

**LAPORAN MAGANG INDUSTRI
SELF GRADING SYSTEM
WITH LEARNING ASSISTANT "PHILO"
DI PT BOER TECHNOLOGY**



Diajukan untuk memenuhi syarat mengikuti
Uji Kompetensi Keahlian Tahun Pelajaran 2022/2023

Disusun oleh:
Ahmad Bayhaqi Saputra (1019009048)
Kelas : XIII SIJA

**PEMERINTAH DAERAH PROVINSI JAWA BARAT
CABANG DINAS PENDIDIKAN WILAYAH 1
SMK NEGERI 1 CIBINONG**
BIDANG KEAHLIAN TEKNOLOGI DAN REKAYASA
PROGRAM KEAHLIAN TEKNIK KOMPUTER DAN INFORMATIKA
KOMPETENSI KEAHLIAN SISTEM INFORMATIKA, JARINGAN, DAN APLIKASI
2023

LEMBAR PENGESAHAN

PEMBIMBING PENULISAN LAPORAN

Laporan Magang Industri di PT Boer Technology ini telah diperiksa dan
disahkan pada:

Hari : Jum'at
Tanggal : 3 Maret 2023

Oleh:

Pembimbing Teknis



Dayu Destamy Ariefin, S.Pd.
NIP. 198712112014012001

Pembimbing Non Teknis



Arini Salsabila, S.Pd.
NIP/NUPTK. -

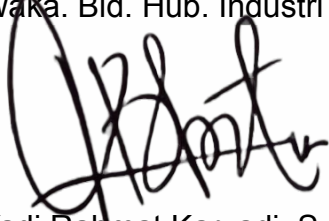
LEMBAR PENGESAHAN SEKOLAH

Laporan Magang Industri di PT Boer Technology ini telah diperiksa dan
disahkan pada:

Hari : Jum'at
Tanggal : 3 Maret 2023

Oleh:

Waka. Bid. Hub. Industri



Yadi Rahmat Karyadi, S.Pd.
NIP. 196705172000121001

Ka. Komp. Keahlian SIJA



Danu Winardi, ST.
NIP. 197111272014051001

Mengetahui,
Kepala SMKN 1 Cibinong

Sugiyo, S.Pd., M.Pd.
NIP. 196604192000031002

LEMBAR PENGESAHAN
DUNIA USAHA/DUNIA INDUSTRI (DU/DI)

Laporan Magang Industri di PT Boer Technology ini telah diperiksa dan
disahkan pada:

Hari : Jum'at
Tanggal : 3 Maret 2023

Oleh:

Pembimbing I

Pembimbing II

Sidarmawan
NIP. -

Muhammad Artur Afriansyah
NIP. -

LEMBAR PENGESAHAN PENGUJI LAPORAN

Laporan Magang Industri di PT Boer Technology ini telah diperiksa dan
disahkan pada:

Hari : Jum'at
Tanggal : 3 Maret 2023

Oleh:

Penguji

A handwritten signature in black ink, appearing to be 'Danu Winardi', written over the printed name.

Danu Winardi, ST.

NIP. 197111272014051001

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa yang telah menganugerahkan banyak nikmat sehingga kami dapat menyusun laporan proyek akhir ini dengan baik. Laporan ini berisi tentang penyelesaian masalah penilaian dengan menggunakan tools yang kami usung yakni “philo”, untuk menunjang kurikulum merdeka belajar.

Laporan ini kami susun secara cepat dengan bantuan dan dukungan berbagai pihak diantaranya; Ibu Dayu Destami, S.Pd selaku pembimbing teknis, Ibu Arini Salsabila, S.Pd selaku pembimbing non-teknisi serta teknologi dan teman-teman yang turut serta membantu saya. Oleh karena itu kami sampaikan terima kasih atas waktu, tenaga, pikiran dan ide yang telah diberikan.

Dalam penyusunan laporan ini, kami menyadari bahwa hasil laporan praktikum ini masih jauh dari kata sempurna. Sehingga kami selaku penyusun sangat mengharapkan kritik dan saran yang membangun dari pembaca sekalian. Akhir kata Semoga laporan praktikum ini dapat memberikan manfaat untuk kelompok kami khususnya, dan masyarakat Indonesia umumnya.

