

Praktikum Modul  
Ke-9  
Stack dan Queue

[ Tugas Terbimbing ]

Max Point : 70

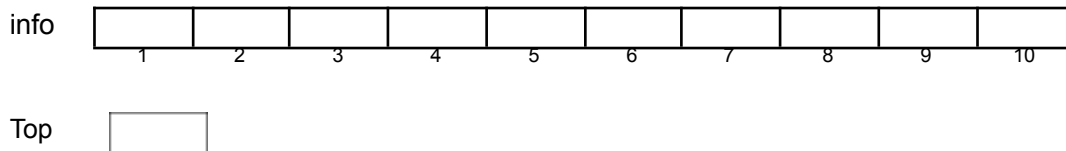
Bagian I: Stack

[POINT : 5]

Dalam praktikum ini Anda diminta untuk membuat stack dengan representasi statis, yaitu menggunakan array. Untuk itu buatlah ADT stack pada file “stack.h” dengan struktur berikut:

Notasi Algoritmik	Bahasa C++
<pre>type infotype : integer type stack : &lt;     info : array[1..10] of         integer, top : integer &gt;</pre>	<pre>typedef int infotype;  struct stack {     infotype info[10];     int Top; };</pre>

Ilustrasi untuk struktur di atas adalah:



Elemen info[1..10] akan digunakan untuk menampung data bilangan bulat. Sedangkan Top akan digunakan untuk menampung informasi mengenai indeks elemen info yang paling atas. Jadi, Top = 0 artinya stack dalam keadaan kosong. Top = 1, artinya stack berisi satu elemen.

Prosedur dan fungsi yang digunakan adalah

Notasi Algoritmik	Bahasa C++
<pre>procedure createStack(input/output S : stack) { I.S. sembarang   F.S. terbentuk stack dengan Top = 0 }</pre>	<pre>void createStack(stack &amp;S);</pre>
<pre>function isEmpty(S : stack): boolean { Mengembalikan nilai true jika stack kosong }</pre>	<pre>bool isEmpty(stack S);</pre>
<pre>function isFull(S : stack): boolean { Mengembalikan nilai true jika stack penuh }</pre>	<pre>bool isFull(stack S);</pre>

<u>procedure</u> push( <u>input/output</u> S : stack, <u>input</u> x : <u>infotype</u> ) { I.S. mungkin kosong F.S. menambahkan elemen pada stack dengan nilai x, Top = Top + 1 }	<b>void</b> push( <b>stack</b> &S, <b>infotype</b> x);
<u>function</u> pop( <u>input/output</u> S : stack): <u>infotype</u> { Mengembalikan nilai pada indeks Top, Top = Top - 1 }	<b>int</b> pop( <b>stack</b> &S);
<u>procedure</u> printInfo( <u>input</u> S : stack) { I.S. stack mungkin kosong F.S. Jika stack tidak kosong, maka menampilkan semua info yang ada pada stack }	<b>void</b> printInfo( <b>stack</b> S);

Gunakan alias sebagai berikut:

Top(S) untuk  
 (S).Top info(S)  
 untuk (S).info

Buatlah implementasi ADT stack pada file “stack.cpp”.

Notasi Algoritmik	Bahasa C++
<p>[POINT : 8]</p> <p><u>procedure</u> createStack(<u>input/output</u> S : stack)  { I.S. sembarang  F.S. terbentuk stack dengan Top  = 0 } Kamus:  { Tidak ada  kamus } Algoritma:  Top(S) ← 0</p>	<pre>void createStack(stack &amp;S) {     /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<p>[POINT : 9]</p> <p><u>function</u> isEmpty(S : stack): <u>boolean</u>  { Mengembalikan nilai true jika stack  kosong } Kamus:  { Tidak ada  kamus } Algoritma:  if Top(S) = 0 <u>then</u>  → <u>true</u>  <u>else</u>  → <u>false</u>  { <u>end if</u> }</p>	<pre>bool isEmpty(stack S) {     /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<p>[POINT : 9]</p> <p><u>function</u> isFull(S : stack): <u>boolean</u>  { Mengembalikan nilai true jika stack  penuh } Kamus:  { Tidak ada  kamus } Algoritma:  if Top(S) = 10 <u>then</u>  → <u>true</u>  <u>else</u>  → <u>false</u>  { <u>end if</u> }</p>	<pre>bool isFull(stack S) {     /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
-------------------	------------

