

1. Gambaran Umum Kegiatan

Tujuan aktivitas ini adalah untuk memberikan pemahaman kepada siswa mengenai salah satu bentuk representasi data, yang dapat digunakan untuk membantu berpikir komputasional. Selama pembelajaran berlangsung, guru dapat memantau dan menilai proses diskusi kelompok yang dilakukan.

A. Pendahuluan

Guru membagi kelompok diskusi, memberikan topik yang akan didiskusikan, dan membagi lembar kerja siswa

B. Kegiatan Inti

Setiap kelompok mendiskusikan topik diskusi tentang contoh soal representasi data *stack* yang sudah diberikan guru. Guru juga dapat mencari soal lain yang relevan dengan representasi data *stack*.

Sudut pandang :

Untuk memahami sejumlah data, maka dapat dilakukan dengan representasi data, yaitu tumpukan (*stack*)

Protokol diskusi :

1. Alokasi waktu 60 menit
2. Guru sebagai moderator juga membimbing dan memantau kinerja siswa
3. Siswa dapat mengajukan pertanyaan dengan mengacungkan jari telunjuk sebelah kanan terlebih dulu

CONTOH Soal 1 representasi data tumpukan (*stack*)

Inspirasi dari soal Bebras I-2018-CY-03– Antrian Mobil

Ari mempunyai jalanan di halamannya yang cukup panjang. Tetangganya dapat parkir di jalan tersebut, namun hanya bisa mundur untuk keluar sebab jalannya sempit. Karena ia hanya memiliki sebuah mobil, tetangga minta izin untuk ikut parkir di jalan tersebut. Supaya yakin tidak ada yang terblokir, ia membuat tabel kapan tetangga boleh parkir, dan kapan harus pergi.



Setiap pagi, mobil yang akan pergi harus keluar sebelum mobil lainnya masuk. Ari parkir duluan, kemudian Bob parkir setelah Ari. Seperti dapat dilihat pada tabel, tak ada yang meninggalkan jalan pada hari Senin.

Hari	Jumlah Mobil Pergi	Jumlah Mobil Masuk	Pemilik Mobil dan Urutan Mereka masuk
Senin	0	2	Ari, Bob
Selasa	1	3	Kati, Ben, Roi
Rabu	2	1	Desi
Kamis	0	2	Fina, Rosa
Jumat	3	1	Vino

Pertanyaan :

Mobil siapa yang akan diparkir di jalanan pada akhir hari Jumat?

Jawaban : Ari, Kati, Vino.

Penjelasan :

Jika diurutkan sepanjang minggu, berikut ini adalah urutan parkir mobil:

- ✦ Akhir Senin: Ari, Bob
- ✦ Akhir Selasa: Ari, Kati, Ben, Roi
- ✦ Akhir Rabu: Ari, Kati, Desi
- ✦ Akhir Kamis: Ari, Kati, Desi, Fina, Rosa
- ✦ Akhir Jumat: Ari, Kati, Vino **Ini Informatika!**

Soal ini menggunakan konsep *stack* (tumpukan). Tumpukan adalah tipe data abstrak tempat elemen terakhir yang dimasukkan dimana elemen yang pertama akan keluar. Pengoperasian *stack* melibatkan dua fungsi yaitu *push* (memasukkan item ke dalam *stack*) dan *pop* (hapus elemen dari *stack*). Operasi tumpukan digambarkan sebagai LIFO (terakhir masuk pertama keluar).

CONTOH Soal 2 representasi data tumpukan (stack)

Inspirasi dari soal Bebras I-2016-CZ-026 – Karung dalam Elevator

Sejumlah karung diletakkan di koridor, di dekat suatu lift (elevator). Koridor sangat sempit sehingga karung-karung harus dibariskan satu-satu. Dengan lift tersebut, karung-karung hendak dikirimkan ke toko di lantai dasar. Sekali angkut, Lift hanya dapat mengangkat karungkarung dengan total berat tidak kurang dari 80 kg dan tidak lebih dari 100 kg. Setelah terkirim maka lift akan kembali ke lantai tersebut.