Bogor, Maret 2020

Penulis

DAFTAR ISI

Cover	i
Lembar Pengesahan Pembimbing Laporan	ii
Lembar Pengesahan Sekolah	iii
Lembar Pengesahan Dunia Usaha/Dunia Industri (DU/DI)	iv
Lembar Pengesahan Penguji	v
Kata Pengantar	vi
Daftar Isi	vii
Daftar Gambar	viii
Daftar Tabel	ix
Identitas Siswa	x
Bab I Pendahuluan	1
A. Latar Belakang Pemilihan Judul	1
B. Sejarah PT Boer Technology	2
C. Struktur Organisasi PT Boer Technology	2
D. Visi dan Misi PT Boer Technology	3
E. Disiplin Kerja PT Boer Technology	4
Bab II Proses dan Hasil Proyek Akhir	5
A. Landasan Teori	6
B. Sistem Desain	8
C. Implementasi	11
D. Panduan Pengguna	17
Bab III Penutup	20
A. Kesimpulan	20
B. Saran	20
Daftar Pustaka	22
Lampiran	23

DAFTAR GAMBAR

Gambar 1.1 Struktur Organisasi	2
Gambar 2.1 Rumus Penentuan <i>Score</i>	5
Gambar 2.2 Arsitektur <i>Self Grading System</i>	9
Gambar 2.3 Shopiea Log	15
Gambar 2.4 Cek Versi Philo	16
Gambar 2.5 Philo Panduan Pengguna	19
Gambar 2.6 Hasil Rekapitulasi Nilai	19

DAFTAR TABEL

Tabel 2.1 Spesifikasi minimal	11
-------------------------------	----

IDENTITAS SISWA



Nama Lengkap : Ahmad Bayhaqi Saputra
Tempat Lahir : Jakarta
Tanggal Lahir : 14 April 2004
Alamat Lengkap : KP. Kelapa RT 06 / 20 No. 32 A
No. Kontak : +62-851-5614-4384
E-mail : bayhaqisp04@gmail.com

BAB I

PENDAHULUAN

A. Latar Belakang Pemilihan Judul

Bagi seorang guru, permasalahan yang sering terjadi dalam kegiatan pembelajaran adalah melakukan penilaian secara manual. Guru dituntut untuk membaca dan menilai satu per satu tugas yang diterima dari setiap siswa. Proses ini membutuhkan waktu dan tenaga yang cukup banyak, terutama jika jumlah siswa yang harus dinilai sangat banyak. Berdasarkan penjelasan tersebut, dapat disimpulkan beberapa kendala guru dalam melakukan penilaian yaitu:

1. Waktu yang terbatas: Guru harus menyelesaikan proses penilaian tugas siswa dalam waktu yang sangat terbatas, sehingga membuat mereka terburu-buru dan kurang memperhatikan hasil pekerjaan siswa secara detail.
2. Kesalahan dalam penilaian: Terbatasnya waktu dalam melakukan penilaian dapat mengakibatkan terjadinya kesalahan dalam penilaian, misalnya melupakan beberapa tugas atau salah memberikan nilai pada tugas siswa.
3. Objektivitas penilaian. Guru kesulitan untuk melakukan penilaian secara adil berdasarkan kemampuan siswa.
4. Keterbatasan sumber daya: Guru mungkin tidak memiliki sumber daya yang cukup untuk menilai tugas siswa secara manual, seperti kurangnya waktu, staf, atau peralatan yang dibutuhkan.
5. Kelelahan: Proses menilai tugas siswa secara manual dapat menyebabkan kelelahan dan stres bagi guru, terutama jika jumlah tugas yang harus dinilai sangat banyak.

Untuk mengatasi masalah ini, beberapa solusi yang dapat dilakukan adalah dengan menggunakan teknologi untuk membantu proses penilaian, menyediakan sumber daya yang lebih banyak untuk membantu guru, dan memberikan pelatihan kepada guru untuk membantu mereka menilai tugas siswa secara efisien dan efektif.

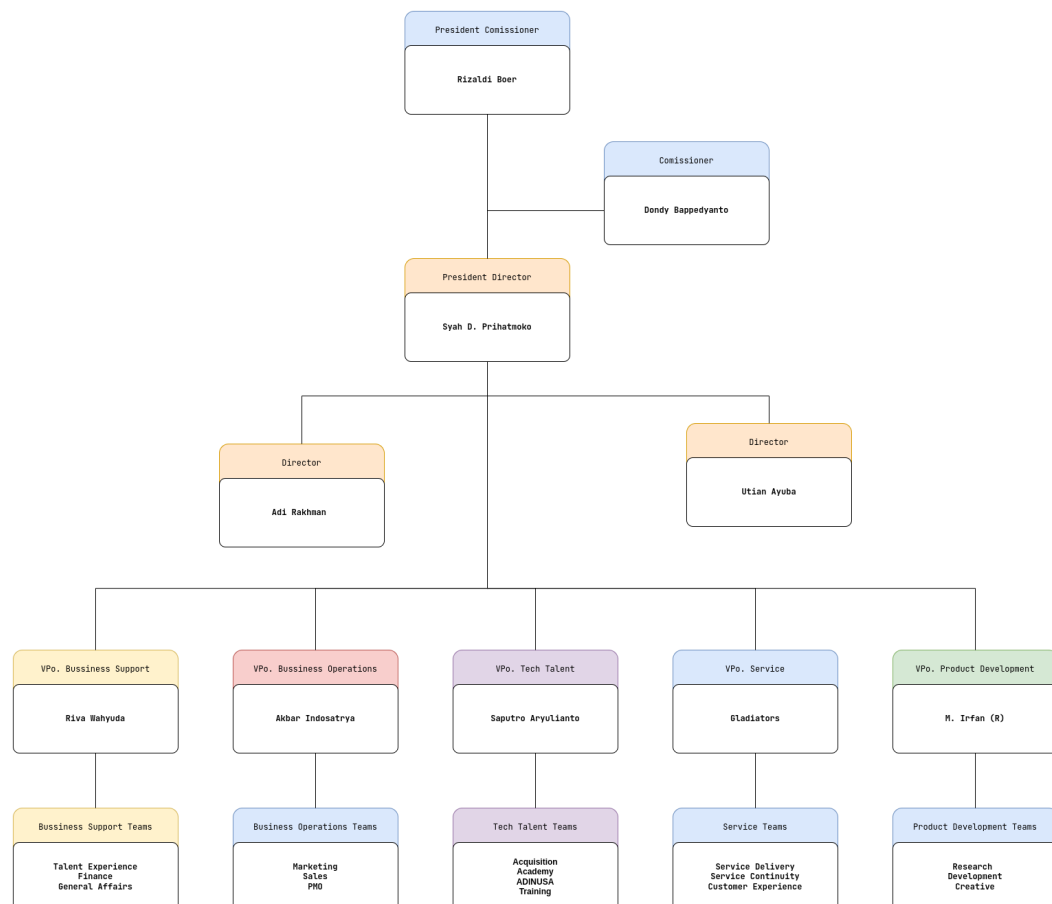
Self Grading System adalah solusi yang efektif untuk mengatasi masalah kesulitan guru dalam menilai tugas siswa secara manual. Sistem ini

