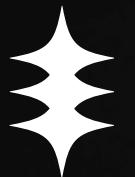
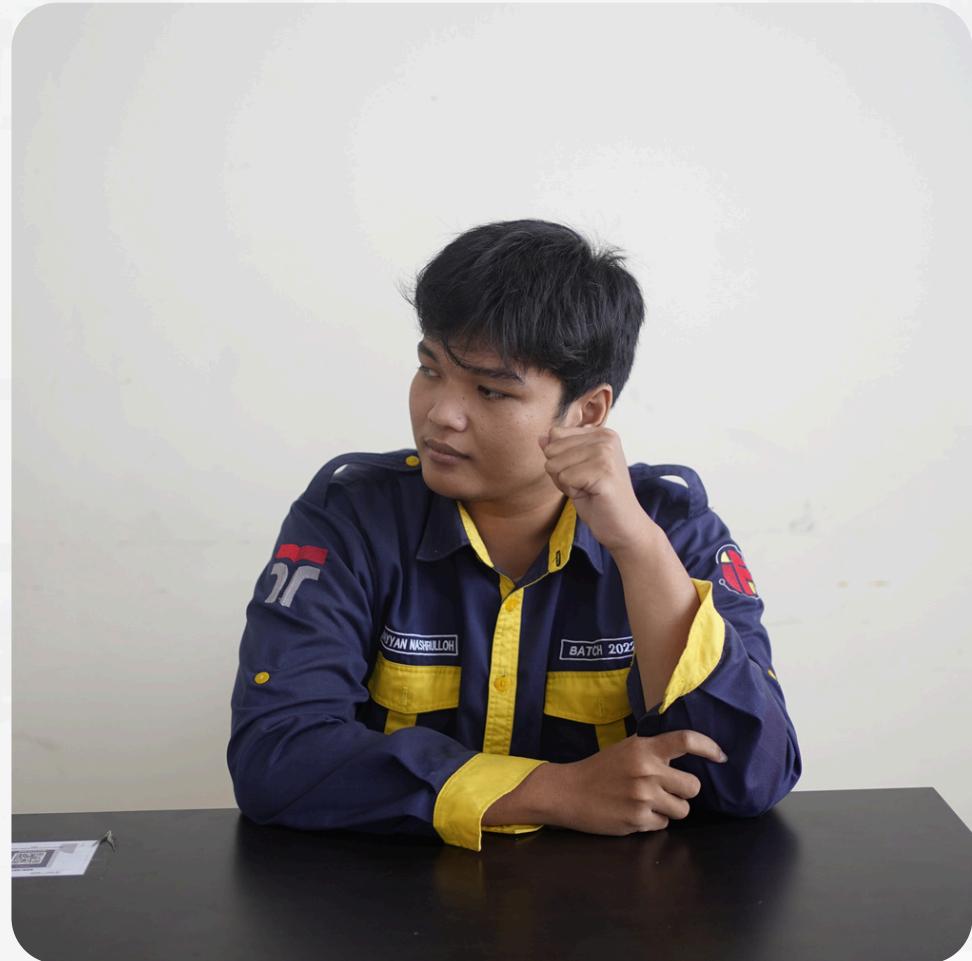


PORTOFOLIO

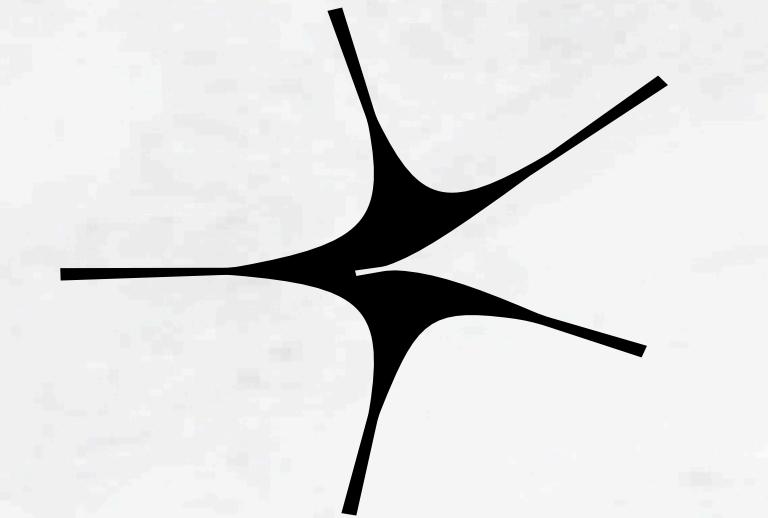


INTRODUCTION

02



Halo ! Saya Hayyan
Nashrulloh seorang
mahasiswa dari Institut
Teknologi Telkom Purwokerto
dari jurusan Teknik
Informatika



