

LAPORAN PRAKTIKUM
Modul 05
“Single Linked List Bagian Kedua”



Disusun Oleh:
Yoga Eka Pratama - 2311104023
Kelas
SE-07-1
Dosen :
Yudha Islami Sulistya, S.Kom., M.Cs.

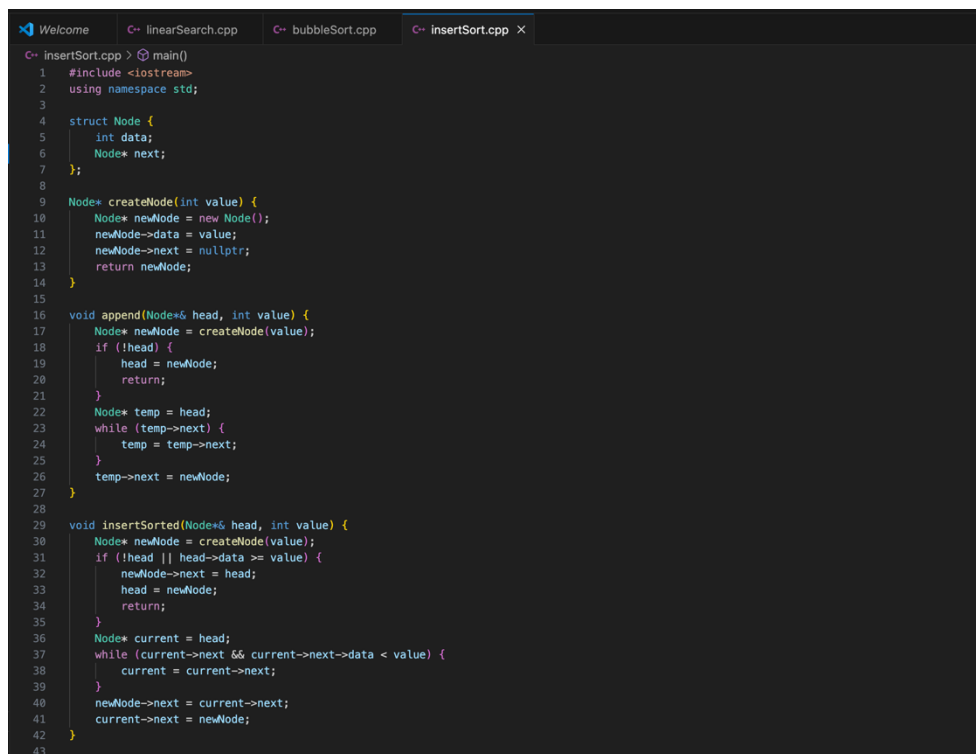
PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

Tugas Pendahuluan

1. Mencari Elemen Tertentu dalam SLL,

Deskripsi Soal: Buatlah program yang mengizinkan pengguna memasukkan 6 elemen integer ke dalam list. Implementasikan function searchElement untuk mencari apakah sebuah nilai tertentu ada dalam list.

Jawab :



```
C++ insertSort.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  Node* createNode(int value) {
10     Node* newNode = new Node();
11     newNode->data = value;
12     newNode->next = nullptr;
13     return newNode;
14 }
15
16 void append(Node*& head, int value) {
17     Node* newNode = createNode(value);
18     if (!head) {
19         head = newNode;
20         return;
21     }
22     Node* temp = head;
23     while (temp->next) {
24         temp = temp->next;
25     }
26     temp->next = newNode;
27 }
28
29 void insertSorted(Node*& head, int value) {
30     Node* newNode = createNode(value);
31     if (!head || head->data >= value) {
32         newNode->next = head;
33         head = newNode;
34         return;
35     }
36     Node* current = head;
37     while (current->next && current->next->data < value) {
38         current = current->next;
39     }
40     newNode->next = current->next;
41     current->next = newNode;
42 }
43
```

```

44 void displayList(Node* head) {
45     Node* temp = head;
46     while (temp) {
47         cout << temp->data << " -> ";
48         temp = temp->next;
49     }
50     cout << "null" << endl;
51 }
52
53 int main() {
54     Node* head = nullptr;
55     int num;
56
57     for (int i = 0; i < 4; i++) {
58         cout << "Masukan Elemen " << i + 1 << ": ";
59         cin >> num;
60         insertSorted(head, num);
61     }
62
63     int newElement;
64     cout << "Masukkan elemen yang akan disisipkan dalam urutan yang sudah diurutkan: ";
65     cin >> newElement;
66     insertSorted(head, newElement);
67
68     cout << "List setelah vdi sisipkan: " << endl;
69     displayList(head);
70
71     return 0;
72 }
73
74
75
76

```

Output :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/MODUL 5/TP/output"
./"insertSort"
macairm1@MacBook-Air-MacAirM1 TP % cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/MODUL 5/TP/output"
macairm1@MacBook-Air-MacAirM1 output % ./"insertSort"
Masukan Elemen 1: 10
Masukan Elemen 2: 23
Masukan Elemen 3: 1
Masukan Elemen 4: 5
Masukkan elemen yang akan disisipkan dalam urutan yang sudah diurutkan: 20
List setelah vdi sisipkan:
1 -> 5 -> 10 -> 20 -> 23 -> null
macairm1@MacBook-Air-MacAirM1 output %

```

2. Mengurutkan List Menggunakan Bubble Sort

Deskripsi Soal: Buatlah program yang mengizinkan pengguna memasukkan 5 elemen integer ke dalam list. Implementasikan procedure bubbleSortList untuk mengurutkan elemen-elemen dalam list dari nilai terkecil ke terbesar.

Jawab :