memungkinkan siswa untuk mengerjakan tugas dan memeriksa hasilnya sendiri, sehingga mempermudah tugas guru dalam menilai tugas siswa. Oleh karena itu, penulis tertarik untuk memilih judul

B. Sejarah PT Boer Technology

Sejak tahun 2009, Boer Technology (Btech) telah berkomitmen kepada pelanggan dari berbagai industri di Indonesia untuk terus memberikan pelayanan dan solusi terbaik dengan menggunakan *Open Source Software* sebagai alat untuk mewujudkan perusahaan yang ramah lingkungan. Didukung oleh sumber daya manusia yang andal, Kami kini fokus ke Cloud, *DevOps*, *Security*, dan *Tech Professional Delivery* untuk mewujudkan perusahaan yang efisien dan aman serta membuat bisnis berjalan lebih cepat.

C. Struktur Organisasi PT Boer Technology



Gambar 1.1 Struktur Organisasi

D. Visi dan Misi PT Boer Technology

Dibangun dari Komunitas IT untuk ekosistem IT yang lebih luas, Btech memiliki hubungan dan interaksi yang solid dengan komunitas yang memiliki visi yang sama karena kami selalu percaya bahwa bersama-sama kita melakukan sesuatu dengan lebih baik.

1. *Together is better*

Leads to effective communication and collaboration. Komunikasi dan kolaborasi yang efektif sangat penting untuk tim atau organisasi yang sukses. Ketika individu dapat secara efektif mengkomunikasikan pemikiran, ide, dan keprihatinan mereka satu sama lain, mereka dapat bekerja sama dengan lebih efisien dan efektif. Ini mengarah pada pemahaman yang lebih baik tentang perspektif masing-masing, serta rasa kepercayaan dan kohesi yang lebih kuat dalam kelompok. Selain itu, ketika anggota tim dapat berkolaborasi secara efektif, mereka dapat memanfaatkan kekuatan dan keahlian satu sama lain, sehingga menghasilkan ide dan solusi yang lebih beragam. Pada akhirnya, komunikasi dan kolaborasi yang efektif mengarah pada pengambilan keputusan yang lebih baik, peningkatan produktivitas, dan lingkungan kerja yang lebih positif dan harmonis.

2. Continuous learning

Leads to increased knowledge, skills, and innovation. pembelajaran dan pengembangan berkelanjutan sangat penting untuk pertumbuhan pribadi dan profesional, dan hal itu dapat mengarah pada peningkatan pengetahuan, keterampilan, dan inovasi. Ketika individu mencari kesempatan belajar baru, mereka memiliki kesempatan untuk memperluas pemahaman mereka tentang dunia di sekitar mereka dan mengembangkan kompetensi baru yang dapat membantu mereka unggul dalam karir mereka. Ini juga dapat menumbuhkan budaya inovasi, karena individu yang terus belajar lebih mungkin menghasilkan solusi kreatif untuk masalah dan tantangan. Selain itu, ketika individu mendapatkan pengetahuan dan keterampilan baru, mereka dapat membagikannya dengan orang lain, yang mengarah ke efek riak pembelajaran dan pengembangan dalam suatu organisasi. Pada akhirnya, fokus pada pembelajaran dan pengembangan berkelanjutan

mengarah pada tenaga kerja yang lebih berpengetahuan, terampil, dan inovatif.

