

# PELATIHAN MENGENALKAN STRUKTUR DATA PADA SMK NEGERI 2 CIMAH

*Training on Introducing Data Structure at SMK Negeri 2 Cimahi*

## **Shandy Tresnawati**

Politeknik TEDC Bandung, Cimahi, Indonesia

e-mail: [shandy.tresnawati@poltektedc.ac.id](mailto:shandy.tresnawati@poltektedc.ac.id)

## **Nia Ekawati**

Politeknik TEDC Bandung, Cimahi, Indonesia

e-mail: [niaekawati@poltektedc.ac.id](mailto:niaekawati@poltektedc.ac.id)

## **Novita Lestari Anggreini**

Politeknik TEDC Bandung, Cimahi, Indonesia

e-mail: [novitalestari@poltektedc.ac.id](mailto:novitalestari@poltektedc.ac.id)

## **Dini Rohmayani**

Politeknik TEDC Bandung, Cimahi, Indonesia

e-mail: [dinirohmayani@poltektedc.ac.id](mailto:dinirohmayani@poltektedc.ac.id)

### **Abstract**

*Data structures and algorithms are important, integrated, and mutually supportive parts of building good and efficient application programs. Paying more attention to this can provide very important results, especially in increasing access speed and providing insight and guidance on how to improve problem solving and find better solutions. The service team conducts community service in Software Engineering expertise, this certainly supports the material that will be delivered to the Software Engineering department about learning Data Structures. In addition, the service team received an explanation from the teacher of SMK Negeri 2 Cimahi, there is data structure learning material so that it is sustainable in the training held by the service team. The material presented by 3 (three) speakers included linked lists, stacks and queues. The following are the results done by the service participants. The conclusion on community service at SMK Negeri 2 Cimahi, the service participants, namely school students, were enthusiastic in learning data structure material. Service participants understand better and can implement the results that have been done in the training process.*

**Keywords**—Data structure, Algorithms, Training.

## **1. PENDAHULUAN**

Struktur data dan algoritma merupakan bagian yang penting, terintegrasi, dan saling mendukung dalam membangun program aplikasi yang baik dan efisien. Memberikan perhatian lebih terhadap hal ini dapat memberikan hasil yang sangat penting, terutama dalam meningkatkan kecepatan akses dan memberikan wawasan serta panduan tentang cara meningkatkan pemecahan masalah dan menemukan solusi yang lebih baik. Algoritma sangat penting dan merupakan kunci yang menjadi dasar pemrograman meskipun algoritma berbeda dapat digunakan

untuk menyelesaikan masalah yang sama, memilih algoritma dengan strategi yang sesuai dan efisien akan mempercepat proses eksekusi. Algoritma dan struktur data memiliki hubungan yang kuat dan bekerja sama untuk mencapai hasil optimal dan memungkinkan solusi. Dinamika proses eksekusi yang cepat dan algoritma dalam menyelesaikan masalah. Meskipun membangun struktur data yang sesuai meningkatkan fungsionalitas algoritme, algoritme yang dikembangkan dengan benar mengurangi kompleksitas algoritme itu sendiri, dan kesederhanaan algoritme secara signifikan meningkatkan kinerja algoritme (Winarsih & Wahono, 2022).

Pengabdian kepada Masyarakat saat ini dilaksanakan di SMK Negeri 2 Cimahi. Kompetensi keahlian yang ada pada sekolah tersebut diantaranya: Animasi, Multimedia, Rekayasa Perangkat Lunak, Teknik Kimia Industri, Teknik Mekatronika, dan Teknik Pemesinan. Tim pengabdian melakukan pengabdian kepada Masyarakat pada keahlian Rekayasa Perangkat Lunak, hal ini tentu menunjang dari materi yang akan disampaikan pada jurusan Rekayasa Perangkat Lunak tentang pembelajaran Struktur Data. Selain itu, tim pengabdian mendapatkan penjelasan dari guru SMK Negeri 2 Cimahi, ada materi pembelajaran struktur data sehingga berkesinambungan pada pelatihan yang diadakan oleh tim pengabdian.

Ada banyak sekali sumber daya yang tersedia di internet, namun mungkin kurang sesuai dengan apa yang ingin Anda sampaikan. Mengingat karakteristik siswa berbeda-beda di setiap daerah, akan lebih efektif jika guru di setiap sekolah menyiapkan media pembelajaran. Hal ini memerlukan inovasi bagi guru untuk mengembangkan media pembelajaran yang dapat menunjang proses pembelajaran di masa pandemi (Purba et al., 2024).

