Purwadhika
Digital Technology School

2025

Presentasi Capstone Modul 1

Dipresentasikan Oleh:
Sulaeman Nurhakim

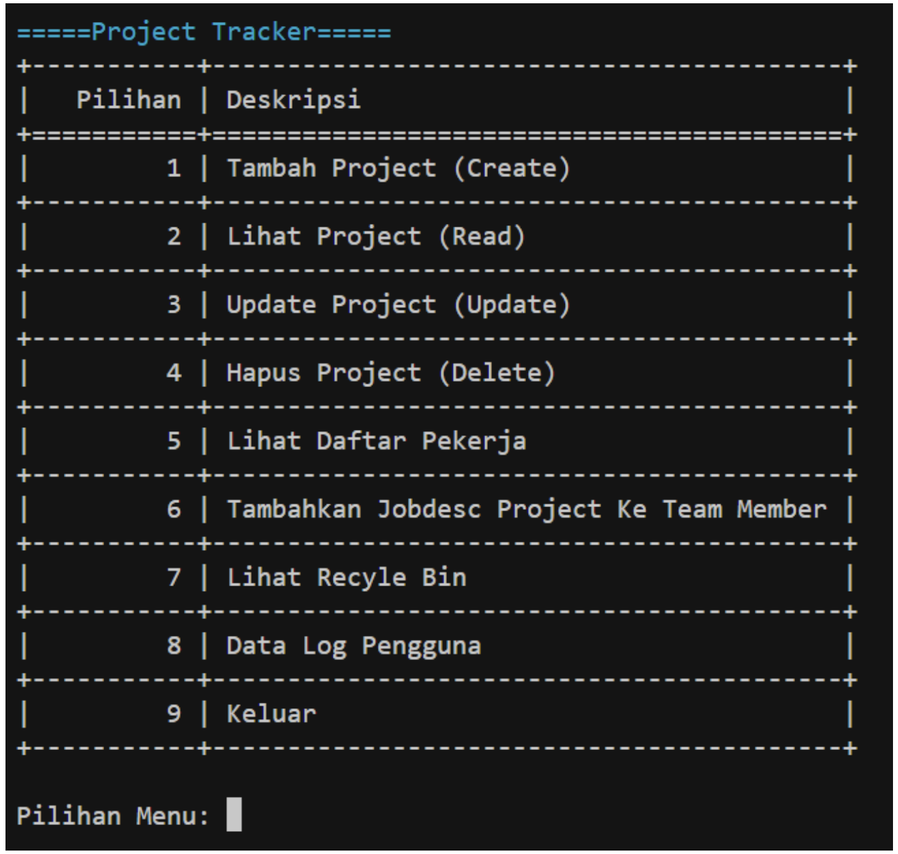
Program Bootcamp:
JCDS-2804-001

Pendahuluan

Latar Belakang dan Tujuan

- Membuat sebuah program sederhana berbasis phyton untuk melakukan project tracker bagi sebuah team untuk dapat mengawal satu atau banyaknya project yang sedang dilakukan.

- Ditujukan untuk melihat seberapa jauh pemahaman student mengerti bahasa pemograman phyton



Fitur Create

```python
def create_project(user_id):
    '''Menambahkan Project Baru'''
    project_id = input("Masukkan ID Project: ").strip()

    if project_id in projects:
        print("ID Project sudah ada")
        return

    name = input("Masukkan Nama Project: ").strip()
    deadline = input("Masukkan Deadline Project (YYYY-MM-DD): ").strip()
    status = input("Masukkan Status Project (Belum Dimulai/ Berjalan/ Selesai): ").strip()

    projects[project_id] = {
        "name": name,
        "deadline": deadline,
        "status": status,
        "owner" : user_id,
        "assigned_members" : [],
        "tasks" : {}
    }
    print("Project Berhasil Ditambahkan!")
    save_projects()
```

Fitur Read

```python
def read_project():
    '''Menampilkan Semua Project'''
    if not projects:
        print("Belum ada Project")
        return

    project_data = []
    for project_id, project in projects.items():
        countdown = calculate_deadline_countdown(project.get('deadline', "Not Set"))
        assigned_members = project.get('assigned_members', [])

        team_members = ', '.join([users[member_id]['name'] for member_id in assigned_members if member_id in users]) if assigned_members else "Tidak ada anggota"

        tasks = ', '.join([task["Name"] for task in project.get("tasks", {}).values()])
        total_tasks = len(project.get("tasks", {}))
        completed_tasks = sum(1 for task in project.get("tasks", {}).values() if task["Status"] == "Selesai")
        progress = (completed_tasks / total_tasks * 100) if total_tasks > 0 else 0

        project_data.append([
            project_id,
            project['name'],
            project['deadline'],
            countdown,
            project['status'],
            project['owner'],
            tasks,
            team_members,
            f"{progress:.2f}%"
        ])

    headers = ["ID Project", "Nama Project", "Deadline Project", "Countdown", "Status Project", "Owner", "Pekerjaan", "Team Members", "Progress"]
    print(tabulate(project_data, headers, tablefmt="grid"))
```

