



INSTITUT TEKNOLOGI BANDUNG

STEI - TEKNIK INFORMATIKA



# OPTIMISASI METODE CROWDSOURCING PADA DOMAIN SOFTWARE TESTING

Presentation by :

13520025 FRANSISKUS DAVIN ANWARI





# Latar Belakang



- Crowdsourcing adalah suatu paradigma yang belum diutilisasi secara maksimal
- Pada bidang software development, memiliki banyak kegunaan
- Penggunaan pekerja inhouse masih menjadi pilihan pertama



# Rumusan Masalah



## PROBLEM 01

Apakah crowdsourcing efektif diutilisasikan pada pekerjaan pada domain software testing?

## PROBLEM 02

Bagaimana requester dapat melakukan monitoring atas alur kerja worker selama jalannya crowdsourcing?

## PROBLEM 03

Apakah monitoring jalan kerja crowdsourcing dapat meningkatkan efektivitas crowdsourcing?



# Tujuan



### OBJECTIVES

01

Membuat sistem yang membantu aktivitas crowdsourcing pada domain software testing.

### OBJECTIVES

02

Membuat sistem bagi requester untuk mengawasi progress worker selama penggerjaan task berjalan.

### OBJECTIVES

03

Membuat sistem yang dapat menjamin efektivitas penggerjaan crowdsourcing dengan cara monitoring.



# Batasan



### BATASAN 01

Crowdsourcing hanya dilakukan pada tahap pelaksanaan testing software development.

### BATASAN 02

Sistem mengutamakan optimisasi pada monitoring atau pengawasan pada pelaksanaan crowdsourcing.

### BATASAN 03

Sistem hanya melakukan crowdsourcing pada tahap Test Execution pada jenis testing yang menial seperti SIT dan endpoint testing.



INSTITUT TEKNOLOGI BANDUNG

STEI - TEKNIK INFORMATIKA



# Methodology

IDENTIFIKASI MASALAH

ANALISIS SOLUSI

PENGEMBANGAN SISTEM

PENGETESAN SISTEM

EVALUASI



# Studi Literatur

- Crowdsourcing -> Pengalihan pekerjaan ke pihak luar secara terbuka
- Aktor yang bersangkutan; Requester, Worker, Platform
- Implementasi banyak pada bidang software engineering -> terutama software testing
- Terdapat 3 model; Peer Production, Competition, Microtasking





# Peluang dan Tantangan

## PELUANG

- Meningkatkan kecepatan pekerjaan
- Mendapat banyak alternatif solusi
- Mengurangi beban pekerja
- Kesempatan mempelajari teknologi baru

## TANTANGAN

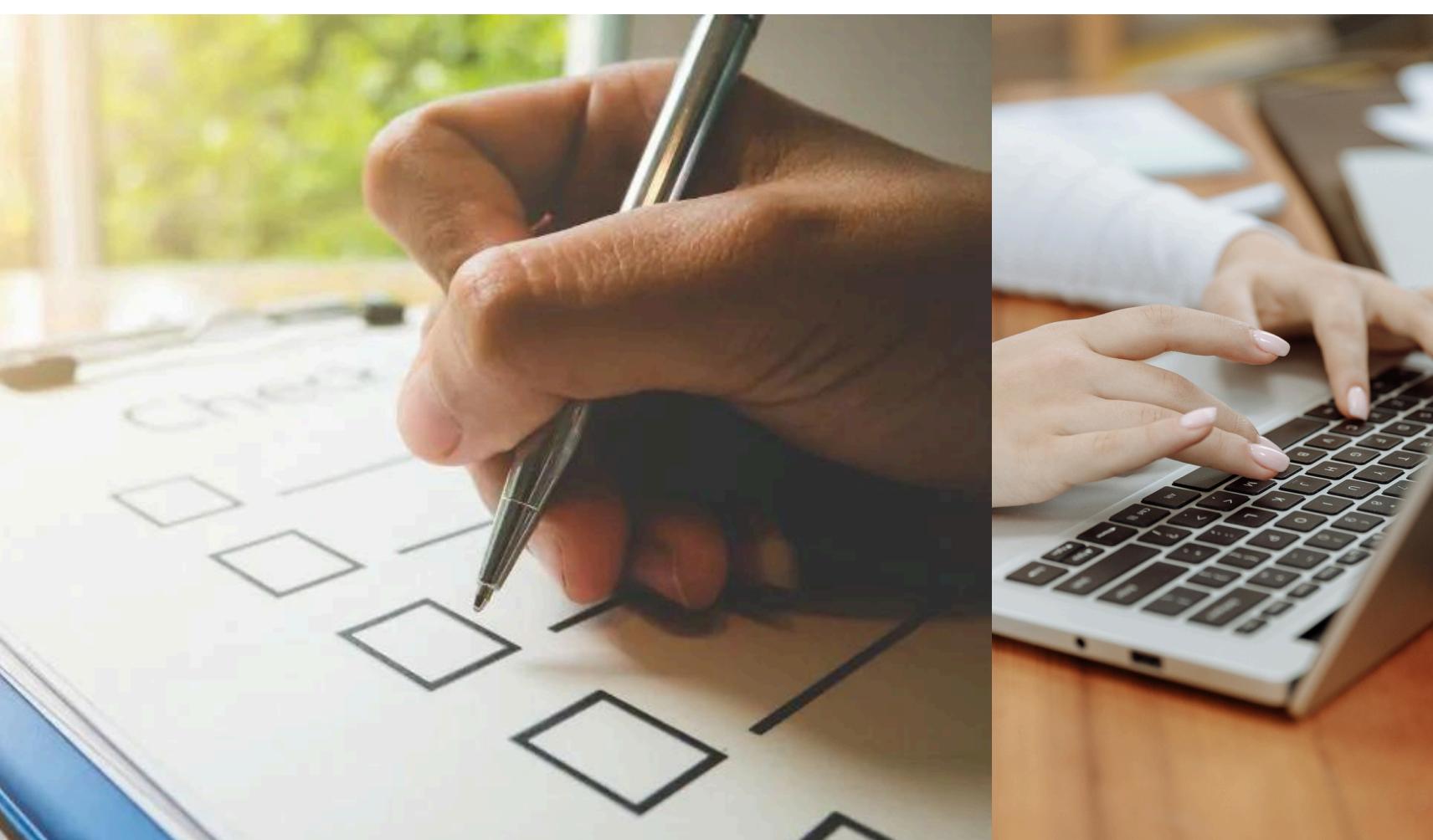
- Pilihan jenis pekerjaan terbatas
- Resiko penggunaan pekerja asing



