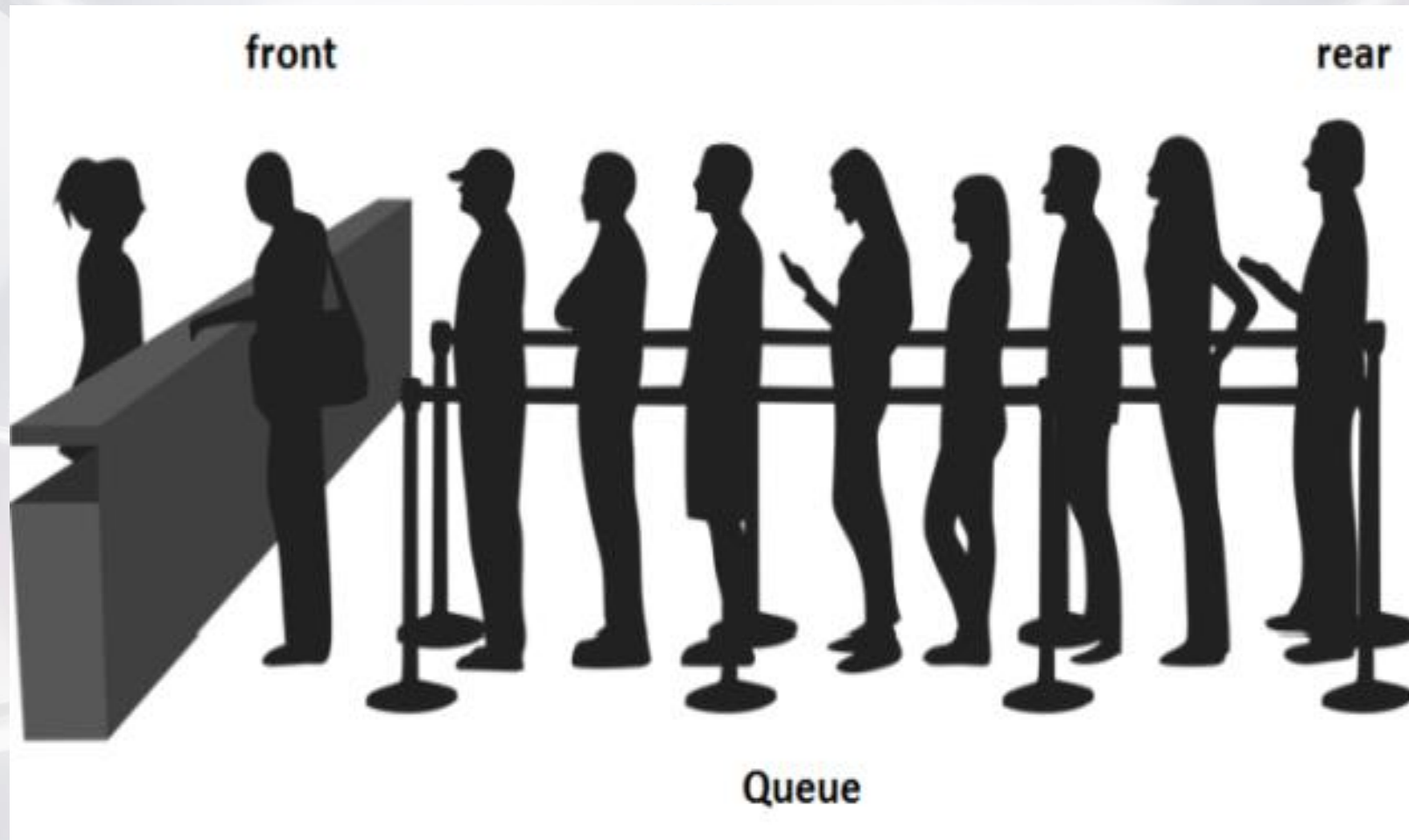


Struktur Antrian



Struktur Data

Fitri Nuraeni, M.Kom

<http://latihan.nufi3.com>

Reguler Pagi 2020

Struktur Antrian (Queue)

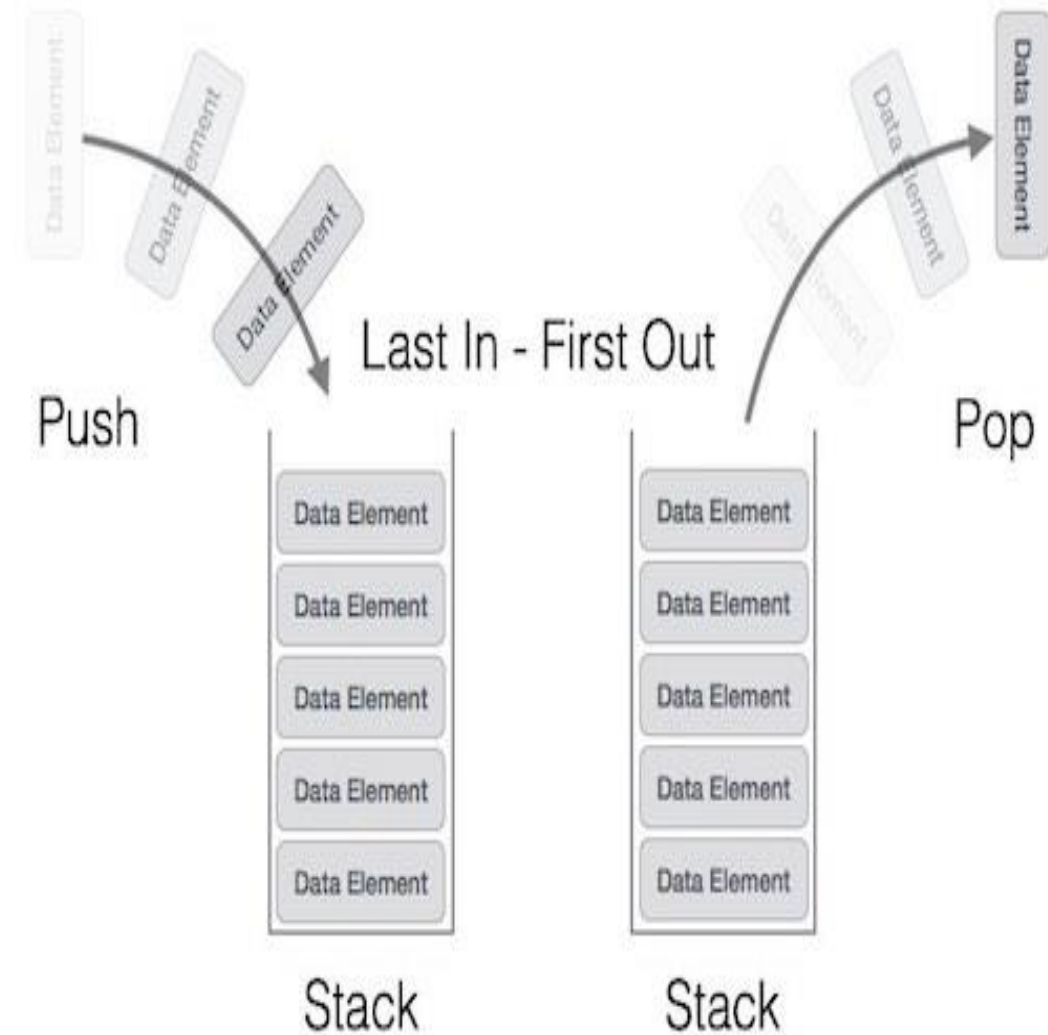
Pertemuan ke-12



Kilas Balik Pertemuan Terakhir

Pertemuan Ke-10 : Struktur Tumpukan (Stack)

- Stack adalah?
- Konsep Utama Stack?
- Operasi pada Struktur Stack:
 - PUSH
 - POP
 - Tambahan
 - Inisialisasi
 - cekKosong
 - cekPenuh
 - cariTop
- Masalah terakhir:
 - Bagaimana implementasi pengambil data pada stack yang bukan elemen TOP dengan bahasa C++?



