Nama : Teguh Agung Prabowo

NIM : 512121230007

Jurusan : Sistem Informasi

Matkul : UTS - Struktur Data

1. Tipe data atomik, adalah tipe data yg dipandang sebagai satu kesatuan tunggal dan tidak dapat dipecah-pecah lagi (non decomposible entity).Contoh : Integer, Char, float/real.

Tipe data berstruktur, adalah tipe data yang dipandang sebagai satu kesatuan tunggal dan dapat dipecah-pecah lagi (decomposible entity).Contoh : Array, Structure, dll.

2. Diketahui :

char nim[10]

char nama[20]

char alamat[30]

int semester

float ipk

Base address (b) = 2000

tipe char butuh 1 byte

tipe int butuh 2 byte

float butuh 4 byte

- Tentukan kapasitas memory untuk menampung 1 record mhs.

10 + 20 + 30 + 2 + 4 = 66 byte

- Tentukan kapasitas memory untuk menampung variabel X.

X [100]

base location + offset

2000 + (100*66) = 8600

- Tentukan Address :

Addr (X[10]) = 2000 + (10*66) = 2660

Addr (X[20].nama) = 2000 + (20*66 + 10) = 3330

Addr (X[30].alamat) = 2000 + (30*66 + 10 + 20) = 4010

Addr (X[40].ipk) = 2000 + (40*66 + 10 + 20 + 30 + 2) = 4702

Addr (X[50].nim) = 2000 + (50*66 + 0) = 5300

3. Membuat stack pada cpp :

```
#include
#define MAX 10
using namespace std;

//Deklarasi struct tumpukan
struct Stack {
int top, data[MAX];
}Tumpukan;

void init(){
Tumpukan.top = -1;
}

bool isEmpty() {
return Tumpukan.top == -1;
}

bool isFull() {
return Tumpukan.top == MAX-1;
}

void push() {
if (isFull()) {
cout << "\nTumpukan penuh"<< endl;
```

```
 }
else {
Tumpukan.top++;
cout << "\nMasukkan data = "; cin >> Tumpukan.data[Tumpukan.top];
cout << "Data " << Tumpukan.data[Tumpukan.top] << " masuk ke stack"< }
}

void pop() {
if (isEmpty()) {
cout << "\nData kosong\n"< }
else {
cout << "\nData "< Tumpukan.top--;
 }
}

void printStack() {
if (isEmpty()) {
cout << "Tumpukan kosong";
 }
else {
cout << "\nTumpukan : ";
for (int i = Tumpukan.top; i >= 0; i--)
cout << Tumpukan.data[i] << ((i == 0) ? "" : ",");
 }
}

int main() {
int pilihan;
init();
do {
```

```
printStack();
cout << "\n1. Input (Push)\n"
<<"2. Hapus (Pop)\n"
<<"3. Keluar\n"
<<"Masukkan Pilihan: ";
cin >> pilihan;
switch (pilihan)
{
case 1:
push();
break;
case 2:
pop();
break;
default:
cout << "Pilihan tidak tersedia" << endl;
break;
  }
 }while (pilihan!=3); }
```

4. Membuat notasi prefix, postfix dan infix

a. (A + B) * C – D * E => notasi infix

- prefix => *+AB*C-DE
- postfix => AB+C-D*E*

b. + * - A B C * D E => notasi prefix

- infix => (((A-B)*C)+(D*E))
- postfix => AB-C*DE*+

c. A B + C * D E * + => notasi postfix

- infix => (((A+B)*C)+(D*E))
- prefix = > +*+ABC*DE