PSO Final Project

To do meter

Jason Ho – 5026221005

Fresnel Siregar – 5026221076

I Gusti Ngurah Adhya Pradipta – 5026221184

project scope & objectives

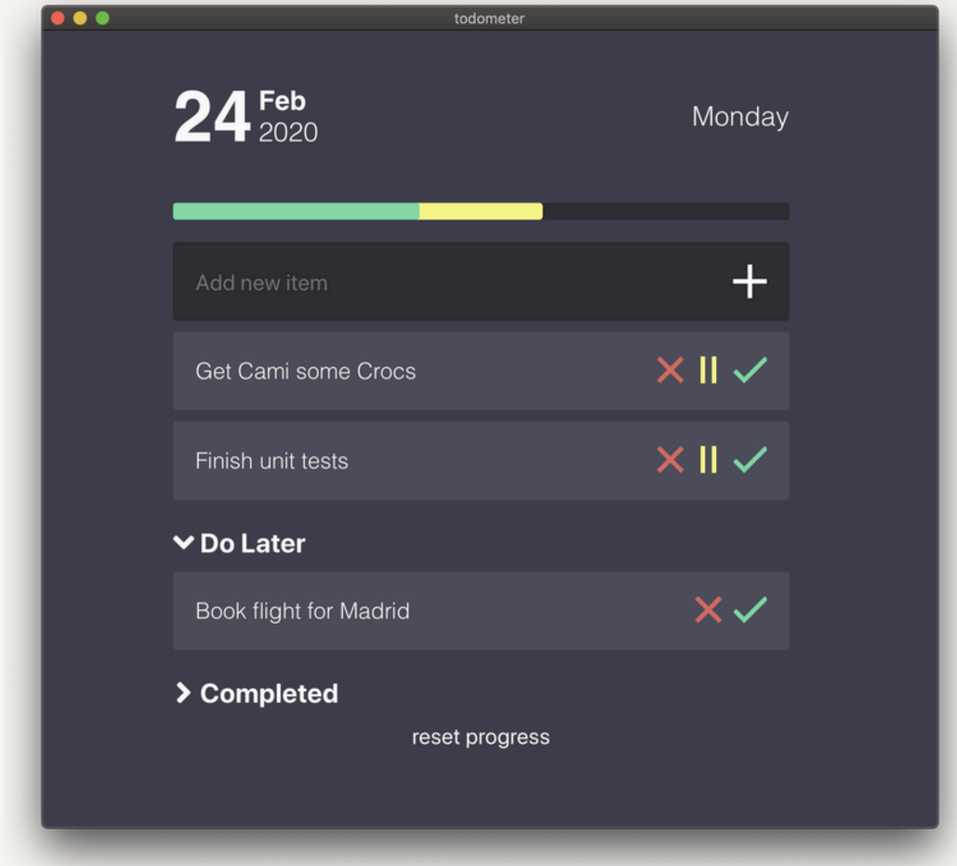
scope

Proyek ini bertujuan untuk mengembangkan sebuah website **To Do Meter** sederhana dan fungsional bagi individu yang membutuhkan cara efisien untuk mengatur aktivitas harian mereka. Website ini akan memiliki fitur manajemen tugas, termasuk pengelompokan tugas ke dalam kategori "Aktif," "Do Later," dan "Completed," serta menampilkan progres harian secara visual. Website ini dirancang dengan antarmuka pengguna yang ramah dengan penyimpanan data lokal agar tugas dapat tersimpan dengan baik. Proyek ini akan diselesaikan dalam waktu 5 minggu, meliputi tahap pengembangan dan pengujian, termasuk pengaturan pipeline CI/CD.

objectives

- Mengembangkan sistem manajemen tugas yang sepenuhnya fungsional untuk tugas.
- Menyelesaikan proyek dalam waktu 5 minggu, mencakup pengembangan kode, deployment, dan pengujian.
- Membagi tanggung jawab di antara anggota tim untuk memaksimalkan efisiensi dan memastikan semua tugas tertangani dengan baik.

Initial Step



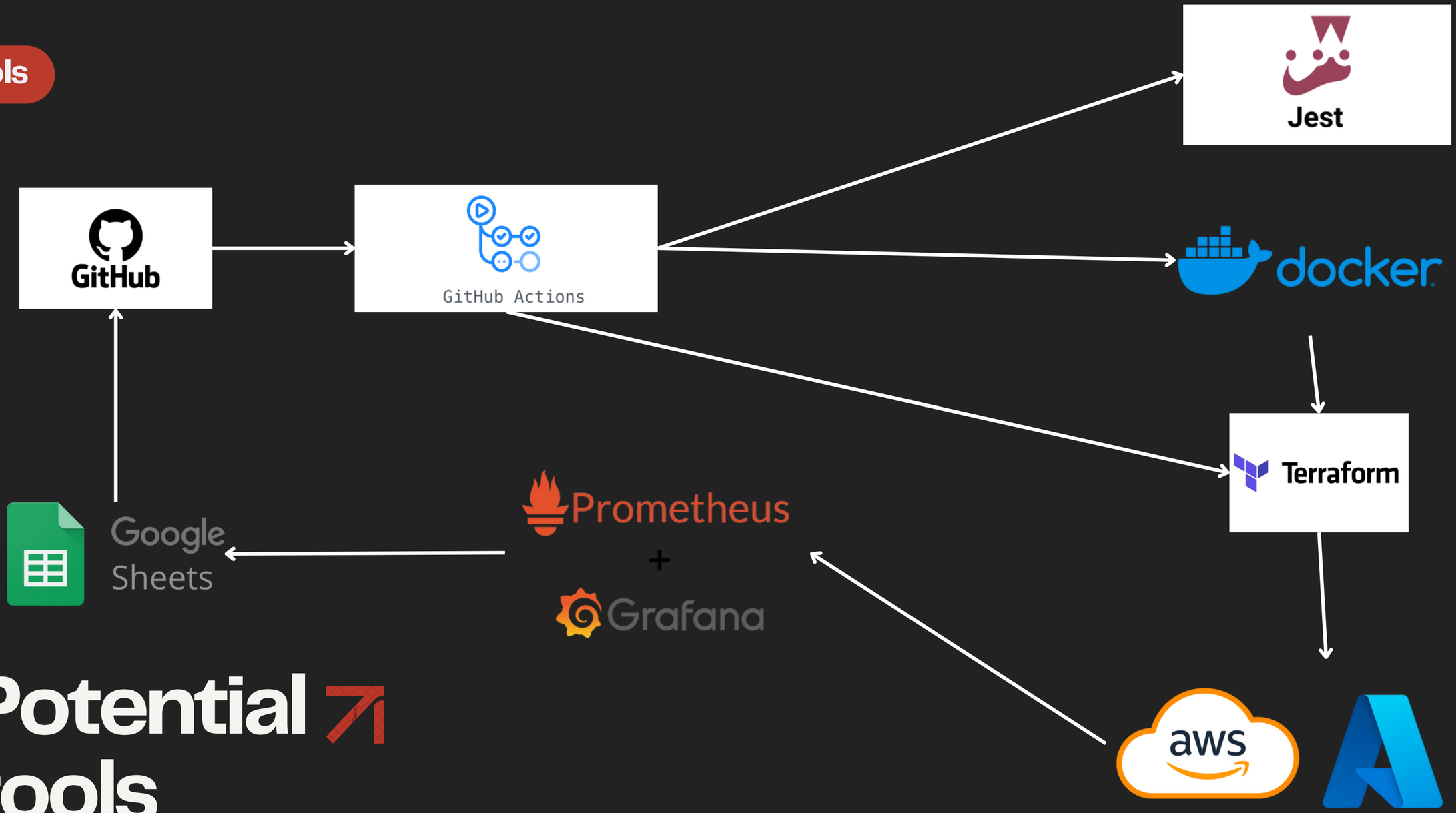
Tampilan awal website to do meter yang dijalankan di lokal

GitHub

Link github
https://github.com/jasonnho/DevOps-PSO

Source
https://github.com/cassidoo/todometer

tools



Potential tools

Google Spreadsheet

Google Spreadsheet di sini berfungsi sebagai alat untuk tracking progress dan tugas-tugas yang perlu dikerjakan dalam proyek. Daftar tugas, siapa yang bertanggung jawab atas tugas tertentu, milestone proyek, dan catatan penting lainnya akan dillist pada google spreadsheet.

Potential tools

Github – Version Control

GitHub berperan sebagai tempat penyimpanan dan pengelolaan kode project. Perubahan pada kode bisa dicatat dan dilacak, sehingga memudahkan untuk kolaborasi antar anggota tim. Jika terjadi kesalahan atau perubahan yang tidak diinginkan, rollback ke versi sebelumnya bisa dilakukan

Potential tools

Terraform – Infrastucture as Code

Terraform adalah alat yang kita pakai untuk mengelola infrastruktur di cloud (AWS atau Azure) secara otomatis. Dengan Terraform, kita bisa mendefinisikan infrastruktur seperti server atau jaringan dalam bentuk kode. Sehingga setup infrastruktur dapat dijalankan dengan praktis

Potential tools

Docker – Containerization

Docker digunakan untuk mengemas aplikasi dalam container agar bisa berjalan di berbagai environment tanpa perlu khawatir masalah dependency. Sehingga aplikasi lebih portable dan bisa berjalan konsisten di mana saja.

Potential tools

Github Actions – CI/CD

GitHub Actions digunakan untuk mengotomatisasi pipeline CI/CD. Setiap kali ada perubahan kode, GitHub Actions akan otomatis membangun, mengetes, dan melakukan deployment aplikasi.

Potential tools

Jest – Code Test

Framework Testing untuk aplikasi Javascript dan akan memastikan setiap bagian dari aplikasi berjalan dengan benar. Jest akan mencari error pada aplikasi jika perubahan yang dilakukan mengganggu fitur/bagian lainnya

Potential tools

AWS / Azure – Deployment Environment

AWS akan dipakai sebagai tempat untuk menjalankan aplikasi di cloud. AWS memungkinkan aplikasi berjalan dengan infrastruktur yang scalable, jadi kalau aplikasi butuh resource lebih banyak, kita bisa dengan mudah menambah kapasitas. Azure juga akan dipertimbangkan sebagai pilihan (bergantung pada credit gratis kedua service)

Potential tools

Grafana dan Prometheus – Monitoring

Grafana dan Prometheus adalah tools yang akan digunakan untuk memantau performa aplikasi Prometheus mengumpulkan data dalam bentuk angka, sementara Grafana menampilkannya dalam bentuk visual yang mudah dipahami.

Potential tools

project schedule ↘

Project Roadmap Plan

Project Title	To do Meter Website
Group Name	Adhya, Fresnel, Jason

Week	Objective	Assign To	Due date	% Task completion	Task Status	Week 1					Week 2							Week 3							Week 4					Week 5				
						S	S	R	K	J	S	M	S	S	R	K	J	S	M	S	S	R	K	J	S	S	R	K	J	S	S	R	K	J
Project Planning and Initial Setup																																		
Week 1	Define the Project Scope and Objectives	Adhya	15/11/2024	100,00%	Done																													
	Create a Project Plan with Timelines and Milestones	Fresnel	15/11/2024	100,00%	Done																													
	Initial research on relevant tools and technologies.	Jason	15/11/2024	100,00%	Done																													
	Set Up a Version Control System	Jason	15/11/2024	100,00%	Done																													
Begin drafting the pipeline design and select tools																																		
Week 2	Finalize tools and architecture for chosen approach	Jason	22/11/2024	0,00%	Open																													
	Initial Configuration of CI/CD Tools	Adhya	22/11/2024	0,00%	Open																													
	Configure a CI tool to automate the build and testing process.	Fresnel	22/11/2024	0,00%	Open																													
	Implement branching strategies and code review processes	Fresnel	22/11/2024	0,00%	Open																													
Initial Prototype and Documentation																																		
Week 3	Start initial development and unit testing of the application or model.	Fresnel	29/11/2024	0,00%	Open																													
	Develop scripts for automated deployment to a staging environment.	Jason	29/11/2024	0,00%	Open																													
	Implement a CD tool to automate deployment processes.	Adhya	29/11/2024	0,00%	Open																													
	Use Infrastructure as Code (IaC) tools to define and provision infrastructure.	Jason	29/11/2024	0,00%	Open																													
Testing and Refinement																																		
Week 4	Run the CI/CD pipeline through basic tests.	All	06/12/2024	0,00%	Open																													
	Troubleshoot and optimize deployment scripts for staging.	All	06/12/2024	0,00%	Open																													
	Set Up Monitoring Tools	Adhya	06/12/2024	0,00%	Open																													
	Implement Logging Mechanisms	Fresnel	06/12/2024	0,00%	Open																													
	Integrate Security Practices	Jason	06/12/2024	0,00%	Open																													
Final Presentation Preparation																																		
Week 5	Prepare Documentation and Final Presentation	All	13/12/2024	0,00%	Open																													

