

Soal Assessment CLO 2 [Stack dan Queue]

CCH1D4 - Struktur Data

A. STACK

Buatlah sebuah project code::block, kemudian lengkapi dengan implementasi dari spesifikasi ADT stack berikut ini!

```
constant nMax integer = 5
type infotype : integer
type stack <
    info : array [0..nMax-1] of infotype,
    top : integer >

procedure createStack(output S:stack)
{ IS. –
  FS. terbentuk sebuah stack kosong S}

procedure push(input/output S:stack, input x:infotype)
{ IS. terdefinisi sebuah Stack S (mungkin penuh) dan infotype x
  FS. x ditambahkan ke dalam stack S}

Procedure pop(input/output S:stack, output x:infotype)
{ IS. terdefinisi sebuah Stack S (mungkin kosong)
  FS. elemen top dari stack dihapus dan nilainya disalin ke x, nilai top berubah}

Procedure printStack(input S: stack)
{ IS. terdefinisi sebuah Stack S (mungkin kosong)
  FS. isi dari stack ditampilkan ke layar
  Catatan: hanya diperbolehkan operasi loop, ifthen, pop, push dan boleh menambahkan stack temporari sebagai bantuan}

Procedure deleteData(input/output S:stack, input x:infotype)
{ IS. terdefinisi sebuah Stack S (mungkin kosong atau penuh) dan infotype x
  FS. mencari x dari dalam Stack S, kemudian hapus x tersebut dari S, tampilkan string "delete berhasil" apabila x ditemukan, atau "data tidak ditemukan" apabila sebaliknya.
  Catatan: hanya diperbolehkan operasi loop, ifthen, pop, push dan boleh menambahkan stack temporari sebagai bantuan}
```

Kemudian buatlah main program berikut ini!

```
int main()
{
    char NIM[11]= "1234567890";
    char Nama[50] = "Praktikan";
    char Kelas[20] = "XX-43-YY";
    cout << "-----"<<endl;
    cout << " Assessment CLO 2 - STACK dan QUEUE"<<endl;
    cout << "-----"<<endl;
    cout << " NIM : "<< NIM << "\n Nama : "<< Nama << "\n Kelas : "<< Kelas <<endl;
    cout << "-----"<<endl;
    stack S;
    createStack(S);
    printStack(S);

    push(S,43);
    push(S,430);
    push(S,4300);
    push(S,43000);
    push(S,430000);
    push(S,4300000);
    printStack(S);
    cout<< "\n hapus data 123 "<<endl;
```

Soal Assessment CLO 2 [Stack dan Queue]

CCH1D4 - Struktur Data

```
infotype y = 123;
deleteData(S,y);
cout<<endl;
printStack(S);
cout<<"\n hapus data 4300 "<<endl;
y = 4300;
deleteData(S,y);
cout<<endl;
printStack(S);
cout<<"\n hapus data 43 "<<endl;
y = 43;
deleteData(S,y);
cout<<endl;
printStack(S);
return 0;
}
```

```
-----
Assessment CLO 2 - STACK dan QUEUE
-----
NIM : 1234567890
Nama : Praktikan
Kelas : XX-43-YY
-----
(top)
(top) 430000 | 43000 | 4300 | 430 | 43 |

hapus data 123
data tidak ditemukan
(top) 430000 | 43000 | 4300 | 430 | 43 |

hapus data 4300
delete berhasil
(top) 430000 | 43000 | 430 | 43 |

hapus data 43
delete berhasil
(top) 430000 | 43000 | 430 |

Process returned 0 (0x0)   execution time : 0.113 s
Press any key to continue.
```

Gambar 1 Hasil running program apabila dijalankan

B. QUEUE

Secara sederhana konsep pengiriman data di dalam jaringan komputer adalah memecah data-data tersebut menjadi paket-paket data dengan ukuran yang lebih kecil. Kemudian paket-paket data dikirim secara serial sesuai antrian di dalam jaringan komputer. Setelah sampai di komputer tujuan maka paket-paket data tersebut disusun kembali menjadi data-data yang utuh.