Saat memasukkan karung-karung ke dalam lift, karung yang terdekat dengan lift yang akan diambil terlebih dulu. Seandainya penambahan suatu karung dapat menyebabkan overload (terlalu berat, karena total beratnya lebih dari 100 kg), karung itu untuk sementara tidak dimasukkan ke dalam lift tapi ditaruh di koridor pada arah berlawanan dari semula sejauhjauhnya (jika tidak overload tentu akan dimasukkan ke dalam lift!).

Bila karung-karung dari barisan awal telah diambil, hal yang sama kemudian dilakukan pada barisan karung yang terbentuk pada koridor arah berlawanan dengan semula hingga seluruh karung di situ berhasil dikirim ke toko atau dipindah ke ujung koridor berlawanan dengannya. Hal itu terus-menerus dilakukan sampai semua karung berhasil dikirim ke toko.

Pertanyaan :

Sampai semua karung dikirimkan ke toko dengan prosedur tersebut di atas, berapa kali lift turun-naik mengangkat karung-karung itu semula karung-karung ada seperti pada gambar di atas dengan berat masing-masing seperti yang tercantum pada setiap karung?

Jawaban : 4

Penjelasan :

- ✦ Pada pemuatan pertama, 3 karung dengan berat $40+20+34=94$ kg.
- ✦ Pada pemuatan kedua, karung 55 kg dimasukkan ke dalam lift.
- ✦ Lalu karung berikutnya (50 kg), yang kemudian akan mengakibatkan overload, sehingga ditaruh pada ujung yang berlawanan.
- ✦ Karung berikutnya (23 kg) ditaruh kembali dalam lift, namun lift tak dapat pergi hanya karena total berat $55+23= 78$ kg.

- ✦ Karung berikutnya adalah 45 kg, yang akan membuat beban lift terlalu berat, maka karung itu dibawa ke ujung yang berlawanan. Hal yang sama terjadi untuk karung berikutnya (30 kg).
- ✦ Akhirnya, karung berikutnya 10 kg dapat ditaruh dalam lift, dan berangkat dengan total $55+23+10=88$ kg.
- ✦ Ketiga karung terakhir ($25+30+15 = 70$ kg) ditaruh dalam lift.
- ✦ Demikian seterusnya pemuatan karung-karung ke dalam lift dilakukan hingga semua karung terkirim.

Ini Informatika!

Pada soal ini, kita perlu memakai algoritma dan menggunakan *stack* (tumpukan). *Stack* adalah struktur untuk menyusun benda dengan menaruh dan mengambil elemen yang terakhir.

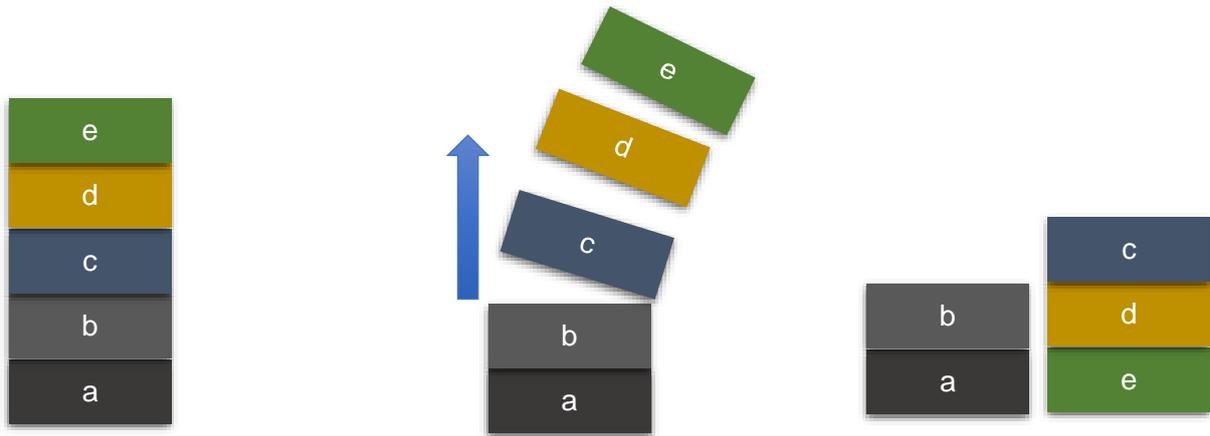
C. Penutup

Guru dan siswa melakukan refleksi bersama dari konsep representasi data tumpukan (*stack*) yang sudah dipelajari

