

```
"D:\Telkom University\SE-43-03\Kelas\Semester 2\Struktur Data\TP\TP 5\circulardll-first\bin\Debug\circulardll-first.exe"
Tampilkan semua data: List kosong
Tampilkan semua data: t c a t a s c a
Cari kata cat: 2

Process returned 0 (0x0)   execution time : 0.071 s
Press any key to continue.
```

```
*circulardll-first.h  X  circulardll-first.cpp  X  main.cpp  X

1  #ifndef CIRCULARDLL-FIRST_H_INCLUDED
2  #define CIRCULARDLL-FIRST_H_INCLUDED
3  #include <iostream>
4  #define first(L) L.first
5  #define prev(P) P->prev
6  #define info(P) P->info
7  #define next(P) P->next
8
9  /*
10     Name      : Muhamad Dwiki Riswanda
11     NIM       : 1302194015
12  */
13
14     using namespace std;
15
16     typedef char infotype;
17     typedef struct elmList *address;
18
19     struct elmList {
20         infotype info;
21         address prev;
22         address next;
23     };
24
25     struct list {
26         address first;
27     };
28
29     bool isEmpty(list L); //1
30     void createList(list &L); //2
31     void createNewElmt(address &P, infotype X); //3
32     void insertFirst(list &L, address P); // 4
```

```

29     bool isEmpty(list L); //1
30     void createList(list &L); //2
31     void createNewElmt(address &P, infotype X); //3
32     void insertFirst(list &L, address P); // 4
33     void insertAfter(list &L, address &P, address Prec); //5
34     void deleteFirst(list &L, address &P); //6
35     void deleteAfter(list &L, address &P, address Prec); //7
36     int countWord(char data[], list L); //8
37     void printInfo(list L); //9
38     void insertLast(list &L, address P); //10
39     address cariElmt(list L, infotype X); //11
40
41     #endif // CIRCULARDLL-FIRST_H_INCLUDED

```

```

*circulardll-first.h  x circulardll-first.cpp  x main.cpp  x
1     #include "circulardll-first.h"
2
3     /*
4     Name      : Muhamad Dwiki Biswanda
5     NIM       : 1302194015
6     */
7
8     bool isEmpty(list L) {
9     if (first(L) == NULL) {
10         return true;
11     } else {
12         return false;
13     }
14     }; //1
15
16     void createList(list &L) {
17         first(L) = NULL;
18     }; //2
19
20     void createNewElmt(address &P, infotype X) {
21         P = new elmList;
22         prev(P) = NULL;
23         info(P) = X;
24         next(P) = NULL;
25     }; //3
26
27     void insertFirst(list &L, address P) {
28     if (isEmpty(L) == true) {
29         first(L) = P;
30         next(P) = first(L);
31     } else {
32         address Q = first(L);

```

```
*circulardll-first.h  x  circulardll-first.cpp  x  main.cpp  x

26
27 void insertFirst(list &L, address P) {
28     if (isEmpty(L) == true) {
29         first(L) = P;
30         next(P) = first(L);
31     } else {
32         address Q = first(L);
33         while (next(Q) != first(L)) {
34             Q = next(Q);
35         }
36         next(P) = first(L);
37         prev(first(L)) = P;
38         first(L) = P;
39         next(Q) = first(L);
40     }
41 }; // 4
42
43 void insertAfter(list &L, address &P, address Prec) {
44     if (isEmpty(L) == true) {
45         cout << "Data kosong gan" << endl;
46     } else if (next(Prec) == first(L)) {
47         next(Prec) = P;
48         prev(P) = P;
49         next(P) = first(L);
50     } else {
51         next(P) = next(Prec);
52         prev(P) = Prec;
53         prev(next(Prec)) = P;
54         next(Prec) = P;
55     }
56 }; //5
57
```

```
*circulardll-first.h  x  circulardll-first.cpp  x  main.cpp  x

55     }
56 }; //5
57
58 void deleteFirst(list &L, address &P) {
59     if (isEmpty(L) == true) {
60         cout << "Kosong listnya, gan";
61         cout << endl;
62         P = NULL;
63     } else if (next(first(L)) == first(L)) {
64         P = first(L);
65         first(L) = NULL;
66     } else {
67         address Q = first(L);
68         while (next(Q) != first(L)) {
69             Q = next(Q);
70         }
71         P = first(L);
72         first(L) = next(P);
73         next(P) = NULL;
74         prev(first(L)) = NULL;
