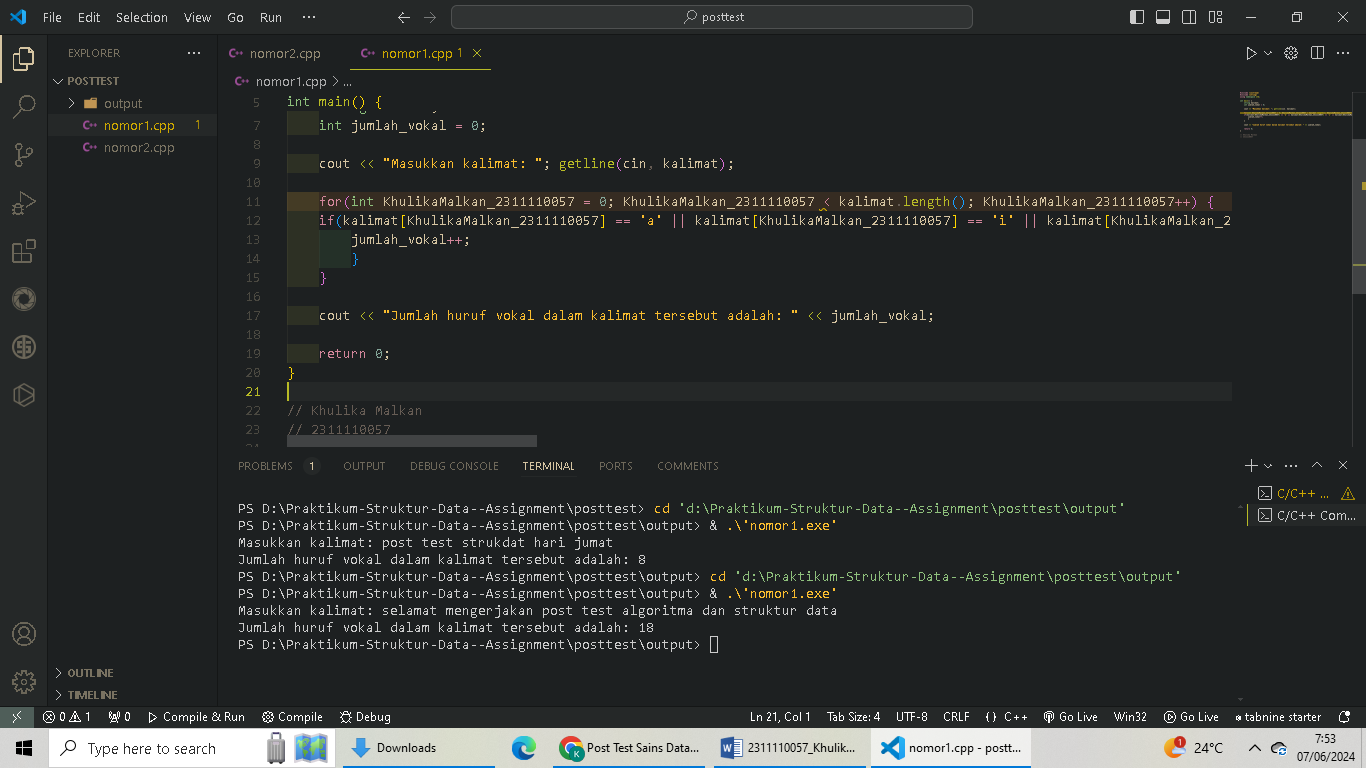
**Nama : Khulika Malkan**

**NIM : 2311110057**

Soal 1[

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  int main() {      string KhulikaMalkan\_2311110057;      int jumlah\_vokal = 0;      cout << "Masukkan kalimat: "; getline(cin, KhulikaMalkan\_2311110057);      for(int i = 0; i < KhulikaMalkan\_2311110057.length(); i++) {      if(KhulikaMalkan\_2311110057[i] == 'a' || KhulikaMalkan\_2311110057[i] == 'i' || KhulikaMalkan\_2311110057[i] == 'u' || KhulikaMalkan\_2311110057[i] == 'e' || KhulikaMalkan\_2311110057[i] == 'o' || KhulikaMalkan\_2311110057[i] == 'A' || KhulikaMalkan\_2311110057[i] == 'I' || KhulikaMalkan\_2311110057[i] == 'U' || KhulikaMalkan\_2311110057[i] == 'E' || KhulikaMalkan\_2311110057[i] == 'O') {          jumlah\_vokal++;          }      }      cout << "Jumlah huruf vokal dalam kalimat tersebut adalah: " << jumlah\_vokal;      return 0;  }  // Khulika Malkan  // 2311110057 |