# Analisis Solusi

## SOLUSI YANG DIDAPATKAN

- Platform pelaksanaan Crowdsourcing software testing dengan fokus pada monitoring menggunakan sistem progress report
- Platform pelaksanaan Crowdsourcing Software Testing dengan utilisasi verifikasi worker
- Platform pelaksanaan Crowdsourcing Software Testing dengan fokus pada komunikasi antara requester dan worker



# Rancangan Solusi

## GAMBARAN UMUM

- Platform pelaksanaan Crowdsourcing software testing dengan fokus pada monitoring
- Implementasi monitoring menggunakan sistem progress reporting

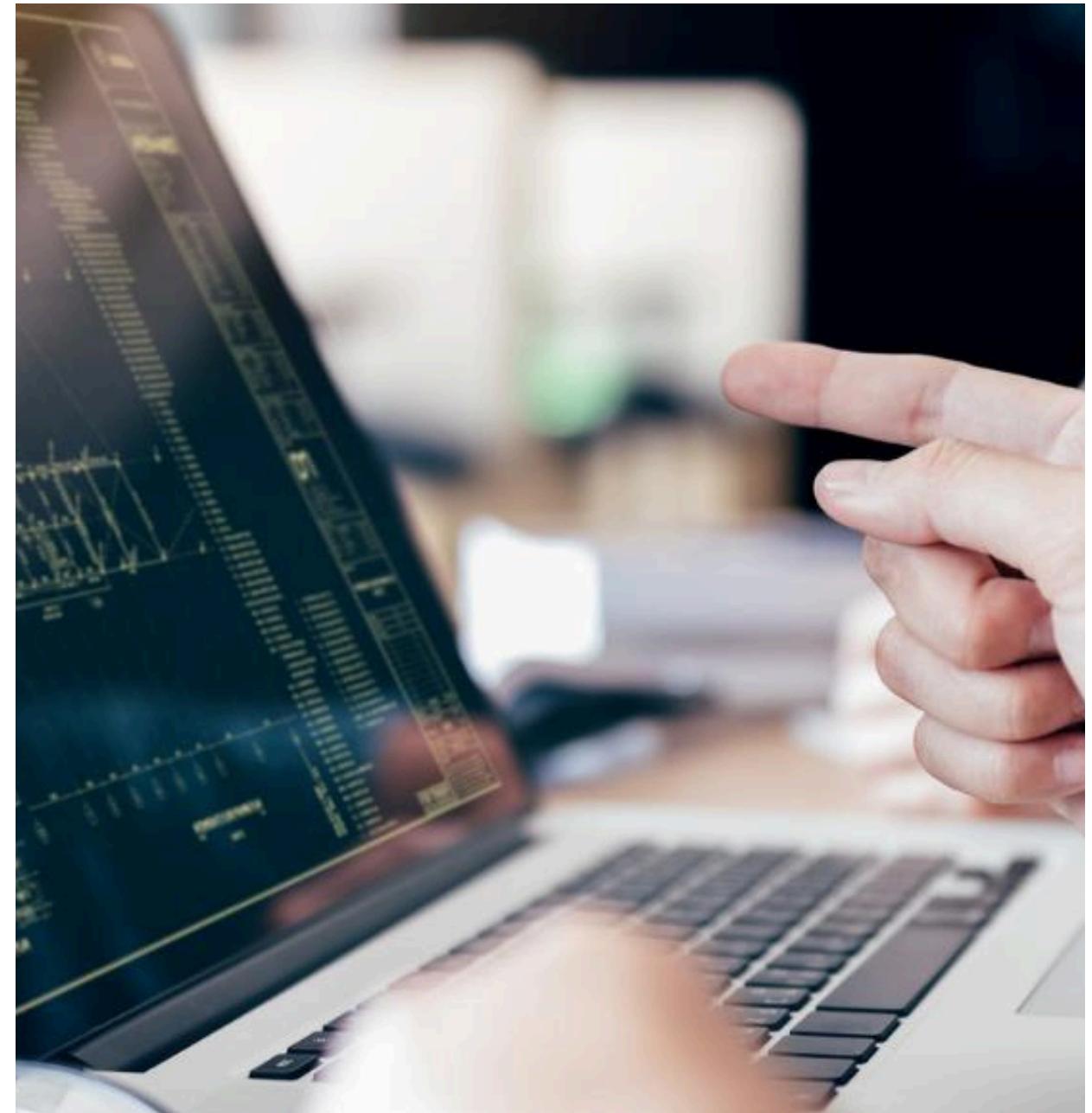


### REQUESTER

- Membuat project
- Memasang task dan mengatur deadline
- Cek progress worker dan menerima task yang selesai

