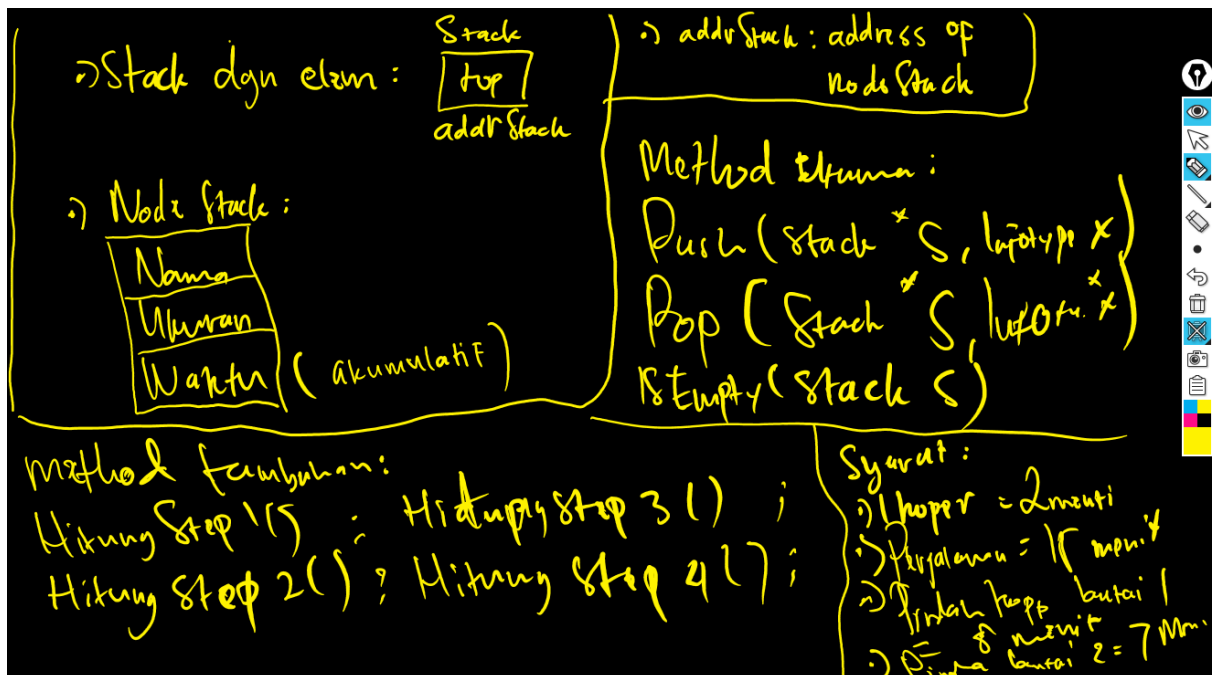
	Nama: Arsal Fadilah	Mata Kuliah: Struktur Data dan Algoritma
	NIM: 201524036 / 1B	Tanggal: 24 Juni 2021
	Prodi: D-IV Teknik Informatika	TUGAS

Pertanyaan Pada Kasus Tidak Terpilih

1. Apa nama Kasus yang tidak Saudara Pilih ?
Kasus 2 (waktu pemindahan koper)
2. Menurut Saudara ADT apa yang dapat digunakan pada kasus yang tidak terpilih?
Sertakan alasannya?
ADT Stack dengan alokasi dinamis yang menurut saya tepat untuk digunakan di kasus tersebut karena proses penyusunan koper akan ditumpuk dan dikeluarkan dari yang paling terakhir dimasukkan ke penyimpanan itu sama dengan konsep LIFO (last In First Out) ADT Stack pada umumnya. Kenapa dinamis ? karena jumlah koper tidak tentu, tergantung dari banyaknya koper yang dibawa oleh penumpang pada saat check-in.
3. Berdasarkan pertanyaan nomor 2, bagaimana penggunaan ADT pada kasus tersebut?
Hint: Jelaskan logic penggunaan ADT tersebut secara umum. Boleh menggunakan visualisasi.
Visualisasi :



Stack dgn elem: $\boxed{\text{top}}$
addr Stack

Node Stack:

Nama
Umuran
Waktu (akumulatif)

Method Eksternal:

- Push (Stack * S, info type x)
- Pop (Stack * S, info type x)
- IsEmpty (Stack S)

Method Tambahan:

- Hitung Step 1 (5) : Hitung step 3 (1) ;
- Hitung Step 2 (1) : Hitung step 4 (1) ;

Syarat:

- 1) Koper = 2 menit
- 2) Pergantian = 15 menit
- 3) Tidak perlu butai 1
- 4) 8 menit
- 5) Pindah butai 2 = 7 menit

Logik dari Method Utama:

- Push
Menambahkan elemen stack dengan posisi selalu berada di top
- Pop
Menghapus elemen stack dari posisi top. Jika top 1 maka stack kosong. Jika tidak 1 maka top diisi oleh element di next dari elemen top yang dihapus.
- isEmpty
Pengecekan apakah stack kosong atau tidak
- isFull
pengecekan apakah stack penuh atau tidak

Perhitungan kira-kira dari setiap step pada kasus tersebut :

- HitungStep1()=> masukPenyimpanan(Stack S, infotype X)
masukan setiap koper ke dalam penyimpanan sementara (Stack S) dengan infotype X sebanyak N penumpang.
- HitungStep2() => pecahStack(Stack S, Stack *s1, Stack *s2)
untuk dapat menghitung step 1 langkah-langkah yang dapat dilakukan adalah lakukan pengecekan untuk setiap elemen dari Stack S (penyimpanan sementara), jika ada elemen di S yang berukuran B di Push ke dalam Stack s1 (troli 1) atau jika ada elemen yang berukuran S maka di Push ke dalam Stack s2 (troli 2). Sebelum di push isikan waktu elemen tersebut 2 menit. kemudian totalkan lama proses step 1 dari banyaknya elemen Stack S di kali dengan 2 menit.
- HitungStep3() => copyStack(Stack *s1, Stack *s2)
setelah melakukan step 1 copy kan stack yang ada di (troli 1) ke stack (bagasi 1) dan untuk stack (troli 2) pindahkan ke (bagasi 2) kemudian tambahkan waktu untuk setiap elemen baik dari troli 1 atau troli 2 dengan waktu 15 menit.
- HitungStep4() => hitungJumlahWaktu(Stack S)
Mengkalkulasikan semua waktu yang ada di elemen stack S