

Muhamad Fachri Haikal
1301202398
IF-44-01
AJF MODUL 8
Header:

```
SLL_Circular.cpp X main.cpp X SLL_Circular.h X
1  #ifndef SLL_CIRCULAR_H_INCLUDED
2  #define SLL_CIRCULAR_H_INCLUDED
3
4  #include <string>
5  #include <iostream>
6  using namespace std;
7
8  #define info(P) (P)->info
9  #define next(P) (P)->next
10 #define first(L) ((L).first)
11
12 struct data{
13     string nama;
14     int prioritas;
15     int sisa_durasi;
16 };
17
18 typedef data infotype;
19 typedef struct elmList *adr;
20
21 struct elmList{
22     infotype info;
23     adr next;
24 };
25
26 struct List{
27     adr first;
28 };
29
30 void createList_1301202398(List &L);
31 adr createElemen_1301202398(infotype dataBaru);
32 void insertFirst_1301202398(List &L, adr p);
33 void insertLast_1301202398(List &L, adr p);
34 void insertAfter_1301202398(List &L, adr p);
35 void insertDescending_1301202398(List &L, infotype dataBaru);
36 void deleteFirst_1301202398(List &L);
37 void deleteLast_1301202398(List &L);
38 void deleteAfter_1301202398(List &L, adr prec);
39 void deleteElm_1301202398(List &L, adr p);
40 void printList_1301202398(List L);
41 int panjangList_1301202398(List L);
42 void insertAplikasi_1301202398(List &L, infotype dataBaru);
43 int eksekusi_1301202398(adr p, int durasi);
44 void eksekusiMulti_1301202398(List &L, adr &current, int N, int duration)
45
46 #endif // SLL_CIRCULAR_H_INCLUDED
47
```

Muhamad Fachri Haikal
1301202398
IF-44-01
AJF MODUL 8

Body:

```
*SLL_Circular.cpp X SLL_Circular.h X *main.cpp X
1  #include "SLL_Circular.h"
2
3  void createList_1301202398(List &L){
4      first(L) = NULL;
5  }
6
7  adr createElemen_1301202398(infotype dataBaru){
8      adr p = new elmList;
9
10     info(p) = dataBaru;
11     next(p) = NULL;
12     return p;
13 }
14
15 void insertFirst_1301202398(List &L, adr p){
16     adr q;
17
18     if (first(L) == NULL){
19         first(L) = p;
20         next(p) = first(L);
21     }else{
22         next(p) = first(L);
23         q = first(L);
24         while (next(q) != first(L)){
25             q = next(q);
26         }
27         next(q) = p;
28         first(L) = p;
29     }
30 }
31
32 void insertLast_1301202398(List &L, adr p){
33     adr q;
34
35     if (first(L) == NULL){
36         first(L) = p;
37         next(p) = first(L);
38     }else{
39         next(p) = first(L);
40         q = first(L);
41         while (next(q) != first(L)){
42             q = next(q);
43         }
44         next(q) = p;
45     }
46 }
```

```
*SLL_Circular.cpp X SLL_Circular.h X *main.cpp X
46 }
47 void insertAfter_1301202398(List &L, adr prec, adr p){
48     next(p) = next(prec);
49     next(prec) = p;
50 }
51
52 void insertDescending_1301202398(List &L, infotype dataBaru){
53     adr p, q, prec;
54
55     p = createElemen_1301202398(dataBaru);
56     q = first(L);
57     if (first(L) == NULL && last(L) == NULL){
58         insertFirst_1301202398(L, p);
59     }else{
60         while (next(q) != first(L) && info(p).prioritas > info(q).prioritas){
61             prec = q;
62             q = next(q);
63         }
64         if (q == first(L)){
65             insertFirst_1301202398(L, p);
66         }else if (next(q) == first(L)){
67             insertLast_1301202398(L, p);
68         }else{
69             insertAfter_1301202398(L, prec, p);
70         }
71     }
72 }
73 void deleteFirst_1301202398(List &L){
74     adr p, q;
75
76     p = first(L);
77     if (next(first(L)) == first(L)){
78         first(L) = NULL;
79         next(p) = NULL;
80     }else{
81         first(L) = next(first(L));
82         next(p) = NULL;
83         q = first(L);
84         while (next(q) != p){
85             q = next(q);
86         }
87         next(q) = first(L);
88     }
89 }
```

```
*SLL_Circular.cpp X SLL_Circular.h X *main.cpp X
138 void printList_1301202398(List L){
139     adr p;
140     int i;
141
142     i = 1;
143     if (first(L) == NULL){
144         cout << "List Kosong!" << endl;
145     }else{
146         p = first(L);
147         while (next(p) != first(L)){
148             cout << "[" << i << " ] ";
149             cout << info(p).nama;
150             cout << info(p).prioritas;
151             cout << info(p).sisa_durasi;
152             cout << endl;
153             i++;
154             p = next(p);
155         }
156         cout << "[" << i << " ] ";
157         cout << info(p).nama;
158         cout << info(p).prioritas;
159         cout << info(p).sisa_durasi;
160         cout << endl;
161     }
162 }
163
164 int panjangList_1301202398(List L){
165     int jum;
166     adr p;
167
168     jum = 0;
169     p = first(L);
170     while (next(p) != first(L)){
171         jum++;
172         p = next(p);
173     }
174     jum++;
175     return jum;
176 }
```

```
*SLL_Circular.cpp X SLL_Circular.h X *main.cpp X
88 }
89 void deleteLast_1301202398(List &L){
90     adr p, q;
91
92     q = first(L);
93     p = first(L);
94     while (next(p) != first(L)){
95         q = p;
96         p = next(p);
97     }
98     p = next(q);
99     next(q) = first(L);
100     next(p) = NULL;
101 }
102 void deleteAfter_1301202398(List &L){
103     adr p;
104
105     p = next(prec);
106     next(prec) = next(p);
107     next(p) = NULL;
108 }
109 void deleteElem_1301202398(List &L, adr p){
110     adr q, prec;
111     bool ketemu;
112
113     q = first(L);
114     ketemu = false;
115     while (next(q) != first(L) && ketemu == false){
116         if (info(q) == info(p)){
117             ketemu = true;
118         }else{
119             prec = q;
120             q = next(q);
121         }
122     }
123     if (ketemu == true){
124         if (q == first(L)){
125             deleteFirst_1301202398(L);
126         }else if (next(q) == first(L)){
127             deleteLast_1301202398(L);
128         }else{
129             deleteAfter_1301202398(L, prec);
130         }
131     }else{
132         cout << "Data tidak ditemukan" << endl;
133     }
134 }
135 }
136 }
137 }
```

