

## 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

The screenshot displays the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the file structure with 'TUGAS POST DAY 2' expanded, containing 'drone\_simulation\_output.exe', 'drone\_simulation.cpp', and 'history.txt'.
- EDITOR:** Displays 'drone\_simulation.cpp' with the following code:
 

```

1 #include <iostream>
2 #include <fstream>
3 #include <cmath>
4 #include <vector>
5 #include <iomanip>
6
7 // Misalkan drone bergerak dengan kecepatan 5 kotak/s selama 3 detik pada sudut 45 derajat dari horizontal
8 // Deklarasi variabel global
9 std::vector<std::pair<double, double>> history; // Untuk menyimpan history koordinat drone
10 int current_index = -1; // Indeks saat ini di history
11
12 // Fungsi untuk menampilkan lokasi drone
13 void lokasi() {
14     std::cout << "lokasi => (" << std::fixed << std::setprecision(2) << history[current_index].first
15     << ", " << history[current_index].second << ")\n";
16 }

```
- TERMINAL:** Shows the execution output:
 

```

Karakter pertama: J
Nama lengkap: John Doe Smith
Panjang nama lengkap: 14
PS C:\Users\KPU\Downloads\Unit Robotika ITB\Tugas Post Day 2> g++ drone_simulation.cpp -o drone_simulation_outp
ut
PS C:\Users\KPU\Downloads\Unit Robotika ITB\Tugas Post Day 2> .\drone_simulation_output.exe
lokasi => (0.00, 0.00)
lokasi => (10.61, 10.61)
undo() => Undo berhasil
lokasi => (0.00, 0.00)
redo() => Redo berhasil
lokasi => (10.61, 10.61)
PS C:\Users\KPU\Downloads\Unit Robotika ITB\Tugas Post Day 2>

```
- RIGHT SIDEBAR:** Includes a 'RUN AND DEBUG' section with a 'console' tab showing the same terminal output, and a 'TERMINAL' tab with a dropdown menu for 'powershell', 'Python', and 'powershell'.

```
drone_simulation.cpp  history.txt
1 0 0
2 10.6066 10.6066
3

Karakter pertama: J
Nama lengkap: John Doe Smith
Panjang nama lengkap: 14
PS C:\Users\KPU\Downloads\Unit Robotika ITB\Tugas Post Day 2> g++ drone_simulation.cpp -o drone_simulation_outp
ut
PS C:\Users\KPU\Downloads\Unit Robotika ITB\Tugas Post Day 2> .\drone_simulation_output.exe
lokasi => (0.00, 0.00)
lokasi => (10.61, 10.61)
undo() => Undo berhasil
lokasi => (0.00, 0.00)
redo() => Redo berhasil
lokasi => (10.61, 10.61)
PS C:\Users\KPU\Downloads\Unit Robotika ITB\Tugas Post Day 2> ]
```

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.