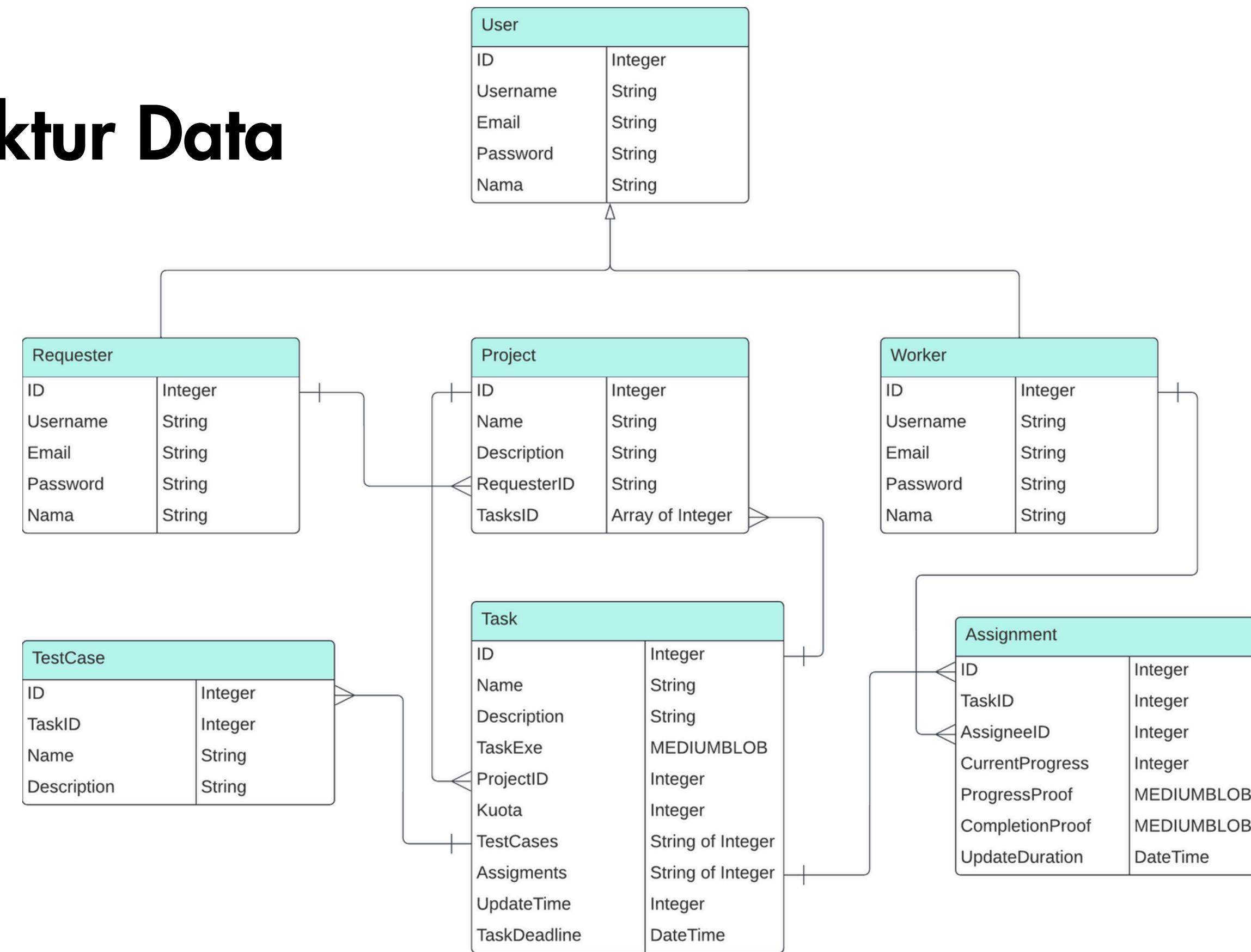
### WORKER

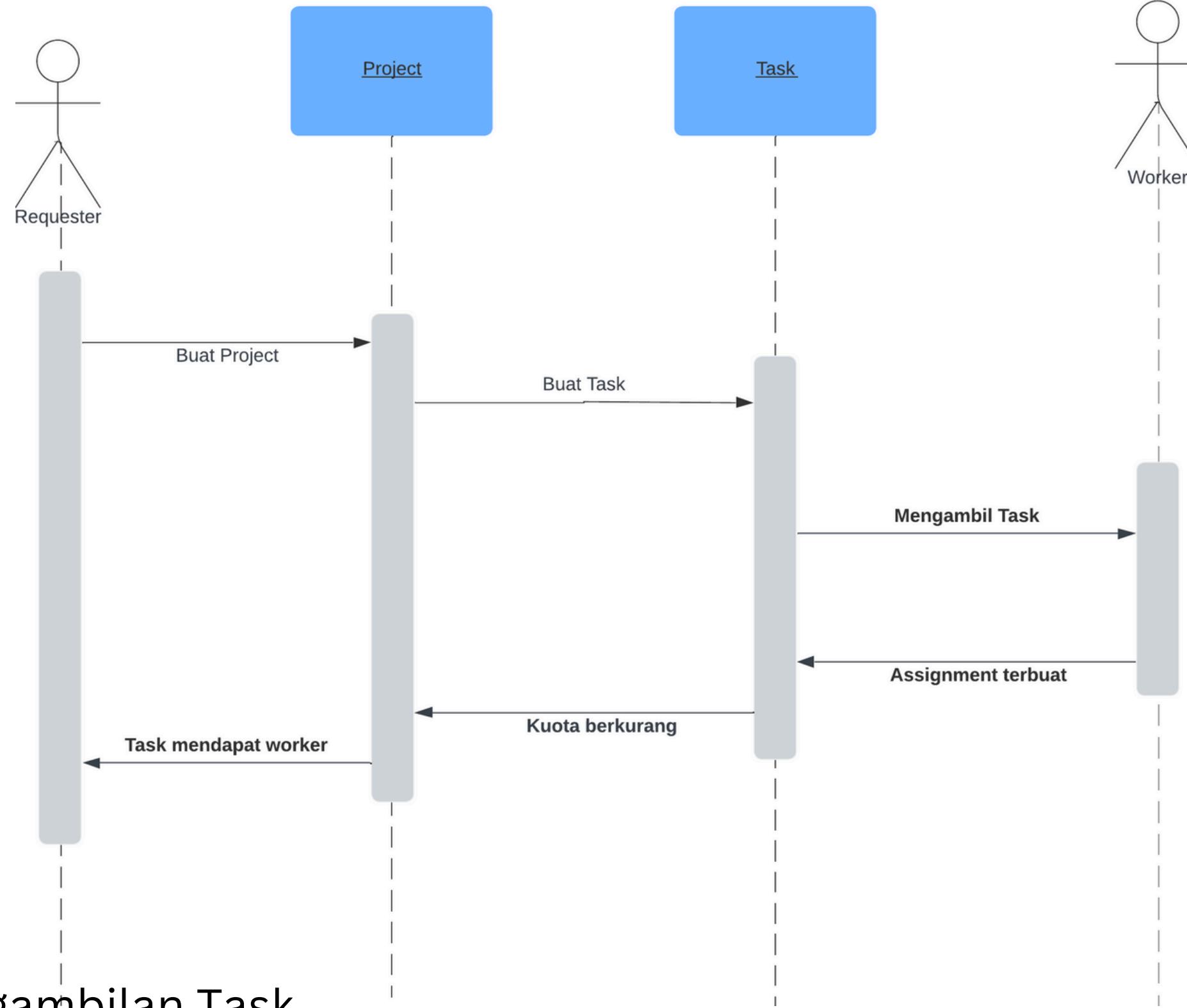
- Mengambil Task menjadi assignment
- Mengerjakan Testing dan melaporkan progress sesuai deadline
- Mengirimkan Bukti penggerjaan seiring pelaporan progress