## 2. METODE

Pelaksanaan kegiatan pengabdian kepada masyarakat yang dilakukan pada sekolah SMK Negeri 2 Cimahi menggunakan metode pelatihan yaitu beberapa peserta pengabdian pada jurusan Rekayasa Perangkat Lunak dilatih dalam pembelajaran Struktur Data dengan memberikan materi dan praktik. Berikut adalah susunan kegiatan acara pelatihan yang dilaksanakan di SMK Negeri 2 Cimahi.

**Tabel 1.** Susunan Acara Kegiatan Pengabdian kepada Masyarakat

No	Kegiatan	Waktu	Keterangan
1.	Kumpul tim pengabdian	06.30-07.00	Tim pengabdian kumpul di lobby kampus Politeknik TEDC Bandung
2.	Pemberangkatan tim pengabdian	07.00-07.30	Tempat pengabdian (SMK Negeri 2 Cimahi)
3.	Briefing	07.30-08.00	Briefing tim pengabdian
4.	Pembukaan	08.00-08.05	Moderator/Ketua
5.	Perkenalan tim pengabdian	08.05-08.10	Seluruh tim pengabdian
6.	Perkenalan peserta pengabdian	08.10-08.20	Seluruh peserta pengabdian
7.	Pengenalan sistem pembelajaran	08.20-08.30	Moderator/Narasumber/Notulen
8.	Materi Linked List dan tanya jawab	08.30-09.15	Narasumber 1
9.	Materi Stack dan tanya jawab	09.15-10.00	Narasumber 2

10.	Istirahat	10.00-10.30	Seluruh peserta dan tim pengabdian
11.	Materi Queue dan tanya jawab	10.30-11.15	Narasumber 3
12.	Games (Hiburan) + Pembagian Doorprize	11.15-11.30	Notulensi/Moderator
13.	Pembagian e-sertifikat	11.30-11.35	Notulensi/Moderator
14.	Penyerahan piagam	11.35-11.40	Ketua dan pihak sekolah
15.	Foto bersama	11.40-11.50	Seksi dokumentasi
16.	Penutup	11.50-12.00	Ketua/Notulen

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil

Materi yang disampaikan oleh 3 (tiga) narasumber diantaranya adalah linked list, stack dan queue. Berikut hasil yang dikerjakan oleh para peserta pengabdian.

```
Nodes of singly linked list:
1 2 3 4
```

**Gambar 1.** Hasil pengerjaan dari materi linked list

Pada gambar 1 peserta pengabdian mempelajari source code/coding pembuatan linked list yang disampaikan oleh narasumber 1.

```
Apakah stack kosongtrue
Elemen dalam stack :[6, 2]
Elemen dalam stack :[6, 2, 8]
Elemen setelah pop :[6, 2]
Ini adalah item paling atas :2
```

**Gambar 2.** Hasil pengerjaan dari materi stack

Pada gambar 2 peserta pengabdian mempelajari source code/coding pembuatan stack yang disampaikan oleh narasumber 2.

```
Elemen dalam queue [0, 1, 2, 3, 4]
menghapus elemen : 0
[1, 2, 3, 4]
data dari depan queue : 1
ukuran queue : 4
```

**Gambar 3.** Hasil pengerjaan dari materi queue

Pada gambar 3 peserta pengabdian mempelajari source code/coding pembuatan queue yang disampaikan oleh narasumber 3.

Berdasarkan 3 materi yang disampaikan oleh narasumber, para peserta pengabdian antusias dalam mengerjakan dan jika ada kendala dari source code/coding para peserta pengabdian tidak segan bertanya.



Gambar 4. Tim Pengabdi telah selesai memberikan pelatihan

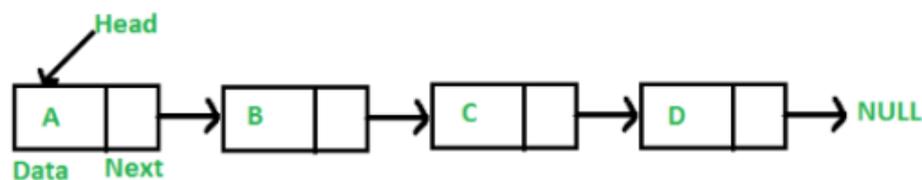
Pada gambar 4 foto bersama dengan guru serta dosen di halaman sekolah SMK Negeri 2 Cimahi sebagai simbolis penutup dari kegiatan pengabdian kepada masyarakat ini.

### 3.2 Pembahasan

Pembahasan yang dilakukan oleh tim pengabdi ada beberapa materi yang disampaikan, diantaranya:

#### 1. Linked List

Linked list adalah struktur data linier berbentuk rantai simpul di mana setiap simpul menyimpan 2 item, yaitu nilai data dan pointer ke simpul elemen berikutnya. Berbeda dengan array, elemen linked list tidak ditempatkan dalam alamat memori yang berdekatan melainkan elemen ditautkan menggunakan pointer.