```

Welcome | C++ linearSearch.cpp | C++ bubbleSort.cpp X | C++ insertSort.cpp
C++ bubbleSort.cpp > displayList(Node*)
1 #include <iostream>
2 using namespace std;
3
4 struct Node {
5     int data;
6     Node* next;
7 };
8
9 Node* createNode(int value) {
10     Node* newNode = new Node();
11     newNode->data = value;
12     newNode->next = nullptr;
13     return newNode;
14 }
15
16 void append(Node*& head, int value) {
17     Node* newNode = createNode(value);
18     if (!head) {
19         head = newNode;
20         return;
21     }
22     Node* temp = head;
23     while (temp->next) {
24         temp = temp->next;
25     }
26     temp->next = newNode;
27 }
28
29 void bubbleSort(Node*& head) {
30     if (!head) return;
31     bool swapped;
32     Node* current;
33     Node* lastSorted = nullptr;
34
35     do {
36         swapped = false;
37         current = head;
38
39         while (current->next != lastSorted) {
40             if (current->data > current->next->data) {
41                 swap(current->data, current->next->data);
42                 swapped = true;
43             }
44             current = current->next;
45         }
46         lastSorted = current;
47     } while (swapped);
48 }
49
50 void displayList(Node* head) {
51     Node* temp = head;
52     while (temp) {
53         cout << temp->data << " -> ";
54         temp = temp->next;
55     }
56     cout << "null" << endl;
57 }
58
59 int main() {
60     Node* head = nullptr;
61     int num;
62
63     for (int i = 0; i < 5; i++) {
64         cout << "Masukan Elemen " << i + 1 << ": ";
65         cin >> num;
66         append(head, num);
67     }
68
69     bubbleSort(head);
70
71     cout << "Sorted list: " << endl;
72     displayList(head);
73
74     return 0;
75 }

```

Output :

```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS
cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/MODUL 5/TP/output"
./bubbleSort
macairm1@MacBook-Air-MacAirM1 TP % cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/MODUL 5/TP/output"
macairm1@MacBook-Air-MacAirM1 output % ./"bubbleSort"
Masukan Elemen 1: 20
Masukan Elemen 2: 50
Masukan Elemen 3: 12
Masukan Elemen 4: 1
Masukan Elemen 5: 25
Sorted list:
1 -> 12 -> 20 -> 25 -> 50 -> null
macairm1@MacBook-Air-MacAirM1 output %

```

3. Menambahkan Elemen Secara Terurut

Deskripsi Soal: Buatlah program yang mengizinkan pengguna memasukkan 4 elemen integer ke dalam list secara manual. Kemudian, minta pengguna memasukkan elemen tambahan yang harus ditempatkan di posisi yang sesuai sehingga list tetap terurut.

Jawab :

```

C++ linearSearch.cpp  C++ bubbleSort.cpp  C++ insertSort.cpp X
C++ insertSort.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  Node* createNode(int value) {
10     Node* newNode = new Node();
11     newNode->data = value;
12     newNode->next = nullptr;
13     return newNode;
14 }
15
16 void append(Node*& head, int value) {
17     Node* newNode = createNode(value);
18     if (!head) {
19         head = newNode;
20         return;
21     }
22     Node* temp = head;
23     while (temp->next) {
24         temp = temp->next;
25     }
26     temp->next = newNode;
27 }
28
29 void insertSorted(Node*& head, int value) {
30     Node* newNode = createNode(value);
31     if (!head || head->data >= value) {
32         newNode->next = head;
33         head = newNode;
34         return;
35     }
36     Node* current = head;
37     while (current->next && current->next->data < value) {
38         current = current->next;
39     }
40     newNode->next = current->next;
41     current->next = newNode;
42 }
43
44 void displayList(Node* head) {
45     Node* temp = head;
46     while (temp) {
47         cout << temp->data << " -> ";
48         temp = temp->next;
49     }
50     cout << "null" << endl;
51 }
52
53 int main() {
54     Node* head = nullptr;
55     int num;
56
57     for (int i = 0; i < 4; i++) {
58         cout << "Masukan Elemen " << i + 1 << ": ";
59         cin >> num;
60         insertSorted(head, num);
61     }
62
63     int newElement;
64     cout << "Masukkan elemen yang akan disisipkan dalam urutan yang sudah diurutkan: ";
65     cin >> newElement;
66     insertSorted(head, newElement);
67
68     cout << "List setelah vdi sisipkan: " << endl;
69     displayList(head);
70
71     return 0;
72 }
73
74
75
```

Output :

```
EA | Module: kernel.kexts.kexts
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/MODUL 5/TP/output"
./"InsertSort"
● macairm1@MacBook-Air-MacAirM1 TP % cd "/Users/macairm1/STD_YOGA_EKA_PRATAMA_2311104023/MODUL 5/TP/output"
● macairm1@MacBook-Air-MacAirM1 output % ./"InsertSort"
Masukan Elemen 1: 10
Masukan Elemen 2: 2
Masukan Elemen 3: 30
Masukan Elemen 4: 25
Masukkan elemen yang akan disisipkan dalam urutan yang sudah diurutkan: 22
List setelah vdi sisipkan:
2 -> 10 -> 22 -> 25 -> 30 -> null
○ macairm1@MacBook-Air-MacAirM1 output % █
```