[POINT : 10]

procedure push(input/output S : stack, input x : infotype)  
{ I.S. mungkin kosong  
F.S. menambahkan elemen pada stack dengan  
nilai x, Top = Top + 1}  
Kamus:  
{ Tidak ada  
kamus } Algoritma:  
  if isFull(S) = false then  
    Top(S) ← Top(S)  
    + 1 Info(S)[Top(S)]  
    ← x  
  end if

```
void push(stack &S, infotype x) {  
  
    /* Lengkapi kodenya */  
  
}
```

Notasi Algoritmik	Bahasa C++
----------------------	---------------

<p>[POINT : 10]</p> <p><u>function</u> pop(<u>input/output</u> S : stack): infotype  { Mengembalikan nilai infotype yang ada pada indeks Top, Top = Top - 1 }  Kamus:  x :  infotype  Algoritma:  x ← info(S)[Top(S)]  Top(S) ← Top(S) - 1  → x</p>	<pre>int pop(stack &amp;S) {      /* Lengkapi kodenya */  }</pre>
---	---

Notasi Algoritmik	Bahasa C++
<p>[POINT : 9]</p> <p><u>procedure</u> printInfo(<u>input</u> S : stack)  { I.S. stack mungkin kosong  F.S. Jika stack tidak kosong, maka menampilkan semua info yang ada pada stack }</p> <p>Kamus:  i :  integer  Algoritma:  for i ← Top(S) <u>downto</u> 1  do <u>output</u>(info(S)[i])  { end for }</p>	<pre>void printInfo(stack S) {      /* Lengkapi kodenya */  }</pre>

[POINT : 5]

Untuk menguji implementasi stack buatlah kode pada “main.cpp” sebagai berikut (Silakan copy paste):

```
int main() {  
    stack S;  
    createStack(S);  
    push(S, 2);  
    push(S, 3);  
    push(S, 4);  
    push(S, 5);  
    printInfo(S);  
    pop(S);  
    printInfo(S);  
  
    return 0;  
}
```

Output yang diharapkan:

```
5 4 3 2  
4 3 2
```

Selanjutnya buatlah procedure ascending dan procedure descending yang masing-masing membuat info stack terurut menaik dan menurun.

[POINT : 5]

Untuk menguji implementasi stack buatlah kode pada “main.cpp” sebagai berikut (Silakan copy paste):

```
int
main
() {
    stack
    S;
    createStack
    (S); push(S,
    12);
    push(S, 17);
    push(S, 5);
    push(S, 10);
    push(S, 15);
    push(S, 25);
    push(S, 11);
    push(S, 22);
    push(S,
    19);
    printInfo(S);
    ascending(
    S);
    printInfo(S);
    descending
    (S);
    printInfo(S);

    return 0;
}
```

Output yang diharapkan:

```
19 22 11 25 15 10 5 17 12
25 22 19 17 15 12 11 10 5
5 10 11 12 15 17 19 22 25
```

