

**PROPOSAL PENGEMBANGAN APLIKASI MANAJEMEN RENCANA  
PEMBANGUNAN PROYEK DENGAN STRUKTUR DATA QUEUE PRIORITAS DAN  
HASH TABLE**

**MATAKULIAH STRUKTUR DATA**

**UNIVERSITAS JEMBER**



Diajukan oleh:

Mohammad Rizky Ramdhani (222410103042)

Ahmad Ilyas (222410103049)

Paulus Aditya Wicaksono (222410103052)

Dosen pembimbing:

Januar Adi Putra S.Kom, M.Kom

Tio Dharmawan S.Kom, M.Kom

Semester 3

Kata Kunci: Manajemen Proyek, Rencana Pembangunan, Aplikasi Teknologi Informasi, Queue Prioritas, Hash Table, Struktur Data, Pengelolaan Informasi, Kecepatan Akses, Efisiensi, Implementasi Sistem, Pengujian Aplikasi, Pembelajaran, Teknologi Informasi, Kelancaran Proyek, Manajemen Informasi.

## DAFTAR ISI

## PENDAHULUAN

Dalam era kemajuan teknologi informasi, pengelolaan proyek menjadi suatu aspek yang krusial untuk memastikan kelancaran dan efisiensi suatu pembangunan. Melalui penerapan teknologi informasi, aplikasi rencana pembangunan proyek dapat menjadi solusi cerdas dalam mendukung proses perencanaan dan pelaksanaan proyek.

Laporan ini merinci langkah-langkah dalam merancang dan mengimplementasikan aplikasi tersebut, dengan menggunakan struktur data berupa queue prioritas dan hash table. Adopsi struktur data ini diharapkan dapat meningkatkan kecepatan akses dan pengelolaan informasi terkait rencana pembangunan proyek.

### 1.1. Latar Belakang

Pada era perkembangan teknologi informasi, pengelolaan proyek menjadi tantangan utama dalam memastikan keberhasilan dan efisiensi suatu pembangunan. Manajemen rencana pembangunan proyek memerlukan pendekatan yang cermat untuk mengatasi kompleksitas dan dinamika yang melekat pada proses tersebut. Dalam konteks ini, penerapan aplikasi berbasis teknologi informasi menjadi kebutuhan yang mendesak untuk meningkatkan transparansi, kecepatan akses informasi, dan pengambilan keputusan yang tepat.

Melalui penggabungan struktur data queue prioritas dan hash table, aplikasi ini bertujuan untuk memberikan solusi cerdas dalam pengelolaan rencana pembangunan. Queue prioritas digunakan untuk mengatur prioritas tugas, memastikan penanganan yang efisien, sementara hash table memfasilitasi penyimpanan informasi tambahan yang relevan dengan setiap rencana. Dengan demikian, aplikasi ini diharapkan dapat mengoptimalkan proses perencanaan proyek, meningkatkan kualitas pengambilan keputusan, dan memberikan kontribusi positif terhadap keseluruhan kelancaran pelaksanaan proyek pembangunan.

### 1.2. Tujuan Laporan

#### 1) **Memperkenalkan Konsep Aplikasi**

Memberikan pemahaman yang komprehensif terkait konsep dan desain aplikasi manajemen rencana pembangunan proyek menggunakan struktur data queue prioritas dan hash table.

#### 2) **Menyajikan Implementasi Praktis**

Menguraikan langkah-langkah implementasi aplikasi secara praktis, termasuk rinciannya dalam mengadopsi struktur data yang sesuai dengan kebutuhan manajemen proyek.

### 3) **Menganalisis Keuntungan Penggunaan Struktur Data**

Menjelaskan manfaat penggunaan struktur data queue prioritas dan hash table dalam konteks pengelolaan rencana pembangunan proyek, dengan fokus pada kecepatan akses dan efisiensi.

### 4) **Memberikan Panduan Penggunaan**

Memberikan panduan praktis tentang cara menggunakan aplikasi, termasuk langkah-langkah penambahan, penghapusan, dan pengecekan detail rencana.

Dengan demikian, laporan ini bertujuan memberikan kontribusi pemahaman mendalam terhadap implementasi aplikasi serta memberikan landasan bagi pengembangan dan peningkatan berkelanjutan di bidang manajemen rencana pembangunan proyek.

## 1.3. Ruang Lingkup Aplikasi

### 1) **Desain Aplikasi**

- Pengembangan aplikasi manajemen rencana pembangunan proyek dengan fokus pada penggunaan struktur data queue prioritas dan hash table.
- Penjelasan desain umum aplikasi dan alur kerja pengguna.

### 2) **Implementasi Struktur Data**

- Penggunaan queue prioritas untuk mengelola prioritas rencana pembangunan.
- Penerapan hash table untuk penyimpanan informasi tambahan terkait setiap rencana.

### 3) **Fungsi Utama Aplikasi:**

- **add\_plan(nama\_rencana, prioritas, detail\_rencana):** Menambahkan rencana baru ke dalam antrian prioritas dan hash.
- **get\_next\_plan():** Mengambil dan menampilkan rencana berikutnya dengan prioritas tertinggi.
- **get\_plan\_details(nama\_rencana):** Menampilkan detail lengkap dari suatu rencana berdasarkan nama rencana.
- **remove\_plan(nama\_rencana):** Menghapus rencana dari antrian prioritas dan hash.