Gambar 5. Proses linked list

Simpul pertama dari linked list disebut sebagai head atau simpul kepala. Apabila linked list berisi elemen kosong, maka nilai pointer dari head menunjuk ke NULL. Begitu juga untuk pointer berikutnya dari simpul terakhir atau simpul ekor akan menunjuk ke NULL.

Ukuran elemen dari linked list dapat bertambah secara dinamis dan mudah untuk menyisipkan dan menghapus elemen karena tidak seperti array, kita hanya perlu mengubah pointer elemen sebelumnya dan elemen berikutnya untuk menyisipkan atau menghapus elemen.

Implementasi Dalam Java:

```
1 → public class SinglyLinkedList {
2     //Represent a node of the singly linked list
3     class Node{
4         int data;
5         Node next;
6
7         public Node(int data) {
8             this.data = data;
9             this.next = null;
10        }
11    }
12
13    //Represent the head and tail of the singly linked list
14    public Node head = null;
15    public Node tail = null;
16
17    //addNode() will add a new node to the list
18    public void addNode(int data) {
19        //Create a new node
20        Node newNode = new Node(data);
21
22        //Checks if the list is empty
23        if(head == null) {
24            //If list is empty, both head and tail will point to new node
25            head = newNode;
26            tail = newNode;
27        }
28        else {
29            //newNode will be added after tail such that tail's next will point to newNode
30            tail.next = newNode;
31            //newNode will become new tail of the list
32            tail = newNode;
33        }
34    }
35
36    //display() will display all the nodes present in the list
37    public void display() {
38        //Node current will point to head
39        Node current = head;
40
41        if(head == null) {
42            System.out.println("List is empty");
43            return;
44        }
45        System.out.println("Nodes of singly linked list: ");
46        while(current != null) {
47            //Prints each node by incrementing pointer
48            System.out.print(current.data + " ");
49            current = current.next;
50        }
51        System.out.println();
52    }
53
54    public static void main(String[] args) {
55
56        SinglyLinkedList sList = new SinglyLinkedList();
57
58        //Add nodes to the list
59        sList.addNode(1);
60        sList.addNode(2);
61        sList.addNode(3);
62        sList.addNode(4);
63
64        //Displays the nodes present in the list
65        sList.display();
66    }
67 }
```

Gambar 6. Implementasi dalam Java

## 2. Stack

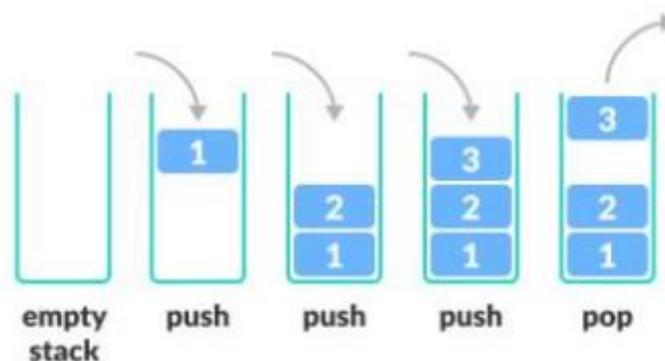
Stack atau dalam Bahasa Indonesia diartikan tumpukan, adalah struktur data linier yang mengikuti prinsip Last In First Out (LIFO). Artinya elemen yang terakhir disisipkan akan menjadi elemen pertama yang keluar.



**Gambar 7.** Struktur Data Stack

Cara struktur data stack dalam menyimpan sebuah nilai dapat kita bayangkan seperti piring yang disusun rapi secara bertumpuk ke atas. Apabila kita ingin mengambil piring bagian bawah, kita harus terlebih dahulu menyisihkan semua piring yang ada di atas.

Dalam istilah pemrograman, upaya menambahkan elemen pada struktur data stack disebut dengan push. Sedangkan proses menghapus atau menghilangkan elemen data dari stack disebut pop.



**Gambar 8.** Simulasi Pop

Dari gambar di atas, dapat terlihat bahwa meskipun elemen ke-3 adalah yang paling terakhir ditambahkan, namun elemen tersebut justru yang pertama dihapus. Operasi inilah yang kemudian disebut sebagai prinsip operasi LIFO (Last In First Out).

Kita dapat mengimplementasikan stack dengan bahasa pemrograman seperti C, C++, Java, Python, atau C#.