Fitur Update dan Delete

```python
def update_project(user_id, role):
    '''Memperbarui Status Project'''
    project_id = input("Masukkan ID Project yang akan diperbarui: ").strip()

    if project_id not in projects:
        print("ID Project tidak ditemukan")
        return

    if role == "Team Member" and projects[project_id]["owner"] != user_id:
        print("Anda tidak bisa memperbarui project ini karena Anda bukan pemiliknya.")
        return

    status = input("Masukkan Status Project Baru (Belum Dimulai/ Berjalan/ Selesai): ").strip()
    if status not in ["Belum Dimulai", "Berjalan", "Selesai"]:
        print("Status tidak valid. Pilih dari: Belum Dimulai, Berjalan, atau Selesai.")
        return

    projects[project_id]["status"] = status
    print("Status Project Berhasil Diperbarui!")
    save_projects()
```

```python
def delete_project():
    '''Menghapus Project'''
    project_id = input("Masukkan ID Project yang akan dihapus: ").strip()

    if project_id not in projects:
        print("ID Project tidak ditemukan")
        return

    recycle_bin[project_id] = projects.pop(project_id)
    print("Project Berhasil Dihapus!")
    save_projects()
```

Fitur Update dan Delete

```
def restore_project():
    if not recycle_bin:
        print("Recycle Bin Kosong")
        return
    print("\n=====Daftar Project di Recycle Bin")
    project_data = [[pid, p['name'], p['deadline'], p['status'], p['owner']] for pid, p in recycle_bin.items()]
    print(tabulate(project_data, ["ID Project", "Nama Project", "Deadline", "Status", "Owner"], tablefmt="grid"))
    project_id = input("\nMasukkan ID Project yang ingin di-restore: ").strip()
    if project_id in recycle_bin:
        projects[project_id] = recycle_bin.pop(project_id)
        print(f"Project dengan ID '{project_id}' berhasil dipulihkan!")
    else:
        print("ID Project tidak ditemukan di Recycle Bin.")
```

Fitur Login, Save, dan Load

```python
def login():
    user_id = input("Masukkan ID Anda: ").strip()

    if user_id not in users:
        print("ID Pengguna tidak ditemukan.")
        return

    password = getpass.getpass("Masukkan Password: ")

    if password == users[user_id]["password"]:

        log_login(user_id)

        user_data = users[user_id]
        name = user_data["name"]
        role = user_data["role"]
        print(f"Selamat datang, {name} ({role})!")
        return user_id, role
    else:
        print("Password yang Anda masukkan salah. Silakan coba lagi.")
        return None, None
```

```python
def save_projects():
    '''Save projects data to a JSON file'''
    with open('projects.json', 'w') as f:
        json.dump(projects, f, indent=4)

def load_projects():
    '''Load projects data from a JSON file if exists'''
    global projects
    if os.path.exists('projects.json'):
        with open('projects.json', 'r') as f:
            projects = json.load(f)
```

Fitur Backlog User

```python
def log_login(user_id):
    '''Mencatat login pengguna ke dalam log'''
    timestamp = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    login_log.append({"user_id": user_id, "timestamp": timestamp})
    print(f"{users[user_id]['name']} berhasil login pada {timestamp}")

def view_login_backlog():
    '''Menampilkan backlog login'''
    if not login_log:
        print("Tidak ada data login.")
        return

    print("\n===== Backlog Login =====")
    backlog_data = []
    for log in login_log:
        user_name = users[log["user_id"]]["name"]
        backlog_data.append([user_name, log["timestamp"]])

    print(tabulate(backlog_data, headers=["Nama Pengguna", "Waktu Login"], tablefmt="grid"))
```

Fitur Menampilkan Tugas yang Diberikan

```python
def view_and_update_task(user_id):
    '''Menampilkan dan Memperbarui Status Tugas yang Ditugaskan kepada Team Member'''
    assigned_tasks = []
    task_mapping = {}

    for project_id, project in projects.items():
        total_tasks = 0
        completed_tasks = 0
        for task_id, task in project["tasks"].items():
            if task["Assigned To"] == user_id:
                assigned_tasks.append([
                    project_id,
                    project["name"],
                    task_id,
                    task["Name"],
                    task["Status"],
                    task["Description"]
                ])
                task_mapping[task_id] = (project_id, task)

            total_tasks += 1
            if task["Status"] == "Selesai":
                completed_tasks += 1

        progress = (completed_tasks / total_tasks * 100) if total_tasks > 0 else 0
        for task in assigned_tasks:
            if task[0] == project_id:
                task.append(f"{progress:.2f}%")

    if not assigned_tasks:
        print("Anda belum memiliki tugas yang ditugaskan.")
        input("Tekan Enter untuk melanjutkan...")
        return

    headers = ["ID Project", "Nama Project", "ID Tugas", "Tugas", "Status", "Deskripsi", "Progress"]
    print(tabulate(assigned_tasks, headers, tablefmt="grid"))
```

```python
task_id = input("Masukkan ID Tugas yang akan diperbarui (atau tekan Enter untuk keluar): ").strip()
if not task_id:
    return

if task_id not in task_mapping:
    print("Tugas tidak valid atau bukan milik Anda.")
    return

status = input("Masukkan Status Baru (Belum Dimulai/ Berjalan/ Selesai): ").strip()
if status not in ["Belum Dimulai", "Berjalan", "Selesai"]:
    print("Status tidak valid.")
    return

project_id, task = task_mapping[task_id]
task["Status"] = status
print(f"Status tugas '{task['Name']}' diperbarui menjadi {status}.")
save_projects()

input("Tekan Enter untuk melanjutkan...")
```

Purwadhika
Digital Technology School

2025

Terima Kasih

"Inovasi adalah hasil dari keberanian untuk mendesain solusi yang belum pernah ada sebelumnya."

Ketut Susilo