**Program**

#include <iostream>

#include <fstream>

#include <cmath>

#include <vector>

#include <iomanip>

// Misalkan drone bergerak dengan kecepatan 5 kotak/s selama 3 detik pada sudut 45 derajat dari horizontal positif searah jarum jam

// Deklarasi variabel global

std::vector<std::pair<double, double>> history;  // Untuk menyimpan history koordinat drone

int current\_index = -1;  // Indeks saat ini di history

// Fungsi untuk menampilkan lokasi drone

void lokasi() {

    std::cout << "lokasi => (" << std::fixed << std::setprecision(2) << history[current\_index].first

              << ", " << history[current\_index].second << ")\n";

}

// Fungsi untuk melakukan gerakan dengan kecepatan dan waktu

void gerak\_2(double v, double t, double theta) {

    // Menghitung perubahan koordinat berdasarkan kecepatan dan waktu

    double delta\_x = v \* t \* cos(theta);

    double delta\_y = v \* t \* sin(theta);

    // Menambahkan koordinat baru ke dalam history

    history.push\_back(std::make\_pair(history[current\_index].first + delta\_x, history[current\_index].second + delta\_y));

    current\_index = history.size() - 1;

}

// Fungsi untuk menyimpan koordinat ke dalam file

void save() {

    std::ofstream file("history.txt");

    for (const auto& coord : history) {

        file << coord.first << " " << coord.second << "\n";

    }

    file.close();

}

// Fungsi untuk memuat koordinat dari file

void load() {

    std::ifstream file("history.txt");

    double x, y;

    history.clear();

    current\_index = -1;

    while (file >> x >> y) {

        history.push\_back(std::make\_pair(x, y));

    }

    file.close();

}

// Fungsi untuk melakukan undo

void undo() {

    if (current\_index > 0) {

        current\_index--;

        std::cout << "undo() => Undo berhasil\n";

    } else {

        std::cout << "undo() => Tidak dapat undo!\n";

    }

}

// Fungsi untuk melakukan redo

void redo() {

    if (current\_index < history.size() - 1) {

        current\_index++;

        std::cout << "redo() => Redo berhasil\n";

    } else {

        std::cout << "redo() => Tidak dapat redo\n";

    }

}

int main() {

    // Memuat koordinat dari file saat program dimulai

    load();

    // Jika history kosong (misalnya, file tidak ada atau kosong), inisialisasi koordinat awal

    if (history.empty()) {

        history.push\_back(std::make\_pair(0.0, 0.0));

        current\_index = 0;

    } else {

        // Jika ada data dalam history, atur indeks saat ini ke posisi terakhir

        current\_index = history.size() - 1;

    }

    // Contoh penggunaan fungsi gerak\_2

    lokasi();

    gerak\_2(5.0, 3.0, M\_PI / 4);  // Kecepatan 5 kotak/s, waktu 3 detik, sudut 45 derajat

    lokasi();

    // Menyimpan koordinat ke dalam file saat program ditutup

    save();

    // Melakukan undo dan redo

    undo();

    lokasi();