Prosedur POP bukan elemen TOP/ puncak

```
93 void popStackNotTop(int posisi){
94     //siapkan Stack lain untuk penyimpanan elemen sementara
95     int temp[n];
96     int id = 0;
97     while(getTop() > posisi){
98         if(isKosong() == false){
99             //pindahkan elemen puncak sampai sebelum posisi ke Stack temp;
100             temp[id] = S[Top];
101             id++;
102             cout<<S[Top]<<" telah diambil dari Stack S ";
103
104             S[Top] = 0;
105             Top -= 1;
106             cout<<" | Top = "<<getTop()<<" \n";
107         }else{
108             cout<<"Stack S KOSONG | Top = "<<getTop()<<" \n";
109         }
110     }
111
112     //diambil elemen pada posisi
113     if(getTop() == posisi){
114         popStack();
115     }
116
117     //push elemen kembali ke Stack S
118     id -= 1;
119     while(id >= 0){
120         pushStack(temp[id]);
121         id--;
122     }
123 }
```

Prosedur Pencarian Linear

```
79 ▾ int cariAngka(int cari){  
80     int id = 0;  
81     int posisi = -1;  
82 ▾ while(posisi < 0 && id < n ){  
83 ▾     if(S[id] == cari){  
84         posisi = id;  
85 ▾     }else{  
86         id++;  
87     }  
88     }  
89     return posisi;  
91 }
```

Linear Search

Main Program

```
125 int main(){
126     int x;
127     char pop;
128
129     buatKosong();
130     tampilkanStack();
131
132     x = 0;
133
134     //isi Stack sampai PENUH
135     while(isPenuh() == false){
136         cout<<"Masukan angka (0-99) : ";
137         cin>>x;
138         pushStack(x);
139     }
140
141     tampilkanStack();
142
143     //minta angka yg akan diambil
144     cout<<"Masukan angka yang akan diambil dari Stack : ";
145     cin>>x;
146     int posisi = cariAngka(x);
147
148     if(posisi >=0){ //ditemukan pada Stack
149         popStackNotTop(posisi);
150     }else{
151         cout<<x<<" bukan elemen Stack\n";
152     }
153
154     tampilkanStack();
155
156     return 0;
157 }
```

Update bagian Main()

Demo Program

OnlineGDB beta
online compiler and debugger for c/c++

Welcome, Fitri Nuraeni

StrukturData_Stack

Create New Project

My Projects

Classroom **new**

Learn Programming

Programming Questions

Logout

Remote Ruby Podcast Episode #79 "Railsconf Couch Edition" ads served ethically

About • FAQ • Blog • Terms of Use • Contact Us
• GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online

main.cpp

```
1 #include <iostream>
2 #define n 10
3
4 using namespace std;
5
6 //global variabel
7 int Top;
8 int S[n];
9
10 void buatKosong(){
11     Top = -1;
12 }
13
14 bool isKosong(){
15     if(Top < 0){
16         return true;
17     }else{
18         return false;
19     }
20 }
21
22 bool isPenuh(){
```

input

Command line arguments:

Standard Input: ☒ Interactive Console ☐ Text

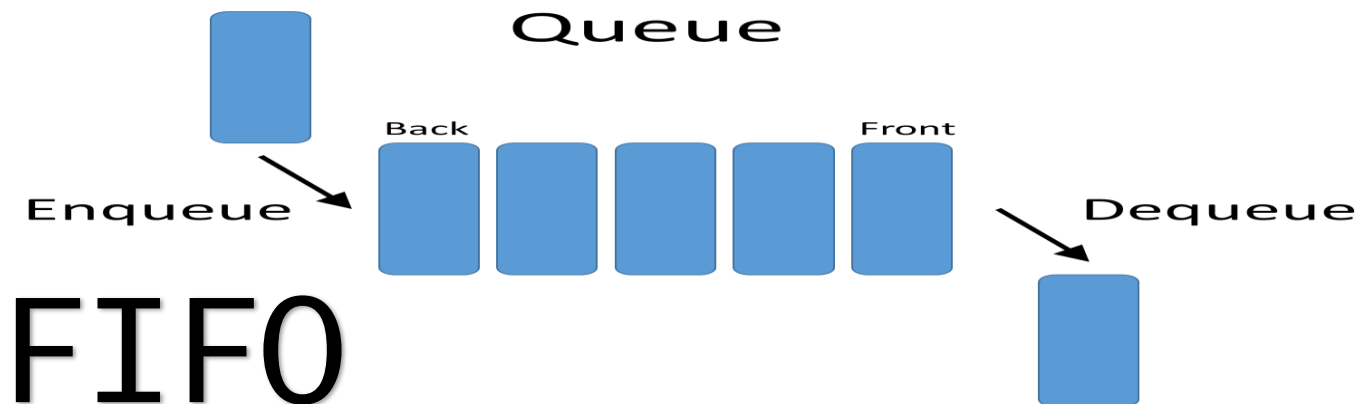
20.24
16/06/2020

POP bukan elemen TOP Stack

Struktur Queue

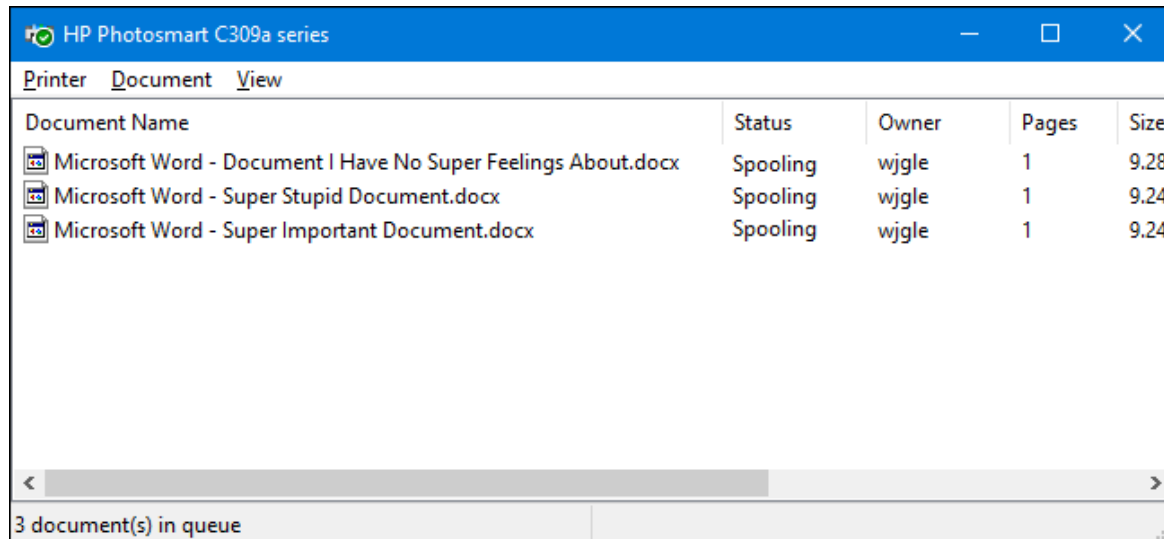
Pengertian

- Antrean adalah himpunan elemen bertipe sama dimana:
 - elemen ditambahkan pada bagian akhir yang dinamakan dengan **belakang/ back/ rear**.
 - Elemen dihapus/ diambil dari bagian lainnya yang dinamakan dengan **depan/ front**.
- Sama seperti pada dunia nyata, saat nasabah menunggu untuk dilayani oleh petugas bank. Setiap nasabah yang baru datang harus mengantri dibagian belakang nasabah terakhir. Sedangkan petugas bank akan melayani nasabah yang berada diurutan depan terlebih dahulu.