E. Disiplin Kerja PT Boer Technology

Continuous Learning adalah proses memperoleh pengetahuan, keterampilan, dan kemampuan baru sepanjang hidup seseorang. Ini adalah proses yang berkelanjutan dan mandiri, yang membantu individu untuk beradaptasi dengan keadaan yang berubah, meningkatkan kinerja mereka dan mencapai tujuan mereka. Pembelajaran berkelanjutan dapat terjadi melalui pendidikan formal, pelatihan kerja, atau belajar mandiri, dan dapat melibatkan berbagai mata pelajaran, mulai dari keterampilan teknis hingga pengembangan pribadi. Ini adalah komponen kunci dari pertumbuhan pribadi dan profesional, dan semakin dipandang penting di dunia yang serba cepat dan selalu berubah saat ini. Pembelajaran berkelanjutan membantu memastikan bahwa individu dapat mengikuti perkembangan baru dan tetap kompetitif di dunia kerja.

Pada PT Boer Technology tidak diwajibkan untuk melakukan presensi rutin, meski begitu presensi dilihat dari tiket *issue* yang sudah diselesaikan oleh setiap karyawan. Hal ini menjadikan lingkungan kerja yang lebih fleksibel namun tetap produktif.

BAB II

PROSES DAN HASIL PROJEK AKHIR

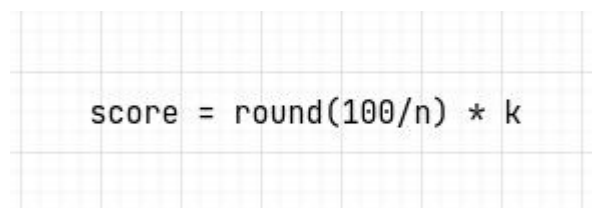
A. Landasan Teori

Philo adalah sebuah CLI (*Command-Line Interface*) aplikasi yang memungkinkan pengguna untuk melakukan *self-grading* atau penilaian secara otomatis pada tugas-tugas yang telah diberikan oleh guru. Aplikasi ini menggunakan bahasa pemrograman Go dan berjalan pada sistem operasi Linux, Windows, dan MacOS. Philo juga memanfaatkan teknologi seperti VirtualBox dan Vagrant untuk membuat *environment* yang terisolasi dan terstandarisasi bagi para siswa, serta GitHub untuk menyimpan tugas dan test case yang telah diberikan oleh guru. Selain itu, Philo juga memungkinkan siswa untuk melakukan scoring dan mengecek status tugas yang telah dikerjakan melalui sub command yang tersedia di dalam aplikasi ini.

1. Self Grading System

Self-grading system adalah sistem yang secara otomatis mengevaluasi tugas atau proyek siswa dan memberikan nilai berdasarkan kriteria yang ditentukan sebelumnya. Dalam kasus yang dijelaskan, *self-grading system* menggunakan case yang sudah dibuat dan di-hosting pada platform Github sebagai kriteria untuk mengevaluasi tugas siswa.

Sistem ini menggunakan *remote command* untuk menjalankan tugas siswa di *virtual machine* target dan kemudian membandingkan hasilnya dengan ekspektasi yang sudah diatur pada case.yaml. Setelah semua *test case* dijalankan, sistem akan menghitung jumlah test case yang berhasil dilalui dan menghitung nilai siswa berdasarkan rumus.


$$\text{score} = \text{round}(100/n) * k$$

Gambar 2.1 rumus penentuan score

Keterangan:

- n = jumlah total test case
- k = jumlah test case yang berhasil dijalankan dengan hasil sesuai expectation
- score = nilai akhir yang diperoleh, dalam rentang 0-100

Contoh: Jika terdapat 10 test case dan hanya 8 yang berhasil dijalankan dengan hasil sesuai ekspektasi, maka nilai akhir yang diperoleh adalah: $\text{score} = (100 / 10) \times 8 = 80$

Dengan adanya *self-grading system*, evaluasi tugas siswa dapat dilakukan dengan cepat dan akurat, tanpa perlu menunggu guru untuk mengevaluasi tugas satu per satu. Hal ini juga dapat membantu siswa untuk mengetahui performa mereka secara langsung dan memperbaiki kesalahan dengan cepat.

2. Infrastruktur sebagai kode

Infrastruktur sebagai kode (*Infrastructure as a Code*) adalah sebuah konsep dalam pengembangan perangkat lunak yang mengacu pada penggunaan kode untuk mengelola, menyediakan, dan mengotomatisasi infrastruktur pada suatu lingkungan. Dengan menggunakan pendekatan ini, infrastruktur diatur dan diimplementasikan dengan cara yang serupa seperti pengembangan perangkat lunak, dengan menggunakan versi kontrol dan proses otomatisasi yang sama.