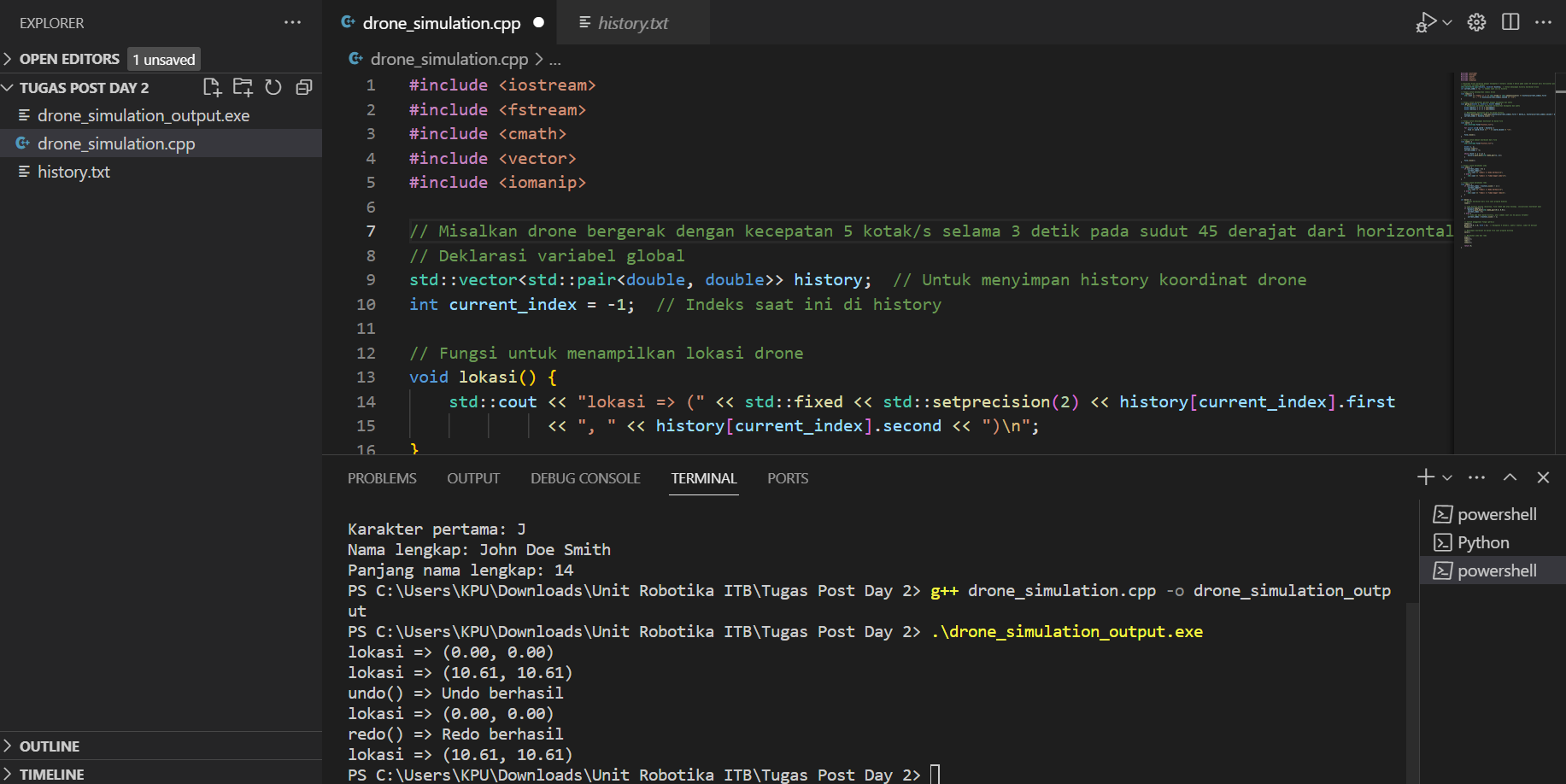
    redo();

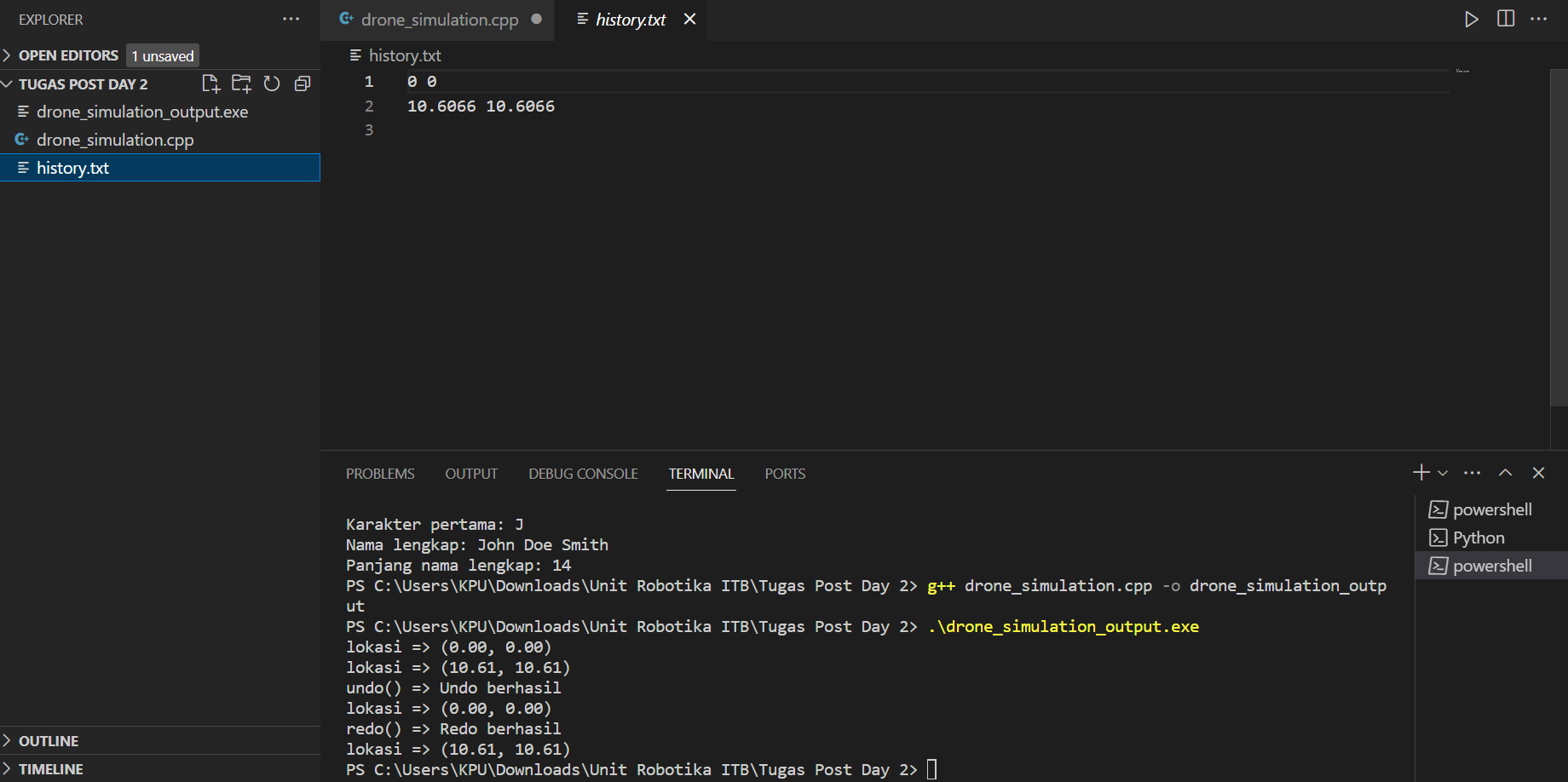
    lokasi();