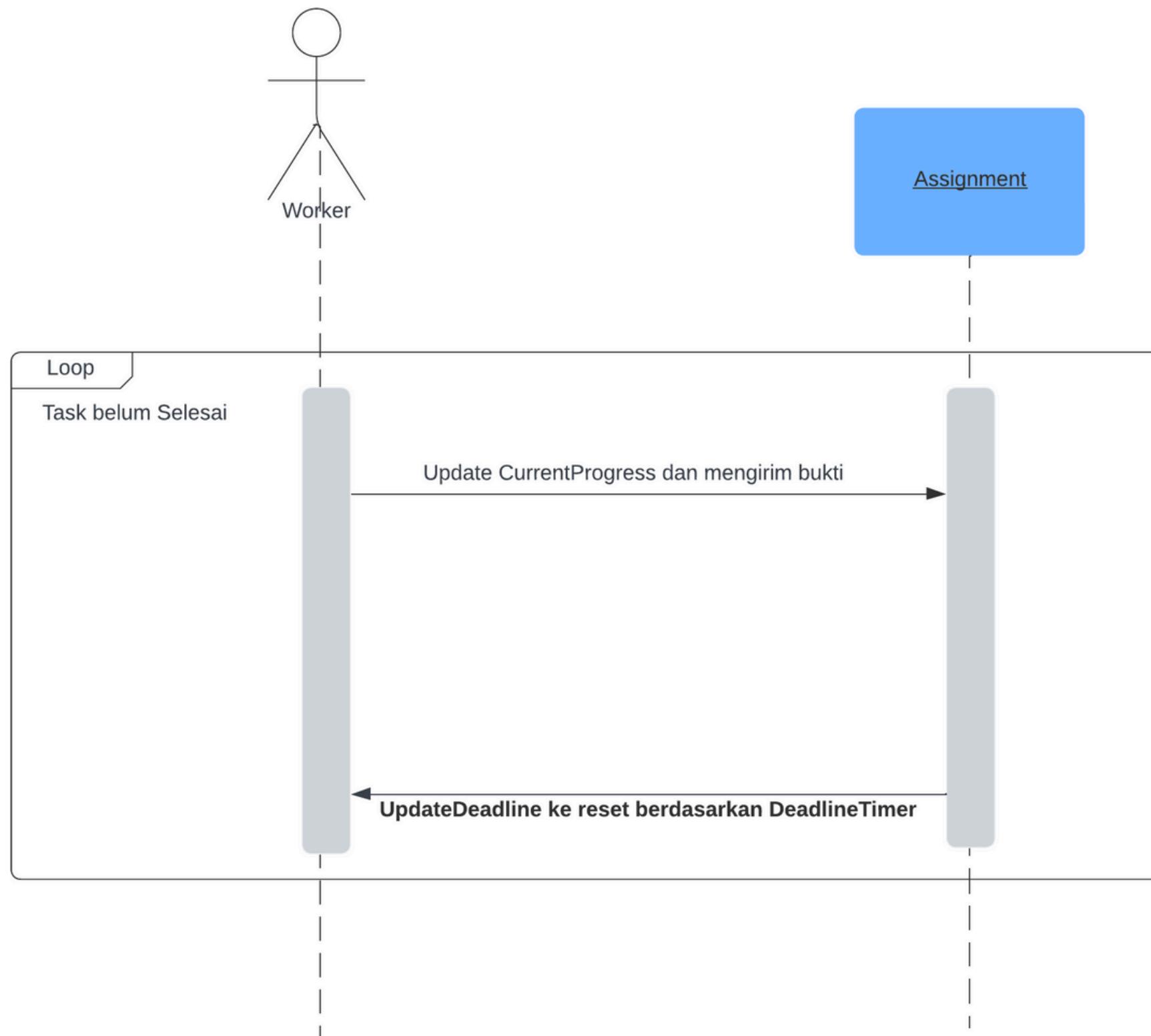
# Desain Struktur Data





## Jalan Kerja Sistem

Alur Pembuatan dan Pengambilan Task

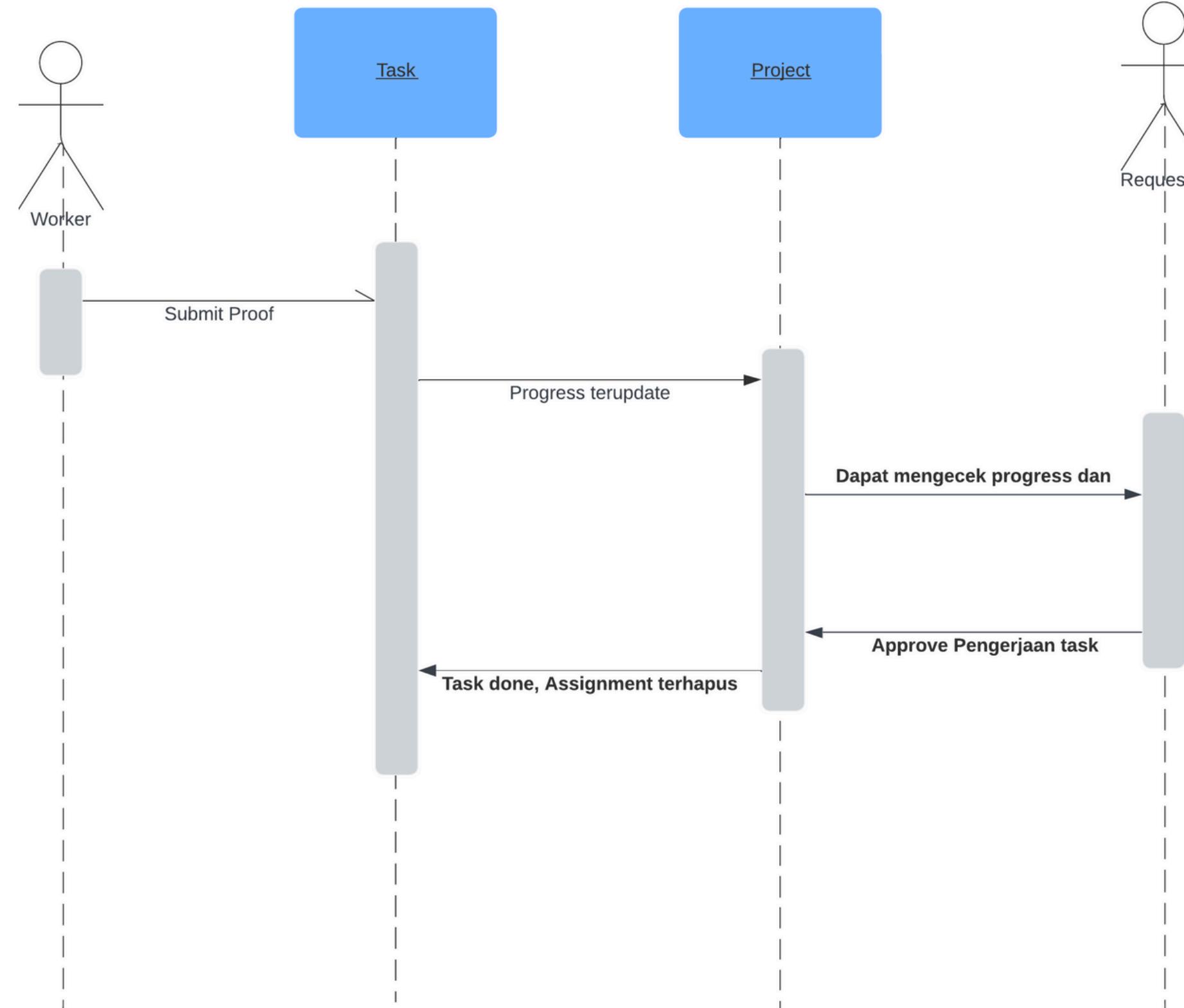


Alur Pelaporan Progress Task

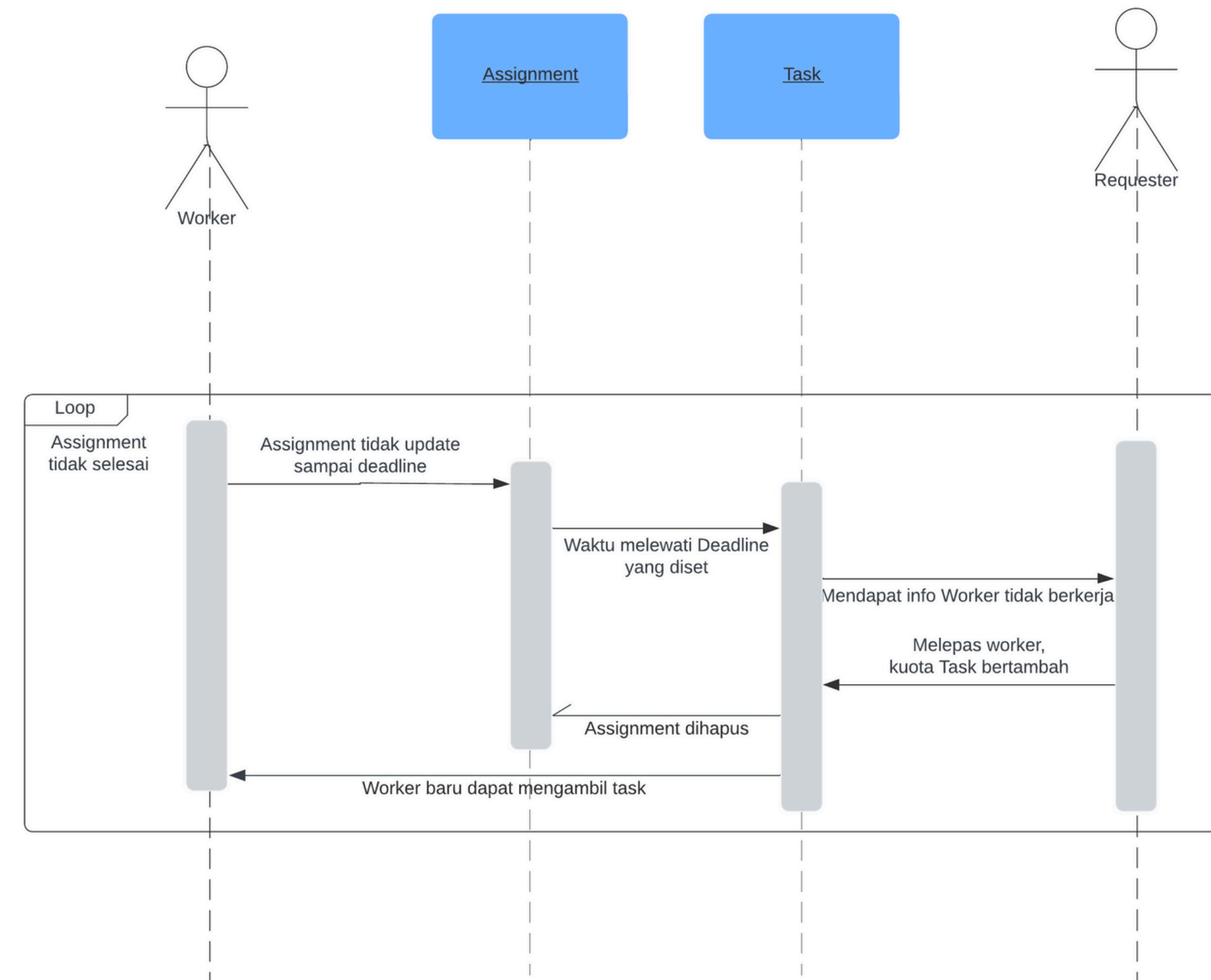


# INSTITUT TEKNOLOGI BANDUNG

STEI - TEKNIK INFORMATIKA



Alur Penyelesaian Task



Alur Deadline Terlewat



# IMPLEMENTASI

Website penyedia platform *crowdsourcing* dengan fokus pada monitoring dan progress reporting

Kakas yang digunakan :

- Backend: Java, Framework Springboot, H2 Database
- Frontend: HTML, CSS, Javascript, http-server

Batasan Implementasi

- Deployment hanya local
- Database tidak tersimpan antar session
- Frontend hanya bisa pada port tertentu



# Backend Services

## Domain User:

- GET /users/{id}
  - Mendapatkan data user
- POST /users
  - Menambah user
- POST /login
  - login sebagai user

## Domain Project:

- GET /projects/{id}
  - Mendapatkan data project
- GET /projects/user/{id}
  - menambahkan data project dari user tertentu
- POST /projects
  - Menambahkan project
- function addTask( projectId, taskId )
  - Menambahkan task tertentu ke suatu project



# **Backend Services**

## **Domain Task:**

- GET /task/{id}
  - Mendapatkan data task
- GET /availableTask
  - Mendapatkan data semua task yang dapat diambil
- POST /task/project/{id}
  - Mendapatkan data task dari project tertentu
- POST /task
  - Menambah task
- Function reduceQuota(id) dan addQuota(id)
  - Menambah dan mengurangi kuota dari task tertentu