Dalam konteks pengembangan perangkat lunak, infrastruktur dapat meliputi sistem operasi, aplikasi, basis data, jaringan, dan sebagainya. Dengan mengadopsi konsep ini, tim pengembang dapat mengelola infrastruktur sebagai bagian dari proses pengembangan perangkat lunak secara keseluruhan, dan memungkinkan untuk pengujian dan deployment yang lebih konsisten dan lebih mudah diulang.

Salah satu contoh implementasi infrastruktur sebagai kode adalah penggunaan alat seperti Vagrant atau Docker Compose untuk menyiapkan lingkungan pengembangan secara otomatis. Dengan menggunakan kode yang dapat dikelola, lingkungan pengembangan dapat didefinisikan dengan jelas dan diimplementasikan secara otomatis, sehingga meminimalkan kesalahan manusia dan menghindari

masalah yang muncul akibat perbedaan konfigurasi pada lingkungan yang berbeda.

3. *Learning Assistant*

Learning assistant adalah teknologi yang membantu dalam mendukung pembelajaran secara online atau jarak jauh dengan menyediakan lingkungan yang sesuai untuk belajar dan mendorong kemandirian siswa dalam belajar. Salah satu aspek dari *learning assistant* adalah menyediakan lingkungan belajar yang optimal untuk siswa.

Dalam konteks pengembangan perangkat lunak, *learning assistant* dapat membantu siswa dengan menyiapkan lingkungan laboratorium atau *preconfigured lab* yang diperlukan untuk belajar dan mengerjakan tugas. Namun, masalah yang sering terjadi adalah terjadinya *bottleneck* pada lingkungan laboratorium, di mana semua siswa menggunakan lingkungan yang sama dan dapat menyebabkan keterbatasan kapasitas dan kinerja.

Untuk mengatasi masalah ini, dapat digunakan Vagrant sebagai *infrastructure as a code* untuk membuat dan mengelola lingkungan laboratorium secara terdistribusi. Dalam konteks ini, Vagrant digunakan untuk membuat lingkungan laboratorium yang sama untuk setiap siswa, sehingga menghindari *bottleneck* pada lingkungan yang sama dan meningkatkan kinerja dan kecepatan dalam mengerjakan tugas.

Selain itu, Vagrant juga memungkinkan untuk membuat *environment lab* yang sesuai dengan spesifikasi yang diinginkan dengan mudah dan cepat, sehingga memudahkan siswa untuk belajar dan mengerjakan tugas tanpa menghabiskan waktu dan sumber daya yang berlebihan. Dengan demikian, *learning assistant* dengan bantuan Vagrant dapat membantu siswa untuk fokus pada pembelajaran dan tugas yang diberikan tanpa terganggu oleh masalah teknis yang mungkin terjadi pada lingkungan laboratorium.

B. Sistem Desain

Sistem Desain adalah proses mendefinisikan arsitektur, komponen, antarmuka, dan data untuk sistem perangkat lunak untuk memenuhi persyaratan yang ditentukan. Ini adalah langkah penting dalam proses

pengembangan perangkat lunak karena meletakkan dasar untuk keseluruhan proyek. Desain sistem adalah cetak biru terperinci yang menggambarkan fungsionalitas, kinerja, skalabilitas, dan keamanan sistem. Proses desain sistem melibatkan pemahaman domain masalah, mengidentifikasi persyaratan sistem, dan membuat desain tingkat tinggi yang menggambarkan keseluruhan struktur sistem. Desain tingkat tinggi ini kemudian disempurnakan menjadi desain detail, menentukan interaksi antara berbagai komponen dan antarmuka. Desain sistem harus fleksibel dan modular untuk memudahkan pemeliharaan dan skalabilitas di masa mendatang. Hasil akhir dari desain sistem harus berupa desain yang terdokumentasi dengan baik, jelas, dan mudah dipahami yang dapat digunakan sebagai panduan implementasi dan pengujian sistem.

1. MVP

Minimum Viable Product (MVP) adalah versi minimal dari produk yang memiliki fitur utama dan dapat digunakan oleh pengguna untuk melakukan tes pasar. Dalam hal ini, MVP Philo adalah solusi untuk membantu guru menilai tugas siswa secara efisien dan efektif.