### 4) **Pengujian Aplikasi:**

- Rincian strategi pengujian yang mencakup keandalan antrian prioritas, kecocokan hasil pencarian pada hash table, dan pengujian fungsionalitas utama.
- Analisis hasil pengujian dan temuan.

## 5) Pembahasan dan Kesimpulan:

- Evaluasi keberhasilan implementasi dan dampak penggunaan struktur data terpilih.
- Kesimpulan dari analisis pengujian dan temuan.

Dengan membatasi ruang lingkup pada poin-poin tersebut, laporan ini bertujuan untuk memberikan gambaran yang komprehensif terhadap implementasi dan kegunaan aplikasi manajemen rencana pembangunan proyek ini.

### 1.4. Desain Umum Aplikasi

Aplikasi manajemen rencana pembangunan proyek ini dirancang untuk memberikan penggunaan yang intuitif dan efisien. Berikut adalah desain umum aplikasi, mencakup antarmuka pengguna dan alur kerja utama:

#### 1) Antarmuka Pengguna:

##### a) Beranda (Home)

- Tampilkan rencana berikutnya dengan prioritas tertinggi.
- Sediakan tombol atau opsi untuk menambahkan rencana baru.

##### b) Tambah Rencana (Add Plan)

- Form input untuk nama rencana, prioritas, dan detail rencana.
- Tombol "Tambahkan" untuk memasukkan rencana ke dalam antrian prioritas.

##### c) Lihat Rencana (View Plan)

- Tampilkan daftar rencana dengan prioritas dan opsi untuk melihat detail.
- Pilihan untuk menghapus rencana setelah selesai atau tidak relevan.

##### d) Detail Rencana (Plan Details)

- Tampilkan informasi lengkap tentang suatu rencana, termasuk prioritas dan detail.

#### 2) Alur Kerja Aplikasi

##### a) Tambahkan Rencana

- Pengguna membuka aplikasi dan memilih opsi "Tambah Rencana".
- Pengguna mengisi formulir dengan nama rencana, prioritas, dan detail rencana.

- Aplikasi menyimpan rencana ke dalam antrian prioritas dan hash table.

**b) Tampilkan Rencana Berikutnya**

- Pengguna melihat beranda untuk mendapatkan rencana berikutnya dengan prioritas tertinggi.
- Aplikasi menampilkan rencana tersebut dengan opsi untuk melihat detail.

**c) Lihat dan Hapus Rencana**

- Pengguna memilih opsi "Lihat Rencana" untuk melihat daftar rencana.
- Dari daftar, pengguna bisa melihat detail atau menghapus rencana yang sudah selesai.

**d) Melihat Detail Rencana**

- Pengguna memilih suatu rencana untuk melihat detail lengkap.
- Aplikasi menampilkan prioritas dan informasi tambahan terkait rencana tersebut.

**e) Menghapus Rencana**

- Pengguna memilih opsi "Lihat Rencana" dan memilih rencana yang ingin dihapus.
- Aplikasi menghapus rencana dari antrian prioritas dan hash table.

Desain umum ini bertujuan untuk memberikan pengalaman pengguna yang sederhana namun efektif dalam mengelola rencana pembangunan proyek. Dengan fokus pada kejelasan informasi dan kemudahan penggunaan, diharapkan aplikasi ini dapat menjadi alat yang efisien bagi manajer proyek dan tim pengembangan.

## PERENCANAAN APLIKASI

Pada tahap perencanaan aplikasi ini, diputuskan untuk menggunakan bahasa pemrograman Python sebagai basis pengembangan. Keputusan ini didasarkan pada keluwesan dan kejelasan sintaksis Python, yang membuatnya menjadi pilihan yang populer untuk pengembangan aplikasi dengan tingkat kompleksitas beragam.

Selain itu, untuk memastikan antarmuka pengguna (GUI) aplikasi memiliki tampilan yang modern, intuitif, dan mudah digunakan, rencananya adalah memanfaatkan modul GUI modern yang telah terbukti efektif. Sebagai contoh, modul seperti Tkinter, PyQt, atau Kivy dapat menjadi pilihan yang sesuai. Pemilihan modul ini akan bergantung pada kebutuhan spesifik dan preferensi pengembang.

### 2.1. Tujuan Aplikasi

Aplikasi ini bertujuan untuk meningkatkan efisiensi dan kecepatan pengelolaan rencana pembangunan proyek melalui penerapan teknologi informasi. Dengan menggabungkan struktur data queue prioritas dan hash table, tujuan utama aplikasi adalah memberikan solusi yang cerdas dalam perencanaan, pengelolaan prioritas, dan pelaporan progres proyek.