Penggunaan Antrian (Queue)

- Antrian digunakan ketika hal-hal tidak harus segera diproses, tetapi harus diproses dalam urutan **First In First Out**.
 - 1) Ketika sumber daya dibagi di antara banyak konsumen. Contohnya termasuk penjadwalan CPU, Penjadwalan Disk.
 - 2) Ketika data ditransfer secara tidak sinkron (data tidak harus diterima dengan kecepatan yang sama dengan yang dikirim) antara dua proses. Contohnya termasuk IO Buffer, pipa, file IO, dll.

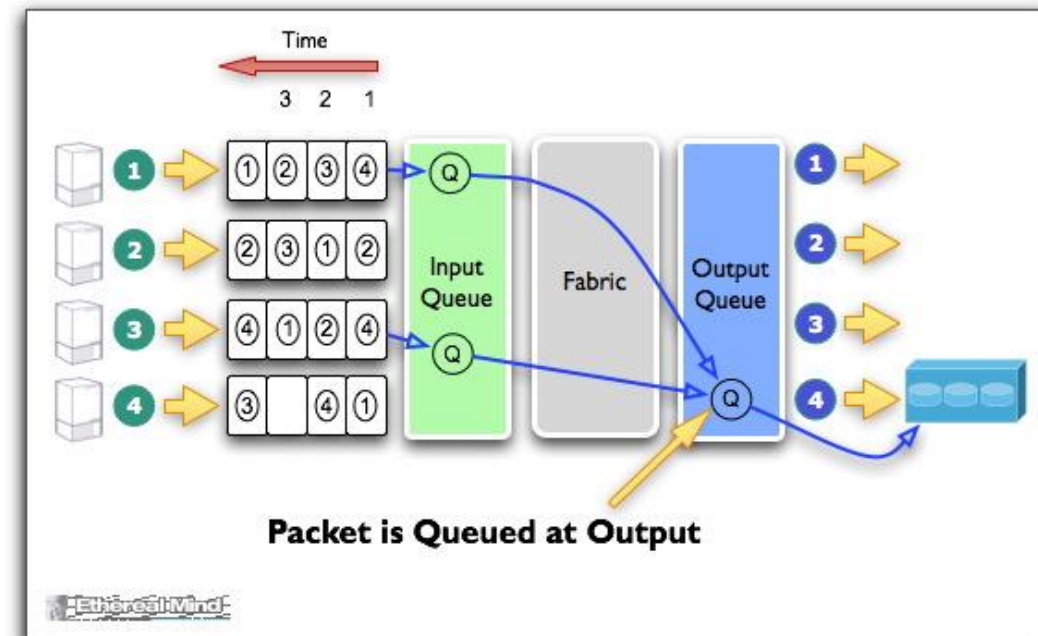
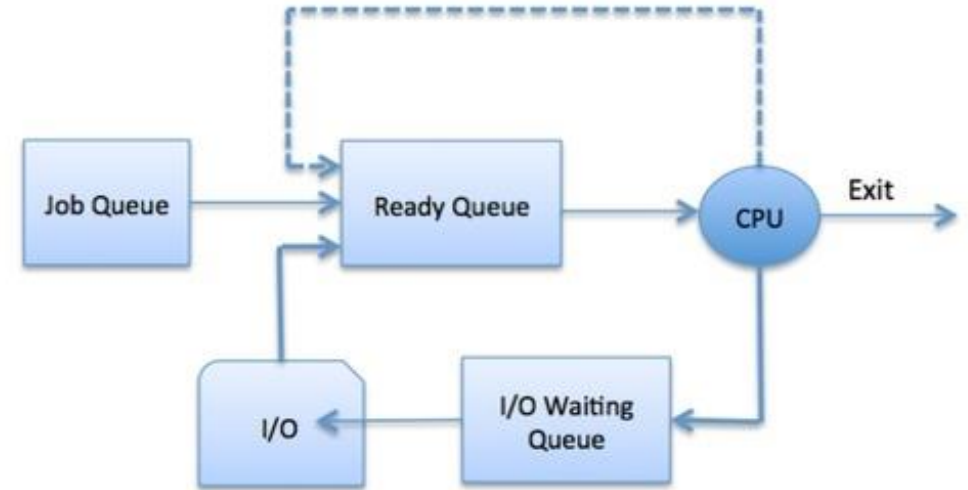


HP Photosmart C309a series

Printer Document View

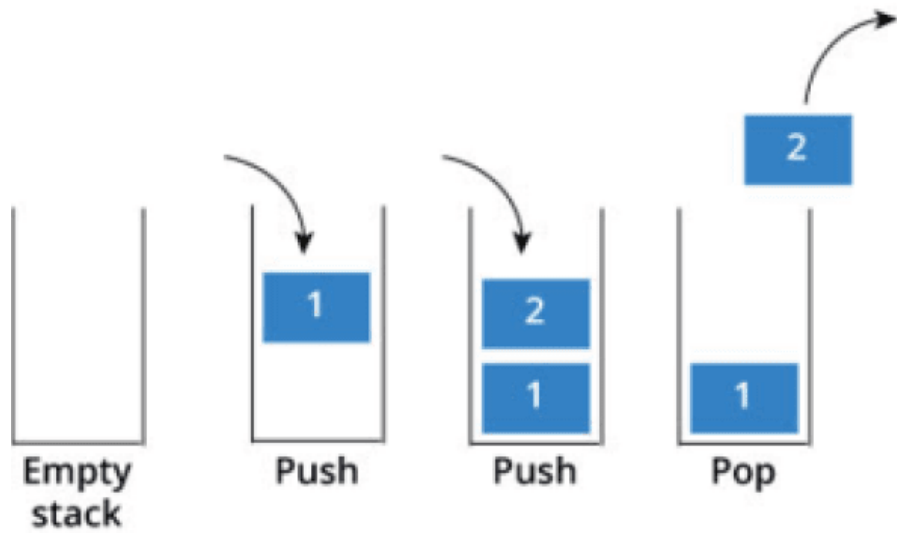
Document Name	Status	Owner	Pages	Size
Microsoft Word - Document I Have No Super Feelings About.docx	Spooling	wjgle	1	9.28
Microsoft Word - Super Stupid Document.docx	Spooling	wjgle	1	9.24
Microsoft Word - Super Important Document.docx	Spooling	wjgle	1	9.24