File: Z No: 1 Ukuran: 2	File: A No: 1 Ukuran: 8	File: A No: 2 Ukuran: 8	File: A No: 3 Ukuran: 2	File: E No: 1 Ukuran: 8	File: E No: 2 Ukuran: 6	...
0	1	2	3	4	5	...

Gambar 2 ilustrasi paket file di dalam Queue

Buatlah sebuah project code::block yang merupakan simulasi antrian paket dalam pengiriman data sesuai dengan petunjuk berikut ini!

Soal Assessment CLO 2 [Stack dan Queue]

CCH1D4 - Struktur Data

1. Buatlah ADT Queue dengan Array Alternatif III (Circular)
2. Kapasitas Queue adalah 10 elemen atau paket.
3. Satuan ukuran data yang digunakan adalah byte. Asumsikan byte sebagai tipe data integer.
4. Setiap elemen array dari queue hanya bisa menampung maksimal 8 byte data (ukuran paket data).
5. infotype dari elemen queue adalah nama file dan nomor urut paket dari file tersebut dan ukuran file dalam paket (maksimal 8).
6. Lengkapi ADT tersebut dengan subprogram berikut ini

```
void createQueue(Queue &Q);
/* IS. -
   FS. terbentuk suatu Queue kosong di mana head dan tail bernilai -1 */

bool isFull(Queue Q);
/* return TRUE apabila queue penuh, atau FALSE apabila sebaliknya */

bool isEmpty(Queue Q);
/* return TRUE apabila queue kosong, atau FALSE apabila sebaliknya */

void Add(Queue &Q, int ukuran, int no, string filename);
/* IS. terdefinisi sebuah Queue Q (mungkin penuh) dan sebuah paket baru yang memiliki filename, no urut,
   dan ukuran paket
   FS. data paket baru tersebut di-EnQueue ke dalam Queue Q */

void Del(Queue &Q, int &ukuran, int &no, string &filename);
/* IS. terdefinisi sebuah Queue Q (mungkin kosong)
   FS. filename, no dan ukuran berisi data hasil DeQueue dari Queue Q */

void printQueue(Queue Q);
/* IS. terdefinisi sebuah Queue Q (mungkin kosong)
   FS. menampilkan head dan tail dari Queue, kemudian isi dari Queue Q , perhatikan contoh ilustrasi gambar
   hasil running */

void kirimData(Queue &Q, int fileSize, string filename);
/* IS. terdefinisi sebuah Queue Q (mungkin penuh) dan fileSize yang menyatakan ukuran data dari filename
   yang akan dikirim
   FS. data dipecah menjadi beberapa paket dan dimasukkan ke Add ke dalam Queue, Tampilkan "berhasil"
   apabila semua paket berhasil masuk ke dalam Queue, atau "Queue Penuh" apabila Q penuh atau hanya
   sebagian paket yang berhasil di Add ke dalam Queue */
```

7. Kemudian buatlah main program berikut ini!

```
void main(){
    string NIM = "1234567890";
    string Nama = "Praktikan";
    string Kelas = "XX-43-YY";
    cout << " -----" << endl;
    cout << " Assessment CLO 2 - STACK dan QUEUE" << endl;
    cout << " -----" << endl;
    cout << " NIM : " << NIM << "\n Nama : " << Nama << "\n Kelas : " << Kelas << endl;
    cout << " -----" << endl;
    Queue Q; int ukuran; int no; string file;
    createQueue(Q);
    printQueue(Q);
    cout << "\nadd A ukuran 20" << endl;
    kirimData(Q,20,"A");
    printQueue(Q);
    cout << "\nadd C ukuran 5" << endl;
    kirimData(Q,5,"C");
    printQueue(Q);
    cout << "\nadd E ukuran 16" << endl;
    kirimData(Q,16,"E");
```

