

## Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)  
Tahun / Semester : 2023-2024 / Genap  
Modul : Tugas Besar  
Nama Asisten / NIM : Agape D' Sky/23223031  
Nama Praktikan / NIM : Poggy Macello Gultom/13222102

---

### Log Book Tugas Besar

#### **Jumat, 10 Mei 2024: Definisi Ruang Lingkup Masalah**

Pada hari pertama, saya fokus pada mendefinisikan ruang lingkup masalah yang akan diselesaikan. Saya mulai dengan menjelaskan bahwa masalah utama adalah menemukan jalur dari titik awal ke titik akhir dalam sebuah labirin yang direpresentasikan dalam bentuk file teks. Saya mengidentifikasi beberapa kasus yang perlu ditangani: jika file masukkan tidak ada, program harus memberikan pesan kesalahan yang sesuai; jika tidak ada solusi untuk test case tertentu, program harus menginformasikan bahwa tidak ada jalur yang ditemukan; jika format masukkan tidak valid, program harus memberikan pesan error. Selain itu, saya menentukan bahwa labirin dengan ukuran kurang dari 7x7 tidak akan ditangani oleh program saya.

#### **Minggu, 12 Mei 2024: Merancang Interaksi Pengguna dan Struktur Data**

Pada hari kedua, saya merancang bagaimana pengguna dapat berinteraksi dengan program. Pengguna akan memasukkan nama file yang berisi labirin, dan program akan menampilkan semua jalur yang mungkin, jalur terpendek, dan jalur terpanjang dari titik awal ke titik akhir. Saya memecah proses utama program menjadi beberapa proses yang lebih kecil, seperti pembacaan file, validasi posisi, pencarian jalur, dan pencetakan jalur. Saya juga menggambarkan proses-proses tersebut menggunakan Diagram Alir Data (DFD) dan merancang algoritma Greedy untuk menyelesaikan masalah ini.

#### **Senin, 13 Mei 2024: Merancang Flowchart dan Mendefinisikan Struktur Data**

Pada hari ketiga, saya membuat flowchart untuk setiap proses yang telah saya tentukan sebelumnya. Flowchart ini membantu saya memahami alur kerja program dengan lebih baik dan memastikan bahwa setiap langkah sudah terdefinisi dengan jelas. Saya juga mendefinisikan struktur data yang akan digunakan, seperti `Point` untuk menyimpan koordinat `(x, y)` dan `Path` untuk menyimpan jalur yang terdiri dari array `Point` dan panjang jalur.

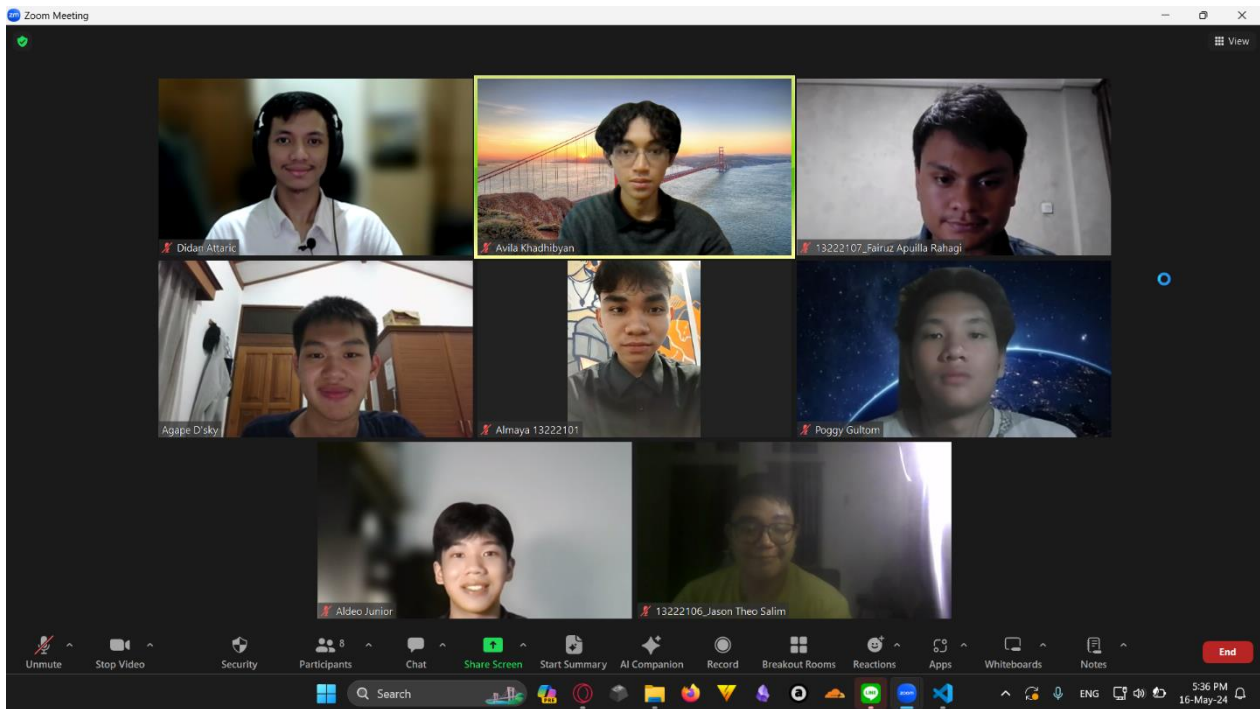
#### **Selasa, 14 Mei 2024: Implementasi Pembacaan Labirin dan Validasi Posisi**