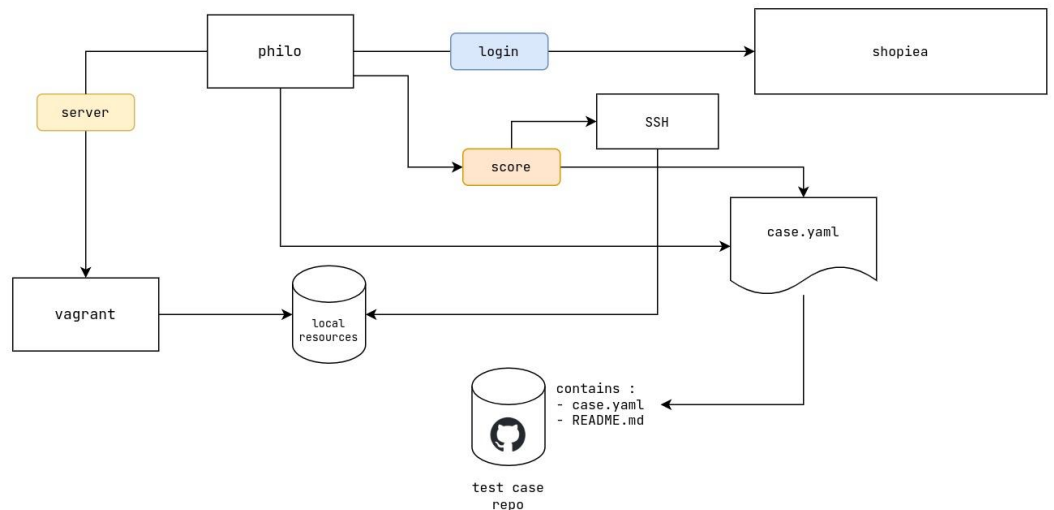
Menggunakan GitHub sebagai platform utama dan *dashboard* guru memungkinkan MVP Philo untuk memiliki beberapa fitur penting sebagai berikut:

1. Penyimpanan Test Case: Guru dapat menambahkan dan menyimpan test case pada platform GitHub, sehingga mudah diakses dan dapat berkolaborasi.
2. Dashboard Guru: Guru dapat menggunakan dashboard pada *subcommand* philo admin untuk melihat serta mengeksport nilai siswa
3. Integrasi dengan Repository Siswa: MVP Philo dapat terintegrasi dengan repository siswa, sehingga tugas siswa dapat dinilai secara otomatis dan hasil penilaian dapat dilihat oleh guru dan siswa.
4. Laporan Hasil Penilaian: MVP Philo dapat menghasilkan laporan hasil penilaian secara otomatis, sehingga mempermudah tugas guru dan memastikan bahwa hasil penilaian benar dan akurat.
5. Kemudahan Berkolaborasi: Guru dan siswa dapat berkolaborasi dengan mudah melalui platform GitHub, sehingga mempermudah proses pembelajaran.

Dengan demikian, MVP Philo menggunakan GitHub sebagai platform utama dan dashboard guru memungkinkan guru dan siswa untuk melakukan tugas dan menilai hasil pekerjaan secara efisien dan efektif, sehingga mempermudah proses pembelajaran dan mengurangi beban kerja guru.

a. Arsitektur

Arsitektur diatur sedemikian rupa dengan memperhatikan sumber daya serta penjenjangan skalabilitas serta universalitas pengguna. proyek ini menggunakan 3 komponen utama yakni philo sebagai *client*, github (atau *public repository provider* lainnya) sebagai media penyimpanan *test case* yang di buat, serta shopiea sebagai *scoring server*.



Gambar 2.2 Arsitektur self grading system

Arsitektur yang mengintegrasikan antara Philo sebagai CLI App client dan Shopiea sebagai server scoring memungkinkan untuk menampung nilai siswa dalam sistem *centralize*. Arsitektur ini terdiri dari beberapa komponen, yaitu:

1. Philo CLI App: Komponen utama yang digunakan oleh siswa untuk menilai tugas mereka dan mengirimkan hasil nilai ke *server scoring*. Philo CLI App juga memiliki sub command philo login yang digunakan untuk login ke server scoring.
2. Shopiea Server: Komponen server yang menyimpan data nilai siswa dalam basis data dan menyediakan REST API endpoint untuk mengambil data nilai siswa. Shopiea Server akan

menerima data nilai siswa dari Philo CLI App melalui fitur sub command philo login.

3. Database: Komponen yang digunakan untuk menyimpan data nilai siswa. Shopiea Server akan menulis data nilai siswa ke dalam basis data.

Dalam arsitektur ini, setiap kali seorang siswa menyelesaikan tugas dan menilai tugas tersebut menggunakan Philo CLI App, hasil nilai siswa akan dikirimkan ke Shopiea Server melalui sub command philo login. Shopiea Server akan menulis data nilai siswa ke dalam basis data. Kemudian, guru atau pengajar dapat melihat data nilai siswa dari dashboard yang terpusat.

Dengan arsitektur ini, nilai siswa dapat diintegrasikan secara otomatis ke dalam sistem yang terpusat dan mudah dikelola. Selain itu, arsitektur ini juga memungkinkan untuk dilakukan analisis data yang lebih akurat dan efektif, sehingga dapat membantu meningkatkan kualitas pendidikan secara keseluruhan.

b. Fitur

Philo adalah sebuah self grading system yang membantu guru dalam menilai tugas siswa secara otomatis. Aplikasi ini memiliki beberapa fitur minimal yang memungkinkan untuk menunjang fungsionalitas sebagai *self grading system*. Berikut adalah penjelasan fitur pada philo:

1. *self grading with bash scripting executor*, fitur ini memungkinkan eksekusi dan memberikan efek fleksibilitas kepada guru karena dapat menyesuaikan dengan tugas yang diberikan serta kustomisasi yang luas seperti *bash scripting* pada umumnya
2. *self provisioning with vagrant*, fitur ini memanfaatkan *backend Infrastructure as a Code Vagrant* untuk manage *virtual machine* yang ada pada Virtualbox. Virtualbox dipilih lantaran mudah digunakan serta *free software* sehingga tidak perlu memiliki *license* untuk menggunakannya.
3. *SSH to Virtual Machine*, selain dapat *manage virtual machine* fitur lain adalah dapat mengakses langsung ke *virtual machine* sehingga memudahkan kita dalam mengerjakan lab tanpa harus ke *dashboard* virtualbox secara langsung.

