks].namaField = nilai
```

```
daftar_belanja[0].jumlah = 1;  
daftar_belanja[0].barang = "Air Mineral";  
daftar_belanja[0].harga = 6000;
```


Mengakses Array of Struct

```
namaArray[indeks].namaField
```

```
cout << daftar_belanja[0].barang << endl;
```

Pencarian



Pencarian Beruntun

Sequential Search

```
1 AirMineral 6000
3 Kopi 1500
1 Roti 12000
1 Keju 17000
3 Telur 3000
```

```
nota cariBarang (nota daftar[], string kriteria, int ukuranArray)
{
    for(int i=0; i<ukuranArray; i++)
    {
        if(daftar[i].barang == kriteria)
        {
            return daftar[i];
        }
    }
}
```

```
hasil = cariBarang(daftar_belanja, barang, 5);
cout << "Harga " << hasil.barang << " adalah Rp " << to_string(hasil.harga);
```