Link timeline proyek

PSO Final Project

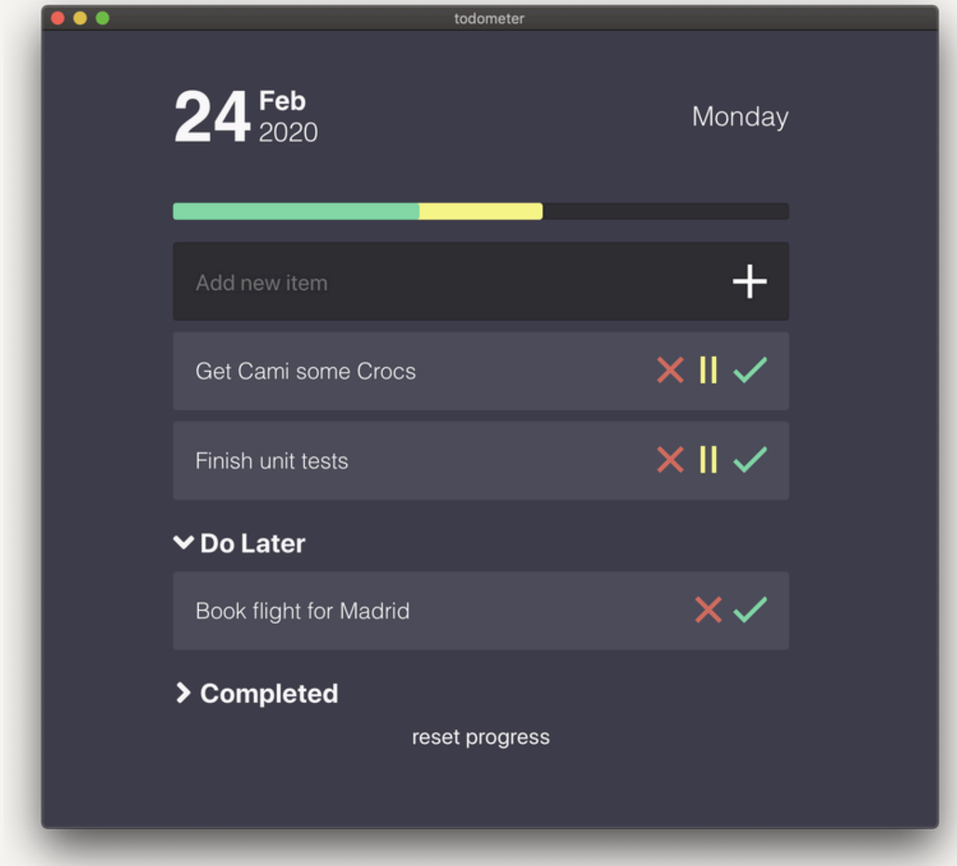
To do meter

Jason Ho – 5026221005

Fresnel Siregar – 5026221076

I Gusti Ngurah Adhya Pradipta – 5026221184

Todometer



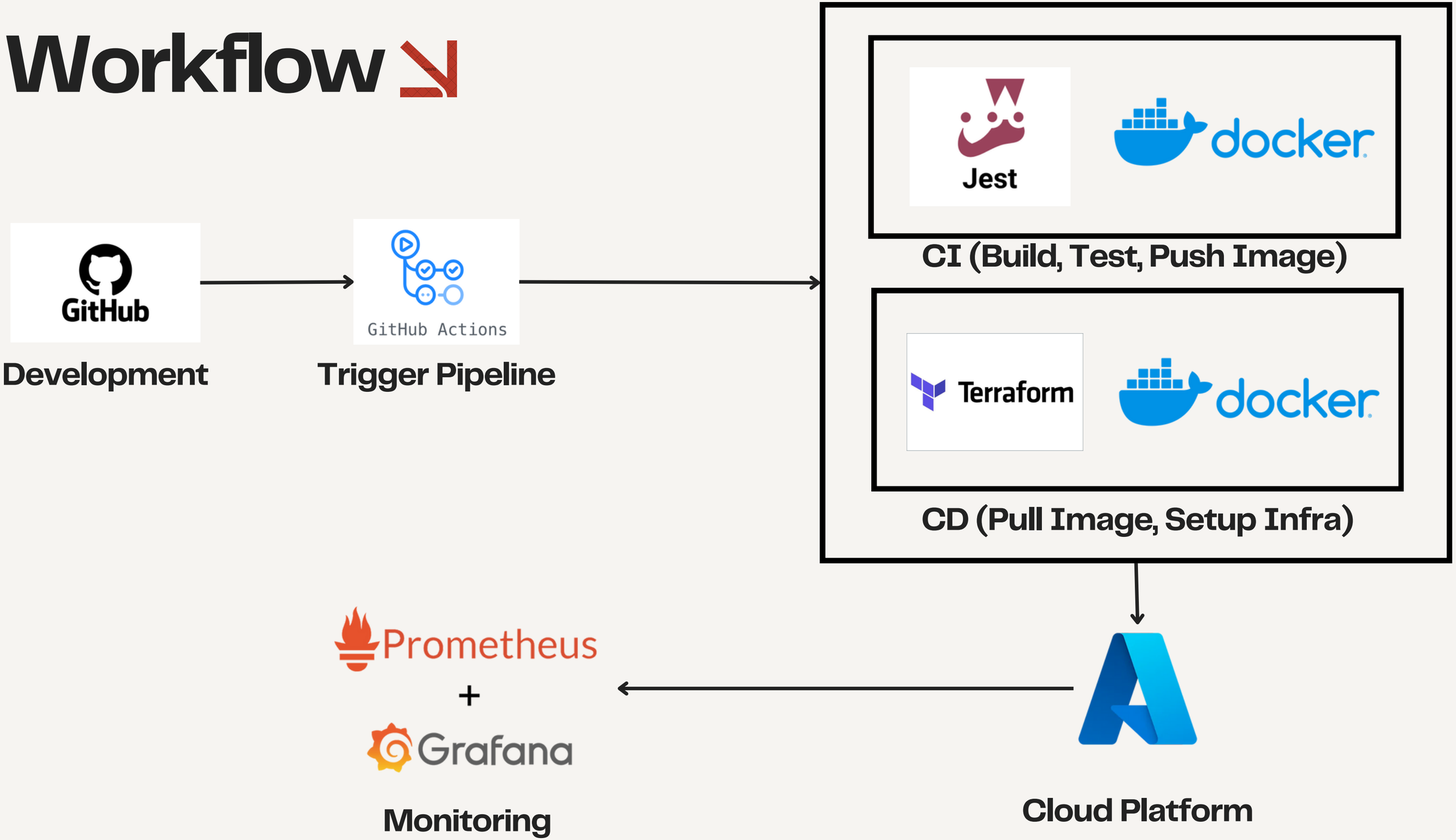
Tampilan awal website to do meter yang dijalankan di lokal

Link github
https://github.com/jasonnho/DevOps-PSO

Source
https://github.com/cassidoo/todometer

Week 2

Workflow ↘


GitHub
Development
GitHub Actions
Trigger Pipeline
Jest
docker
CI (Build, Test, Push Image)
Terraform
docker
CD (Pull Image, Setup Infra)
Prometheus
+
Grafana
Monitoring
Cloud Platform

Tools

- Github
  - Mendukung kolaborasi
  - Integrasi dengan Github Actions untuk CI/CD
- Github Actions (Trigger Pipeline)
  - Native CI/CD tool dari Github
  - Otomasi pipeline secara langsung
- Docker (Containerization, Build and Test)
  - Menjaga konsistensi dengan containerization
- Jest (Testing Framework)
  - Framework testing aplikasi Javascript
  - Mendukung mocking, assertion, dan coverage.
- Terraform (IaC)
  - Melakukan pengelolaan infrastruktur
  - Mendukung berbagai cloud platform termasuk Azure
- Azure (Cloud Deployment)
  - Mendukung deployment berbasis container
  - Dapat diintegrasi dengan monitoring tools
- Prometheus dan Grafana (Monitoring)
  - Open Source
  - Dashboarding berdasarkan data yang dikumpulkan

Week 2 Progress

01

Finalisasi workflow

- Melakukan research pada aplikasi potensial
- Finalisasi dan membuat skema workflow CI/CD

02

Initial configuration

- Melakukan konfigurasi awal pada Github Action
- Implementasi build dan testing otomatis dengan initial pipeline

03

Implement Docker (ongoing)

- Dalam tahapan integrasi docker dalam pipeline

Thank you 

PSO Final Project

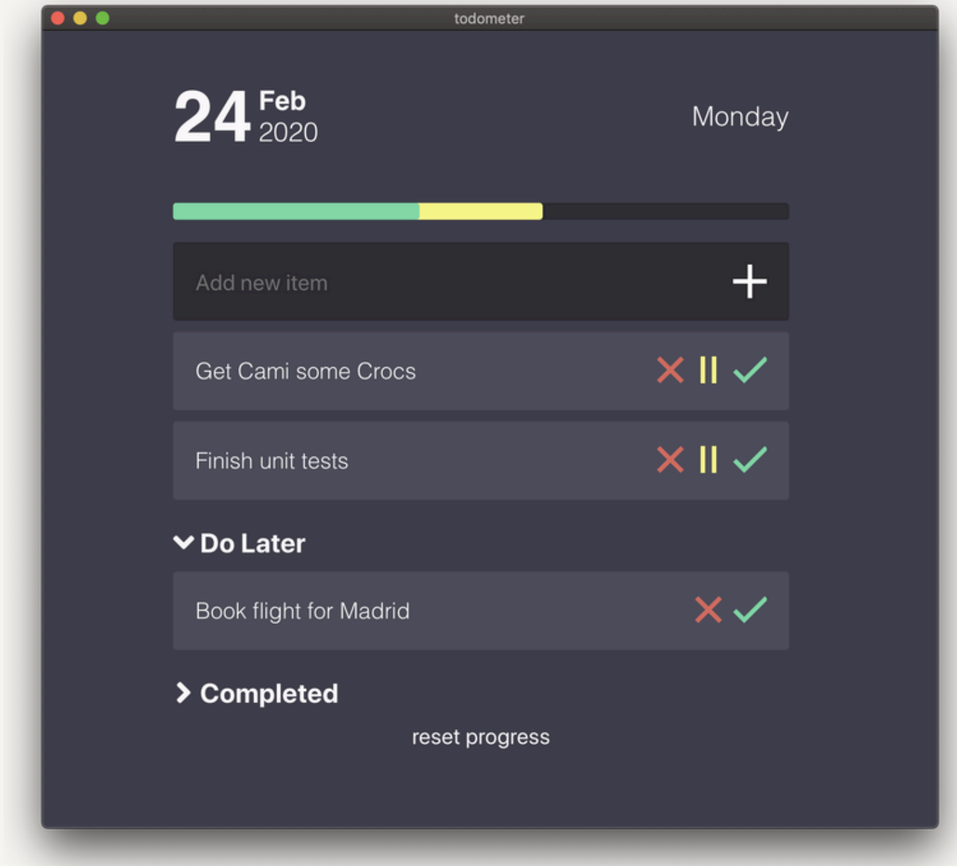
To do meter

Jason Ho – 5026221005

Fresnel Siregar – 5026221076

I Gusti Ngurah Adhya Pradipta – 5026221184

Todometer



Tampilan awal website to do meter yang dijalankan di lokal

Link github
https://github.com/jasonnho/DevOps-PSO

Source
https://github.com/cassidoo/todometer

project schedule ↘

Project Roadmap Plan

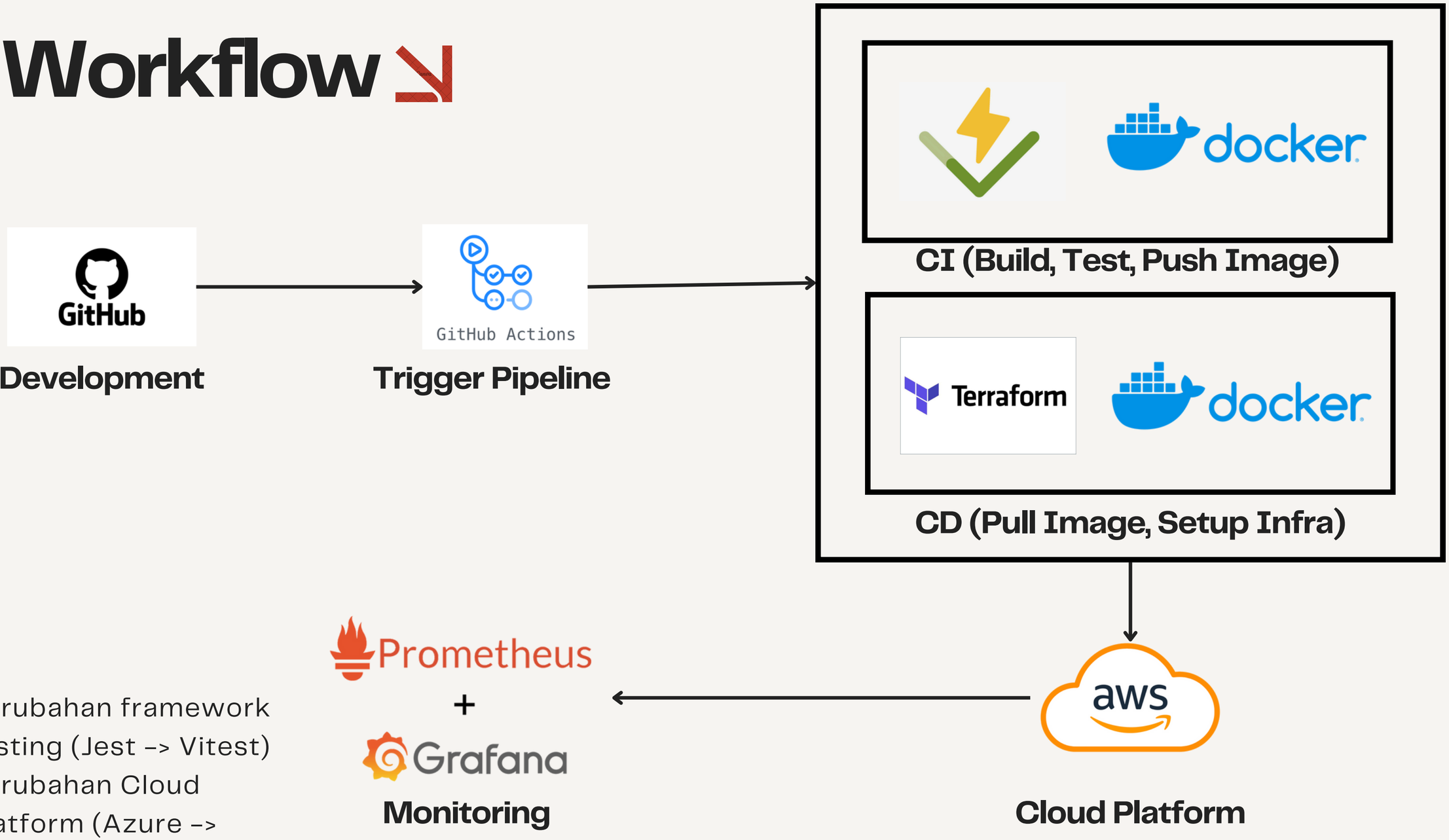
Project Title	To do Meter Website
Group Name	Adhya, Fresnel, Jason