3 document(s) in queue



Stack Vs Queue

Perbandingan



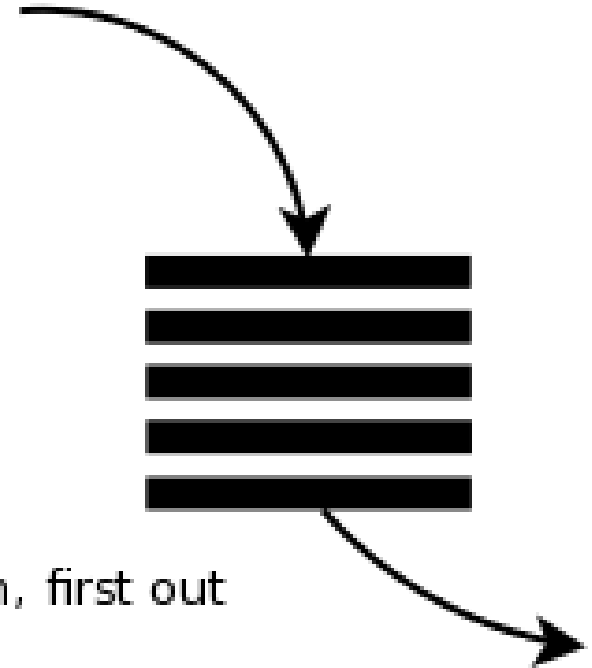
Stack



Queue

Operasi Pada Antrian

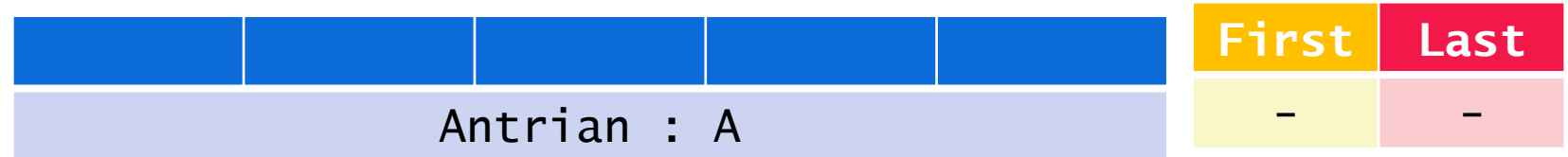
Queue:



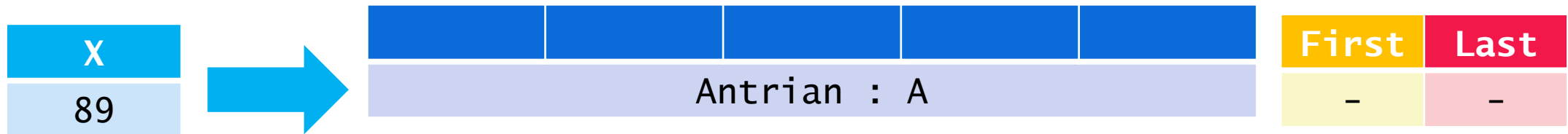
Tambah Elemen Antrian

- Proses menambahkan sebuah elemen baru ke **belakang** antrian
- Sebelumnya harus dicek apakah antrian ada dan **tidak penuh**.

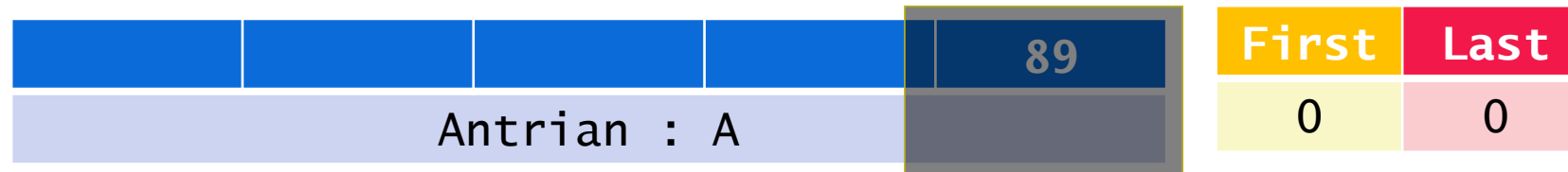
Antrian memiliki 5 slot dan masih kosong



Elemen baru siap dimasukan yaitu : 89



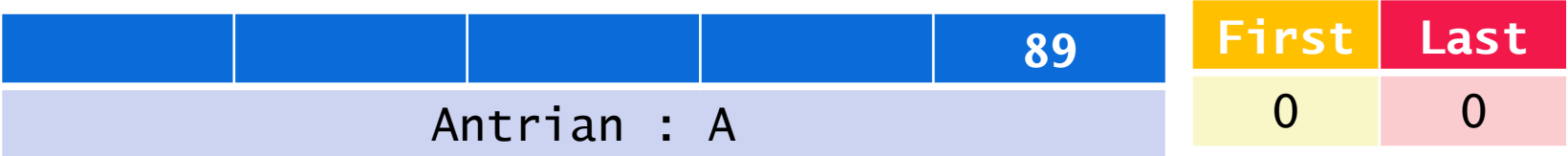
89 disimpan pada bagian depan



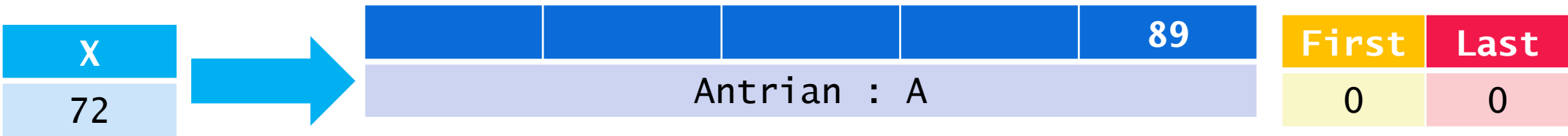
Tambah Elemen Antrian (2)

Menambahkan elemen lainnya

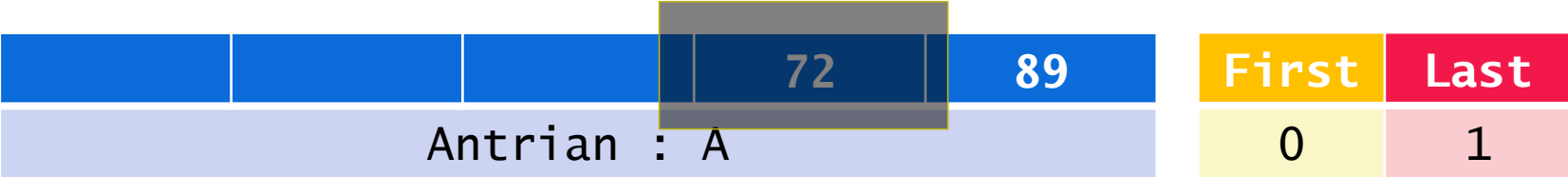
Antrian memiliki 5 slot dan sudah terisi 1



Elemen baru siap dimasukan yaitu : 72



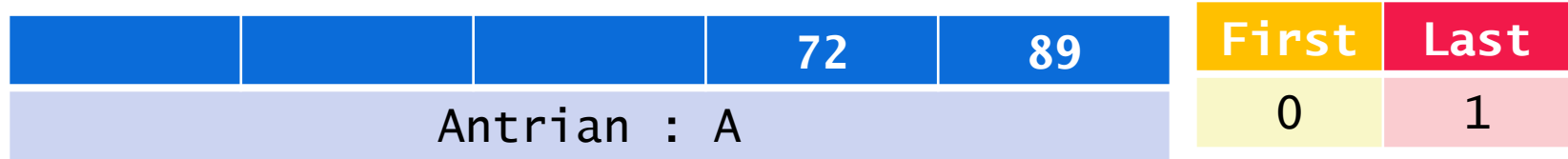
72 disimpan pada setelah elemen posisi **belakang**



Tambah Elemen Antrian (3)