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="title" content="Page" />
<meta name="description" content="Description" />
<meta name="keywords" content="Keywords" />
</head>
<body>
<div class="feature-item">
<div class="navbar-header">
<header class="container">
<div class="navbar">
<nav class="navbar">
<ul class="nav">
```

KETERTARIKAN

Saya memiliki ketertarikan dalam bidang pemrograman Website, sehingga saya mempelajari beberapa bahasa pemrograman untuk membuat website seperti HTML, CSS, JavaScript walaupun saya rasa masih belum ahli, namun saya akan berusaha menjadi lebih baik lagi .

PENGALAMAN KERJA



ANGGOTA KELOMPOK PENYELENGGARA
PEMUNGUTAN SUARA(KPPS)-14 FEBRUARI
2024

1. Bertanggung jawab dalam proses pengorganisasian pemungutan suara pada hari pemilihan, termasuk persiapan dan penyelesaian tugas dengan cermat.
2. Memastikan proses pemilihan berjalan lancar, melakukan perhitungan suara dengan teliti dan terlibat dalam pelaporan hasil pemilihan keada instansi terkait sesuai dengan dengan prosedur yang berlaku

MUNGKIN BAPAK ATAU IBU MERASA PENGALAMAN KERJA INI TIDAK ADA HUBUNGANNYA DENGAN PEMROGRAMAN, NAMUN SAYA RASA PENGALAMAN KERJA INI DAPAT DI PERTIMBANGKAN UNTUK PERTIMBANGAN SOFT SKILL CONTOHNYA DAPAT BEKERJA DALAM TIM, MANAJEMEN WAKTU DAN PEMECAHAN MASALAH.

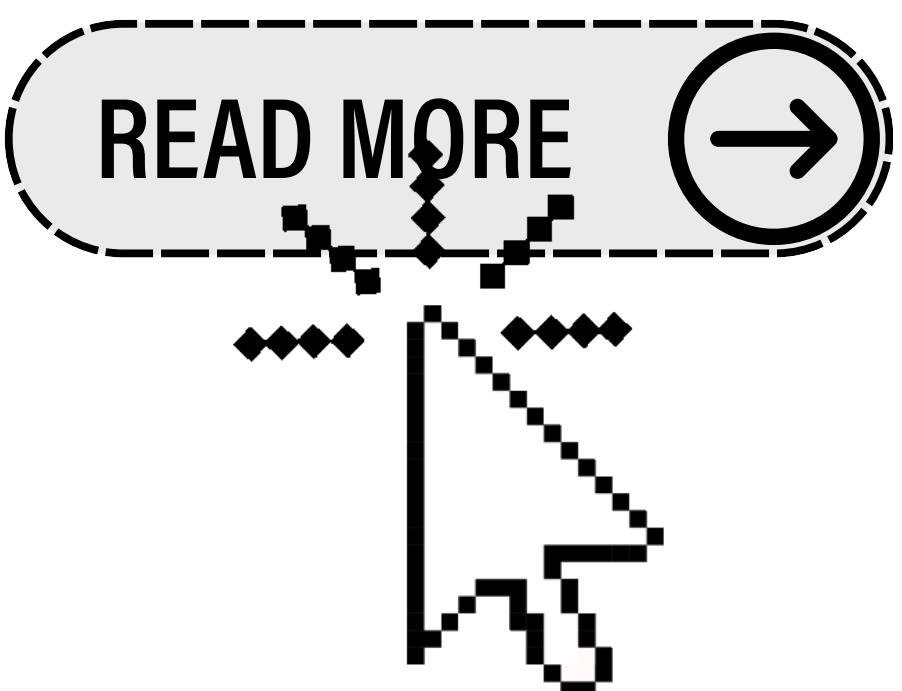


Personal Skills

- ◆ **PROGRAMMING LANGUAGE**
Java, HTML, CSS, JavaScript
- ◆ **TOOL SKILL**
Figma
- ◆ **SOFT SKILL**
Dapat bekerja dalam tim, Manajemen waktu.

PROJECT PORTOFOLIO

- Submission akhir Dicoding kelas AWS Back End Academy
- Submission akhir modul Dicoding Dasar Pemrograman Web
- Tugas besar Struktur data



PROJECT 01 MEMBUAT APLIKASI BACK END

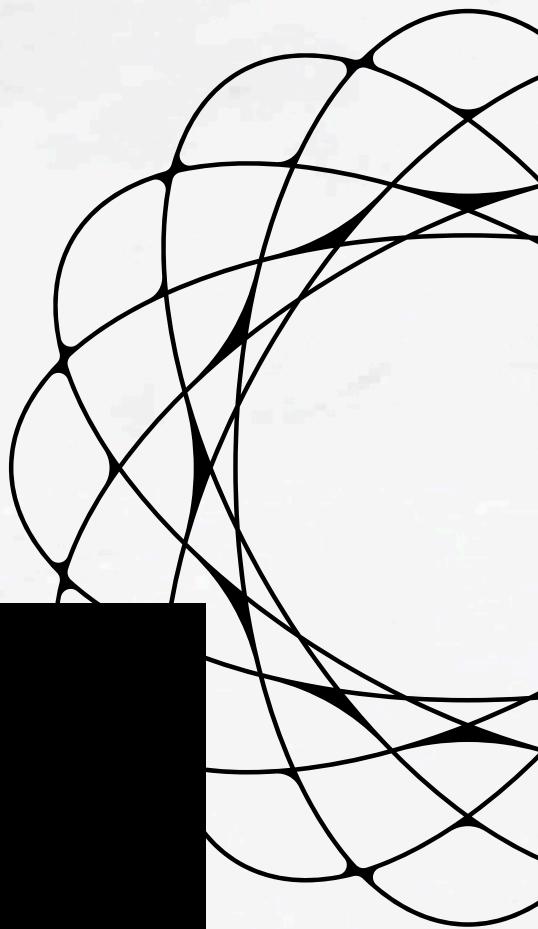
Projek ini adalah sebuah API untuk manajemen data buku yang memungkinkan pengguna untuk menambahkan, membaca, memperbarui, dan menghapus buku. Dibangun menggunakan Node.js dengan bantuan Hapi.js, API ini memiliki struktur direktori yang terorganisir dengan baik untuk memisahkan logika aplikasi. Dengan penanganan kesalahan yang baik dan respons informatif, pengguna dapat dengan mudah mengelola daftar buku mereka dengan API ini.

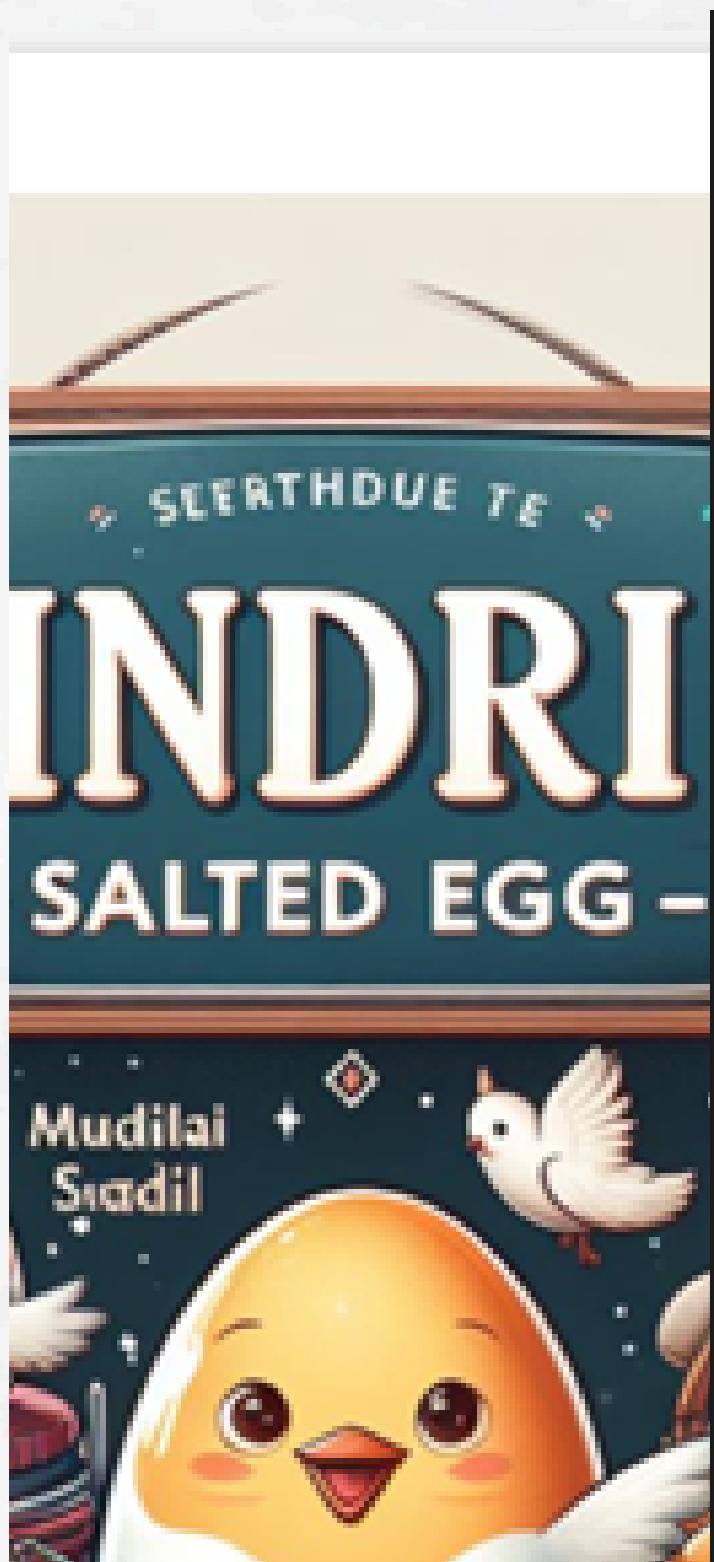
```
    string genre;
    int jumlahEpisode;
    Anime* prev;
    Anime* next;
};