### 2.2. Fitur Utama

#### 1) **Tambah Rencana**

Pengguna dapat dengan mudah menambahkan rencana baru dengan menyediakan informasi seperti nama, prioritas, dan detail rencana. Fitur ini dirancang untuk memudahkan pengguna dalam merencanakan tugas atau proyek baru dengan cepat dan akurat. Dengan formulir input yang jelas, pengguna dapat dengan mudah mengisi detail rencana dan menambahkannya ke dalam antrian prioritas.

#### 2) **Tampilkan Rencana**

Aplikasi menyajikan rencana berikutnya dengan prioritas tertinggi, memungkinkan pengguna untuk fokus pada tugas yang paling penting. Fitur ini membantu pengguna dalam menentukan langkah selanjutnya dengan jelas dan efisien. Dengan memberikan visibilitas terhadap rencana yang memiliki prioritas tinggi, pengguna dapat mengoptimalkan waktu dan sumber daya mereka.

#### 3) **Tampilkan detail Rencana**

Pengguna dapat melihat detail lengkap dari suatu rencana, termasuk prioritas dan informasi tambahan terkait. Fitur ini memberikan pengguna pemahaman yang mendalam tentang setiap rencana, memungkinkan mereka untuk mengambil

keputusan yang terinformasi dan menyeluruh. Dengan informasi tambahan yang tersedia, pengguna dapat merencanakan eksekusi rencana dengan lebih baik.

#### 4) **Hapus Rencana**

Setelah selesai atau tidak relevan, pengguna dapat menghapus rencana dari antrian, menjaga kebersihan dan keteraturan rencana. Fitur ini memastikan bahwa antrian prioritas tetap terorganisir dan hanya berisi rencana-rencana yang masih relevan. Pengguna dapat dengan mudah mengelola daftar rencana mereka dengan menghapus yang sudah selesai atau tidak diperlukan lagi.

### 2.3. Alur Kerja Umum

#### 1) **Tambah Rencana**

- a) Pengguna membuka aplikasi dan memilih opsi "Tambah Rencana".
- b) Mengisi formulir dengan nama rencana, prioritas, dan detail.
- c) Aplikasi menyimpan rencana ke dalam antrian prioritas dan hash table.

#### 2) **Tampilkan Rencana**

- a) Pengguna melihat beranda untuk mendapatkan rencana berikutnya dengan prioritas tertinggi.
- b) Aplikasi menampilkan rencana tersebut dengan opsi untuk melihat detail.

#### 3) **Tampilkan detail Rencana**

- a) Pengguna memilih suatu rencana untuk melihat detail lengkap.
- b) Aplikasi menampilkan prioritas dan informasi tambahan terkait rencana tersebut.

#### 4) **Hapus Rencana**

- a) Pengguna memilih opsi "Lihat Rencana" dan memilih rencana yang ingin dihapus.
- b) Aplikasi menghapus rencana dari antrian prioritas dan hash table.



## IMPLEMENTASI STRUKTUR DATA

### 3.1. Queue Prioritas

Queue Prioritas adalah struktur data linear yang mirip dengan queue (antrian), namun setiap elemennya memiliki prioritas tertentu. Elemen dengan prioritas lebih tinggi akan diproses sebelum elemen dengan prioritas lebih rendah.

#### Implementasi Kode

```
1 class PriorityQueue:
2     def __init__(self):
3         self.queue = []
4
5     def is_empty(self):
6         return len(self.queue) == 0
7
8     def enqueue(self, item, priority):
9         element = (item, priority)
10        if self.is_empty():
11            self.queue.append(element)
12        else:
13            for i in range(len(self.queue)):
14                if self.queue[i][1] < priority:
15                    self.queue.insert(i, element)
16                    return
17            self.queue.append(element)
18
19    def dequeue(self):
20        if not self.is_empty():
21            return self.queue.pop(0)
22        else:
23            return "Queue is empty"
24
25    def peek(self):
26        if not self.is_empty():
27            return self.queue[0]
28        else:
29            return "Queue is empty"
30
31    def size(self):
32        return len(self.queue)
```

### 3.2. Hash Table

Hash Table adalah struktur data yang menyimpan data dalam bentuk pasangan kunci-nilai. Fungsi hash digunakan untuk mengonversi kunci menjadi indeks dalam tabel, memungkinkan akses data dengan waktu konstan.

#### Implementasi Kode

```
1 class HashTable:
2     def __init__(self, size):
3         self.size = size
4         self.table = [None] * size
5
6     def hash_function(self, key):
7         return hash(key) % self.size
8
9     def insert(self, key, value):
10        index = self.hash_function(key)
11        if self.table[index] is None:
12            self.table[index] = [(key, value)]
13        else:
14            for pair in self.table[index]:
15                if pair[0] == key:
16                    pair[1] = value
17                return
18            self.table[index].append((key, value))
19
20    def search(self, key):
21        index = self.hash_function(key)
22        if self.table[index] is not None:
23            for pair in self.table[index]:
24                if pair[0] == key:
25                    return pair[1]
26        return None
27
28    def delete(self, key):
29        index = self.hash_function(key)
30        if self.table[index] is not None:
31            for i, pair in enumerate(self.table[index]):
32                if pair[0] == key:
33                    self.table[index].pop(i)
34            return
35
```