Menambahkan elemen lainnya

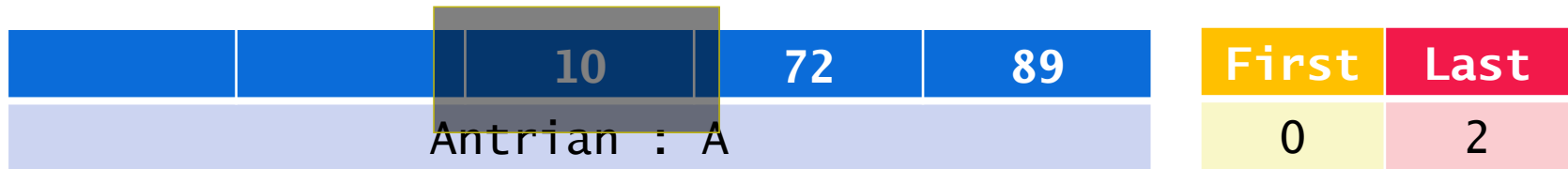
Antrian memiliki 5 slot dan sudah terisi 2



Elemen baru siap dimasukan yaitu : 10



10 disimpan pada setelah elemen posisi **belakang**



Setiap tambah elemen,
posisi **BELAKANG** bertambah

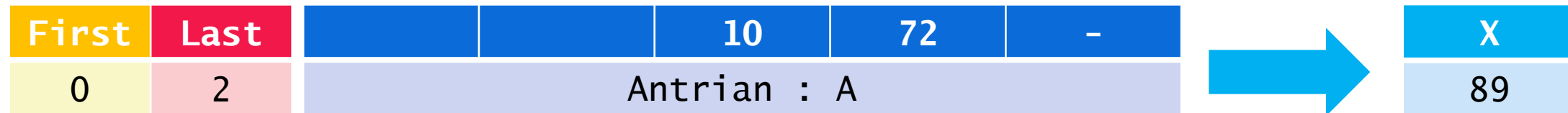
Hapus Elemen Antrian

- Proses menghapus sebuah elemen dari bagian **depan** antrian
 - Ibarat antrian nasabah, saat nasabah paling depan selesai dilayani.
- Sebelum dilakukan penghapusan, pastikan antrian ada dan **tidak kosong**

Antrian memiliki 5 slot dan sudah terisi 3



Hapus satu elemen pada antrian, posisi **depan**



Semua elemen bergeser satu langkah ke posisi **depan**



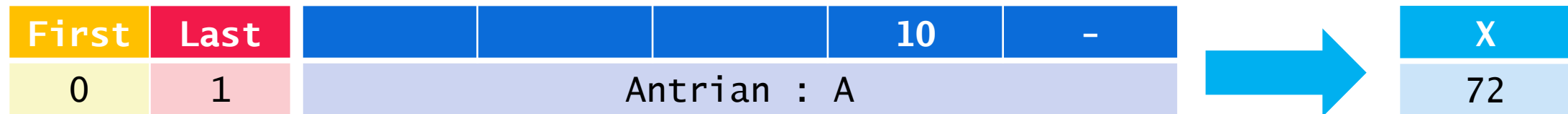
Hapus Elemen Antrian (2)

Menghapus elemen lainnya pada antrian

Antrian memiliki 5 slot dan sudah terisi 2



Hapus satu elemen pada antrian, posisi **depan**



Semua elemen bergeser satu langkah ke posisi **depan**



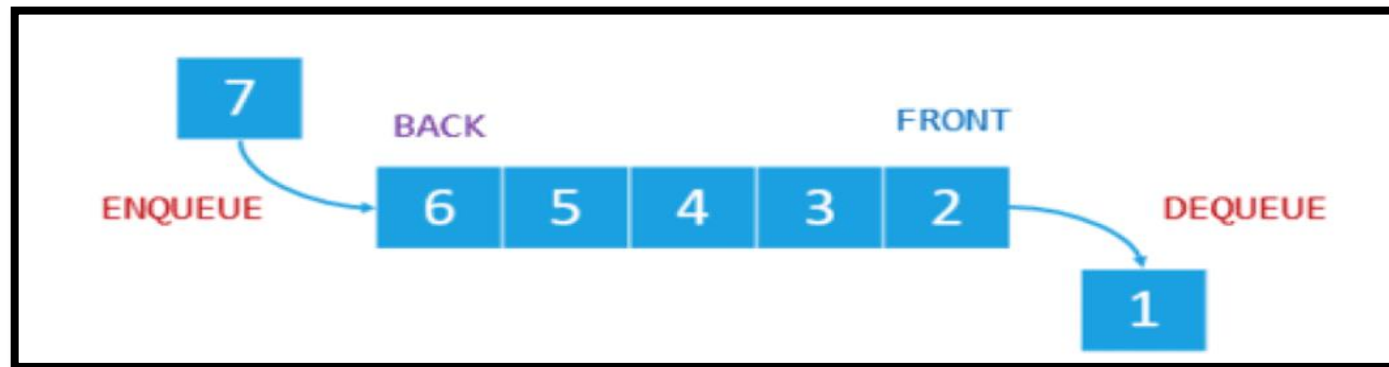
Setiap tambah elemen,
posisi **BELAKANG berkurang**

Operasi Lainnya Pada Antrian

- **Inisialisasi antrian**: menginisialisasi antrian dengan sebuah keadaan kosong: **First** dan **Last** diberi nilai kurang satu dari indeks awal.
- **Apakah Antrian Kosong** : menentukan apakah antrian kosong: jika **First** memiliki nilai kurang dari indeks awal/ terkecil.
 - Jika antrian kosong, mengembalikan nilai **true**.
 - Jika sebaliknya, mengembalikan nilai **false**.
- **Apakah Antrian Penuh** : menentukan apakah tumpukan penuh: jika **Last** memiliki nilai sama dengan ukuran antrian/ indeks paling besar pada array.
 - Jika antrian penuh, mengembalikan nilai **true**
 - Jika sebaliknya, mengembalikan nilai **false**
- **Depan** : menghasilkan elemen pertama (**first**) yang dimasukkan pada antrian.
- **Belakang** : menghasilkan elemen terakhir (**last**) yang dimasukkan pada antrian.

Implementasi Stack pada Array

- Menerapkan Antrian (**Statis**) pada array, karena :
 - Antrian : kumpulan elemen yang bertipe sama
 - Array : kumpulan data yang bertipe sama
- Perlu diingat:
 - Array adalah struktur data akses acak, artinya elemen array dapat diakses secara sembarang (menggunakan indeks).
 - Antrian adalah struktur data dimana **elemen-elemennya diakses (tambah & hapus) menggunakan dua sisi saja** : dimana menambah data pada bagian akhir sedangkan menghapus data pada bagian depan antrian (*First In First Out* / **FIFO**)