Screenshots Output



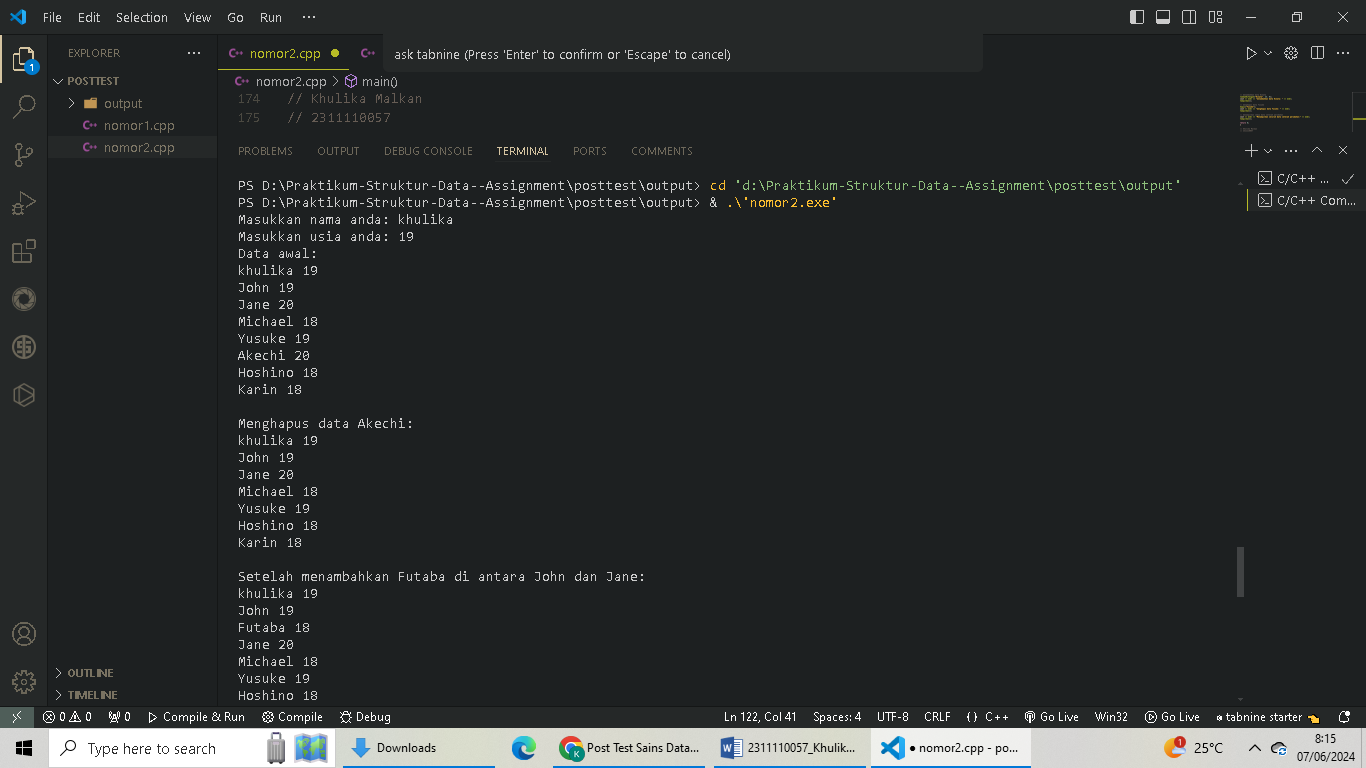
Deskripsi:

Program ini dapat menghitung banyaknya huruf vokal dalam sebuah kalimat. Pertama program akan meminta pengguna untuk memasukkan kalimat menggunakan getline(). Kemudian program akan melakukan perulangan untuk setiap karakter dalam kalimat dan memeriksa apakah karakter tersebut merupakan huruf vokal atau bukan. Jika karakter tersebut merupakan huruf vokal, maka variabel jumlah\_vokal akan ditambah satu. Terakhir, program akan mencetak jumlah huruf vokal yang ditemukan.