4. *Integrated with shopiea scoring server*, untuk memudahkan guru melakukan rekapitulasi serta memberikan fitur *saving score* pada peserta sehingga dapat melihat score yang sebelumnya sudah dikerjakan.
5. Subcommand `admin`, subcommand ini didapatkan dengan menambahkan `-tags admin` pada saat build memudahkan guru dalam melakukan pendaftaran lab, membuat kelas baru, kursus baru serta mengatur users.

C. Implementasi

Dalam pengimplementasiannya philo memerlukan beberapa persyaratan minimal untuk mendukung berjalannya semua fungsi dengan baik. *requirements*-nya antara lain Shopiea, Docker, Vagrant, Go, dan Virtualbox. Docker digunakan untuk menjalankan Shopiea sebagai *container*, Golang digunakan untuk membuat Philo sebagai CLI App client, Vagrant digunakan untuk mengelola environment lab, dan VirtualBox digunakan sebagai hypervisor untuk menjalankan VM. Dengan menggunakan persyaratan ini, self grading system dapat berfungsi dengan semestinya. Oleh karena itu berikut adalah panduan dalam pengimplementasian self grading system serta cara penggunaannya, sebelum itu pastikan kita memiliki Virtual Machine dengan spesifikasi berikut.

Tabel 2.1 spesifikasi minimal

Operating System	Ubuntu 18.04 or higher
RAM	2048 MB
VCPUs	2 core
Disk	20 GB

1. Installing Docker

Docker adalah sekumpulan platform sebagai produk layanan yang menggunakan virtualisasi tingkat OS untuk mengirimkan perangkat lunak dalam paket yang disebut kontainer. Layanan ini memiliki tingkatan gratis dan premium. Perangkat lunak yang menampung kontainer disebut Docker Engine. Berikut adalah langkah-langkah untuk menginstall Docker pada Ubuntu 22.04:

Pastikan sistem Ubuntu 22.04 Anda telah diperbarui dengan perintah berikut:

```
sudo apt update  
sudo apt upgrade
```

Install paket-paket yang dibutuhkan untuk menginstall Docker dengan perintah berikut:

```
sudo apt install apt-transport-https ca-certificates curl  
gnupg-agent software-properties-common
```

Tambahkan kunci GPG resmi Docker ke sistem Anda dengan perintah berikut:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg |  
sudo apt-key add -
```

Tambahkan repository Docker ke daftar repositori APT dengan perintah berikut:

```
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release  
-cs) stable"
```

Update daftar paket dan install Docker dengan perintah berikut:

```
sudo apt update  
sudo apt install docker-ce docker-ce-cli containerd.io
```

Verifikasi bahwa Docker telah terinstall dengan menjalankan perintah berikut:

```
sudo docker run hello-world
```

Jika output menunjukkan pesan "Hello from Docker!", maka Docker telah terinstall dengan benar. Setelah Docker terinstall, pastikan bahwa user Anda terdaftar dalam grup docker dengan perintah berikut:

```
sudo usermod -aG docker $USER
```

Kemudian *logout* dan *login* kembali untuk menerapkan perubahan. untuk versi docker yang terinstall dapat dilihat menggunakan command berikut:

```
docker version
```

Sekarang Anda telah berhasil menginstall Docker pada Ubuntu 22.04 dan siap digunakan.

2. Install shopiea

Shopiea adalah RESTful API yang ditulis dalam Go yang menyediakan fungsionalitas sistem penilaian untuk mengelola dan menilai tugas siswa, otentikasi dan pendaftaran lab dan kursus, serta membuat laporan.

Untuk melakukan instalasi shopiea diperlukan tools git untuk *cloning repository* yang ada pada github <https://github.com/Kyuubang/shopiea>. Git dapat diinstal melalui tautan berikut <https://git-scm.com/download/linux>. Jika sudah terinstall dengan baik lalu eksekusi command berikut ini.

```
git clone https://github.com/Kyuubang/shopiea.git
```

Lalu lakukan perubahan pada file docker-compose sesuai dengan credentials yang ingin di buat

```
version: '3.8'

services:
  db:
    image: postgres:alpine3.17
    restart: always
    environment:
      POSTGRES_USER: shopiea
      POSTGRES_PASSWORD: mys3cretP4ssw0rd
      POSTGRES_DB: shopiea
    volumes:
      - postgres-data:/var/lib/postgresql/data
  networks:
    shopiea-network:
```

lakukan ada service shopiea sesuai dengan credential db yang sebelumnya sudah didefinisikan.