Contoh : Array Sebagai Antrian

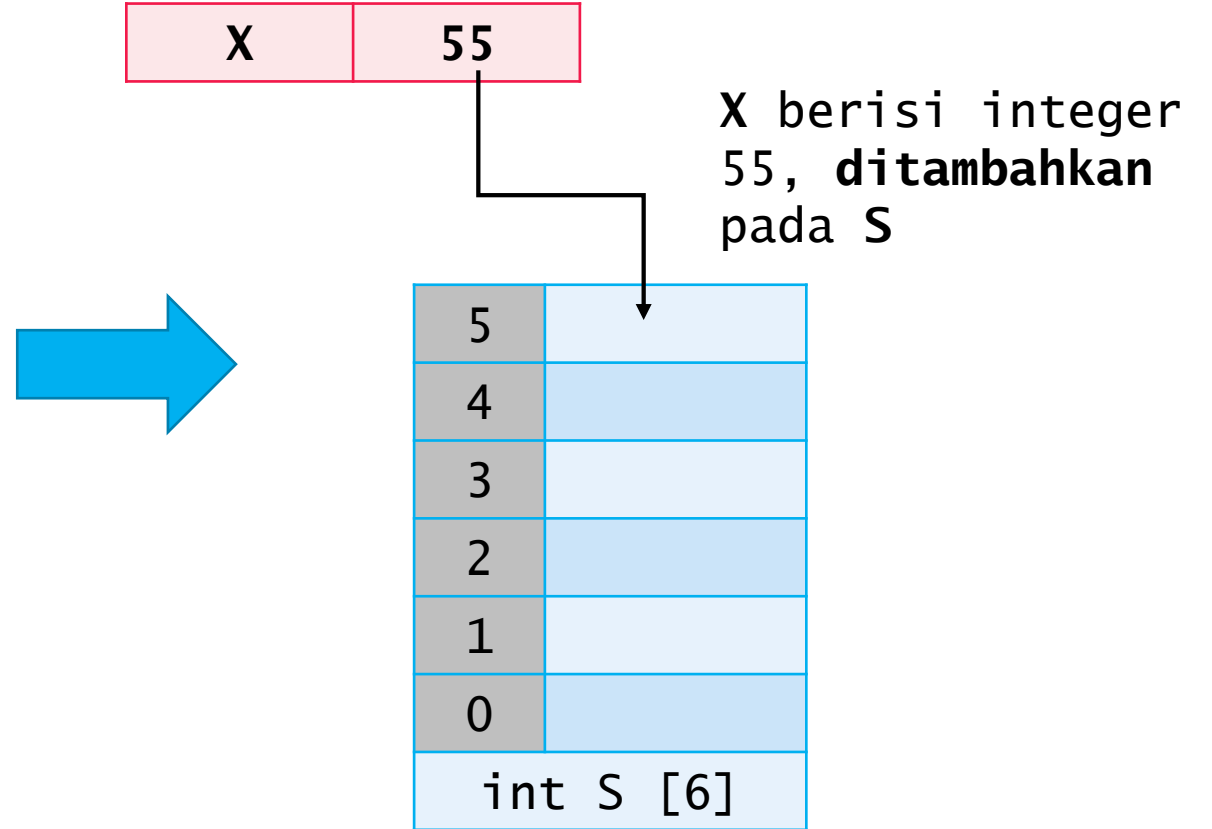
5	
4	
3	
2	
1	
0	
int S [6]	

S adalah **ARRAY** dengan tipe **integer** dengan ukuran 6 elemen

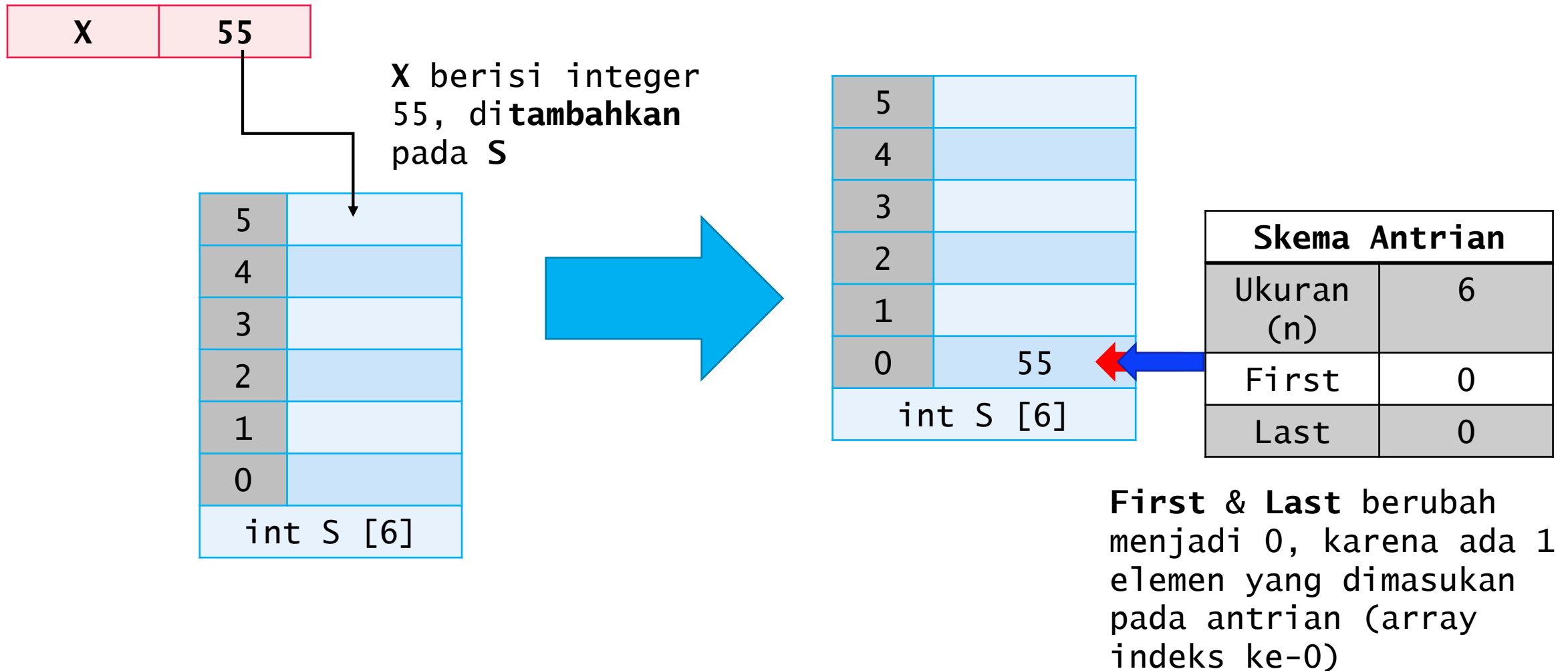
Skema Antrian	
Ukuran (n)	6
First	-1
Last	-1

Karena Antriannya dibangun dengan array, maka memiliki **ukuran (n)**, sehingga **(n-1) = indeks maksimum pada array**

First & Last diisi -1, karena belum ada elemen yang dimasukan pada antrian (**Antrian Kosong**)



Contoh : **TAMBAH** Elemen Ke Antrian



Contoh : **TAMBAH** Elemen Ke Antrian

X	89
---	----

X berisi integer 89,
ditambahkan pada S

5	
4	
3	
2	
1	
0	55
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	0



5	
4	
3	
2	
1	89
0	55
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	1

Last berubah menjadi 1,
karena ada 1 elemen lagi
yang dimasukan pada
antrian (array indeks
ke-1)

Contoh : **TAMBAH** Elemen Ke Antrian

X	45
---	----

X berisi integer 45,
ditambahkan pada S

5	
4	
3	
2	
1	89
0	55
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	1



5	
4	
3	
2	45
1	89
0	55
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	2

Last berubah menjadi 2,
karena ada 1 elemen lagi
yang dimasukan pada
antrian (array indeks
ke-2)

Contoh : **HAPUS** Elemen Dari Antrian

Elemen Pertama
(**First**) berisi 55,
dihapus dari S

5	
4	
3	
2	45
1	89
0	55
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	2



5	
4	
3	
2	
1	45
0	89
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	1

Last berubah menjadi 1,
karena ada 1 elemen
dihapus dari antrian
(array indeks ke-0)