Week	Objective	Assign To	Due date	% Task completion	Task Status	Week 1					Week 2							Week 3							Week 4					Week 5				
						S	S	R	K	J	S	M	S	S	R	K	J	S	M	S	S	R	K	J	S	S	R	K	J	S	S	R	K	J
Project Planning and Initial Setup																																		
Week 1	Define the Project Scope and Objectives	Adhya	15/11/2024	100,00%	Done																													
	Create a Project Plan with Timelines and Milestones	Fresnel	15/11/2024	100,00%	Done																													
	Initial research on relevant tools and technologies.	Jason	15/11/2024	100,00%	Done																													
	Set Up a Version Control System	Jason	15/11/2024	100,00%	Done																													
Begin drafting the pipeline design and select tools																																		
Week 2	Finalize tools and architecture for chosen approach	Jason	22/11/2024	0,00%	Open																													
	Initial Configuration of CI/CD Tools	Adhya	22/11/2024	0,00%	Open																													
	Configure a CI tool to automate the build and testing process.	Fresnel	22/11/2024	0,00%	Open																													
	Implement branching strategies and code review processes	Fresnel	22/11/2024	0,00%	Open																													
Initial Prototype and Documentation																																		
Week 3	Start initial development and unit testing of the application or model.	Fresnel	29/11/2024	0,00%	Open																													
	Develop scripts for automated deployment to a staging environment.	Jason	29/11/2024	0,00%	Open																													
	Implement a CD tool to automate deployment processes.	Adhya	29/11/2024	0,00%	Open																													
	Use Infrastructure as Code (IaC) tools to define and provision infrastructure.	Jason	29/11/2024	0,00%	Open																													
Testing and Refinement																																		
Week 4	Run the CI/CD pipeline through basic tests.	All	06/12/2024	0,00%	Open																													
	Troubleshoot and optimize deployment scripts for staging.	All	06/12/2024	0,00%	Open																													
	Set Up Monitoring Tools	Adhya	06/12/2024	0,00%	Open																													
	Implement Logging Mechanisms	Fresnel	06/12/2024	0,00%	Open																													
	Integrate Security Practices	Jason	06/12/2024	0,00%	Open																													
Final Presentation Preparation																																		
Week 5	Prepare Documentation and Final Presentation	All	13/12/2024	0,00%	Open																													

Link timeline proyek

Week 3

Workflow ↘



- Perubahan framework testing (Jest -> Vitest)
- Perubahan Cloud Platform (Azure -> AWS)

CI/CD Configuration ↘

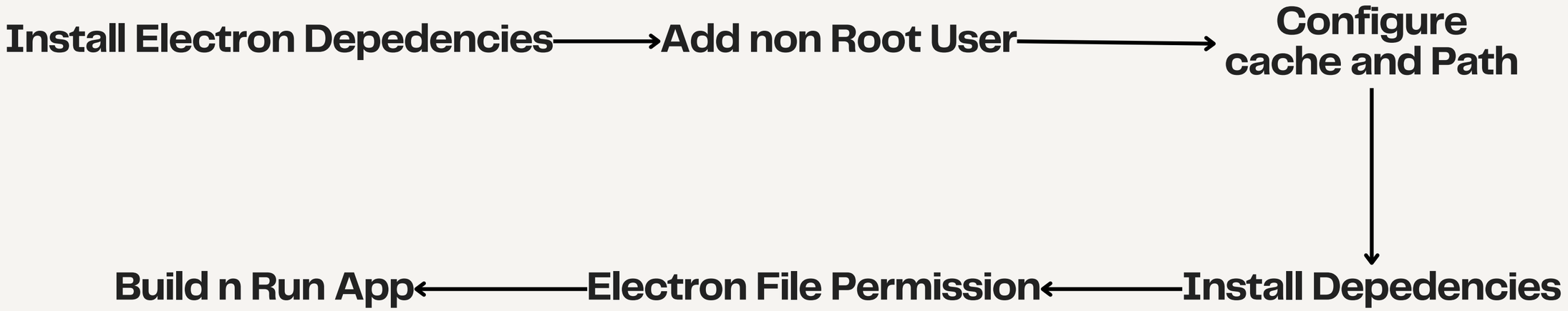
Dockerfile Configuration

```
FROM node:22

# Install dependencies needed for Electron
RUN apt-get update && apt-get install \
    git libx11-xcb1 libxcb-dri3-0 libxtst6 libnss3 libatk-bridge2.0-0 libgtk-3-0 libxss1 libasound2 libdrm2 libgbm1 \
    -yq --no-install-suggests --no-install-recommends \
    && apt-get clean && rm -rf /var/lib/apt/lists/*

# Add a non-root user
RUN useradd -m -d /custom-app custom-app
WORKDIR /custom-app
COPY . .

# Configure npm cache location
RUN npm config set cache /tmp/npm-cache

# Set npm global directory to avoid permission issues
ENV NPM_CONFIG_PREFIX=/home/node/.npm-global
ENV PATH=$PATH:/home/node/.npm-global/bin

# Install dependencies as root
USER root
RUN npm cache clean --force
```

```
# Set ownership of the working directory and node_modules
RUN mkdir -p /custom-app/node_modules && chown -R custom-app:custom-app /custom-app

# Install npm dependencies as the non-root user
USER custom-app
RUN npm install

# Electron-specific permissions
USER root
RUN chown root /custom-app/node_modules/electron/dist/chrome-sandbox
RUN chmod 4755 /custom-app/node_modules/electron/dist/chrome-sandbox

# Switch back to the non-root user
USER custom-app

# Start the application
CMD npm run start
```

Dockerfile Configuration ↘

Base Image : Node 22



CI Configuration

Build Docker Image -> Push Docker Image

```yaml
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Source
        uses: actions/checkout@v4

      - name: Build docker image
        run: docker build -t jasonnho/devops-pso .

      - name: Login to docker hub
        run: echo ${{ secrets.DOCKER_PASSWORD }} | docker login -u ${{ secrets.DOCKER_USERNAME }} --password-stdin

      - name: Publish image to docker hub
        run: docker push jasonnho/devops-pso:latest
```

CI Configuration

Testing (Render Tanggal & Add item)

```
test("renders the day of the month", () => {
  const { getByText } = render(<App />);
  const linkElement = getByText(format(new Date(), "d"));
  expect(linkElement).toBeInTheDocument();
});

test("renders the month", () => {
  const { getByText } = render(<App />);
  const linkElement = getByText(format(new Date(), "MMM"));
  expect(linkElement).toBeInTheDocument();
});

test("renders the year", () => {
  const { getByText } = render(<App />);
  const linkElement = getByText(format(new Date(), "y"));
  expect(linkElement).toBeInTheDocument();
});

test("renders the weekday", () => {
  const { getByText } = render(<App />);
  const linkElement = getByText(format(new Date(), "EEEE"));
  expect(linkElement).toBeInTheDocument();
});
```

```
test("adds a new item to the list", () => {
  const { getByPlaceholderText, getByText } = render(<App />);
  const input = getByPlaceholderText("Add new item");
  const addButton = getByText("", { selector: "button" });

  fireEvent.change(input, { target: { value: "New Task" } });
  fireEvent.click(addButton);

  expect(getByText("New Task")).toBeInTheDocument();
});
```

CI Configuration ↘

Pull Docker Image -> Run Test

```yaml
test:
  needs: build
  runs-on: ubuntu-latest
  steps:
    - name: Login to docker hub
      run: docker login -u ${{ secrets.DOCKER_USERNAME }} -p ${{ secrets.DOCKER_PASSWORD }}
    - name: Pull image from docker hub
      run: docker pull jasonnho/devops-pso:latest
    - name: Run vitest testing
      run: |
        docker run --rm jasonnho/devops-pso:latest sh -c "npm install && npm test"
```

CD Configuration

Terraform Config (main.tf)

```
data "aws_ami" "ubuntu" {
    most_recent = true

    filter {
        name   = "name"
        values = ["ubuntu/images/hvm-ssd/*20.04-amd64-server-*"]
    }

    filter {
        name   = "virtualization-type"
        values = ["hvm"]
    }

    owners = ["099720109477"] # Canonical
}
```

```
provider "aws" {
  region  = "ap-southeast-2"
}

resource "aws_instance" "app_server" {
  ami           = "ami-001f2488b35ca8aad"
  instance_type = "t2.micro"
  key_name      = "key-pair-devopspso"

  tags = {
    Name = "to-do-list"
  }
}
```

CD Configuration

Initialize Terraform -> Plan Infrastructure Changes -> Apply Changes

```yaml
terraform:
  needs: test
  runs-on: ubuntu-latest
  steps:
    - name: Checkout Source
      uses: actions/checkout@v4

    - name: Configure AWS credentials
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: ap-southeast-2

    - name: Set up Terraform
      uses: hashicorp/setup-terraform@v1
```

```yaml
- name: Set up Terraform
  uses: hashicorp/setup-terraform@v1

- name: Terraform Init
  run: terraform init

- name: Terraform Plan
  run: terraform plan
  env:
    AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
    AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}

- name: Terraform Apply
  run: terraform apply -auto-approve
  env:
    AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
    AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
```

CD Configuration ↘

Pull Docker Image (konfirmasi) –> Upload SSH Key (Autentikasi ke EC2)

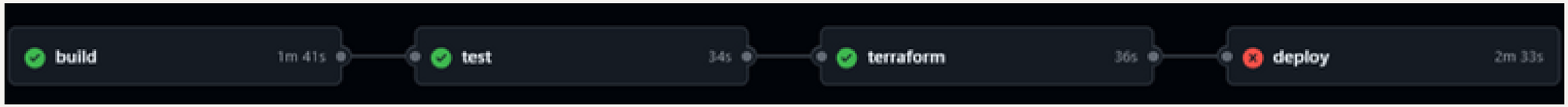
–> Deploy (Pull Image –> Remove Container Lama –> Run Container Baru)

```yaml
deploy:
  needs: terraform
  runs-on: ubuntu-latest
  steps:
    - name: Checkout Source
      uses: actions/checkout@v4

    - name: Configure AWS credentials
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: ap-southeast-2

    - name: Pull image from docker hub
      run: docker pull jasonnho/devops-pso:latest

    - name: Upload SSH Key
      run: |
        echo "${{ secrets.SSH_PRIVATE_KEY }}" > ec2-key.pem
        chmod 400 "ec2-key.pem"
        chmod 600 ec2-key.pem
```

→

```yaml
- name: Upload SSH Key
  run: |
    echo "${{ secrets.SSH_PRIVATE_KEY }}" > ec2-key.pem
    chmod 400 "ec2-key.pem"
    chmod 600 ec2-key.pem

- name: Deploy to EC22
  run: |
    ssh -o StrictHostKeyChecking=no -i "ec2-key.pem" ubuntu@ec2-3-27-126-61.ap-southeast-2.compute.amazonaws.com << 'EOF'
      docker pull jasonnho/devops-pso:latest
      docker rm -f devops-pso-container || true
      docker run -d -p 3000:3000 --name devops-pso-container jasonnho/devops-pso
    EOF
  env:
    EC2_INSTANCE_PUBLIC_DNS: ${{ secrets.EC2_INSTANCE_PUBLIC_DNS }}
```

Week 3 Progress



01

Docker image

- Mencari depedensi pada kode file untuk docker image
- Memperbaiki konfigurasi pada docker file

02

Testing

- Perubahan framework testing (Vitest)
- Menyesuaikan konfigurasi pada package dan docker untuk testing

03

Terraform and Deploy (ongoing)

- Perubahan aplikasi deployment (AWS)
- Melakukan set up pada AWS
- Konfigurasi awal IaC dengan Terraform