class AnimeList {
private:
    Anime* head;
    Anime* tail;
    stack<Anime*> animeStack;
    queue<Anime*> animeQueue;
public:
    AnimeList() {
        head = nullptr;
        tail = nullptr;
    }
    void tambahAnime(Anime anime) {
        Anime* newNode = new Anime;
```

```
        getline(cin, anime->genre);
        cout << "Masukkan jumlah episode baru: ";
        cin >> anime->jumlahEpisode;
    } else {
        cout << "Anime dengan judul " << judul << " tidak
ditemukan." << endl;
    }
}

void hapusAnime(string judul) {
    if (head == nullptr) {
        cout << "Anime tidak ditemukan." << endl;
        return;
    }
    Anime* current = head;
    while (current != nullptr) {
        if (current->judul == judul) {
            if (current == head && current == tail) {
                delete current;
                head = nullptr;
```





PROJECT 02 DASAR PEMROGRAMAN WEB

CLIENT : DICODING

Program ini adalah sebuah halaman web yang dibuat untuk mempromosikan dan menjual produk telur asin dari "Indri Telur Asin". Halaman ini mencakup berbagai informasi tentang produk, lokasi toko, diskon, dan cara menghubungi penjual melalui WhatsApp. Program ini adalah submission akhir dari modul belajar Dicoding Dasar Pemrograman Web. Berikut adalah komponen-komponen utama dari program ini adalah HTML Struktur, CSS Styling dan JavaScript Interaktivitas.

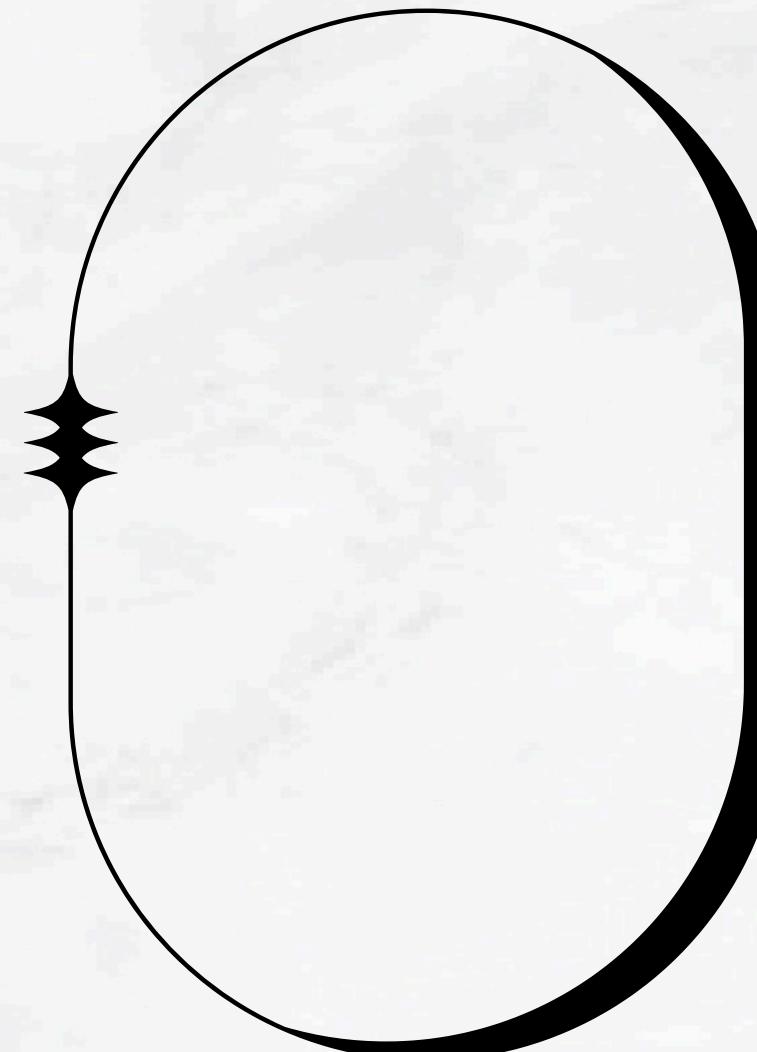
PROJECT 03 TUGAS BESAR STRUKTUR DATA

Proyek ini bertujuan untuk membuat aplikasi sederhana yang dapat digunakan untuk mencatat daftar anime. Aplikasi ini menggunakan struktur data seperti Queue dan Binary Tree untuk mengelola data anime. Struktur data Queue Digunakan untuk menyimpan daftar anime dalam urutan masuk terakhir keluar pertama (FIFO), sementara struktur data Binary Tree digunakan untuk menyimpan daftar anime dalam urutan terurut berdasarkan judul.