Soal Assessment CLO 2 [Stack dan Queue]

CCH1D4 - Struktur Data

```
printQueue(Q);
cout<<"\nadd X ukuran 13"<<endl;
 kirimData(Q,13,"X");
printQueue(Q);
cout<<"\nadd B ukuran 50"<<endl;
 kirimData(Q,50,"B");
printQueue(Q);
cout<<"\ndel Queue 1"<<endl;
 Del(Q,ukuran,no,file);
cout<<"del Queue 2"<<endl;
 Del(Q,ukuran,no,file);
cout<<"del Queue 3"<<endl;
 Del(Q,ukuran,no,file);
cout<<"del Queue 4"<<endl;
 Del(Q,ukuran,no,file);
cout<<"del Queue 5"<<endl;
 Del(Q,ukuran,no,file);
cout<<"del Queue 6"<<endl;
 Del(Q,ukuran,no,file);
cout<<"del Queue 7"<<endl;
 Del(Q,ukuran,no,file);
printQueue(Q);
cout<<"\nadd K ukuran 20"<<endl;
 kirimData(Q,20,"K");
printQueue(Q);
cout<<"\nadd L ukuran 5"<<endl;
 kirimData(Q,5,"L");
printQueue(Q);
cout<<"\nadd M ukuran 16"<<endl;
 kirimData(Q,16,"M");
printQueue(Q);
cout<<"\nadd N ukuran 13"<<endl;
 kirimData(Q,13,"N");
printQueue(Q);
}
```

Soal Assessment CLO 2 [Stack dan Queue]

CCH1D4 - Struktur Data

```
-----
Assessment CLO 2 - STACK dan QUEUE
-----
NIM : 1234567890
Nama : Praktikan
Kelas : XX-43-YY
-----
Head :-1 Tail: -1 >>

add A ukuran 20
Berhasil
Head :0 Tail: 2 >> A;1;8 | A;2;8 | A;3;4 |

add C ukuran 5
Berhasil
Head :0 Tail: 3 >> A;1;8 | A;2;8 | A;3;4 | C;1;5 |

add E ukuran 16
Berhasil
Head :0 Tail: 5 >> A;1;8 | A;2;8 | A;3;4 | C;1;5 | E;1;8 | E;2;8 |

add X ukuran 13
Berhasil
Head :0 Tail: 7 >> A;1;8 | A;2;8 | A;3;4 | C;1;5 | E;1;8 | E;2;8 | X;1;8 | X;2;5 |

add B ukuran 50
Queue Penuh
Head :0 Tail: 9 >> A;1;8 | A;2;8 | A;3;4 | C;1;5 | E;1;8 | E;2;8 | X;1;8 | X;2;5 | B;1;8 | B;2;8 |

del Queue 1
del Queue 2
del Queue 3
del Queue 4
del Queue 5
del Queue 6
del Queue 7
Head :7 Tail: 9 >> X;2;5 | B;1;8 | B;2;8 |

add K ukuran 20
Berhasil
Head :7 Tail: 2 >> X;2;5 | B;1;8 | B;2;8 | K;1;8 | K;2;8 | K;3;4 |

add L ukuran 5
Berhasil
Head :7 Tail: 3 >> X;2;5 | B;1;8 | B;2;8 | K;1;8 | K;2;8 | K;3;4 | L;1;5 |

add M ukuran 16
Berhasil
Head :7 Tail: 5 >> X;2;5 | B;1;8 | B;2;8 | K;1;8 | K;2;8 | K;3;4 | L;1;5 | M;1;8 | M;2;8 |

add N ukuran 13
Queue Penuh
Head :7 Tail: 6 >> X;2;5 | B;1;8 | B;2;8 | K;1;8 | K;2;8 | K;3;4 | L;1;5 | M;1;8 | M;2;8 | N;1;8 |

Process returned 0 (0x0)   execution time : 0.167 s
Press any key to continue.
```

Gambar 3 Hasil running program apabila dijalankan