Thank you 

PSO Final Project

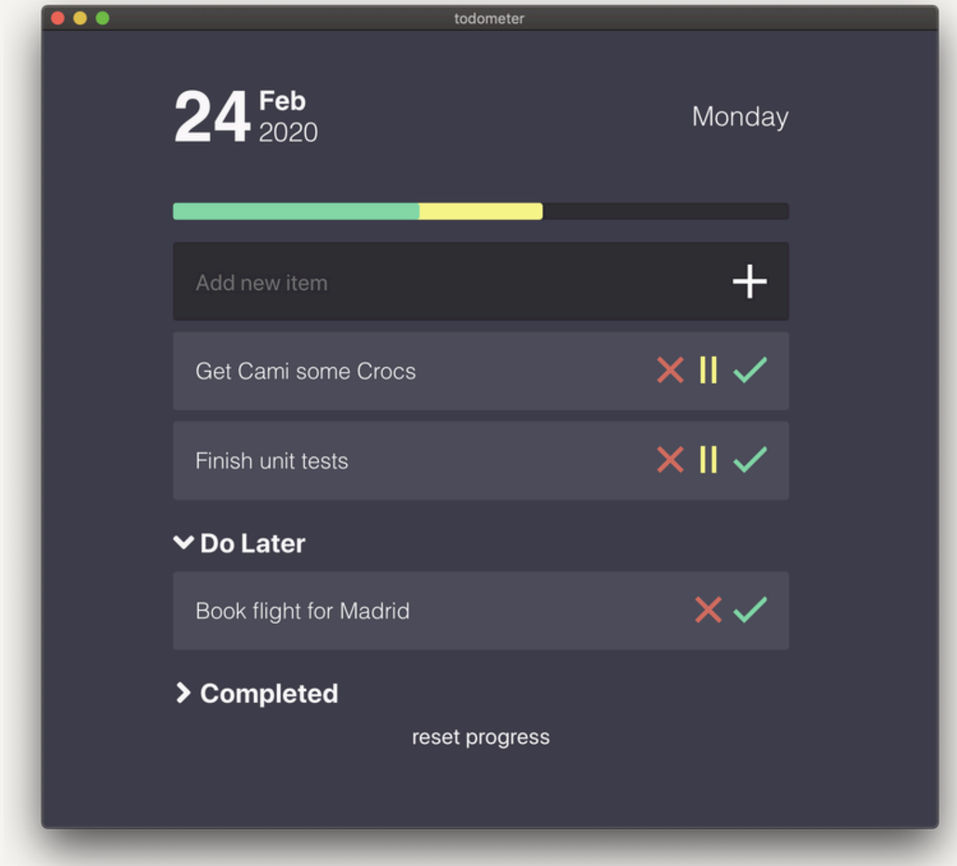
To do meter

Jason Ho – 5026221005

Fresnel Siregar – 5026221076

I Gusti Ngurah Adhya Pradipta – 5026221184

Todometer



Tampilan awal website to do meter yang dijalankan di lokal

Link github
https://github.com/jasonnho/DevOps-PSO

Source
https://github.com/cassidoo/todometer

project schedule ↘

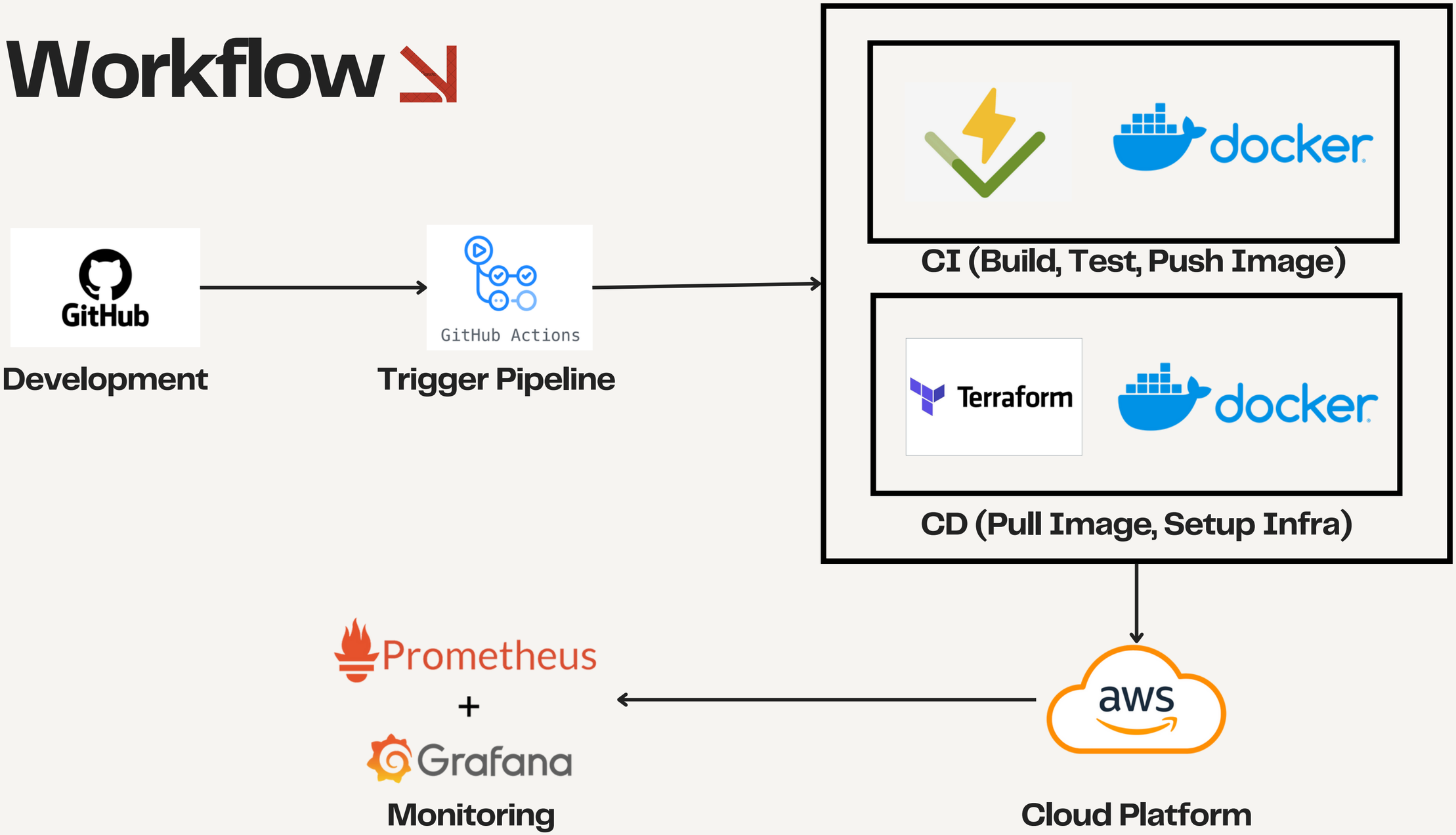
Project Roadmap Plan

Project Title	To do Meter Website
Group Name	Adhya, Fresnel, Jason

Week	Objective	Assign To	Due date	% Task completion	Task Status	Week 1					Week 2							Week 3							Week 4					Week 5				
						S	S	R	K	J	S	M	S	S	R	K	J	S	M	S	S	R	K	J	S	S	R	K	J	S	S	R	K	J
Project Planning and Initial Setup																																		
Week 1	Define the Project Scope and Objectives	Adhya	15/11/2024	100,00%	Done																													
	Create a Project Plan with Timelines and Milestones	Fresnel	15/11/2024	100,00%	Done																													
	Initial research on relevant tools and technologies.	Jason	15/11/2024	100,00%	Done																													
	Set Up a Version Control System	Jason	15/11/2024	100,00%	Done																													
Begin drafting the pipeline design and select tools																																		
Week 2	Finalize tools and architecture for chosen approach	Jason	22/11/2024	0,00%	Open																													
	Initial Configuration of CI/CD Tools	Adhya	22/11/2024	0,00%	Open																													
	Configure a CI tool to automate the build and testing process.	Fresnel	22/11/2024	0,00%	Open																													
	Implement branching strategies and code review processes	Fresnel	22/11/2024	0,00%	Open																													
Initial Prototype and Documentation																																		
Week 3	Start initial development and unit testing of the application or model.	Fresnel	29/11/2024	0,00%	Open																													
	Develop scripts for automated deployment to a staging environment.	Jason	29/11/2024	0,00%	Open																													
	Implement a CD tool to automate deployment processes.	Adhya	29/11/2024	0,00%	Open																													
	Use Infrastructure as Code (IaC) tools to define and provision infrastructure.	Jason	29/11/2024	0,00%	Open																													
Testing and Refinement																																		
Week 4	Run the CI/CD pipeline through basic tests.	All	06/12/2024	0,00%	Open																													
	Troubleshoot and optimize deployment scripts for staging.	All	06/12/2024	0,00%	Open																													
	Set Up Monitoring Tools	Adhya	06/12/2024	0,00%	Open																													
	Implement Logging Mechanisms	Fresnel	06/12/2024	0,00%	Open																													
	Integrate Security Practices	Jason	06/12/2024	0,00%	Open																													
Final Presentation Preparation																																		
Week 5	Prepare Documentation and Final Presentation	All	13/12/2024	0,00%	Open																													

Link timeline proyek

Workflow ↘


GitHub
Development
GitHub Actions
Trigger Pipeline
docker
CI (Build, Test, Push Image)
Terraform
docker
CD (Pull Image, Setup Infra)
aws
Cloud Platform
Prometheus
+
Grafana
Monitoring

CI Configuration

Build Docker Image -> Push Docker Image

```yaml
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Source
        uses: actions/checkout@v4

      - name: Build docker image
        run: docker build -t jasonnho/devops-pso .

      - name: Login to docker hub
        run: echo ${{ secrets.DOCKER_PASSWORD }} | docker login -u ${{ secrets.DOCKER_USERNAME }} --password-stdin

      - name: Publish image to docker hub
        run: docker push jasonnho/devops-pso:latest
```

CI Configuration

Testing (Render Tanggal & Add item)

```
test("renders the day of the month", () => {
  const { getByText } = render(<App />);
  const linkElement = getByText(format(new Date(), "d"));
  expect(linkElement).toBeInTheDocument();
});

test("renders the month", () => {
  const { getByText } = render(<App />);
  const linkElement = getByText(format(new Date(), "MMM"));
  expect(linkElement).toBeInTheDocument();
});

test("renders the year", () => {
  const { getByText } = render(<App />);
  const linkElement = getByText(format(new Date(), "y"));
  expect(linkElement).toBeInTheDocument();
});

test("renders the weekday", () => {
  const { getByText } = render(<App />);
  const linkElement = getByText(format(new Date(), "EEEE"));
  expect(linkElement).toBeInTheDocument();
});
```

```
test("adds a new item to the list", () => {
  const { getByPlaceholderText, getByText } = render(<App />);
  const input = getByPlaceholderText("Add new item");
  const addButton = getByText("", { selector: "button" });

  fireEvent.change(input, { target: { value: "New Task" } });
  fireEvent.click(addButton);

  expect(getByText("New Task")).toBeInTheDocument();
});
```

CI Configuration ↘

Pull Docker Image -> Run Test

```yaml
test:
  needs: build
  runs-on: ubuntu-latest
  steps:
    - name: Login to docker hub
      run: docker login -u ${{ secrets.DOCKER_USERNAME }} -p ${{ secrets.DOCKER_PASSWORD }}
    - name: Pull image from docker hub
      run: docker pull jasonnho/devops-pso:latest
    - name: Run vitest testing
      run: |
        docker run --rm jasonnho/devops-pso:latest sh -c "npm install && npm test"
```

CD Configuration

Terraform Config (main.tf)

```
data "aws_ami" "ubuntu" {
    most_recent = true

    filter {
        name   = "name"
        values = ["ubuntu/images/hvm-ssd/*20.04-amd64-server-*"]
    }

    filter {
        name   = "virtualization-type"
        values = ["hvm"]
    }

    owners = ["099720109477"] # Canonical
}
```

```
provider "aws" {
  region  = "ap-southeast-2"
}

resource "aws_instance" "app_server" {
  ami           = "ami-001f2488b35ca8aad"
  instance_type = "t2.micro"
  key_name      = "key-pair-devopspso"

  tags = {
    Name = "to-do-list"
  }
}
```

CD Configuration

Initialize Terraform -> Plan Infrastructure Changes -> Apply Changes

```yaml
terraform:
  needs: test
  runs-on: ubuntu-latest
  steps:
    - name: Checkout Source
      uses: actions/checkout@v4

    - name: Configure AWS credentials
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: ap-southeast-2

    - name: Set up Terraform
      uses: hashicorp/setup-terraform@v1
```

```yaml
- name: Set up Terraform
  uses: hashicorp/setup-terraform@v1

- name: Terraform Init
  run: terraform init

- name: Terraform Plan
  run: terraform plan
  env:
    AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
    AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}

- name: Terraform Apply
  run: terraform apply -auto-approve
  env:
    AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
    AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
```

CD Configuration ↘

Pull Docker Image (konfirmasi) –> Upload SSH Key (Autentikasi ke EC2)

–> Deploy (Pull Image –> Remove Container Lama –> Run Container Baru)

```yaml
deploy:
  needs: terraform
  runs-on: ubuntu-latest
  steps:
    - name: Checkout Source
      uses: actions/checkout@v4

    - name: Configure AWS credentials
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: ap-southeast-2

    - name: Pull image from docker hub
      run: docker pull jasonnho/devops-pso:latest

    - name: Upload SSH Key
      run: |
        echo "${{ secrets.SSH_PRIVATE_KEY }}" > ec2-key.pem
        chmod 400 "ec2-key.pem"
        chmod 600 ec2-key.pem
```

→

```yaml
- name: Start Xvfb
  run: |
    Xvfb :99 -screen 0 1024x768x16 &
    export DISPLAY=:99
- name: Deploy Application
  run: |
    ssh -o StrictHostKeyChecking=no -T -i "key-pair-devopspso.pem" ubuntu@ec2-52-65-0-117.ap-southeast-2.compute.amazonaws.com << 'EOF'
      docker pull jasonnho/devops-pso:latest
      docker rm -f devops-pso-container || true
      docker run --privileged -d -e DISPLAY=:99 -p 3000:3000 --name devops-pso-container jasonnho/devops-pso
      docker logs devops-pso-container
    EOF
  env:
    EC2_INSTANCE_PUBLIC_DNS: ${{ secrets.EC2_INSTANCE_PUBLIC_DNS }}
```