Soal 2[

|  |
| --- |
| #include <iostream>  using namespace std;  // mendeklarasikan struct node  struct Node {      string KhulikaMalkan\_2311110057;      int usia;      Node\* next;  };  // deklarasi head node  Node\* head = NULL;  // fungsi untuk menambahkan node di awal  void tambahDiAwal(string KhulikaMalkan\_2311110057, int usia) {      Node\* newNode = new Node;      newNode->KhulikaMalkan\_2311110057 = KhulikaMalkan\_2311110057;      newNode->usia = usia;      newNode->next = head;      head = newNode;  }  // fungsi untuk menambahkan node di akhir  void tambahDiAkhir(string KhulikaMalkan\_2311110057, int usia) {      Node\* newNode = new Node;      newNode->KhulikaMalkan\_2311110057 = KhulikaMalkan\_2311110057;      newNode->usia = usia;      newNode->next = NULL;      if (head == NULL) {          head = newNode;          return;      }      Node\* curr = head;      while (curr->next != NULL) {          curr = curr->next;      }      curr->next = newNode;  }  // fungsi untuk menambahkan node di tengah  void tambahDiTengah(string KhulikaMalkan\_2311110057, int usia, int pos) {      Node\* newNode = new Node;      newNode->KhulikaMalkan\_2311110057 = KhulikaMalkan\_2311110057;      newNode->usia = usia;      newNode->next = NULL;      if (head == NULL) {          head = newNode;          return;      }      Node\* curr = head;      int i = 1;      while (i < pos-1 && curr->next != NULL) {          curr = curr->next;          i++;      }      newNode->next = curr->next;      curr->next = newNode;  }  // fungsi untuk menghapus node dengan variabel nama(KhulikaMalkan\_2311110057) tertentu  void hapus(string KhulikaMalkan\_2311110057) {      Node\* curr = head;      Node\* prev = NULL;      while (curr != NULL && curr->KhulikaMalkan\_2311110057 != KhulikaMalkan\_2311110057) {          prev = curr;          curr = curr->next;      }      if (curr == NULL) {          cout << "Data tidak ditemukan" << endl;          return;      }      if (prev == NULL) {          head = curr->next;      } else {          prev->next = curr->next;      }      delete curr;  }  // fungsi untuk mengubah data node dengan nama(KhulikaMalkan\_2311110057) tertentu  void ubah(string KhulikaMalkan\_2311110057, string newKhulikaMalkan\_2311110057, int newUsia) {      Node\* curr = head;      while (curr != NULL && curr->KhulikaMalkan\_2311110057 != KhulikaMalkan\_2311110057) {          curr = curr->next;      }      if (curr == NULL) {          cout << "Data tidak ditemukan" << endl;          return;      }      curr->KhulikaMalkan\_2311110057 = newKhulikaMalkan\_2311110057;      curr->usia = newUsia;  }  // fungsi untuk menampilkan semua data  void tampilkan() {      if (head == NULL) {          cout << "List kosong" << endl;          return;      }      Node\* curr = head;      while (curr != NULL) {          cout << curr->KhulikaMalkan\_2311110057 << " " << curr->usia << endl;          curr = curr->next;      }  }  int main() {      // memasukkan data pertama (nama(KhulikaMalkan\_2311110057) dan usia anda)      string KhulikaMalkan\_2311110057Anda;      int usiaAnda;      cout << "Masukkan nama anda: ";      cin >> KhulikaMalkan\_2311110057Anda;      cout << "Masukkan usia anda: ";      cin >> usiaAnda;      tambahDiAwal(KhulikaMalkan\_2311110057Anda, usiaAnda);      // memasukkan data lainnya  tambahDiAkhir("John", 19);  tambahDiAkhir("Jane", 20);  tambahDiAkhir("Michael", 18);  tambahDiAkhir("Yusuke", 19);  tambahDiAkhir("Akechi", 20);  tambahDiAkhir("Hoshino", 18);  tambahDiAkhir("Karin", 18);  // menampilkan semua data  cout << "Data awal:" << endl;  tampilkan();  // menghapus data Akechi  hapus("Akechi");  cout << endl << "Menghapus data Akechi: " << endl;  tampilkan();  // menambahkan data Futaba di antara Carol dan Ann  tambahDiTengah("Futaba", 18, 3);  cout << endl << "Setelah menambahkan Futaba di antara John dan Jane: " << endl;  tampilkan();  // menambahkan data Igor di awal  tambahDiAwal("Igor", 20);  cout << endl << "Setelah menambahkan Igor di awal: " << endl;  tampilkan();  // menambahkan data Hinata  tambahDiTengah("Hinata", 23, 9);  cout << endl << "menambahkan data Hinata: " << endl;  tampilkan();  // menghapus data Yusuke  hapus("Yusuke");  cout << endl << "menghapus data Yusuke: " << endl;  tampilkan();  // menampilkan semua data setelah perubahan  cout << endl << "Menampilkan seluruh data setelah perubahan:" << endl;  tampilkan();  return 0;  }  // Khulika Malkan  // 2311110057 |

Screenshots Output







Deskripsi:

Linked list adalah sebuah struktur data dinamis yang terdiri dari sejumlah node yang dihubungkan satu sama lain melalui pointer. Setiap node pada linked list terdiri dari dua komponen utama, yaitu data dan pointer ke node selanjutnya. Pada program ini, terdapat sebuah struct bernama Node yang memiliki tiga atribut yaitu name, usia, dan next. Atribut name dan usia menyimpan data nama dan usia, sedangkan atribut next menyimpan pointer ke node selanjutnya.