2. Konsep terkait aktivitas

Karakteristik penting *stack* adalah bersifat LIFO (*Last In First Out*) artinya data yang terakhir masuk merupakan data yang akan keluar terlebih dahulu. Contoh sederhana adalah ketika memasukkan balok bertuliskan huruf alfabet secara urut dari a sampai e (a-b-c-d-e). Kemudian jika ingin mengambil alfabet b, maka huruf e, d dan c harus dikeluarkan terlebih dahulu secara berurut hingga akhirnya dapat mengakses b.

Contoh gambar tumpukan balok



Jenis-jenis Stack

Berdasarkan kemampuan menyimpan data, struktur data stack dapat dibagi menjadi 2 jenis, yaitu: **register stack** dan **memory stack**.

1. Register stack

Register stack merupakan stack yang hanya mampu menampung data dalam jumlah yang kecil. Kedalaman maksimum pada register stack cenderung dibatasi karena ukuran unit memorinya sangat kecil dibandingkan dengan memory stack.

2. Memory stack

Pada stack jenis ini, kedalaman dari stack cukup fleksibel dan mampu menangani dalam dalam skala yang lebih besar dibandingkan jenis sebelumnya.

Karakteristik Stack

Struktur data stack memiliki ciri sebagai berikut:

- Stack digunakan pada banyak algoritma yang berbeda seperti Tower of Hanoi, Tree traversal, rekursi dll.
- Stack diimplementasikan dengan struktur data array atau linked list.
- Mengikuti prinsip operasi Last In First Out, yaitu elemen yang dimasukkan pertama akan muncul terakhir dan sebaliknya.
- Penyisipan dan penghapusan terjadi di satu ujung yaitu dari atas tumpukan.
- Apabila ruang memori yang dialokasikan untuk struktur data stack sudah penuh namun masih dilakukan operasi penyisipan elemen maka akan terjadi stack overflow.
- Apabila struktur data tidak memiliki elemen data atau kosong, namun tetap dilakukan operasi penghapusan maka akan terjadi stack underflow

Operasi-operasi Dasar pada Stack

Ada beberapa operasi dasar yang bisa kita lakukan terhadap struktur data stack. Operasi-operasi tersebut meliputi

- **Push:** Menyisipkan elemen ke bagian atas stack
- **Pop:** Menghapus elemen atas dari stack
- **IsEmpty:** Memeriksa apakah stack kosong
- **IsFull:** Memeriksa apakah stack sudah penuh
- **Peek:** Mendapatkan nilai elemen teratas tanpa menghapusnya

Fungsi dan Kegunaan Stack

Adapun fungsi dan kegunaan struktur data stack adalah sebagai berikut:

- Struktur data stack digunakan dalam evaluasi dan konversi ekspresi aritmatika. Proses ini banyak dipakai untuk program kompiler.
- Stack digunakan dalam pemrograman rekursi.
- Digunakan untuk pemeriksaan tanda kurung.
- Stack digunakan dalam manajemen memori.
- Dipakai untuk memproses pemanggilan sebuah fungsi.

Salah satu contoh penerapan struktur data stack adalah fitur tombol back pada browser. Dimana browser akan menyimpan semua URL yang telah kita kunjungi sebelumnya dalam stack.

Setiap kali kita mengunjungi halaman baru, halaman itu ditambahkan di atas stack. Saat kita menekan tombol kembali, URL saat ini dihapus dari tumpukan, dan URL sebelumnya diakses.

