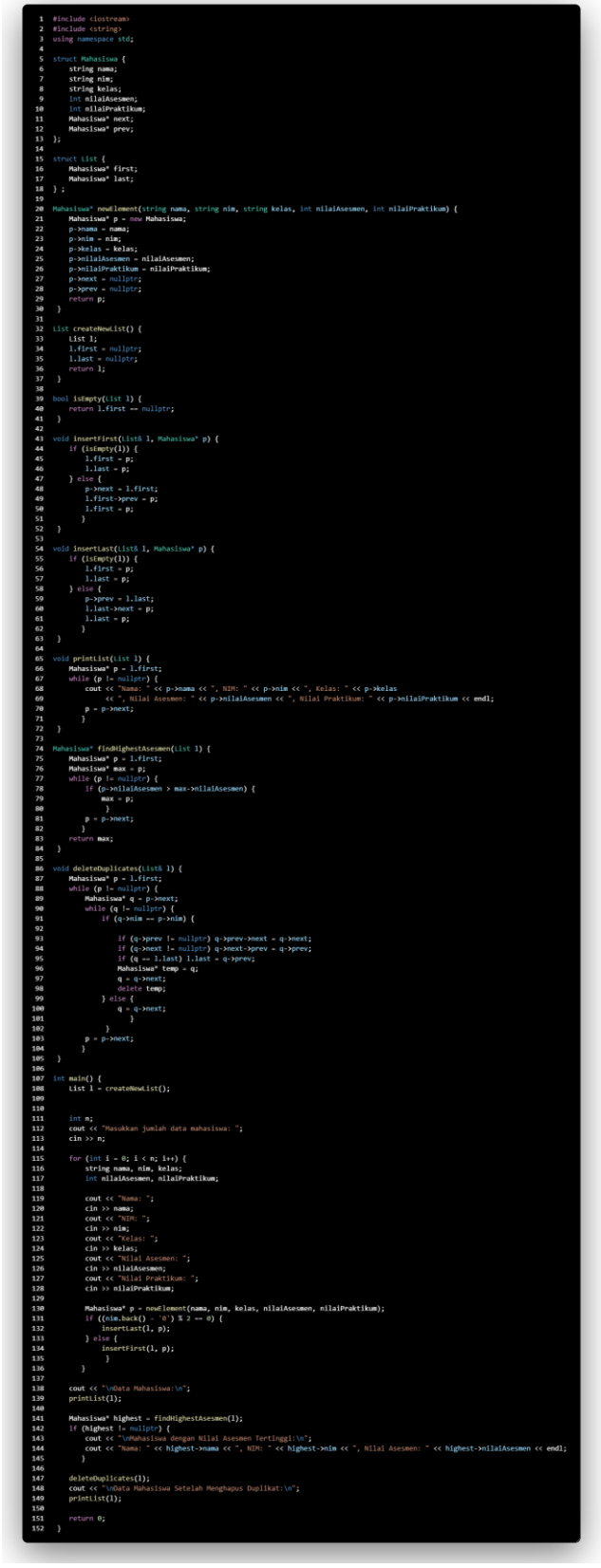
Razhendriya Vania Ramadhan Suganjarsarwat

23111004048



```cpp
#include <iostream>
#include <string>
using namespace std;

struct Mahasiswa {
    string nama;
    string nim;
    string kelas;
    int nilaiAsesmen;
    int nilaiPraktikum;
    Mahasiswa* next;
    Mahasiswa* prev;
};

struct List {
    Mahasiswa* first;
    Mahasiswa* last;
} ;

Mahasiswa* newElement(string nama, string nim, string kelas, int nilaiAsesmen, int nilaiPraktikum) {
    Mahasiswa* p = new Mahasiswa;
    p->nama = nama;
    p->nim = nim;
    p->kelas = kelas;
    p->nilaiAsesmen = nilaiAsesmen;
    p->nilaiPraktikum = nilaiPraktikum;
    p->next = nullptr;
    p->prev = nullptr;
    return p;
}

List createNewList() {
    List l;
    l.first = nullptr;
    l.last = nullptr;
    return l;
}

bool isEmpty(List l) {
    return l.first == nullptr;
}

void insertFirst(List& l, Mahasiswa* p) {
    if (isEmpty(l)) {
        l.first = p;
        l.last = p;
    } else {
        p->next = l.first;
        l.first->prev = p;
        l.first = p;
       }
}

void insertLast(List& l, Mahasiswa* p) {
    if (isEmpty(l)) {
        l.first = p;
        l.last = p;
    } else {
        p->prev = l.last;
        l.last->next = p;
        l.last = p;
       }
}

void printList(List l) {
    Mahasiswa* p = l.first;
    while (p != nullptr) {
        cout << "Nama: " << p->nama << ", NIM: " << p->nim << ", Kelas: " << p->kelas
             << ", Nilai Asesmen: " << p->nilaiAsesmen << ", Nilai Praktikum: " << p->nilaiPraktikum << endl;
        p = p->next;
       }
}

Mahasiswa* findHighestAsesmen(List l) {
    Mahasiswa* p = l.first;
    Mahasiswa* max = p;
    while (p != nullptr) {
        if (p->nilaiAsesmen > max->nilaiAsesmen) {
            max = p;
             }
        p = p->next;
       }
    return max;
}

void deleteDuplicates(List& l) {
    Mahasiswa* p = l.first;
    while (p != nullptr) {
        Mahasiswa* q = p->next;
        while (q != nullptr) {
            if (q->nim == p->nim) {

                if (q->prev != nullptr) q->prev->next = q->next;
                if (q->next != nullptr) q->next->prev = q->prev;
                if (q == l.last) l.last = q->prev;
                Mahasiswa* temp = q;
                q = q->next;
                delete temp;
            } else {
                q = q->next;
                   }
             }
        p = p->next;
       }
}

int main() {
    List l = createNewList();


    int n;
    cout << "Masukkan jumlah data mahasiswa: ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        string nama, nim, kelas;
        int nilaiAsesmen, nilaiPraktikum;

        cout << "Nama: ";
        cin >> nama;
        cout << "NIM: ";
        cin >> nim;
        cout << "Kelas: ";
        cin >> kelas;
        cout << "Nilai Asesmen: ";
        cin >> nilaiAsesmen;
        cout << "Nilai Praktikum: ";
        cin >> nilaiPraktikum;

        Mahasiswa* p = newElement(nama, nim, kelas, nilaiAsesmen, nilaiPraktikum);
        if ((nim.back() - '0') % 2 == 0) {
            insertLast(l, p);
        } else {
            insertFirst(l, p);
              }
    }

    cout << "\nData Mahasiswa:\n";
    printList(l);

    Mahasiswa* highest = findHighestAsesmen(l);
    if (highest != nullptr) {
        cout << "\nMahasiswa dengan Nilai Asesmen Tertinggi:\n";
        cout << "Nama: " << highest->nama << ", NIM: " << highest->nim << ", Nilai Asesmen: " << highest->nilaiAsesmen << endl;
       }

    deleteDuplicates(l);
    cout << "\nData Mahasiswa Setelah Menghapus Duplikat:\n";
    printList(l);

    return 0;
}
```

Contoh outputnya