Implementasi Dalam Java :

```

1 import java.util.Stack;
2 public class StackExample {
3     public static void main(String[] args)
4     {
5         Stack<Integer> stk=new Stack<>();
6
7         boolean hasil=stk.empty();
8         System.out.println("Apakah stack kosong" +hasil);
9
10        // melakukan push pada stack
11        stk.push(6);
12        stk.push(2);
13
14        // menampilkan stack
15        System.out.println("Elemen dalam stack :" +stk);
16
17        stk.push(8);
18        System.out.println("Elemen dalam stack :" +stk);
19
20        // menghapus elemen pada stack
21        stk.pop();
22        System.out.println("Elemen setelah pop :" +stk);
23
24        // melihat elenen paling atas pada stack
25        System.out.println("Ini adalah item paling atas :" +stk.peek());
26    }
27 }

```

Gambar 9. Implementasi dalam Java

3. Queue

Queue adalah struktur data linier yang menerapkan prinsip operasi dimana elemen data yang masuk pertama akan keluar lebih dulu. Prinsip ini dikenal dengan istilah FIFO (First In, First Out).

Berbeda dengan struktur data stack yang menyimpan data secara bertumpuk dimana hanya terdapat satu ujung yang terbuka untuk melakukan operasi data, struktur data queue justru disusun secara horizontal dan terbuka di kedua ujungnya. Ujung pertama (head) digunakan untuk menghapus data sedangkan ujung lainnya (tail) digunakan untuk menyisipkan data.

Contoh nyata dalam kehidupan sehari-hari yang dapat menggambarkan struktur data queue adalah barisan orang yang menunggu untuk membeli tiket di gedung bioskop.

Orang yang baru datang akan bergabung dengan barisan dari ujung dan orang yang berdiri di depan akan menjadi yang pertama mendapatkan tiket dan meninggalkan barisan. Demikian pula dalam struktur data queue, data yang ditambahkan terlebih dahulu akan meninggalkan antrian terlebih dahulu. Berikut ini adalah ilustrasi dari queue



Gambar 10. Sistem antrian/queue

Pada gambar di atas, karena elemen 1 ditambahkan ke antrian lebih dulu daripada 2, maka 1 adalah elemen yang pertama dihapus dari antrian. Hal ini mengikuti aturan operasi FIFO.

Dalam istilah pemrograman, menempatkan item dalam struktur data queue disebut enqueue, sedangkan operasi menghapus item dari queue disebut dequeue.

Kita dapat mengimplementasikan queue dalam bahasa pemrograman apa pun seperti C, C++, Java, Python atau C#, dengan spesifikasi yang hampir sama.

Implementasi Dalam Java :

```
1 import java.util.LinkedList;
2 import java.util.Queue;
3
4 public class QueueExample {
5
6     public static void main(String[] args)
7     {
8         Queue<Integer> q
9             = new LinkedList<>();
10
11         // menambahkan elemen {0, 1, 2, 3, 4}
12         for (int i = 0; i < 5; i++)
13             q.add(i);
14
15         // tampil queue
16         System.out.println("Elemen dalam queue " + q);
17
18         // untuk menghapus data dari depan
19         int removedele = q.remove();
20         System.out.println("menghapus elemen : " + removedele);
21
22         System.out.println(q);
23
24         // untuk melihat data dari depan
25         int head = q.peek();
26         System.out.println("data dari depan queue : " + head);
27
28         // implementasi
29         int size = q.size();
30         System.out.println("ukuran queue : " + size);
31     }
32 }
```

Gambar 11. Implementasi dalam Java

#### 4. KESIMPULAN

Kesimpulan pada pengabdian kepada masyarakat di SMK Negeri 2 Cimahi, para peserta pengabdian yaitu siswa-siswa sekolah, antusias dalam mempelajari materi struktur data. Peserta pengabdian menjadi lebih memahami serta dapat mengimplementasikan pada hasil yang telah dikerjakan pada proses pelatihan yang dapat dilihat dari poin 3.1.

#### 5. SARAN

Berdasarkan hasil kegiatan pengabdian kepada masyarakat di SMK Negeri 2 Cimahi, tentang pembelajaran struktur data. Selanjutnya dapat melakukan pelatihan tentang mengenalkan salah satu bahasa pemrograman.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Politeknik TEDC Bandung yang telah memberikan dukungan **finansial** terhadap pengabdian ini.

**DAFTAR PUSTAKA**

- Purba, A. P., Purba, A. M., Sitanggang, E. S. Y., & Simatupang, E. M. (2024). Pelatihan Media Pembelajaran PowerPoint Berbasis Online Menggunakan Canva Di SMK Swasta Trisula Dolok Sanggul. *Journal Puan Indonesia*, 5(2), 443–448.
- Winarsih, S. S., & Wahono, A. W. (2022). Implementasi Dan Pengujian Struktur Data Berbasis Acuan Untuk Program Aplikasi Mengungkap Kepribadian Berdasarkan Tanggal Lahir Dan Nama. *Jurnal TIKomSIN*, 10(2), 11–17.