Error deployment

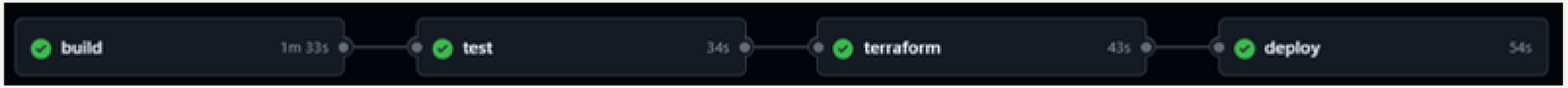
Terdapat beberapa error dalam log container meskipun tahap pipeline sudah berhasil dan saat ini masih proses troubleshooting

```
watching for file changes...

build started...
transforming...
✓ 2 modules transformed.
rendering chunks...
../../dist/main/index.cjs   7.28 KiB
../../dist/main/index.cjs.map 11.87 KiB
built in 195ms.
/custom-app/node_modules/electron-is-dev/index.js:5
        throw new TypeError('Not running in an Electron environment!');
        ^

TypeError: Not running in an Electron environment!
    at Object.<anonymous> (/custom-app/node_modules/electron-is-dev/index.js:5
    at Module._compile (node:internal/modules/cjs/loader:1141:14)
    at Module._extensions..js (node:internal/modules/cjs/loader:1196:10)
    at Module.load (node:internal/modules/cjs/loader:1011:32)
    at Module._load (node:internal/modules/cjs/loader:846:12)
    at f._load (node:electron/js2c/asar_bundle:2:13330)
    at Module.require (node:internal/modules/cjs/loader:1035:19)
    at require (node:internal/modules/cjs/helpers:102:18)
    at Object.<anonymous> (/custom-app/dist/main/index.cjs:4:15)
```

```
> todometer@2.0.1 start
> DEBUG=electron:* npm run dev


> todometer@2.0.1 dev
> node src/scripts/watch.mjs

[54:1201/190332.010766:ERROR:ozone_platform_x11.cc(239)] Missing X server or $DISPLAY
[54:1201/190332.010815:ERROR:env.cc(255)] The platform failed to initialize.  Exiting.
```

```
build started...
transforming...
✓ 1 modules transformed.
rendering chunks...
../../dist/preload/index.cjs   0.29 KiB
../../dist/preload/index.cjs.map 0.67 KiB
built in 56ms.
vite v3.1.8 building SSR bundle for development...

watching for file changes...

build started...
transforming...
✓ 2 modules transformed.
rendering chunks...
../../dist/main/index.cjs   7.28 KiB
../../dist/main/index.cjs.map 11.87 KiB
built in 145ms.
[49:1206/045927.584244:FATAL:setuid_sandbox_host.cc(157)] The SUID sandbox helper binary was found, but is not configure
d correctly. Rather than run without sandboxing I'm aborting now. You need to make sure that /custom-app/node_modules/el
ectron/dist/chrome-sandbox is owned by root and has mode 4755.
npm notice
npm notice New patch version of npm available! 10.9.0 -> 10.9.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.9.2
npm notice To update run: npm install -g npm@10.9.2
npm notice
```

Week 4 Progress



01

Deployment

- Pipeline berhasil dijalankan hingga pada tahap deployment

02

Deployment Error

- App tidak dapat diakses karena error pada container

03

Todo

- Debug Container hingga app bisa diakses melalui public dns
- Setup Monitoring Tools
- Development perubahan tampilan aplikasi

Thank you 

PSO Final Project

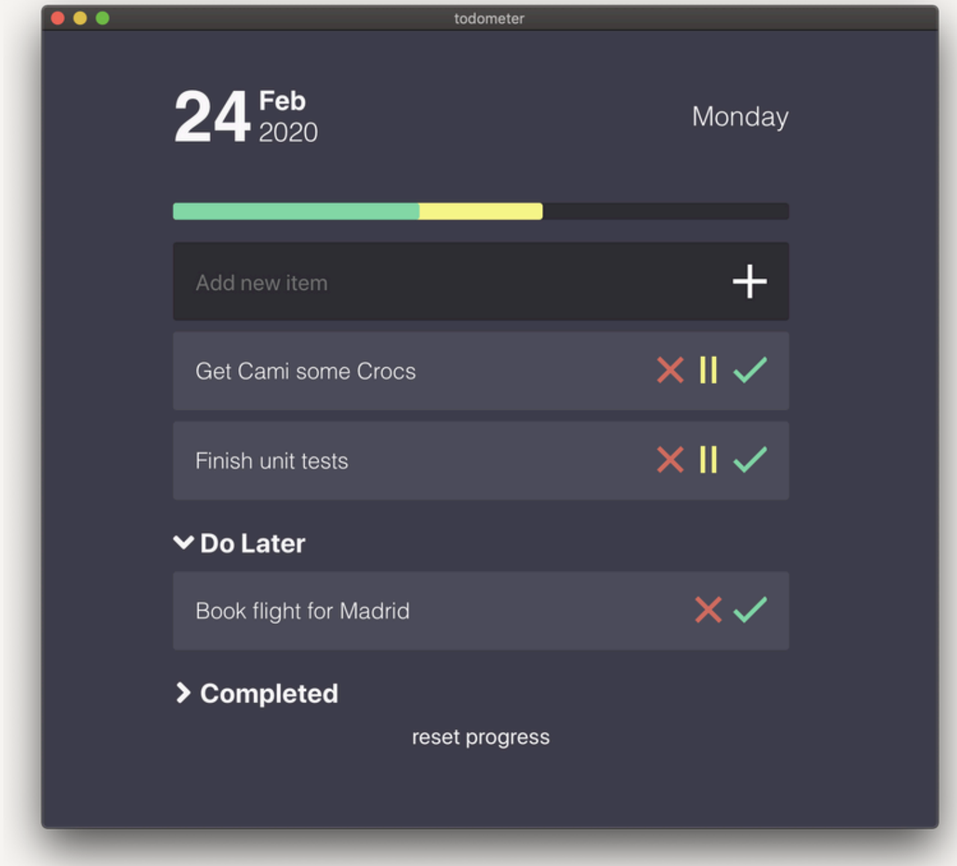
To do meter

Jason Ho – 5026221005

Fresnel Siregar – 5026221076

I Gusti Ngurah Adhya Pradipta – 5026221184

Todometer



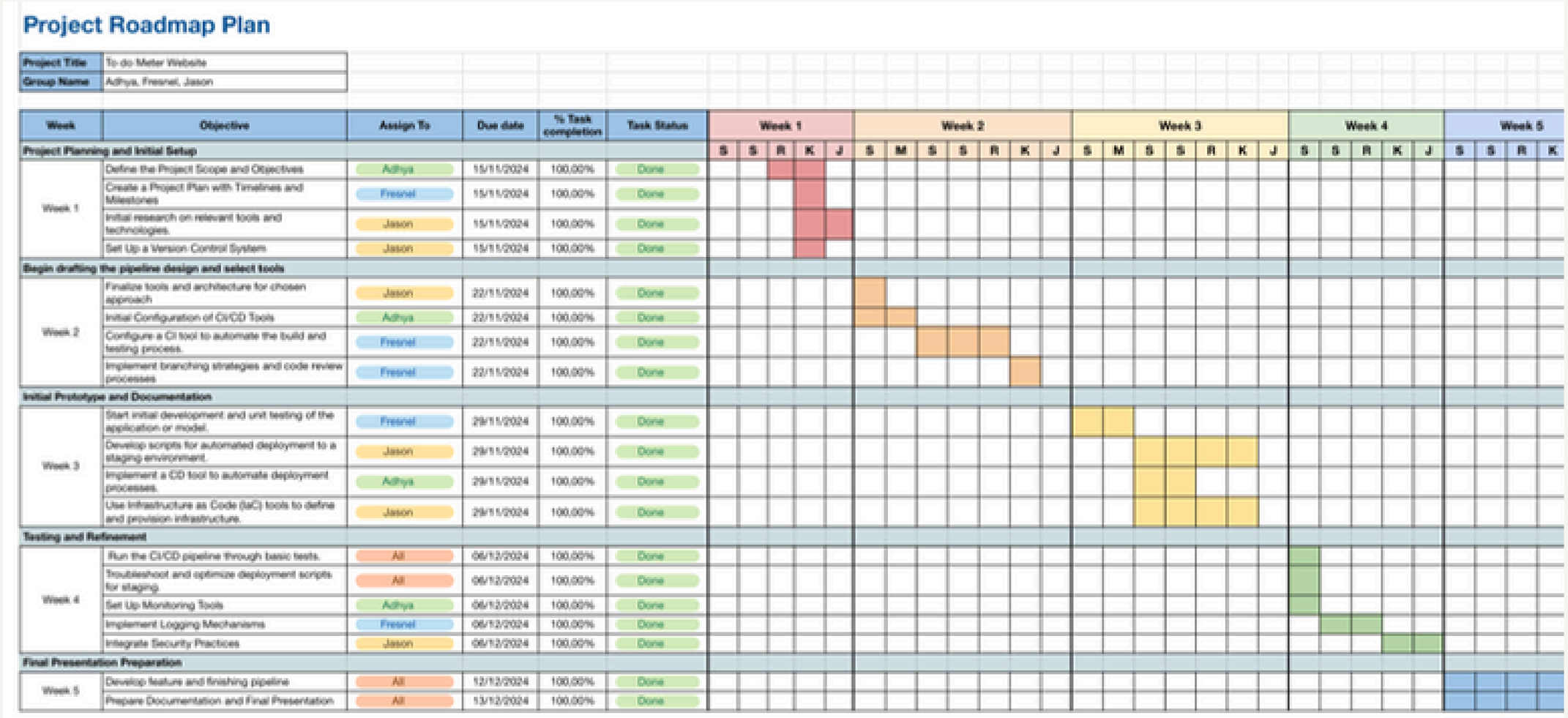
Tampilan awal aplikasi to do meter yang dijalankan di lokal

GitHub

Link github
https://github.com/jasonnho/DevOps-PSO

Source
https://github.com/cassidoo/todometer

Project Roadmap ↘



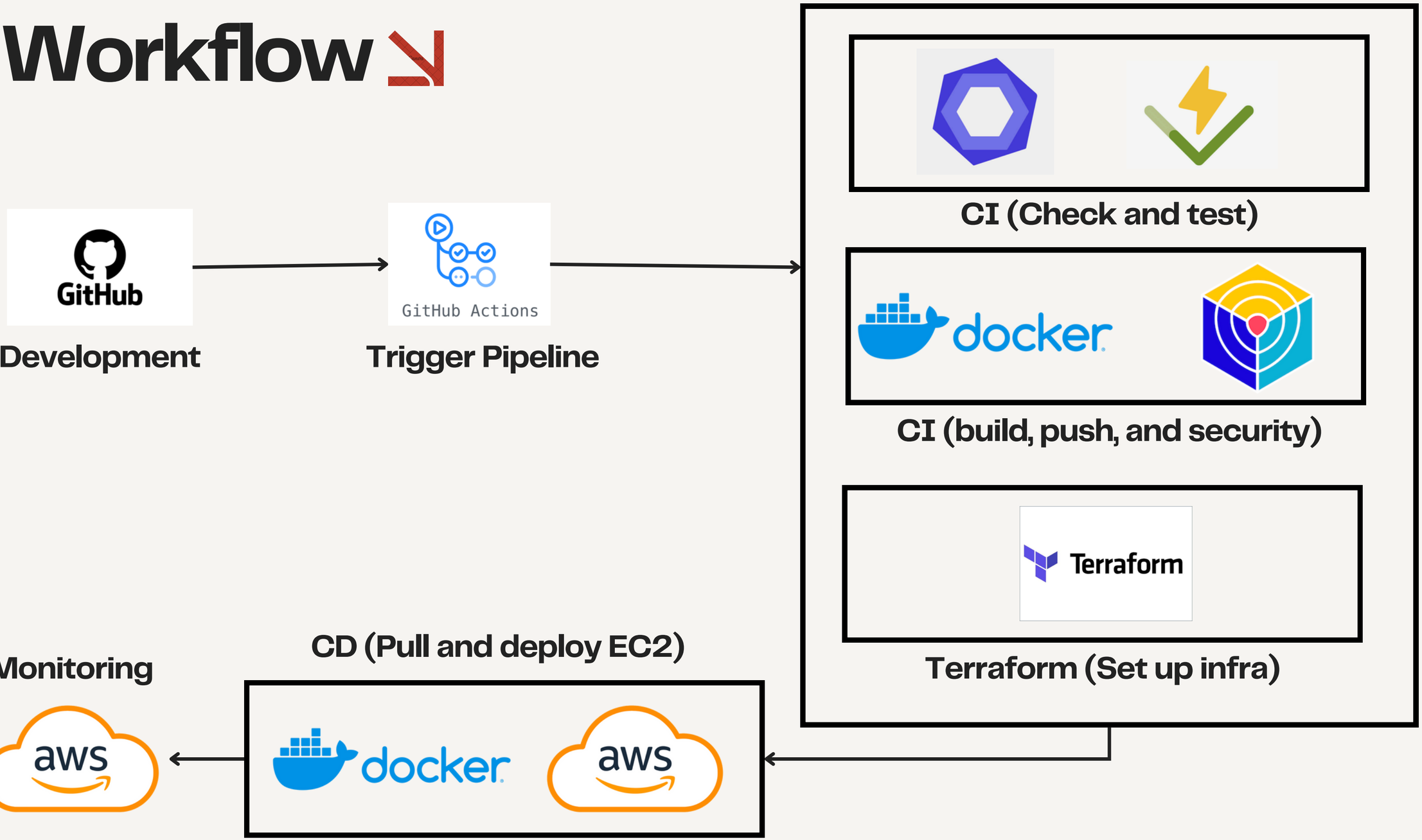
Project Roadmap Plan

Project Title	To do Meter Website
Group Name	Adhya, Fresnel, Jason

Week	Objective	Assign To	Due date	% Task completion	Task Status	Week 1					Week 2							Week 3							Week 4					Week 5				
						S	S	R	K	J	S	M	S	S	R	K	J	S	M	S	S	R	K	J	S	S	R	K	J	S	S	R	K
Project Planning and Initial Setup																																	
Week 1	Define the Project Scope and Objectives	Adhya	15/11/2024	100.00%	Done																												
	Create a Project Plan with Timelines and Milestones	Fresnel	15/11/2024	100.00%	Done																												
	Initial research on relevant tools and technologies.	Jason	15/11/2024	100.00%	Done																												
	Set Up a Version Control System	Jason	15/11/2024	100.00%	Done																												
Begin drafting the pipeline design and select tools																																	
Week 2	Finalize tools and architecture for chosen approach	Jason	22/11/2024	100.00%	Done																												
	Initial Configuration of CI/CD Tools	Adhya	22/11/2024	100.00%	Done																												
	Configure a CI tool to automate the build and testing process.	Fresnel	22/11/2024	100.00%	Done																												
	Implement branching strategies and code review processes	Fresnel	22/11/2024	100.00%	Done																												
Initial Prototype and Documentation																																	
Week 3	Start initial development and unit testing of the application or model.	Fresnel	29/11/2024	100.00%	Done																												
	Develop scripts for automated deployment to a staging environment.	Jason	29/11/2024	100.00%	Done																												
	Implement a CD tool to automate deployment processes.	Adhya	29/11/2024	100.00%	Done																												
	Use Infrastructure as Code (IaC) tools to define and provision infrastructure.	Jason	29/11/2024	100.00%	Done																												
Testing and Refinement																																	
Week 4	Run the CI/CD pipeline through basic tests.	All	06/12/2024	100.00%	Done																												
	Troubleshoot and optimize deployment scripts for staging	All	06/12/2024	100.00%	Done																												
	Set Up Monitoring Tools	Adhya	06/12/2024	100.00%	Done																												
	Implement Logging Mechanisms	Fresnel	06/12/2024	100.00%	Done																												
	Integrate Security Practices	Jason	06/12/2024	100.00%	Done																												
Final Presentation Preparation																																	
Week 5	Develop feature and finishing pipeline	All	12/12/2024	100.00%	Done																												
	Prepare Documentation and Final Presentation	All	13/12/2024	100.00%	Done																												

