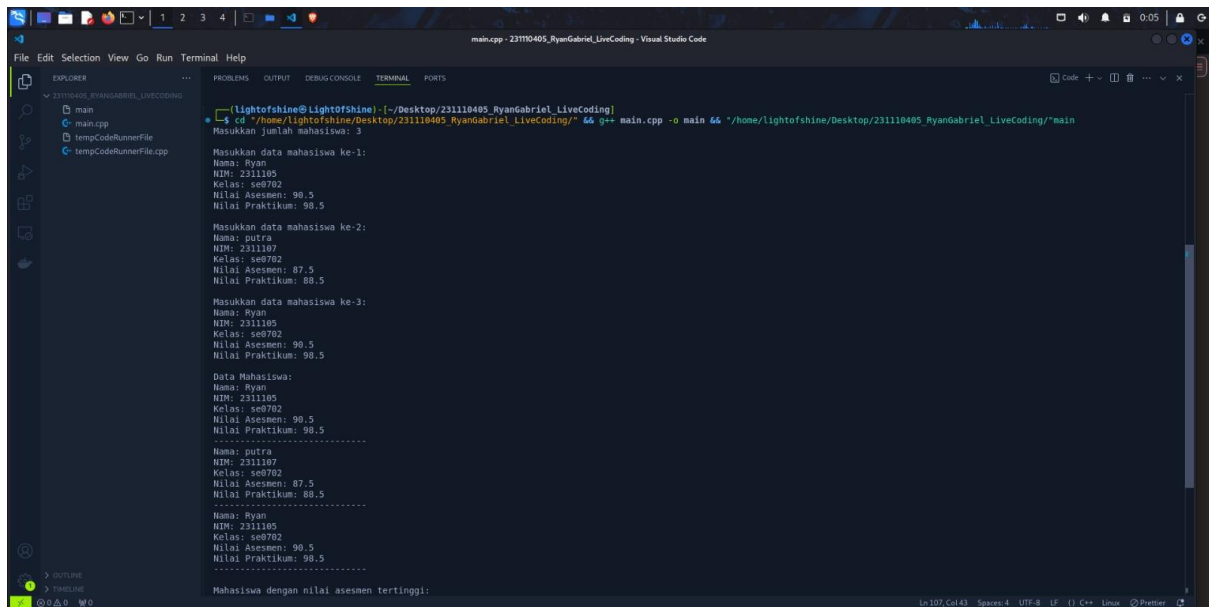


Nama : Ryan Gabriel Togar Simamora

Nim : 2311104045

Kelas : SE0702

Output :



```
(Lightofshine@Lightofshine) [~/Desktop/231110405_RyanGabriel_LiveCoding]
$ cd /home/lightofshine/Desktop/231110405_RyanGabriel_LiveCoding/ && g++ main.cpp -o main && ./main
Masukkan jumlah mahasiswa: 3

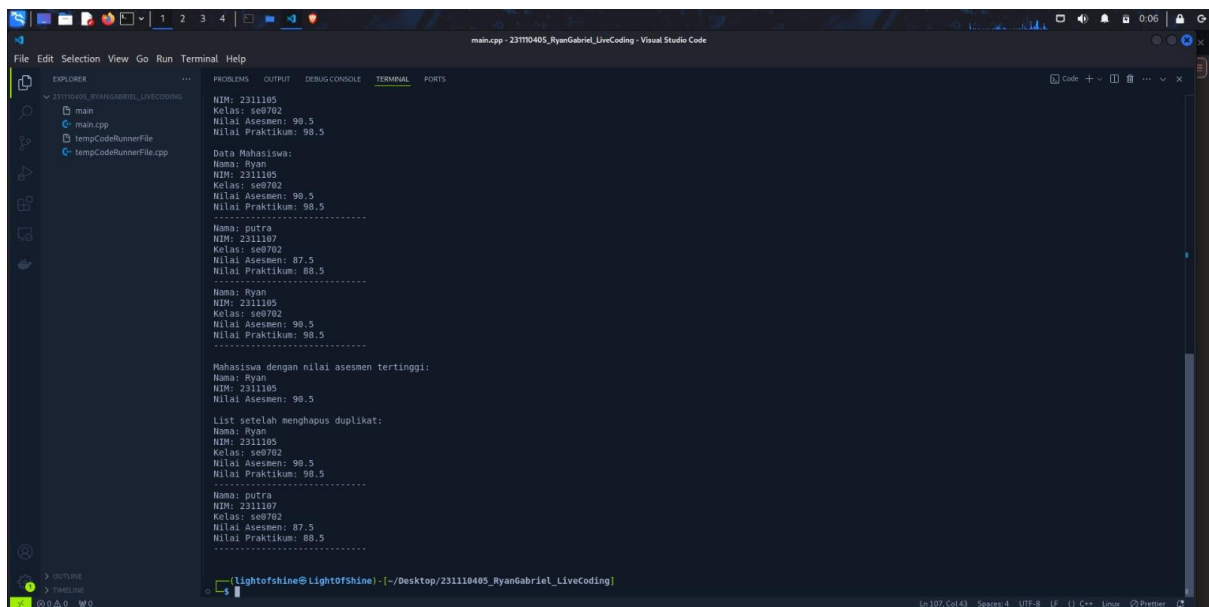
Masukkan data mahasiswa ke-1:
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5

Masukkan data mahasiswa ke-2:
Nama: putra
NIM: 2311107
Kelas: se0702
Nilai Asesmen: 87.5
Nilai Praktikum: 88.5

Masukkan data mahasiswa ke-3:
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5

Data Mahasiswa:
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5
-----
Nama: putra
NIM: 2311107
Kelas: se0702
Nilai Asesmen: 87.5
Nilai Praktikum: 88.5
-----
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5
-----

Mahasiswa dengan nilai asesmen tertinggi:
```



```
(Lightofshine@Lightofshine) [~/Desktop/231110405_RyanGabriel_LiveCoding]
$ cd /home/lightofshine/Desktop/231110405_RyanGabriel_LiveCoding/ && g++ main.cpp -o main && ./main
Masukkan jumlah mahasiswa: 3

Masukkan data mahasiswa ke-1:
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5

Masukkan data mahasiswa ke-2:
Nama: putra
NIM: 2311107
Kelas: se0702
Nilai Asesmen: 87.5
Nilai Praktikum: 88.5

Masukkan data mahasiswa ke-3:
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5

Data Mahasiswa:
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5
-----
Nama: putra
NIM: 2311107
Kelas: se0702
Nilai Asesmen: 87.5
Nilai Praktikum: 88.5
-----
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5
-----

Mahasiswa dengan nilai asesmen tertinggi:
Nama: Ryan
NIM: 2311105
Nilai Asesmen: 90.5

List setelah menghapus duplikat:
Nama: Ryan
NIM: 2311105
Kelas: se0702
Nilai Asesmen: 90.5
Nilai Praktikum: 88.5
-----
Nama: putra
NIM: 2311107
Kelas: se0702
Nilai Asesmen: 87.5
Nilai Praktikum: 88.5
-----
```

Ss Program :