Y	55
---	----

Contoh : **HAPUS** Elemen Dari Antrian

Elemen Pertama
(**First**) berisi **89**,
dihapus dari **S**

5	
4	
3	
2	
1	45
0	89
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	1



5	
4	
3	
2	
1	
0	45
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	0

Last berubah menjadi 0,
karena ada 1 elemen
dihapus dari antrian
(array indeks ke-0)

Y	89
---	----

Contoh : **HAPUS** Elemen Dari Antrian

Elemen Pertama
(**First**) berisi **45**,
dihapus dari **S**

5	
4	
3	
2	
1	
0	45
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	0



5	
4	
3	
2	
1	
0	
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	-1
Last	-1

First & Last berubah menjadi -1, karena ada 1 elemen dihapus dari antrian (array indeks ke-0) dan antrian jadi **KOSONG**

Y	45
---	----

First In First Out

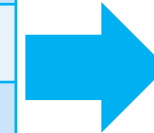
5	
4	
3	
2	
1	
0	
int S [6]	



5	
4	
3	
2	
1	
0	55
int S [6]	



5	
4	
3	
2	
1	89
0	55
int S [6]	

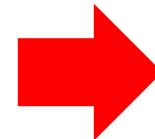


5	
4	
3	
2	45
1	89
0	55
int S [6]	

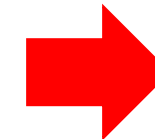
5	
4	
3	
2	45
1	89
0	55
int S [6]	



5	
4	
3	
2	
1	45
0	89
int S [6]	



5	
4	
3	
2	
1	
0	45
int S [6]	



5	
4	
3	
2	
1	
0	
int S [6]	

Contoh : **HAPUS** Elemen tertentu Dari Antrian

Elemen **89** akan **dihapus** dari **S**

Elemen Pertama (**First**) berisi **55**, **dihapus** dari **S**

5	
4	
3	
2	45
1	89
0	55
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	2



5	
4	
3	
2	
1	45
0	89
int S [6]	

Y	55
---	----

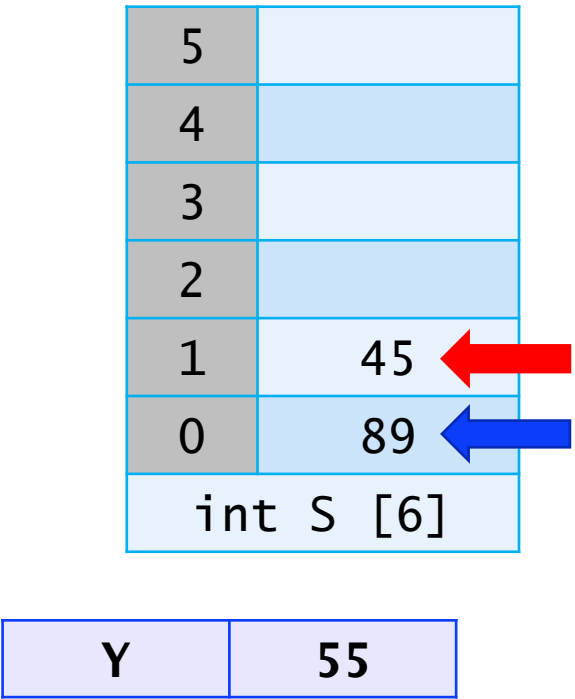
Skema Antrian	
Ukuran (n)	6
First	0
Last	1

Last berubah menjadi 1, karena ada 1 elemen dihapus dari antrian (array indeks ke-0)

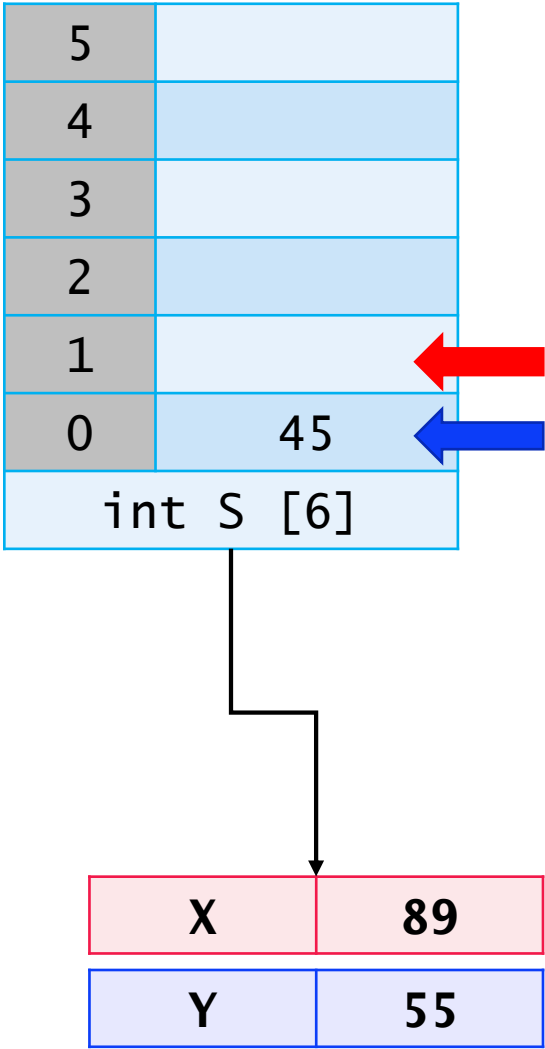
Contoh : **HAPUS** Elemen tertentu Dari Antrian

Elemen **89** akan **dihapus**
dari **S**

Elemen Pertama (**First**)
berisi **89**, **dihapus** dari **S**



Skema Antrian	
Ukuran (n)	6
First	0
Last	1



Skema Antrian	
Ukuran (n)	6
First	0
Last	0

Last berubah menjadi 0,
karena ada 1 elemen
dihapus dari antrian
(array indeks ke-0)

Contoh : **HAPUS** Elemen tertentu Dari Antrian

Elemen **89** akan dihapus dari **S**

Y	55
---	----

Tambahkan kembali elemen sebelum **89** ke Antrian

5	
4	
3	
2	
1	
0	45
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	0

Y	55
---	----

5	
4	
3	
2	
1	55
0	45
int S [6]	

Skema Antrian	
Ukuran (n)	6
First	0
Last	1

Last berubah menjadi 1, karena ada 1 elemen lagi yang dimasukan pada antrian (array indeks ke-1)

Implementasi Pada C++



Implementasi Operasi-Operasi Antrian

```
1  #include <iostream>
2  #define n 10
3
4  using namespace std;
5
6  //global variabel
7  int First, Last;
8  int S[n]; Antrian
9
10 void buatKosong(){
11     First = -1;
12     Last = -1;
13 }
14
```

Deklarasi variable global
agar dapat diakses semua
method

Membuat Antrian menjadi kosong

Operasi-Operasi Pendukung pada Antrian

```
15 bool isKosong(){
16     return (First < 0);
17 }
18
19 bool isPenuh(){
20     return (Last == (n-1));
21 }
22
23 int getDepan(){
24     return First;
25 }
26
27 int getBelakang(){
28     return Last;
29 }
30
```

Menampilkan isi antrian

```
31 void tampilkanAntrian(){
32     if(iskosong() == false){
33         cout<<"Isi stak : ";
34         for(int i=(n-1); i>=0; i--){
35             cout<<S[i]<<" | ";
36         }
37         cout<<"\n";
38     }else{
39         cout<<"Antrian KOSONG | Depan = "<<getDepan()<<" | Belakang = "<<getBelakang()<<"\n";
40     }
41 }
42
```