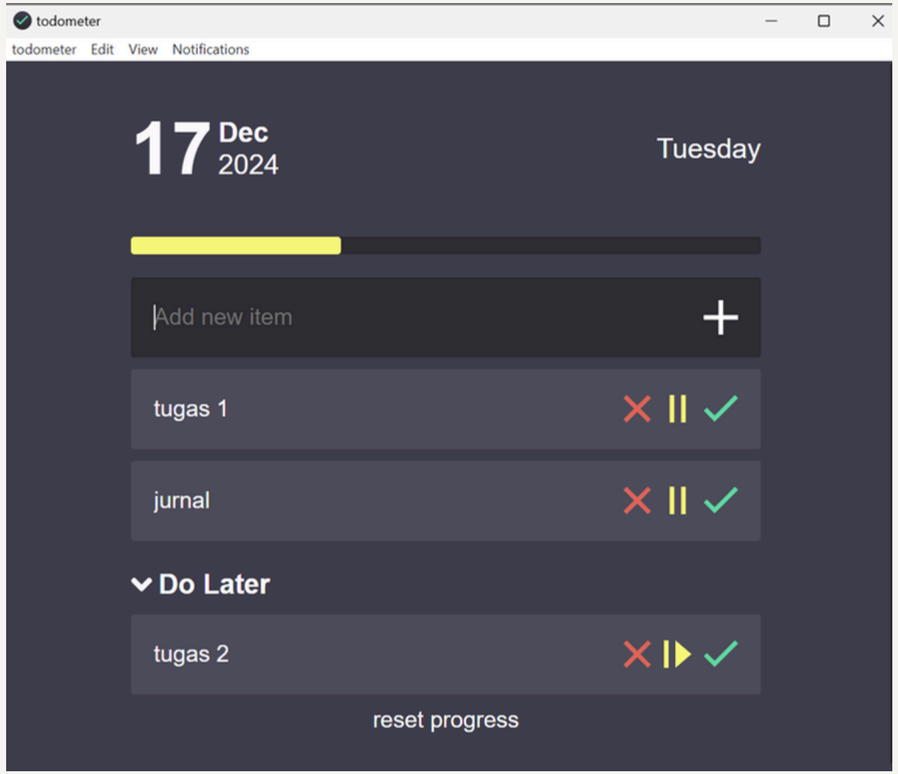
https://docs.google.com/spreadsheets/d/1rlpAe51AVMyojo2cUZJR65ZSFcVW3kUZ6eOOq-I1JuI/edit?usp=sharing

Workflow ↘

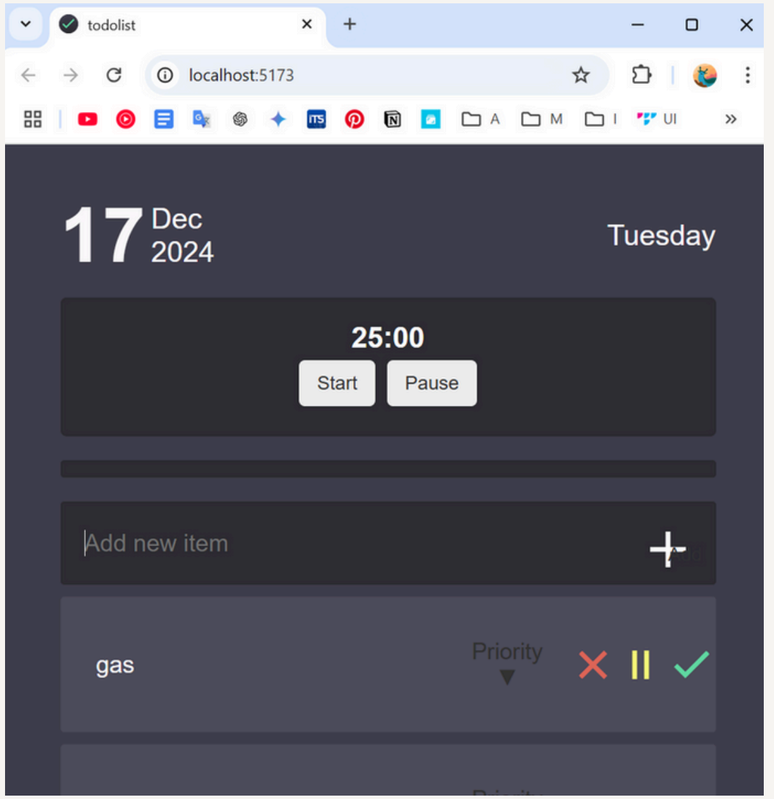

GitHub
Development
GitHub Actions
Trigger Pipeline
CI (Check and test)
docker
CI (build, push, and security)
Terraform
Terraform (Set up infra)
CD (Pull and deploy EC2)
docker
aws
aws
Monitoring

Development Todo Meter ↘

Code awal



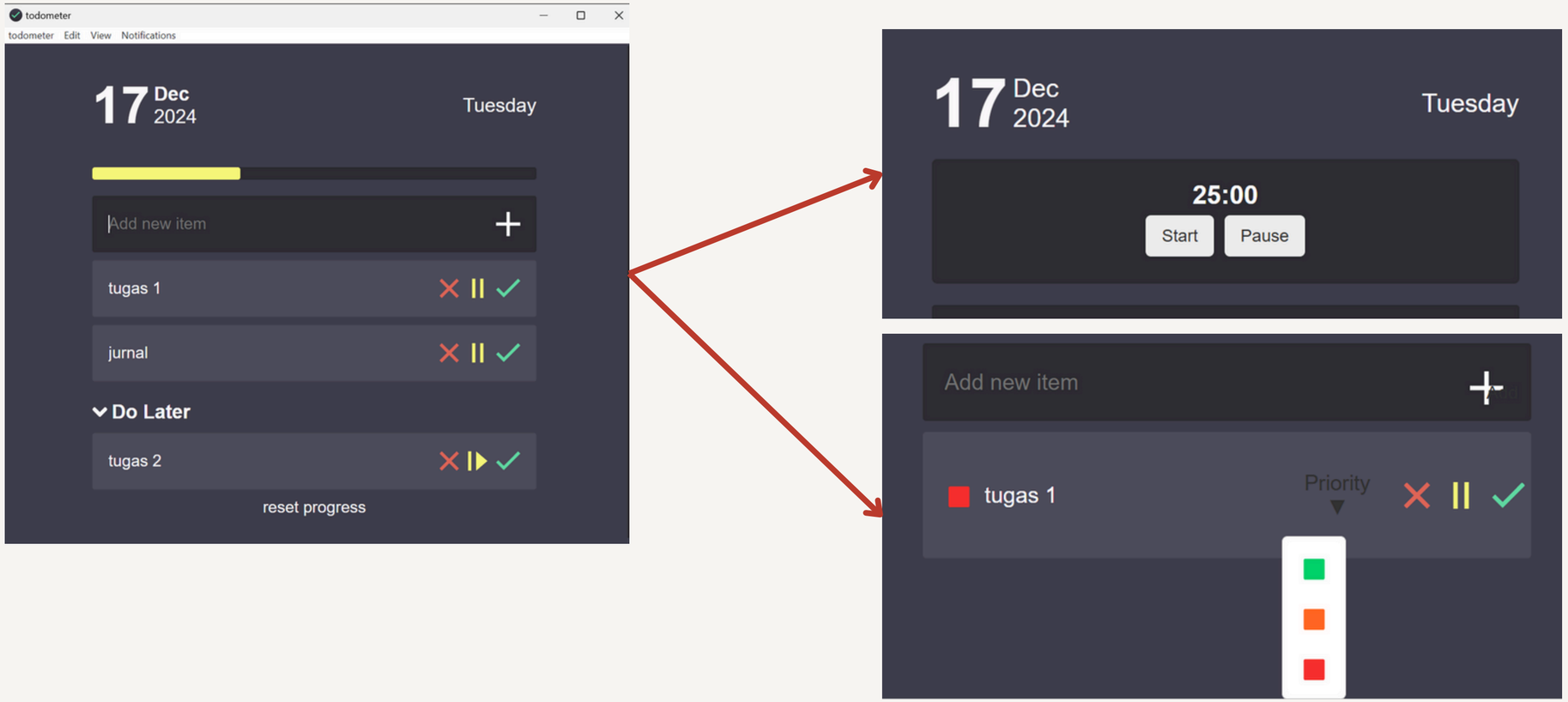
Setelah development



Perubahan aplikasi

Menghapus depedensi electron sehingga aplikasi menjadi berbasis website dengan menggunakan React dan Javascript

Development Todo Meter ↘



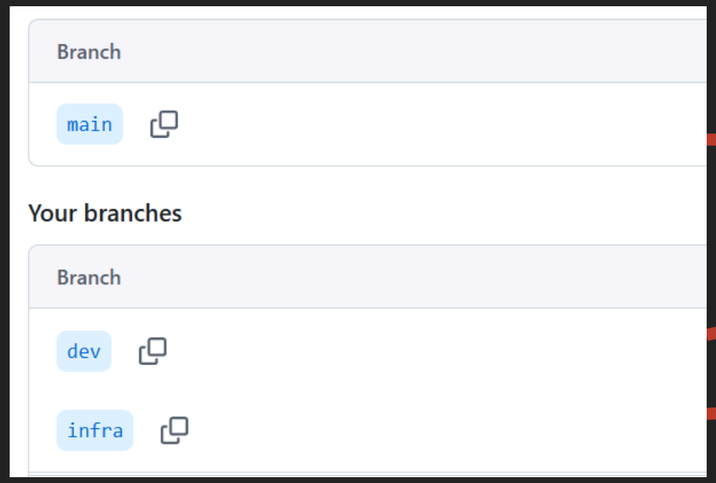
Penambahan timer 25 menit working time dan 5 menit break time

Penambahan fitur opsi warna sebagai indikasi **skala prioritas** tugas

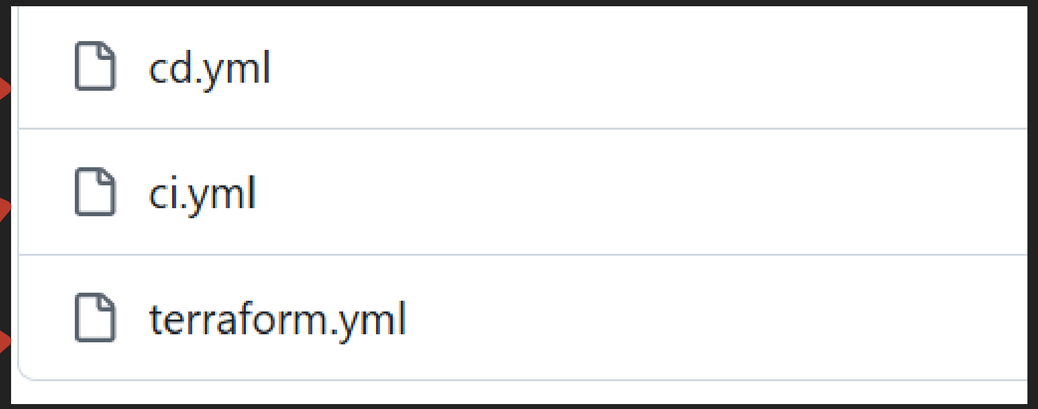
Melakukan pengembangan dari kodeTodolist awal menjadi Pomodoro Todolist dengan penambahan beberapa fitur

Branching Strategies

Branch



Github workflow

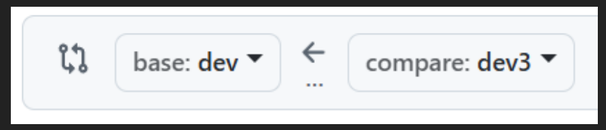


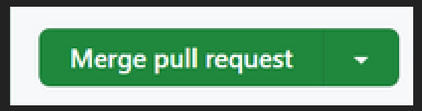
Main: Branch utama yang memicu continuous deployment (CD) agar setiap perubahan kode yang diterima branch ini akan otomatis memicu proses deployment aplikasi.