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 // Struktur Mahasiswa
6 struct Mahasiswa {
7     string Nama;
8     string NIM;
9     string Kelas;
10    float Nilai_Asessen;
11    float Nilai_Praktikum;
12 };
13
14 // Node untuk Single Linked List
15 struct Node {
16     Mahasiswa info;
17     Node* next;
18 };
19
20 // Function: Membuat node baru
21 Node* newElement(Mahasiswa data) {
22     Node* newNode = new Node;
23     newNode->info = data;
24     newNode->next = nullptr;
25     return newNode;
26 }
27
28 // Function: Membuat list baru (kosong)
29 Node* newList() {
30     return nullptr;
31 }
32
33 // Procedure: Insert First untuk NIM ganjil
34 void insertFirst(Node* head, Mahasiswa data) {
35     Node* newNode = newElement(data);
36     newNode->next = head;
37     head = newNode;
38 }
39
40 // Procedure: Menampilkan semua data mahasiswa
41 void printList(Node* head) {
42     if (head == nullptr) {
43         cout << "List kosong!" << endl;
44         return;
45     }
46     Node* current = head;
47     while (current != nullptr) {
48         cout << "Nama: " << current->info.Nama << endl;
49         cout << "NIM: " << current->info.Nim << endl;
50         cout << "Kelas: " << current->info.Kelas << endl;
51         cout << "Nilai Asessen: " << current->info.Nilai_Asessen << endl;
52         cout << "Nilai Praktikum: " << current->info.Nilai_Praktikum << endl;
53         cout << "....." << endl;
54         current = current->next;
55     }
56 }
57
58 // Procedure: Menampilkan mahasiswa dengan nilai asesmen tertinggi
59 void findHighestAsessen(Node* head) {
60     if (head == nullptr) {
61         cout << "List kosong!" << endl;
62         return;
63     }
64     Node* highest = head;
65     Node* current = head->next;
66     while (current != nullptr) {
67         if (current->info.Nilai_Asessen > highest->info.Nilai_Asessen) {
68             highest = current;
69         }
70         current = current->next;
71     }
72     cout << "Mahasiswa dengan nilai asesmen tertinggi:" << endl;
73     cout << "Nama: " << highest->info.Nama << endl;
74     cout << "NIM: " << highest->info.Nim << endl;
75     cout << "Nilai Asessen: " << highest->info.Nilai_Asessen << endl;
76 }
77
78 // Procedure: Menghapus data dengan NIM duplikat
79 void deleteDuplicates(Node* head) {
80     Node* current = head;
81     while (current != nullptr) {
82         Node* prev = current;
83         Node* validasi = current->next;
84         while (validasi != nullptr) {
85             if (validasi->info.Nim == current->info.Nim) {
86                 prev->next = validasi->next;
87                 delete validasi;
88                 validasi = prev->next;
89             } else {
90                 prev = validasi;
91                 validasi = validasi->next;
92             }
93         }
94         current = current->next;
95     }
96 }
97
98 // Program Utama
99 int main() {
100     Node* list = newList(); // Membuat list kosong
101     int N;
102
103     cout << "Masukkan jumlah mahasiswa: ";
104     cin >> N;
105     cin.ignore(); // Menghindari konflik input getline
106
107     // Input data mahasiswa
108     for (int i = 0; i < N; i++) {
109         Mahasiswa data;
110         cout << "Masukkan data mahasiswa ke-" << i + 1 << ": " << endl;
111         cout << "Nama: ";
112         getline(cin, data.Nama);
113         cout << "NIM: ";
114         cin >> data.Nim;
115         cout << "Kelas: ";
116         cin >> data.Kelas;
117         cout << "Nilai Asessen: ";
118         cin >> data.Nilai_Asessen;
119         cout << "Nilai Praktikum: ";
120         cin >> data.Nilai_Praktikum;
121         cin.ignore();
122
123         // Menambahkan data hanya jika NIM ganjil
124         if (stoi(data.Nim) % 2 == 0) {
125             insertFirst(list, data);
126         } else {
127             cout << "Data dengan NIM genap diabaikan." << endl;
128         }
129     }
130
131     // Menampilkan semua data mahasiswa
132     cout << "Data Mahasiswa:" << endl;
133     printList(list);
134
135     // Menampilkan mahasiswa dengan nilai asesmen tertinggi
136     findHighestAsessen(list);
137
138     // Menghapus data dengan NIM duplikat
139     deleteDuplicates(list);
140     cout << "List setelah menghapus duplikat:" << endl;
141     printList(list);
142
143     return 0;
144 }
```