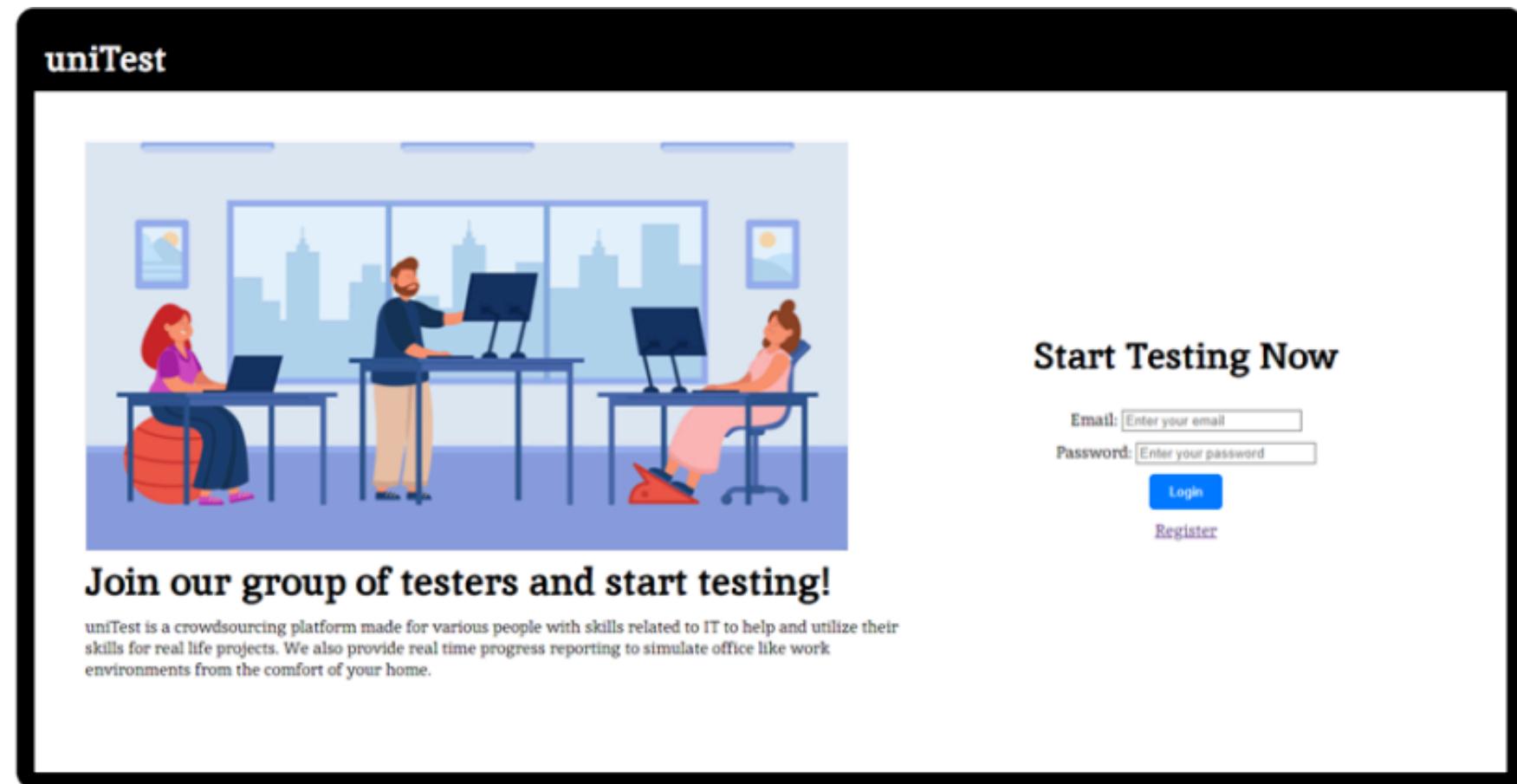
# Backend Services

## Domain Assignment:

- GET /assignment/{id}
  - Mendapatkan data assignment
- GET /activeAssignment
  - Mendapatkan data semua assignment yang sedang aktif
- GET /assignment/task/{id}
  - Mendapatkan data assignment dari task tertentu
- POST /assignment
  - Menambah assignment
- PUT /assignment/update/{id}
  - Mengupdate progress assignment
- DELETE /assignment/finish/{id} & DELETE /assignment/delete/{id}
  - Menyelesaikan atau membatalkan suatu assignment