Muhamad Fachri Haikal

1301202398

IF-44-01

AJF MODUL 8

```
SLL_Circular.cpp X main.cpp X SLL_Circular.h X
163 }
164
165 int panjangList_1301202398(List L){
166     int jum;
167     adr p;
168
169     jum = 0;
170     p = first(L);
171     while (next(p) != first(L)){
172         jum++;
173         p = next(p);
174     }
175     jum++;
176     return jum;
177 }
178
179 void insertAplikasi_1301202398(List &L, infotype dataBaru){
180     adr p;
181     int panjang;
182
183     panjang = panjangList_1301202398(L);
184     if (panjang == 0){
185         insertDescending_1301202398(L, dataBaru);
186     }else if (panjang <= 5){
187         insertDescending_1301202398(L, dataBaru);
188     }else{
189         p = first(L);
190         while (next(p) != first(L)){
191             p = next(p);
192         }
193         if (dataBaru.prioritas > info(p).prioritas){
194             deleteElem_1301202398(L, p);
195             insertAplikasi_1301202398(L, dataBaru);
196         }
197     }
198 }
199
200 int eksekusi_1301202398(adr p, int durasi){
201     int sisa;
202
203     sisa = info(p).sisa_durasi - durasi;
204     return sisa;
205 }
206
207 void eksekusiMulti_1301202398(List &L, adr &current, int N, int duration){
208     int i;
209
210     i = 1;
211     while (i<=N){
212         eksekusi_1301202398(current, duration);
213         if (info(current).sisa_durasi <= 0){
214             deleteElem_1301202398(L, current);
215         }
216         current = next(current);
217         i++;
218     }
219 }
220
```

```
200 int eksekusi_1301202398(adr p, int durasi){
201     int sisa;
202
203     sisa = info(p).sisa_durasi - durasi;
204     return sisa;
205 }
206
207 void eksekusiMulti_1301202398(List &L, adr &current, int N, int duration){
208     int i;
209
210     i = 1;
211     while (i<=N){
212         eksekusi_1301202398(current, duration);
213         if (info(current).sisa_durasi <= 0){
214             deleteElem_1301202398(L, current);
215         }
216         current = next(current);
217         i++;
218     }
219 }
220
```

Main:

```
*SLL_Circular.cpp X main.cpp X SLL_Circular.h X
1 | #include "SLL_Circular.h"
2
3 int main(){
4     List L;
5     adr p;
6     infotype dataBaru;
7
8     const max_application = 5;
9
10    createList_1301202398(L);
11    for (int i=1; i<=5; i++){
12        cout << "Masukkan data ke-" << i << ":" << endl;
13        cout << "Nama: ";
14        cin >> dataBaru.nama;
15        cout << "Prioritas: ";
16        cin >> dataBaru.prioritas;
17        cout << "Sisa Durasi: ";
18        cin >> dataBaru.sisa_durasi;
19        cout << endl;
20        insertAplikasi_1301202398(L, dataBaru);
21    }
22
23    printList_1301202398(L);
24
25    p = first(L);
26    eksekusiMulti_1301202398(p, 3, 10);
27
28    printList_1301202398(L);
29
30    eksekusiMulti_1301202398(p, 3, 10);
31
32    printList_1301202398(L);
33 }
34
```