Pada hari keempat, saya mulai mengimplementasikan program dengan fokus pada pembacaan labirin dari file teks dan validasi posisi. Saya menulis fungsi `readMaze` untuk membaca file, mengisi array `maze`, dan menemukan posisi `start` dan `end`. Selain itu, saya menulis fungsi `isValid` untuk memeriksa apakah suatu sel dalam labirin dapat dilalui, yaitu sel tersebut harus berada dalam batas labirin, bukan dinding (`#`), dan belum pernah dikunjungi.

#### **Kamis, 16 Mei 2024: Asistensi dengan Kak Agape D Sky**

Pada hari kelima, saya mendapatkan asistensi dengan Kak Agape D Sky, asisten kami yang membantu menjelaskan konsep-konsep yang masih membingungkan terkait implementasi algoritma – algoritma kami masing masing dan struktur data yang digunakan. Kami membahas tentang bagaimana mengoptimalkan pencarian jalur dan memastikan bahwa setiap langkah dalam algoritma diimplementasikan dengan benar.

## Dokumentasi :

**Sabtu, 18 Mei 2024: Implementasi Algoritma Pencarian Jalur Greedy**

Pada hari keenam, dengan bantuan dari asistensi sebelumnya, saya mengimplementasikan algoritma pencarian jalur menggunakan metode Greedy. Saya menulis fungsi `findPaths` yang akan mencari semua jalur dari posisi `start` ke `end`. Selain itu, saya menambahkan fungsi `copyPath` untuk menyalin jalur dan `addPath` untuk menambahkan jalur yang ditemukan ke daftar jalur.

**Minggu, 19 Mei 2024: Pengujian Awal, Debugging, dan Pengukuran Waktu Eksekusi**

Pada hari ketujuh, saya melakukan pengujian awal terhadap fungsi-fungsi yang sudah saya buat. Saya mencoba membaca file labirin dan mencari jalur dari `start` ke `end`. Jika terdapat bug atau masalah, saya melakukan debugging untuk memperbaikinya. Selain itu, saya menambahkan pengukuran waktu eksekusi menggunakan fungsi `gettimeofday` untuk melihat berapa lama waktu yang dibutuhkan untuk menemukan semua jalur dari `start` ke `end`. Saya melakukan diskusi dengan teman satu kelompok, dan didapat bahwa alangkah baiknya menggunakan kode dibawah ini untuk `gettimeofday`.

```
struct timeval begin, end;
gettimeofday(&begin, 0);
aStarSearch(maze, row, col, start, dest);
gettimeofday(&end, 0);

long sec = end.tv_sec - begin.tv_sec;
long us = end.tv_usec - begin.tv_usec;
double elapsed = sec * 1e3 + us * 1e-3;
printf("Time: %lf ms\n", elapsed);
```

Saya mengukur waktu mulai sebelum memanggil fungsi `findPaths` dan waktu berakhir setelah pencarian selesai, lalu menghitung selisihnya.

**Senin, 20 Mei 2024 : Pencetakan Jalur, Penentuan Jalur Terpendek dan Terpanjang, serta Pengujian Lanjutan**

Pada hari kedelapan, saya fokus pada pencetakan jalur yang ditemukan dan menentukan jalur terpendek serta terpanjang. Saya menulis fungsi `printPath` untuk mencetak setiap jalur yang ditemukan dengan format yang mudah dibaca. Saya iterasi melalui semua jalur yang ditemukan, membandingkan

panjangnya, dan menyimpan referensi ke jalur terpendek dan terpanjang. Saya juga melakukan pengujian lebih lanjut dengan berbagai kasus labirin untuk memastikan bahwa program berjalan dengan baik dan dapat menangani berbagai ukuran dan kompleksitas labirin. Setelah semua pengujian selesai, saya memperbaiki dan membersihkan kode.

**Senin, 20 Mei 2024 : Pengerjaan Laporan dan Diskusi Algoritma terbaik**

Saya mengerjakan laporan akhir yang mencakup deskripsi masalah, pendekatan yang digunakan, hasil pengujian, dan analisis performa algoritma. Saya mendiskusikan hasil yang telah saya capai dengan teman teman, terutama mengenai efektivitas dan efisiensi algoritma Greedy dibandingkan dengan algoritma lainnya seperti DFS, BFS, A\* dan lain lain. Kami membandingkan kelebihan dan kelemahan masing-masing algoritma dalam konteks labirin dengan ukuran dan kompleksitas yang berbeda.