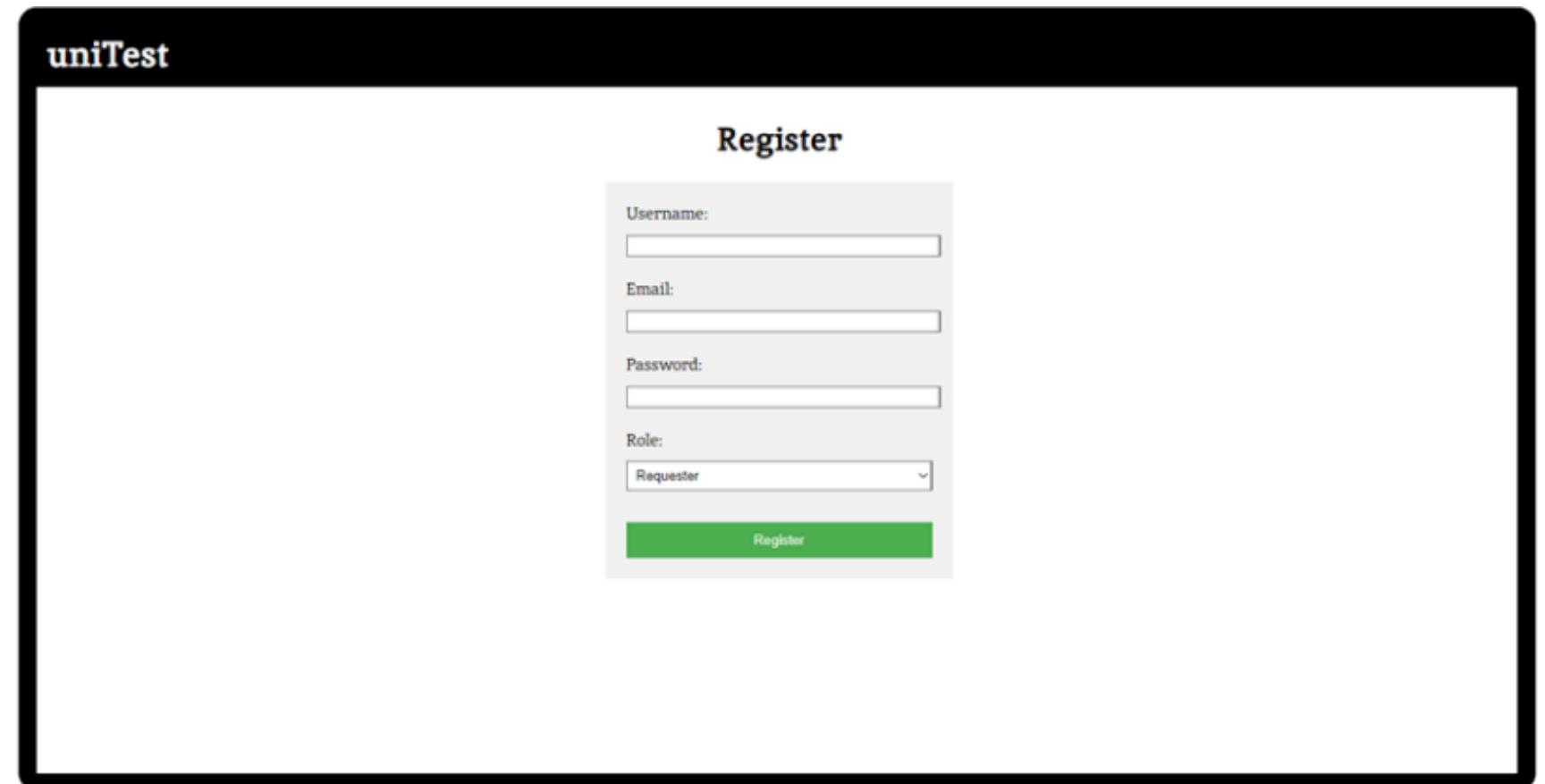
# Frontend



The login page for uniTest features a black header with the text "uniTest". Below the header is a large illustration of three people working at desks in an office setting. To the right of the illustration, the text "Start Testing Now" is displayed above two input fields: "Email: [Enter your email]" and "Password: [Enter your password]". Below these fields are two buttons: "Login" and "Register".

**Join our group of testers and start testing!**

uniTest is a crowdsourcing platform made for various people with skills related to IT to help and utilize their skills for real life projects. We also provide real time progress reporting to simulate office like work environments from the comfort of your home.



The registration page for uniTest features a black header with the text "uniTest". Below the header, the word "Register" is centered. The form consists of several input fields: "Username:", "Email:", "Password:", and "Role:". The "Role:" field contains the value "Requester". At the bottom of the form is a green "Register" button.



# Frontend

uniTest

Logged in as requester [Return to Landing Page](#)

## Projects

**Project 1**  
Description of Project Lorem Ipsum Lorem Ipsum Lorem Ipsum  
[View Project](#)

**Project 2**  
Description of Second Project Lorem Ipsum Lorem Ipsum Lorem Ipsum  
[View Project](#)

**Project 3**  
Description of Third Project Lorem Ipsum Lorem Ipsum Lorem Ipsum  
[View Project](#)

[Add Project](#)

uniTest

Logged in as requester [Return to Landing Page](#)

## Create Project

Project Title:

Project Description:

[Create](#)



# Frontend

uniTest

Logged in as requester [Return to Landing Page](#)

## Project 1

Description of Project Lorem Ipsum Lorem Ipsum Lorem Ipsum

**Task 1**

Description of Task Lorem Ipsum Lorem Ipsum Lorem Ipsum

Quota: 2

[View task](#)

**Task 2**

Description of Second Task Lorem Ipsum Lorem Ipsum Lorem Ipsum

Quota: 0

[View task](#)

[Add Task](#)

uniTest

Logged in as requester [Return to Landing Page](#)

## Create Task

Task Title:

Task Description:

Select a file:

Task Duration (in Days):

Task Quota:

Task Deadline:

[Create](#)

uniTest

Logged in as requester [Return to Landing Page](#)

## Task 2

Description of Second Task Lorem Ipsum Lorem Ipsum Lorem Ipsum

**Assignment worked on by worker**

No progress proof submitted  
No completion proof submitted

Deadline: Mon Jul 15 2024

[View assignment](#)



# Frontend

uniTest

Logged in as requester [Return to Landing Page](#)

## Task 2

Description of Second Task Lorem Ipsum Lorem Ipsum Lorem Ipsum

**Assignment worked on by worker**

No progress proof submitted  
No completion proof submitted

Deadline: Mon Jul 15 2024

[View assignment](#)

uniTest

Logged in as Request [Return to Landing Page](#)

## Assignment from Task Demo

Progress Proof: proof.png



Completion Proof: Not Submitted  
Current Progress: 69%  
Update Deadline: 2024-07-17 03:46:45

## Task Demo

Demo  
Code executable file: bin/task.zip  
Requires Update Every : 2 days



# Frontend

The screenshot displays the uniTest application interface across three main components:

- Available Tasks:** A list of tasks available for workers. One task is shown:
  - Task 1:** Description: Description of Task Lorem Ipsum. Need Update Every: 1 days. Deadline: 2024-07-15T14:30:00.000+00:00. Available quota: 2.
  - [View task](#)
- Active Assignments:** A list of assignments worked on by the current worker. One assignment is shown:
  - Assignment worked on by worker:** No progress proof submitted. No completion proof submitted. Deadline: 2024-07-15T14:30:00.000+00:00.
- Task Demo:** A detailed view of a task.
  - Title:** Task Demo
  - Description:** Demo
  - Code executable file:** bin/task.zip [Download](#)
  - Update Frequency:** Requires Update Every : 2 days
  - Available Quota:** Available Quota : 1
  - Deadline:** Deadline: 2024-07-17 10:44:00
  - [Take Assignment](#)
- Assignment from Task Lewat Deadline:** A detailed view of an assignment from a task that has passed its deadline.
  - Title:** Assignment from Task Lewat Deadline
  - Progress Proof:** tes.jpg
  - Completion Proof:** Not Submitted
  - Current Progress:** 0.32
  - Update Deadline:** 2024-07-15 19:00:00
  - [Update Progress](#)
- Task Lewat Deadline:** A detailed view of a task that has passed its deadline.
  - Description:** Deskripsi untuk task dengan waktu deadline yang dekat
  - Code executable file:** tes.exe [Download](#)
  - Update Frequency:** Requires Update Every : 10 days
  - Available Quota:** Available Quota : 0
  - Deadline:** Deadline: 2024-07-15 19:00:00
- Update Progress:** A modal window for updating the progress of an assignment.
  - Progress Proof:**  Enter Progress Proof
  - Completion Proof:**  Enter Completion Proof
  - Current Progress:**

10%
  - [Update](#)
  - [Close](#)



# Demo

uniTest

## Register

Username:

Email:

Password:

Role:



# Pengujian

Pengujian dilakukan dengan melakukan integration testing pada backend yang menguji interaksi pada service-service.