75         next(Q) = first(L);
76     }
77 }; //6
78
79 void deleteAfter(list &L, address &P, address Prec) {
80     address Q;
81     if (isEmpty(L) == true) {
82         cout << "Listnya kosong gan";
83         cout << endl;
84         P = NULL;
85     } else if (next(Prec) == first(L)) {
86         P = first(L);
```

```
*circulardll-first.h  ×  circulardll-first.cpp  ×  main.cpp  ×

76     }
77 }; //6
78
79 void deleteAfter(list &L, address &P, address Prec) {
80     address Q;
81     if (isEmpty(L) == true) {
82         cout << "Listnya kosong gan";
83         cout << endl;
84         P = NULL;
85     } else if (next(Prec) == first(L)) {
86         P = first(L);
87         while (next(P) != first(L)) {
88             Q = P;
89             P = next(P);
90         }
91         next(Q) = first(L);
92         prev(P) = NULL;
93         next(P) = NULL;
94
95     } else {
96         P = next(Prec);
97         next(Prec) = next(P);
98         prev(next(P)) = Prec;
99         next(P) = NULL;
100        prev(P) = NULL;
101    }
102 }; //7
103
104 int countWord(char data[], list L) {
105     address P;
106     int i = 0;
107     P = first(L);
```

```
*circulardll-first.h  X  circulardll-first.cpp  X  main.cpp  X
103
104 int countWord(char data[], list L) {
105     address P;
106     int i = 0;
107     P = first(L);
108     while (next(P) != first(L)) {
109         if (info(P) == data[0]) {
110             i = i + 1;
111         }
112         P = next(P);
113     }
114     if (info(P) == data[0]) {
115         i = i + 1;
116     }
117     return i;
118 }; //8
119
120
121 void printInfo(list L) {
122     address P = first(L);
123     if (first(L) == NULL) {
124         cout << "List kosong" << endl;
125     } else {
126         P = first(L);
127         cout << info(P) << " ";
128         while (next(P) != first(L)) {
129             P = next(P);
130             cout << info(P) << " ";
131         }
132     }
133 }; //9
134
```

```
*circulardll-first.h  X  circulardll-first.cpp  X  main.cpp  X
130         cout << info(P) << " ";
131     }
132 }
133 }; //9
134
135 void insertLast(list &L, address P) {
136     if (isEmpty(L) != true) {
137         address Q = first(L);
138         while (next(Q) != first(L)) {
139             Q = next(Q);
140         }
141         next(Q) = P;
142         prev(P) = Q;
143         next(P) = first(L);
144     } else {
145         first(L) = P;
146         next(P) = first(L);
147     }
148 }; //10
149
150 address cariElmt(list L, infotype X) {
151     address P = first(L);
152     while (P != NULL && info(P) != X) {
153         P = next(P);
154     }
155     if (P != NULL) {
156         return P;
157     }
158     return NULL;
159 }; //11
160
161
```

```
*circulardll-first.h  X  circulardll-first.cpp  X  *main.cpp  X
1  #include "circulardll-first.h"
2
3  /*
4   Name      : Muhamad Dwiki Riswanda
5   NIM       : 1302194015
6   */
7
8  using namespace std;
9
10 int main()
11 {
12     list L;
13     address Prec, P;
14     char data[] = ("cat");
15     int jumlah;
16
17     createList(L);
18     cout << "Tampilkan semua data: ";
19     printInfo(L);
20
21     createNewElmt(P, 'a');
22     insertFirst(L, P);
23     createNewElmt(P, 't');
24     insertFirst(L, P);
25     Prec = cariElmt(L, 'a');
26     createNewElmt(P, 't');
27     insertAfter(L, P, Prec);
28     Prec = cariElmt(L, 't');
29     createNewElmt(P, 'c');
30     insertAfter(L, P, Prec);
31     createNewElmt(P, 'a');
32     insertLast(L, P);
33
34     Prec = cariElmt(L, 't');
35     createNewElmt(P, 'c');
36     insertAfter(L, P, Prec);
37     createNewElmt(P, 'a');
38     insertLast(L, P);
39     createNewElmt(P, 's');
40     insertLast(L, P);
41     createNewElmt(P, 'c');
42     insertLast(L, P);
43     createNewElmt(P, 'a');
44     insertLast(L, P);
45
46     cout << "Tampilkan semua data: ";
47     printInfo(L);
48     cout << endl;
49
50     jumlah = countWord(data, L);
51     cout << "Cari kata cat: " << jumlah << endl;
52
53     return 0;
54 }
```