## [ Tugas Mandiri ]

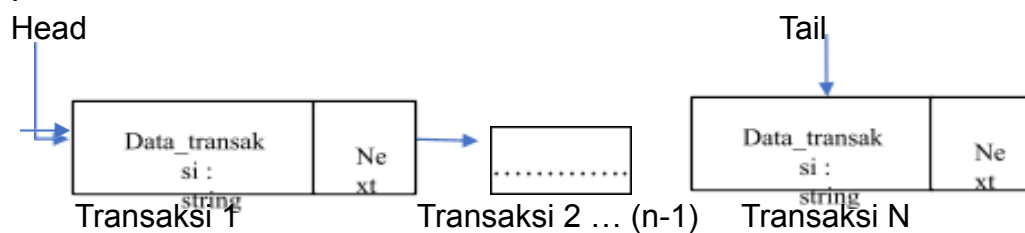
Max Point : 30

### Bagian II: Queue

Blockchain adalah teknologi yang sedang tren saat ini. Bitcoin adalah sistem yang menggunakan blockchain sebagai penyimpanannya. Block chain sebenarnya adalah teknologi yang memanfaatkan prinsip link list sebagai penyimpanan datanya namun digabungkan teknologi kriptografi untuk melindungi datanya.

Sebelum masuk kedalam blockchain, data masuk kedalam pool data sebelum data diproses dan disimpan **menggunakan prinsip queue**. Dalam pool data blockchain, transaksi **terbaru disimpan di akhir** dan transaksi akan **diproses dari depan/awal** ( FIFO First in First Out )

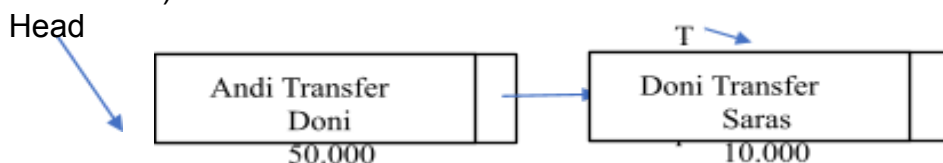
Buatlah simple pool blockchain sederhana berfungsi untuk mengolah data transaksi sebuah bank kecil. Berikut ini adalah struktur pool blockchain yang harus kalian buat :



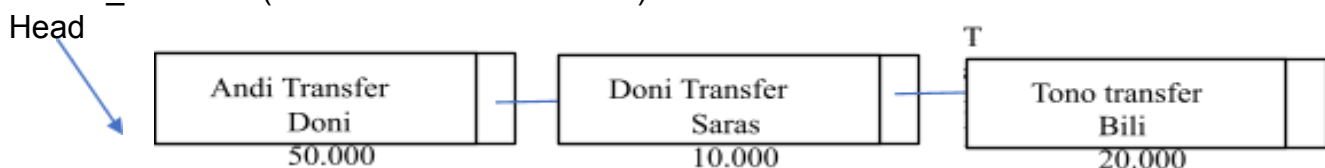
Ada 2 prosedur penting dalam pool blockchain yang berbasis queue yaitu masukkan\_transaksi ( sama dengan prosedur enqueue ) dan proses\_transaksi ( sama dengan prosedur dequeue ).

Contoh

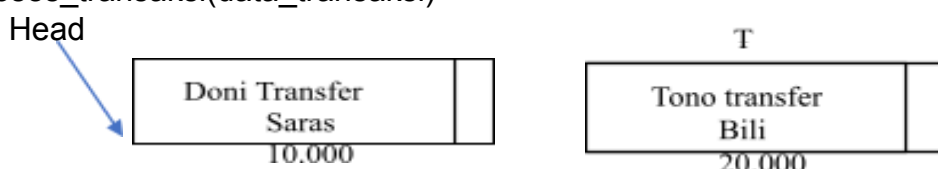
Masukkan\_transaksi ( " Andi transfer doni 50.000" ) Masukkan\_transaksi ( "Doni Trasnfer Saras 20.000" ) Hasil



Masukkan\_transaksi ( "Tono Transfer Bili 20.000" )



Proses\_transaksi(data\_transaksi)





Isi data\_transaksi adalah "Andi Transfer Doni 50.000"

## SOAL

[POINT : 10]

- a. Deklarasi dari ADT queue (LIST) di atas dalam C++(enqueue, dequeue dan printQueue).

[POINT : 20]

- b. Buat prosedur masukan\_transaksi, proses\_transaksi, dan main program seperti contoh yang diberikan, dan tampilkan isi queue.