    return 0;

}

**Dokumentasi Program Simulasi Drone**





Pada contoh penggunaan fungsi gerak\_2(5.0, 3.0, M\_PI / 4);, ini memisalkan bahwa drone bergerak dengan kecepatan 5 kotak per detik selama 3 detik pada sudut 45 derajat dari horizontal positif searah jarum jam.

Untuk menjelaskan lebih detail:

1. Kecepatan horizontal (x) akan dihitung dengan rumus v \* t \* cos(theta), di mana v adalah kecepatan, t adalah waktu, dan theta adalah sudut dalam radian.
2. Kecepatan vertikal (y) akan dihitung dengan rumus v \* t \* sin(theta).
3. Koordinat baru drone kemudian dihitung dengan menambahkan perubahan koordinat (delta x dan delta y) ke koordinat saat ini.
4. Dengan menggunakan nilai v = 5.0 (kecepatan 5 kotak/detik), t = 3.0 detik (waktu 3 detik), dan theta = M\_PI / 4 radian (45 derajat), fungsi gerak\_2 akan menghitung perubahan koordinat dan memperbarui posisi drone sesuai dengan gerakan yang dijelaskan.
5. **Overview:**
   * Program ini adalah simulasi pergerakan drone dengan fitur-fitur seperti penampilan lokasi, gerakan dengan kecepatan dan waktu, penyimpanan koordinat ke dalam file, pengambilan koordinat dari file, serta undo dan redo.
   * Program ditulis dalam bahasa C++ dan menggunakan beberapa fungsi dan struktur data, seperti vektor dan fungsi trigonometri.
   * Dimisalkan drone bergerak dengan kecepatan 5 kotak/s selama 3 detik pada sudut 45 derajat dari horizontal positif searah jarum jam
6. **Fitur-fitur Program:**
   * **lokasi():**
     + Menampilkan lokasi drone dengan koordinat x dan y saat ini.
   * **gerak\_2(double v, double t, double theta):**
     + Menggerakkan drone dengan kecepatan **v** kotak/s selama **t** detik pada sudut **theta** derajat dari horizontal positif searah jarum jam.
   * **save():**
     + Menyimpan koordinat drone ke dalam file "history.txt".
   * **load():**
     + Memuat koordinat drone dari file "history.txt".
   * **undo():**
     + Mengembalikan koordinat drone ke posisi sebelumnya dalam history.
   * **redo():**
     + Mengembalikan koordinat drone setelah perintah undo.
   * **main():**
     + Menginisialisasi program, memuat koordinat dari file jika ada, melakukan gerakan drone, menyimpan koordinat saat program ditutup, serta menunjukkan penggunaan undo dan redo.
7. **Cara Menjalankan Program:**
   * Nyalakan kompiler C++ seperti g++ yang sudah terinstal di VS Code
   * Salin kode program ke dalam berkas **drone\_simulation.cpp**.
   * Buka terminal atau command prompt.
   * Compile program dengan perintah **g++ drone\_simulation.cpp -o drone\_simulation\_output**.
   * Jalankan program dengan perintah **./drone\_simulation\_output** atau **drone\_simulation\_output.exe** (pada Windows).
   * Program akan menampilkan output sesuai dengan fungsi-fungsi yang diimplementasikan.