```
string genre;
int jumlahEpisode;
Anime* prev;
Anime* next;
};

class AnimeList {
private:
    Anime* head;
    Anime* tail;
    stack<Anime*> animeStack;
    queue<Anime*> animeQueue;
public:
    AnimeList() {
        head = nullptr;
        tail = nullptr;
    }
    void tambahAnime(Anime anime) {
        Anime* newNode = new Anime;
        newNode->genre = anime.genre;
        newNode->jumlahEpisode = anime.jumlahEpisode;
        if (head == nullptr) {
            head = newNode;
            tail = newNode;
        } else {
            tail->next = newNode;
            newNode->prev = tail;
            tail = newNode;
        }
    }
    void hapusAnime(string judul) {
        if (head == nullptr) {
            cout << "Anime tidak ditemukan." << endl;
            return;
        }
        Anime* current = head;
        while (current != nullptr) {
            if (current->judul == judul) {
                if (current == head && current == tail) {
                    delete current;
                    head = nullptr;
                    tail = nullptr;
                } else if (current == head) {
                    head = current->next;
                    current->next->prev = nullptr;
                } else if (current == tail) {
                    tail = current->prev;
                    current->prev->next = nullptr;
                } else {
                    current->prev->next = current->next;
                    current->next->prev = current->prev;
                }
                delete current;
            }
            current = current->next;
        }
    }
};
```

**LET'S
WORK
TOGETHER**



+62-895-2989-2817

hayyannashrulloh@gmail.com