Kelebihan Menggunakan Stack

Adapun kelebihan menggunakan struktur data stack di antaranya:

- **Manajemen data yang efisien:** Stack membantu mengelola data berdasarkan prinsip operasi LIFO yang tidak bisa dilakukan dengan linked list dan array.
- **Manajemen fungsi yang efisien:** Ketika suatu fungsi dipanggil, variabel lokal disimpan dalam stack, dan secara otomatis dihancurkan setelah dikembalikan.
- **Kontrol atas memori:** Stack memungkinkan kita untuk mengontrol bagaimana memori dialokasikan dan tidak dialokasikan.
- **Manajemen memori cerdas:** Stack secara otomatis membersihkan objek.
- **Tidak mudah rusak:** Stack tidak mudah rusak, oleh karena itu stack cenderung lebih aman dan dapat diandalkan.
- **Tidak mengizinkan perubahan ukuran variabel:** Variabel pada stack tidak dapat diubah ukurannya.

Kekurangan Menggunakan Stack

Selain kelebihan di atas, stack juga terdapat beberapa kelemahan berikut:

- **Ukuran memori terbatas:** Memori pada stack cukup terbatas.
- **Kemungkinan stack overflow:** Terlalu banyak membuat objek di stack dapat meningkatkan risiko stack overflow.
- **Akses acak tidak dimungkinkan:** Dalam stack, akses data secara acak tidak bisa dilakukan. Data yang dapat diakses adalah data yang berada pada elemen atas.
- **Dapat menyebabkan fungsi tidak terdefinisi:** Ketika penyimpanan variabel akan ditimpa, kadang-kadang akan menyebabkan perilaku fungsi atau program yang tidak terdefinisi.
- **Penghentian yang tidak diinginkan:** Jika stack berada di luar memori maka dapat menyebabkan penghentian yang tidak normal.

3. Lembar Kerja Siswa

Nama anggota kelompok

- 1.**
- 2.**
- 3.**
- 4.**

Setiap kelompok mendiskusikan topik diskusi tentang contoh soal representasi data *stack* yang sudah diberikan guru. Siswa membuat contoh kasus yang relevan dengan representasi data *stack*.

4. Asesmen

Asesmen siswa untuk unit pembelajaran Berpikir Komputasional dapat dilakukan dengan menilai:

- a. Penilaian kelompok
- b. Penilaian individu

Penilaian Kelompok

Indikator	Skor	Nilai
Mampu melakukan dekomposisi dari contoh gambar yang diberikan		A = skor 15-20 B = skor 9-14 C = skor < 9
Mampu melakukan abstraksi dari contoh gambar yang diberikan		
Mampu menyusun algoritma yang masuk akal dari contoh gambar yang diberikan		
Mampu menggambar sesuai algoritma yang sudah dibuat kelompok lain		
Mampu memamerkan hasil gambar dan algoritma yang sudah dibuat		

Kriteria Penilaian

Skor	Kriteria Penilaian
3	Sudah mampu, hasilnya jelas, sistematis, terstruktur
2	Sudah mampu walaupun belum jelas dan belum terstruktur
1	Belum mampu, masih harus dibimbing guru langkah per langkah

Penilaian Individu

Komponen Penilaian	A=Baik Sekali	B=Baik	C=Cukup	D=Kurang
Keaktifan siswa dalam diskusi kelompok	Siswa sangat aktif, dan bersemangat ketika bekerja dalam tim	Siswa bersemangat, tapi berpikir sedikit lambat ketika bekerja dalam tim	Siswa kurang aktif, sibuk yang lain, lebih banyak diam	Siswa pasif dan diam ketika bekerja dalam tim

Kreatif	Selalu memiliki ide kreatif yang orisinal dan berani menampilkan	Memiliki ide kreatif, namun kurang berani menampilkan	Masih suka melihat ide yang lain, sering ikut-ikutan saja	Monoton, tidak mau berpikir dengan ide sendiri
Bernalar Kritis	Dapat berpikir sesuai logika, sistematis	Sudah bisa berpikir masuk akal, namun belum sistematis	Belum bisa berpikir masuk akal	Malas berpikir

Mengetahui,

Kepala Sekolah
SMP IA Bagek Nyaka

Ahmad Yani S.Pd
NIP. 19740924 199903 1 002

Guru Mapel



Usman Irawan S.Pd