Infra: Branch ini digunakan untuk memicu Terraform yang mengelola konfigurasi infrastruktur, khususnya di instance AWS.

Dev: digunakan sebagai branch untuk continous integration (CI) sehingga perubahan yang dilakukan akan diuji terlebih dahulu melalui pull request branch ini termasuk pembuatan docker image.

Workflow implementation

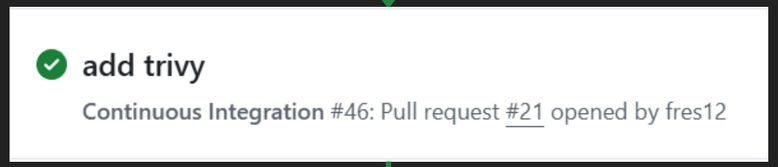




Melakukan development dan publish branch baru

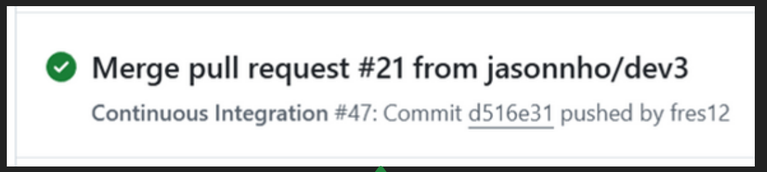
Membuat pull request ke branch dev

Trigger CI



Menggabungkan dengan branch dev

Trigger CI



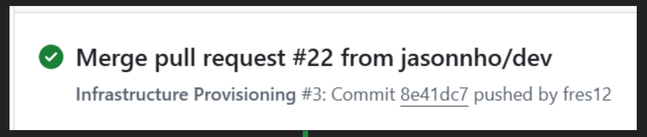
Jika pull sudah berhasil step dilanjutkan ke merging

Workflow implementation

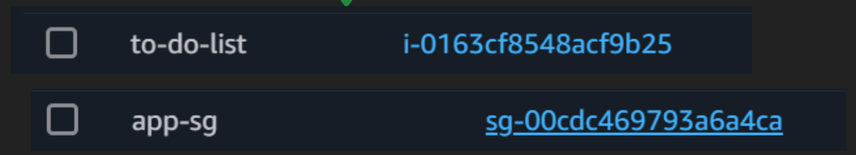
`infra`

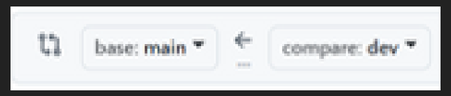
Menjalankan pipeline terraform

Trigger terraform



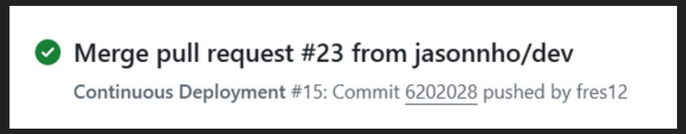
Output instance dan security group baru





Melakukan merging ke branch main

Trigger CD



Aplikasi berhasil dideploy

CI Configuration

Run Lint -> Run Test

```yaml
name: Continuous Integration

on:
  pull_request:
    branches:
      - dev
  push:
    branches:
      - dev

jobs:
  lint:
    name: Run ESLint
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Source Code
        uses: actions/checkout@v4

      - name: Set Up Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '20'

      - name: Install Dependencies for Linting
        run: npm ci --legacy-peer-deps

      - name: Run ESLint
        run: npm run lint
```

```yaml
  unit-test:
    name: Run Unit Tests
    runs-on: ubuntu-latest
    needs: lint
    steps:
      - name: Checkout Source Code
        uses: actions/checkout@v4

      - name: Set Up Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '20'

      - name: Install Dependencies for Testing
        run: npm ci --legacy-peer-deps

      - name: Execute Unit Tests
        run: npm test
```

ESLint Config

- Konfigurasi awal berasal dari file vite yang dibuat
- Penambahan konfigurasi eslint-plugin-security sebagai tools keamanan dan pengecekan kode

```
import js from '@eslint/js';
import globals from 'globals';
import react from 'eslint-plugin-react';
import reactHooks from 'eslint-plugin-react-hooks';
import reactRefresh from 'eslint-plugin-react-refresh';
import security from 'eslint-plugin-security'; // Import

export default [
  { ignores: ['dist'] },
  {
    files: ['**/*.{js,jsx}'],
    languageOptions: {
      ecmaVersion: 2020,
      globals: {
        ...globals.browser,
        describe: 'readonly',
        it: 'readonly',
        expect: 'readonly',
        beforeEach: 'readonly',
        afterEach: 'readonly',
        vi: 'readonly',
      },
      parserOptions: {
        ecmaVersion: 'latest',
        ecmaFeatures: { jsx: true },
        sourceType: 'module',
      },
    },
```

```
    settings: { react: { version: '18.3' } },
    plugins: {
      react,
      'react-hooks': reactHooks,
      'react-refresh': reactRefresh,
      security, // Add the security plugin
    },
    rules: {
      ...js.configs.recommended.rules,
      ...react.configs.recommended.rules,
      ...react.configs['jsx-runtime'].rules,
      ...reactHooks.configs.recommended.rules,
      'react/jsx-no-target-blank': 'off',
      'react-refresh/only-export-components': [
        'warn',
        { allowConstantExport: true },
      ],
      // Add security plugin rules
      ...security.configs.recommended.rules,
    },
  },
];
```

Test file

```
import { test, expect } from "vitest";
import { render, fireEvent } from "@testing-library/react";
import { format } from "date-fns";
import App from "./App.jsx";

  test("renders the day of the month", () => {
    const { getByText } = render(<App />);
    const linkElement = getByText(format(new Date(), "d"));
    expect(linkElement).toBeInTheDocument();
  });

  test("renders the month", () => {
    const { getByText } = render(<App />);
    const linkElement = getByText(format(new Date(), "MMM"));
    expect(linkElement).toBeInTheDocument();
  });

  test("renders the year", () => {
    const { getByText } = render(<App />);
    const linkElement = getByText(format(new Date(), "y"));
    expect(linkElement).toBeInTheDocument();
  });

  test("renders the weekday", () => {
    const { getByText } = render(<App />);
    const linkElement = getByText(format(new Date(), "EEEE"));
    expect(linkElement).toBeInTheDocument();
  });
```

Tes render tanggal/bulan/tahun

Tes tombol add item

```
test("adds a new item to the list", () => {
  const { getByPlaceholderText, getByTestId, getByText } = render(<App />);
  const input = getByPlaceholderText("Add new item");
  const addButton = getByTestId("add-button");

  fireEvent.change(input, { target: { value: "New Task" } });
  fireEvent.click(addButton);

  expect(getByText("New Task")).toBeInTheDocument();
});
```

Test file

Tes tombol delete dan pause

```
vi.mock('../AppContext', () => ({
  useAppReducer: vi.fn(),
}));

describe('Item Component', () => {
  let dispatchMock;

  beforeEach(() => {
    dispatchMock = vi.fn();
    useAppReducer.mockReturnValue(dispatchMock);
  });

  afterEach(() => {
    vi.clearAllMocks();
  });

  const item = {
    id: 1,
    text: 'Test item',
    status: 'pending',
  };
```

```
  it('dispatches DELETE_ITEM when delete button is clicked', () => {
    const { getByTestId } = render(<Item item={item} />);
    const deleteButton = getByTestId('delete-button');

    fireEvent.click(deleteButton);

    expect(dispatchMock).toHaveBeenCalledWith({ type: 'DELETE_ITEM', item });
  });

  it('dispatches UPDATE_ITEM with paused status when pause button is clicked', () => {
    const { getByTestId } = render(<Item item={item} />);
    const pauseButton = getByTestId('pause-button');

    fireEvent.click(pauseButton);

    expect(dispatchMock).toHaveBeenCalledWith({
      type: 'UPDATE_ITEM',
      item: { ...item, status: 'paused' }
    });
  });
```

Test file ↘

Tes tombol pause (untuk resume)

```
  it('dispatches UPDATE_ITEM with pending status when resume button is clicked', () =>
    const pausedItem = { ...item, status: 'paused' };
    const { getByLabelText } = render(<Item item={pausedItem} />);
    const resumeButton = getByLabelText('Resume item');

    fireEvent.click(resumeButton);

    expect(dispatchMock).toHaveBeenCalledWith({
      type: 'UPDATE_ITEM',
      item: { ...pausedItem, status: 'pending' }
    });
  });
});
```

CI Configuration ↘

Build Docker Image (Build -> Push ke Docker Hub)

```yaml
build:
  name: Build Docker Image
  runs-on: ubuntu-latest
  needs: unit-test
  steps:
    - name: Checkout Source Code
      uses: actions/checkout@v4

    - name: Build Docker Image
      run: |
        docker build --no-cache -t jasonnho/devops-pso .
        docker tag jasonnho/devops-pso jasonnho/devops-pso:latest

    - name: Log in to Docker Hub
      run: echo "${{ secrets.DOCKER_PASSWORD }}" | docker login -u "${{ secrets.DOCKER_USERNAME }}" --password-stdin

    - name: Push Docker Image to Docker Hub
      run: |
        docker push jasonnho/devops-pso:latest
```

Dockerfile Config

Install Depedencies -> Run Test -> Build -> Install Serve -> Serve

```
FROM node:20-alpine3.18 as builder
WORKDIR /app

COPY package*.json ./
RUN npm ci --legacy-peer-deps

# Copy the rest of the application code
COPY . .

# Run tests
RUN npm run test
```

```
RUN npm run build

FROM node:20-alpine3.18
WORKDIR /app

# Copy the built application from the builder s
COPY --from=builder /app/dist ./dist

# Install serve globally for serving the built
RUN npm install -g serve

EXPOSE 3000
CMD ["serve", "-s", "dist", "-l", "3000"]
```

CI Configuration

Trivy Security Check

```yaml
scan:
  name: Scan Docker Image with Trivy
  runs-on: ubuntu-latest
  needs: build
  steps:
    - name: Checkout Source Code
      uses: actions/checkout@v4

    - name: Scan and print log docker image with trivy
      uses: aquasecurity/trivy-action@0.28.0
      with:
        image-ref: jasonnho/devops-pso:latest
        severity: HIGH,CRITICAL
        ignore-unfixed: true
        format: table
        exit-code: 0
        quiet: true
        no-progress: true
```

CD Configuration

Terraform Config (main.tf)

```
data "aws_ami" "ubuntu" {
  most_recent = true

  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/*20.04-amd64-server-*"]
  }

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }

  owners = ["099720109477"] # Canonical
}

provider "aws" {
  region = "ap-southeast-2"
}
```

```
# Security Group to allow SSH (22) and HTTP (3000) access
resource "aws_security_group" "app_sg" {
  name        = "app_security_group"
  description = "Allow SSH and HTTP access"

  ingress {
    description = "Allow SSH"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    description = "Allow HTTP on port 3000"
    from_port   = 3000
    to_port     = 3000
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

```
  egress {
    description = "Allow all outbound traffic"
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "app-sg"
  }
}

# EC2 Instance
resource "aws_instance" "app_server" {
  ami           = data.aws_ami.ubuntu.id
  instance_type = "t2.micro"
  key_name      = "key-pair-devopspso"
  security_groups = [aws_security_group.app_sg.name]

  tags = {
    Name = "to-do-list"
  }
}
```

CD Configuration

Initialize Terraform -> Plan Infrastructure Changes -> Apply Changes

```yaml
name: Infrastructure Provisioning

on:
   push:
    branches:
      - infra

jobs:
  terraform:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Source
        uses: actions/checkout@v4

      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: ap-southeast-2
```

```yaml
    - name: Set up Terraform
      uses: hashicorp/setup-terraform@v1

    - name: Terraform Init
      run: terraform init

    - name: Terraform Plan
      run: terraform plan
      env:
        AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
        AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}

    - name: Terraform Apply
      run: terraform apply -auto-approve
      env:
        AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
        AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
```

CD Configuration ↘

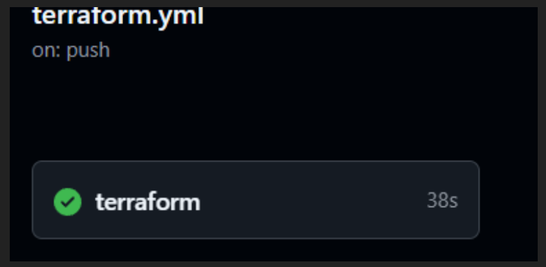
Pull Docker Image (konfirmasi) –> Upload SSH Key (Autentikasi ke EC2)

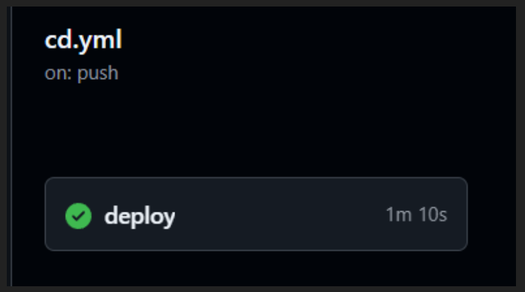
–> Deploy (Pull Image –> Remove Container Lama –> Run Container Baru)