# Skenario Pengambilan Task Hingga Kuota Habis

POST localhost:8080/task Send

Params Auth Headers (9) Body Scripts Tests Settings Cookies

raw JSON Beautify

```
1 {  
2   "name": "Task 1",  
3   "description": "Description of Task Lorem Ipsum Lorem Ipsum Lorem Ipsum",  
4   "exe": "tes.exe",  
5   "projectId": 1,  
6   "updateTime": "1",  
7   "quota": 1,  
8   "deadline": "2024-07-15"  
9 }
```

Body Cookies Headers (8) Test Results 201 Created 91 ms 473 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 2,  
3   "name": "Task 1",  
4   "description": "Description of Task Lorem Ipsum Lorem Ipsum Lorem Ipsum",  
5   "exe": "tes.exe",  
6   "projectId": 1,  
7   "updateTime": "1",  
8   "deadline": "2024-07-15T00:00:00.000+00:00",  
9   "quota": 1,  
10  "assignmentIds": []
```

\* Postbot Runner Start Proxy Cookies Vault Trash

GET localhost:8080/availabletasks Send

Params Auth Headers (9) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	... Bulk Edit
body	Cookies Headers (8) Test Results	200 OK 50 ms 686 B	Save as example

Pretty Raw Preview Visualize JSON

```
1 [  
2   {  
3     "id": 1,  
4     "name": "Task 1",  
5     "description": "Description of Task Lorem Ipsum Lorem Ipsum Lorem Ipsum",  
6     "exe": "tes.exe",  
7     "projectId": 1,  
8     "updateTime": "1",  
9     "deadline": "2024-07-15T00:00:00.000+00:00",  
10    "quota": 1,  
11    "assignmentIds": []  
12  },  
13  {  
14    "id": 2,  
15    "name": "Task 1",  
16    "description": "Description of Task Lorem Ipsum Lorem Ipsum Lorem Ipsum",  
17    "exe": "tes.exe",  
18    "projectId": 1,  
19    "updateTime": "1",
```



# Skenario Pengambilan Task Hingga Kuota Habis

POST localhost:8080/assignment

Send

Params Auth Headers (9) Body Scripts Tests Settings

raw JSON

```
1 {  
2   "taskId": 2,  
3   "assigneeId": 2,  
4   "progressProof": "",  
5   "completionProof": "",  
6   "currentProgress": 0,  
7   "updateDeadline": "2024-07-15T14:30"  
8 }
```

Body Cookies Headers (8) Test Results

201 Created 190 ms 403 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": 1,  
3   "taskId": 2,  
4   "assigneeId": 2,  
5   "progressProof": "",  
6   "completionProof": "",  
7   "currentProgress": 0.0,  
8   "updateDeadline": "2024-07-15T14:30:00.000+00:00"  
9 }
```

GET localhost:8080/availabletasks

Send

Params Auth Headers (9) Body Scripts Tests Settings

Query Params

	Key	Value	Description	... Bulk Edit

Body Cookies Headers (8) Test Results

200 OK 297 ms 470 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [  
2   {  
3     "id": 1,  
4     "name": "Task 1",  
5     "description": "Description of Task Lorem Ipsum Lorem Ipsum Lorem Ipsum",  
6     "exe": "tes.exe",  
7     "projectId": 1,  
8     "updateTime": "1",  
9     "deadline": "2024-07-15T00:00:00.000+00:00",  
10    "quota": 1,  
11    "assignmentIds": []  
12  }  
13 ]
```



# Skenario Update Assignment Mendekati Deadline Task

POST localhost:8080/task Send

Params Auth Headers (9) Body Scripts Tests Settings

raw JSON

```
2 {"name": "Task Lewat Deadline",
3 "description": "Deskripsi untuk task dengan waktu deadline yang dekat",
4 "exe": "tes.exe",
5 "projectId": 1,
6 "updateTime": "10",
7 "quota": 1,
8 "deadline": "2024-07-15 19:00:00"
9 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "name": "Task Lewat Deadline",
4   "description": "Deskripsi untuk task dengan waktu deadline yang dekat",
5   "exe": "tes.exe",
6   "projectId": 1,
7   "updateTime": "10",
8   "deadline": "2024-07-15 19:00:00",
9   "quota": 1,
10  "assignmentIds": []
11 }
```

HTTP crowdsource / createAssignment Save Share

POST localhost:8080/assignment Send

Params Auth Headers (9) Body Scripts Tests Settings

raw JSON

```
1 {
2   "taskId": 2,
3   "assigneeId": 2
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "taskId": 2,
4   "assigneeId": 2,
5   "progressProof": "",
6   "completionProof": "",
7   "currentProgress": 0.0,
8   "updateDeadline": "2024-07-15 19:00:00"
9 }
```



# Skenario Update Assignment Mendekati Deadline Task

SELECT \* FROM ASSIGNMENT

Action	ID	ASSIGNEE_ID	COMPLETION_PROOF	CURRENT_PROGRESS	PROGRESS_PROOF	TASK_ID	UPDATE_DEADLINE
	1	2		0.0		2	2024-07-13 02:00:00

(1 row, 0 ms)

PUT | localhost:8080/assignment/update/1 Send

Params Auth Headers (9) Body Scripts Tests Settings

raw JSON Beautify

```
1 {  
2   "progressProof": "proof.jpg",  
3   "completionProof": "",  
4   "currentProgress": 0.5  
5 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON Save as example

```
1 {  
2   "id": 1,  
3   "taskId": 2,  
4   "assigneeId": 2,  
5   "progressProof": "proof.jpg",  
6   "completionProof": "",  
7   "currentProgress": 0.5,  
8   "updateDeadline": "2024-07-15 19:00:00",  
9   "hibernateLazyInitializer": {}  
10 }
```



# Kesimpulan

1. Sistem crowdsourcing cocok digunakan untuk pengerajan software testing, terutama untuk jenis testing yang berupa task-task menial seperti endpoint testing dan SIT
2. Pelaksanaan crowdsourcing dapat dimonitor dengan mengimplementasikan sistem progress reporting yang menjamin keaktifan worker.
3. Dengan menggunakan sistem progress report, worker akan melaporkan progressnya secara berkala sehingga worker yang berhenti di tengah pengerajan task dapat diketahui dan ditindaklanjuti dengan cepat sehingga proses crowdsourcing efektif.



# Saran

1. Mengimplementasi sistem pemberian reward untuk worker yang menyelesaikan assignmentnya sebelum deadline sampai agar memberi insentif worker bekerja lebih cepat
2. Memberi kuota kepada worker atas banyaknya task yang bisa diambil, dengan pengurangan atau penambahan sesuai performa worker terhadap assignmentnya. Misalkan worker yang sering melewati deadline akan mendapat pengurangan kuota.



INSTITUT TEKNOLOGI BANDUNG

STEI - TEKNIK INFORMATIKA

# Terima Kasih!



Presentation by :

**FRANSISKUS DAVIN ANWARI**





**INSTITUT TEKNOLOGI BANDUNG**

**STEI - TEKNIK INFORMATIKA**

# **Q&A Session**