

Artis : GD
Judul : Croocked

Artis : BigBang
Judul : BLue

Artis : Rm
Judul : Indigo

Artis : Jhope
Judul : Arson

Artis : Jin
Judul : Astronout

Artis : AshIsland
Judul : Everything

Artis : Radiohead
Judul : Creep

===== Menampilkan List =====

Artis : GD
Judul : Croocked

Artis : BigBang
Judul : BLue

Artis : Rm
Judul : Indigo

Artis : Jhope
Judul : Arson

Artis : Jin
Judul : Astronout

Artis : AshIsland
Judul : Everything

Artis : Radiohead
Judul : Creep

===== Menampilkan List yang telah dihapus =====

Artis : BigBang
Judul : BLue

Artis : Rm
Judul : Indigo

Artis : Jhope
Judul : Arson

Artis : Jin
Judul : Astronout

Artis : AshIsland
Judul : Everything

Artis : Radiohead
Judul : Creep

```
CSLLh X CSLLcpp X main.cpp X
1  #ifndef CSLL_H_INCLUDED
2  #define CSLL_H_INCLUDED
3
4  #include <iostream>
5
6  using namespace std;
7
8  #define info(P) (P)->info
9  #define next(P) (P)->next
10 #define head(D) ((D).head)
11 #define tail(D) ((D).tail)
12
13 typedef struct element *adr;
14
15 struct infotype{
16     string Artis, Judul;
17 };
18
19 struct element{
20     infotype info;
21     adr next;
22 };
23
24 struct playlistLagu{
25     adr head, tail;
26 };
27
28 void createPlaylist_1301213072(playlistLagu &Q);
29 void newElement_1301213072(infotype laguBaru, adr &pLagu);
30 void enqueue_1301213072(playlistLagu &Q, adr pLagu);
31 void dequeue_1301213072(playlistLagu &Q, adr &pLagu);
32 void showSemuaLagu_1301213072(playlistLagu Q);
33
34 #endif // CSLL_H_INCLUDED
35
```

```
CSLLh X CSLLcpp X main.cpp X
1  #include "CSLL.h"
2
3  void createPlaylist_1301213072(playlistLagu &Q)
4  {
5      head(Q) = NULL;
6      tail(Q) = NULL;
7  }
8
9  void newElement_1301213072(infotype laguBaru, adr &pLagu)
10 {
11     pLagu = new element;
12     info(pLagu) = laguBaru;
13     next(pLagu) = NULL;
14 }
15
16 void enqueue_1301213072(playlistLagu &Q, adr pLagu){
17
18     if (head(Q) == NULL){
19         head(Q) = pLagu;
20         tail(Q) = pLagu;
21     }else{
22         next(tail(Q)) = pLagu;
23         tail(Q) = pLagu;
24     }
25 }
```

```

26
27 void dequeue_1301213072(playlistLagu &Q, adr &PLagu) {
28     if (head(Q) != NULL) {
29         PLagu = head(Q);
30         head(Q) = next(PLagu);
31         next(PLagu) = NULL;
32     } else {
33         cout << "Queue Kosong" << endl;
34     }
35 }
36
37 void showSemuaLagu_1301213072(playlistLagu Q)
38 {
39     adr P = head(Q);
40
41     while (P != NULL) {
42         cout << "Artis : " << info(P).Artis << endl;
43         cout << "Judul : " << info(P).Judul << endl << endl;
44
45         P = next(P);
46     }
47 }
48

```

```

CSLLh X CSLLcpp X main.cpp X
1 #include "CSLL.h"
2 int main()
3 {
4     playlistLagu Q;
5     createPlaylist_1301213072(Q);
6
7     int N = 7;
8     infotype Lagu;
9     adr P;
10
11
12     for (int i = 1; i <= N; i++) {
13         cout << "Artis : ";
14         cin >> Lagu.Artis;
15         cout << "Judul : ";
16         cin >> Lagu.Judul;
17         cout << endl;
18
19         newElement_1301213072(Lagu, P);
20         enqueue_1301213072(Q, P);
21     }
22     cout << "===== Menampilkan List =====" << endl;
23     showSemuaLagu_1301213072(Q);
24
25     dequeue_1301213072(Q, P);
26
27     cout << "===== Menampilkan List yang telah dihapus =====" << endl;
28     showSemuaLagu_1301213072(Q);
29
30     return 0;
31 }

```