Nama : Joseph Janone Tiwouw

Parallel : A

Lanjutan catatan stack pada hari senin,

STEP 4 Pop data dari stack

Algoritma dimulai dari

Step 1 : Print stack -> top -> bil

Step 2 : Update stack -> top setelah node berikutnya

Step 3 : Update stack -> count –

Ini adalah stack yang diimplementasikan dalam Bahasa C

Pada step yang pertama itu kita mendefinisikan 2 struktur yaitu struktur untuk data node dan struktur untuk head. Untuk informasi di dalam struktur node kita isi dengan informasi dengan link nya dan struktur yang kedua untuk head informasi yang disimpan didalam node nya adalah informasi tentang count dan informasi tentang node. Top diisi dengan alamat lokasi yang diisi dengan data node dengan tipe stack node.

Selanjutnya pada step kedua, kita membuat stack head, logikanya kita membuat 1 head saja, yang lebih dari satu adalah node. Dalam create stack head kita mengalokasikan dengan fungsi malloc untuk mengembalikan alamat dari lokasi ini, kita menyimpan alamat ini dalam varieabel dan di deklarasi secara lokal dalam stack head.

Setelah itu pada step ketiga,kita membuat data node, artinya kita push data kedalam stack, dan logika untuk push data ke dalam stack kita perlu informasi top nya, jadi push data memerlukan informasi dari count. Kita mengirim head ke push data agar bisa mengakses count dengan head. Data dalam node kita bisa kasih masuk lebih dari 1 data. Jika ingin masukan data 2 kali kita push data 2 kali. Kita panggil malloc untuk mengalokasikan 4. Dan kita mendklarasikan secara global diatas.

Akhirnya pada step keempat kita mendisplay semua data (pop), untuk popstack aka nada perulangan didalam stack, dan menampilkan semua data dalam stack dengan fungsi popstack, dan perunlangan akan dikasih masuk didalam fungsi popstack. Untuk akses pada count kita bisa akses dari head, krna sudah dideklarasikan di main.