Tambah Elemen Antrian

```
43 int tambahElemen(int x){
44     if(isPenuh() == false){
45         if(isKosong() == true){
46             First += 1;
47         }
48         Last += 1;
49         S[Last] = x;
50
51         cout<<x<<" telah ditambahkan pada Antrian | Depan = ";
52         cout<<getDepan()<<" | Belakang = "<<getBelakang()<<"\n";
53     }else{
54         cout<<"Antrian PENUH | Depan = "<<getDepan();
55         cout<<" | Belakang = "<<getBelakang()<<"\n";
56     }
57
58     return getBelakang();
59 }
60
```

Hapus Elemen Antrian

```
61 int hapusElemen(){
62     int x = S[First];
63     if( isEmpty() == false){
64         if(getDepan() == getBelakang()){ //satu elemen di Antrian
65             S[First] = 0;
66             buatKosong(); // Antrian jadi KOSONG
67         }else{
68             S[First] = 0; // hapus elemen pertama
69             //geser elemen
70             int posisi = First+1;
71             while(posisi <= Last){
72                 S[posisi-1] = S[posisi];
73                 posisi++;
74             }
75             S[Last] = 0;
76             Last -= 1;
77         }
78
79         cout<<x<<" telah dihapus dari Antrian | Depan = ";
80         cout<<getDepan()<<" | Belakang = "<<getBelakang()<<"\n";
81     }else{
82         cout<<"Antrian PENUH | Depan = "<<getDepan();
83         cout<<" | Belakang = "<<getBelakang()<<"\n";
84     }
85
86     return getBelakang();
87 }
```


Main Program

```
89 int main(){
90     int x;
91     char Y;
92
93     buatKosong();
94     tampilkanAntrian();
95     x = 0;
96
97     //isi Antrian sampai PENUH
98     while(isPenuh() == false){
99         cout<<"Masukan angka (0-99) = ";
100         cin>>x;
101         tambahElemen(x);
102         tampilkanAntrian();
103     }
104
105     Y = 'Y';
106     while(Y == 'Y'){
107         cout<<"Hapus elemen dari Antrian (Y/ T) = ";
108         cin>>Y;
109         if(Y == 'Y'){
110             hapusElemen();
111         }
112         tampilkanAntrian();
113     }
114     return 0;
115 }
```

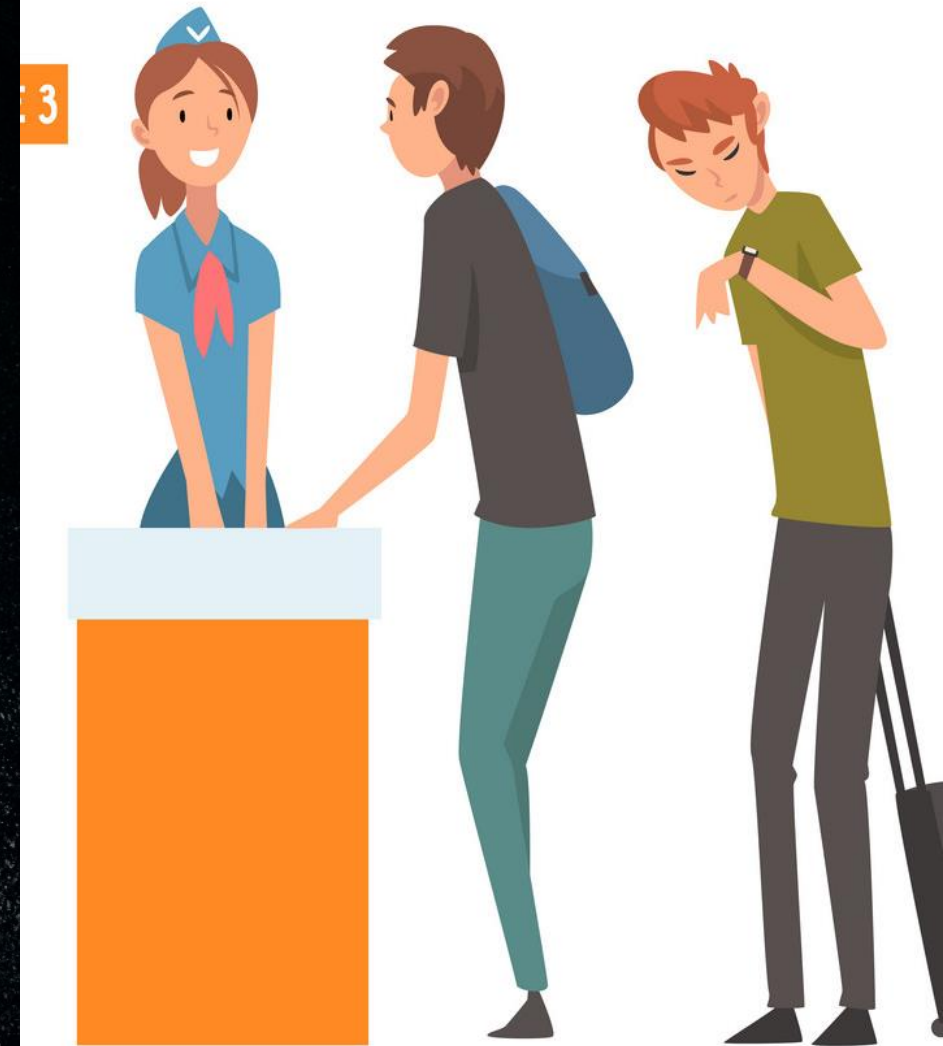
Running Program

The screenshot shows the OnlineGDB web interface. The browser tabs include 'Queue Data Structure - Geeksfor...', 'Queue | Set 1 (Introduction and ...', and 'GDB online Debugger | Compiler'. The address bar shows 'onlinegdb.com'. The left sidebar contains navigation links: 'Welcome, Fitri Nuraeni', 'Create New Project', 'My Projects', 'Classroom new', 'Learn Programming', 'Programming Questions', and 'Logout'. Below these are social media icons for Facebook and Twitter, and a '65.2K' badge. A 'GITHUB' logo and 'VIRTUAL HACKATHON' banner are also visible. The main area displays a C++ program in 'main.cpp' with line numbers 88 to 116. The code includes functions for creating an empty queue, displaying the queue, adding elements, and removing elements. The program uses a while loop to repeatedly add and remove elements from the queue. The language is set to 'C++'. The bottom status bar shows the time '23.50' and date '16/06/2020'.

```
88
89     buatKosong();
90     tampilkanAntrian();
91
92     x = 0;
93
94     //isi Antrian sampai PENUH
95     while(isPenuh() == false){
96         cout<<"Masukan angka (0-99) = ";
97         cin>>x;
98         tambahElemen(x);
99         tampilkanAntrian();
100     }
101
102
103     Y = 'Y';
104     while(Y == 'Y'){
105         cout<<"Hapus elemen dari Antrian (Y/ T) = ";
106         cin>>Y;
107         if(Y == 'Y'){
108             hapusElemen();
109         }
110         tampilkanAntrian();
111     }
112
113
114
115     return 0;
116 }
```

input

Bagaimana menghapus
elemen tertentu yang
bukan elemen Pertama?





Thank You

Fitri Nuraeni, M.Kom

✉ nufi3@stmik-tasikmalaya.ac.id