```yaml
name: Continuous Deployment

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: ap-southeast-2

      - name: Pull Docker Image
        run: docker pull jasonnho/devops-pso:latest
```

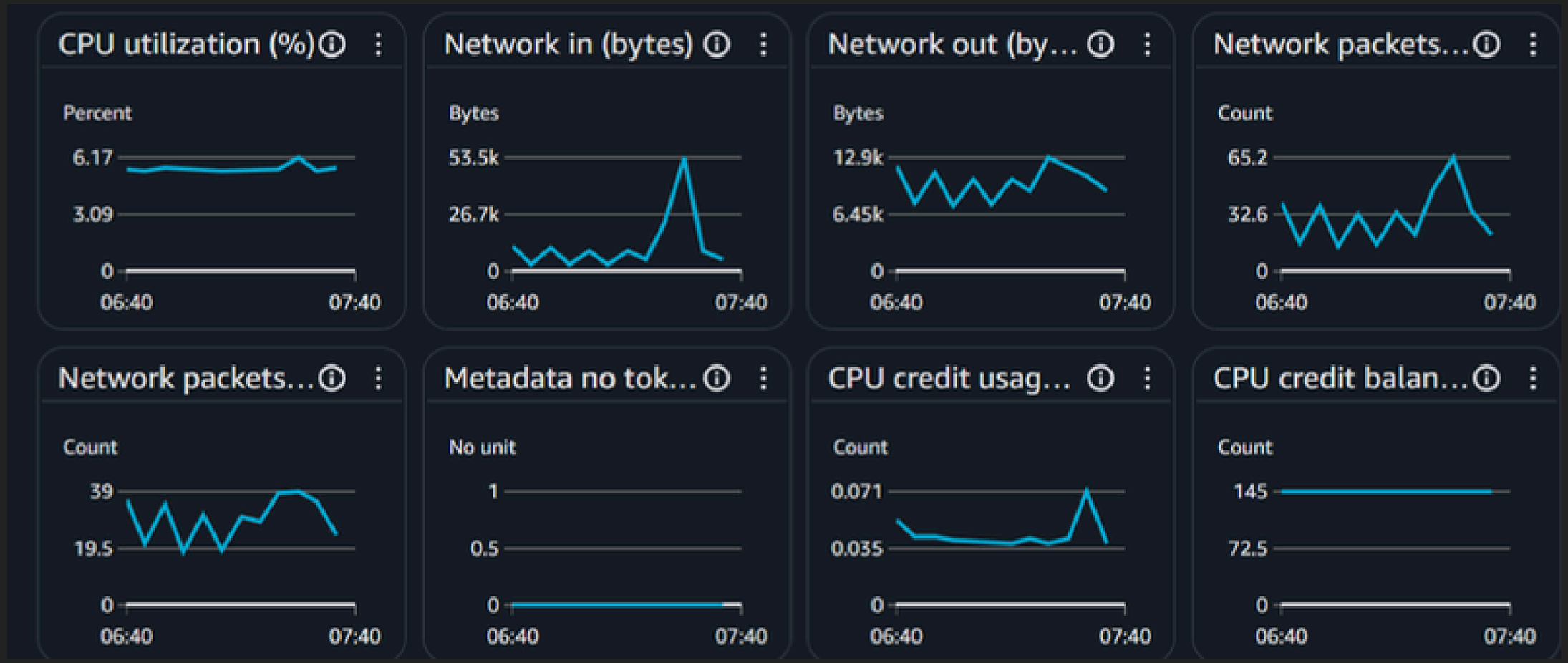
```yaml
- name: Upload SSH Key
  run: |
    echo "${{ secrets.SSH_PRIVATE_KEY }}" > key-pair-devopspso.pem
    chmod 400 "key-pair-devopspso.pem"
    chmod 600 "key-pair-devopspso.pem"

- name: Deploy to EC2
  run: |
    ssh -o StrictHostKeyChecking=no -i key-pair-devopspso.pem ubuntu@${{ secrets.EC2_INSTANCE_PUBLIC_DNS }} << 'EOF'
      docker pull jasonnho/devops-pso:latest
      docker rm -f devops-pso-container || true
      docker run -d -p 3000:3000 --name devops-pso-container jasonnho/devops-pso
    EOF
```

CI-Terraform-CD Pipeline Run ↘

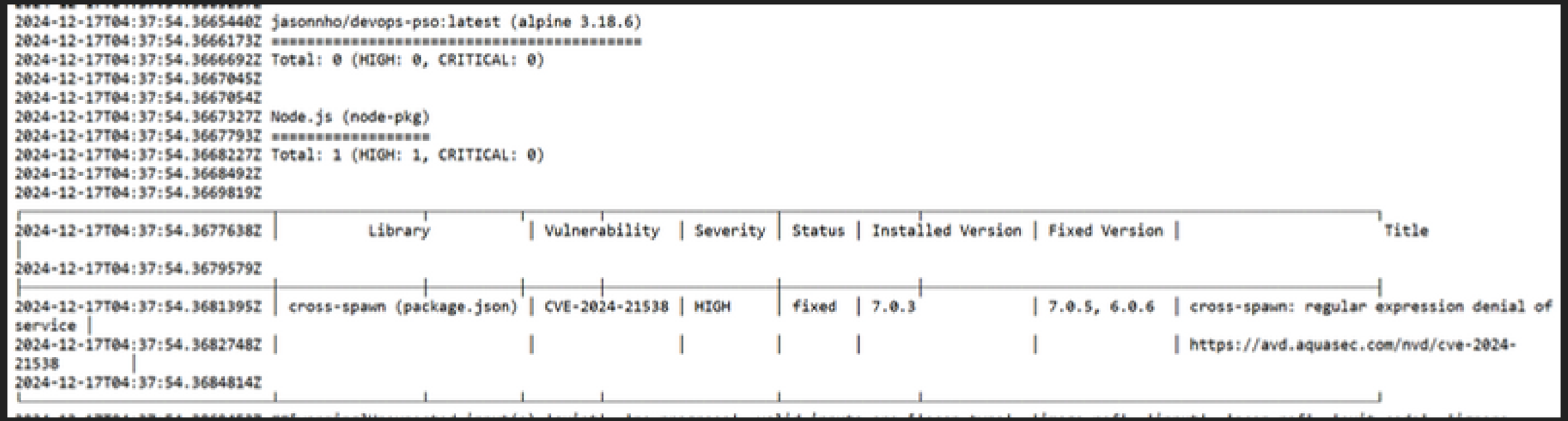




Monitoring

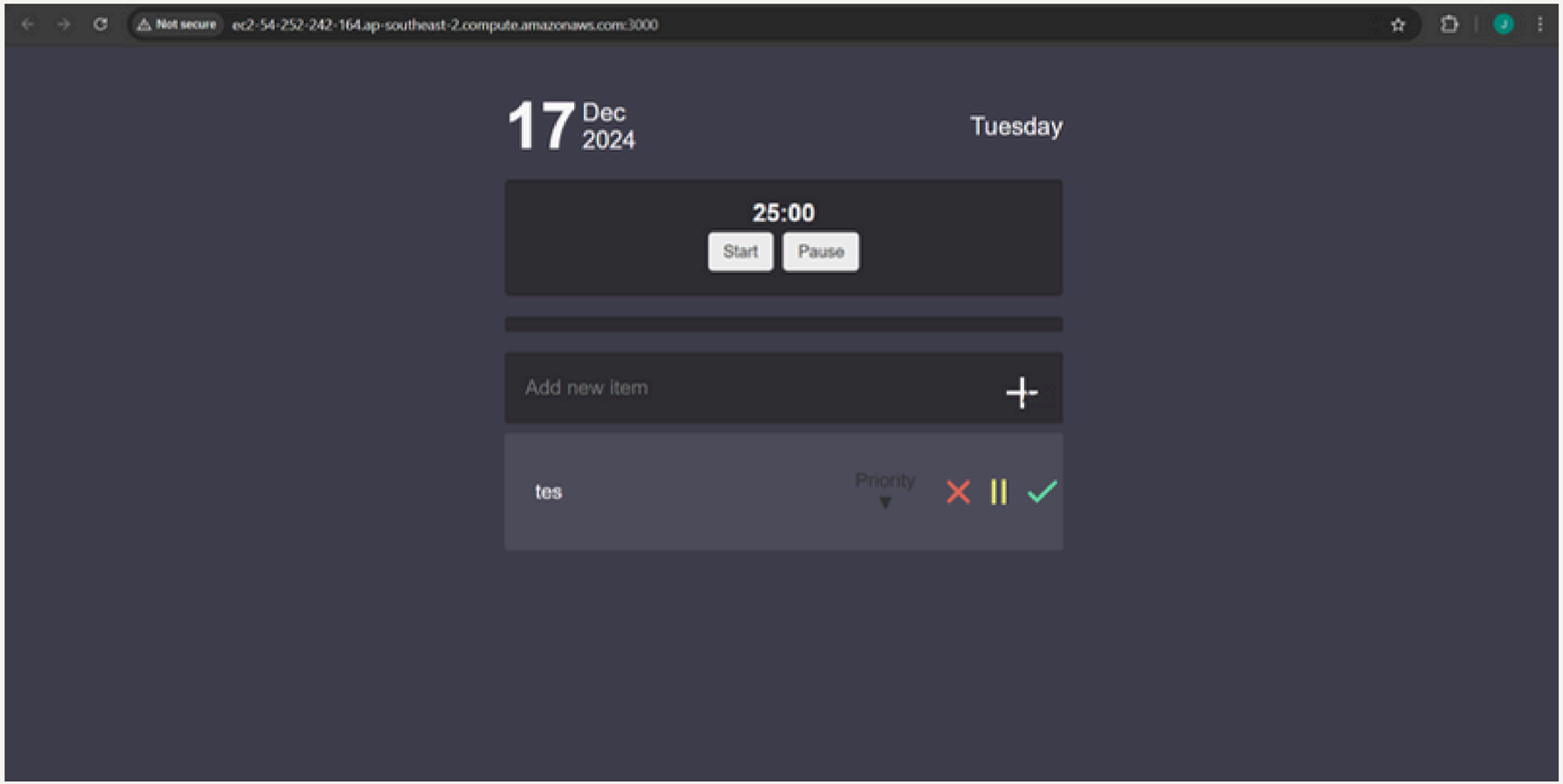

CPU utilization (%)
Percent
6.17
3.09
0
06:40
07:40
Network in (bytes)
Bytes
53.5k
26.7k
0
06:40
07:40
Network out (by…
Bytes
12.9k
6.45k
0
06:40
07:40
Network packets…
Count
65.2
32.6
0
06:40
07:40
Network packets…
Count
39
19.5
0
06:40
07:40
Metadata no tok…
No unit
1
0.5
0
06:40
07:40
CPU credit usag…
Count
0.071
0.035
0
06:40
07:40
CPU credit balan…
Count
145
72.5
0
06:40
07:40

Trivy Log



```
2024-12-17T04:37:54.3665440Z jasonnho/devops-pso:latest (alpine 3.18.6)
2024-12-17T04:37:54.3666173Z ========================================
2024-12-17T04:37:54.3666692Z Total: 0 (HIGH: 0, CRITICAL: 0)
2024-12-17T04:37:54.3667045Z
2024-12-17T04:37:54.3667054Z
2024-12-17T04:37:54.3667327Z Node.js (node-pkg)
2024-12-17T04:37:54.3667793Z ==================
2024-12-17T04:37:54.3668227Z Total: 1 (HIGH: 1, CRITICAL: 0)
2024-12-17T04:37:54.3668492Z
2024-12-17T04:37:54.3669819Z
┌──────────────────────┬──────────────────────┬─────────────────┬──────────┬────────┬───────────────────┬─────────────┬──────────────────────
2024-12-17T04:37:54.3677638Z │       Library        │ Vulnerability   │ Severity │ Status │ Installed Version │ Fixed Version │                      Title
│
2024-12-17T04:37:54.3679579Z
├──────────────────────┼──────────────────────┼─────────────────┼──────────┼────────┼───────────────────┼─────────────┼──────────────────────
2024-12-17T04:37:54.3681395Z │ cross-spawn (package.json) │ CVE-2024-21538 │ HIGH     │ fixed  │ 7.0.3             │ 7.0.5, 6.0.6  │ cross-spawn: regular expression denial of
service │
2024-12-17T04:37:54.3682748Z │                      │                 │          │        │                   │             │ https://avd.aquasec.com/nvd/cve-2024-
21538        │
2024-12-17T04:37:54.3684814Z
└──────────────────────┴──────────────────────┴─────────────────┴──────────┴────────┴───────────────────┴─────────────┴──────────────────────
```

Final Output ↘



http://ec2-54-252-242-164.ap-southeast-2.compute.amazonaws.com:3000/

https://its.id/m/TodometerKel12

Lesson Learned ↘

Memahami kompleksitas suatu project mulai dari mendefinisikan hingga tahapan planning. Project dimulai dengan mencari aplikasi yang ingin dikembangkan. Kemudian dilakukan build dan test yang relevan dengan fungsi aplikasi menggunakan Vitest dan React Testing Library. Kode juga diuji dengan beberapa tools seperti ESlint dan Trivy. Aplikasi kemudian dibuild dalam sebuah Image yang akan digunakan pada tahapan deploy. Konfigurasi infrastruktur yang akan digunakan dilakukan menggunakan Terraform. Kemudian deploy akan menggunakan image yang telah dibuild sebelumnya. Seluruh tahapan dijalankan melalui Github Actions dengan trigger pada masing-masing branch.

Challenges Encountered

Membutuhkan waktu untuk melakukan eksplorasi tools yang tersedia dan memilih tools tepat untuk diimplementasikan dan digunakan dalam project ini. Seperti misalnya penggunaan Docker dan pemilihan Cloud Services yang membutuhkan eksplorasi lebih dengan keterbatasan dalam Storage (Docker) dan credit gratis (cloud services) serta eksplorasi tools lainnya seperti Github Actions yang membutuhkan waktu. Selain itu, project yang digunakan juga harus disesuaikan mulai dari file package.json yang telah tersedia untuk menyesuaikan depedencies serta menghapus electron dari project existing. Selain itu, minimnya pengalaman dan pengetahuan terkait konfigurasi yang terbaik dan ideal juga menjadi tantangan untuk menyusun pipeline yang efektif.

Thank you