```
shopiea:
  image: shopiea-api:latest
  restart: always
  environment:
    SHOPIEA_DB_HOST: db
    SHOPIEA_DB_PORT: 5432
    SHOPIEA_DB_USER: shopiea
    SHOPIEA_DB_PASSWORD: mys3cretP4ssw0rd
    SHOPIEA_DB_NAME: shopiea
    SHOPIEA_CASE_REPO: github.com/Kyuubang/philos-sample-case
    SHOPIEA_CASE_BRANCH: master
    SHOPIEA_INFRA_REPO:
github.com/Kyuubang/philos-sample-infra
    SHOPIEA_INFRA_BRANCH: master
  ports:
    - "9898:9898"
  depends_on:
    - db
  networks:
    shopiea-network:
```

setelah selesai dan di simpan selanjut nya bisa dijalankan docker compose dengan command berikut

```
docker compose up
```

jika tidak terdapat error dan tampilan output seperti berikut, maka kita telah berhasil menginstall shopiea

```

[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:   export GIN_MODE=release
- using code:  gin.SetMode(gin.ReleaseMode)

[GIN-debug] POST    /auth/login          → github.com/Kyuubang/shopiea/handlers.Login (3 handlers)
[GIN-debug] GET     /info                → main.main.func1 (3 handlers)
[GIN-debug] GET     /v1/course           → github.com/Kyuubang/shopiea/handlers.GetCourses (4 handlers)
[GIN-debug] GET     /v1/labs             → github.com/Kyuubang/shopiea/handlers.GetLabs (4 handlers)
[GIN-debug] POST    /v1/score            → github.com/Kyuubang/shopiea/handlers.PushScore (4 handlers)
[GIN-debug] GET     /v1/score            → github.com/Kyuubang/shopiea/handlers.GetScore (4 handlers)
[GIN-debug] POST    /v1/auth/check       → github.com/Kyuubang/shopiea/handlers.CheckToken (4 handlers)
[GIN-debug] POST    /v1/admin/check      → github.com/Kyuubang/shopiea/handlers.CheckToken (5 handlers)
[GIN-debug] GET     /v1/admin/user       → github.com/Kyuubang/shopiea/handlers.GetUsers (5 handlers)
[GIN-debug] POST    /v1/admin/user       → github.com/Kyuubang/shopiea/handlers.CreateUser (5 handlers)
[GIN-debug] DELETE  /v1/admin/user       → github.com/Kyuubang/shopiea/handlers.DeleteUser (5 handlers)
[GIN-debug] PUT     /v1/admin/user       → github.com/Kyuubang/shopiea/handlers.UpdateUser (5 handlers)
[GIN-debug] GET     /v1/admin/class      → github.com/Kyuubang/shopiea/handlers.GetClasses (5 handlers)
[GIN-debug] POST    /v1/admin/class      → github.com/Kyuubang/shopiea/handlers.CreateClass (5 handlers)
[GIN-debug] DELETE  /v1/admin/class      → github.com/Kyuubang/shopiea/handlers.DeleteClass (5 handlers)
[GIN-debug] PUT     /v1/admin/class      → github.com/Kyuubang/shopiea/handlers.UpdateClass (5 handlers)
[GIN-debug] POST    /v1/admin/course     → github.com/Kyuubang/shopiea/handlers.CreateCourse (5 handlers)
[GIN-debug] PUT     /v1/admin/course     → github.com/Kyuubang/shopiea/handlers.UpdateCourse (5 handlers)
[GIN-debug] DELETE  /v1/admin/course     → github.com/Kyuubang/shopiea/handlers.DeleteCourse (5 handlers)
[GIN-debug] POST    /v1/admin/labs       → github.com/Kyuubang/shopiea/handlers.CreateLabs (5 handlers)
[GIN-debug] PUT     /v1/admin/labs       → github.com/Kyuubang/shopiea/handlers.UpdateLabs (5 handlers)
[GIN-debug] DELETE  /v1/admin/labs       → github.com/Kyuubang/shopiea/handlers.DeleteLabs (5 handlers)
[GIN-debug] GET     /v1/admin/export     → github.com/Kyuubang/shopiea/handlers.ExportScore (5 handlers)

```

Gambar 2.3 shopiea log

3. menginstall fungsional server

untuk mendapatkan fungsional server dengan Infrastructure as a code, maka peserta diwajibkan untuk menginstall vagrant. Vagrant adalah produk perangkat lunak sumber terbuka untuk membangun dan memelihara lingkungan pengembangan perangkat lunak virtual *portable*; misalnya, untuk VirtualBox, KVM, Hyper-V, *container* Docker, VMware, dan AWS. Ini mencoba menyederhanakan manajemen konfigurasi perangkat lunak virtualisasi untuk meningkatkan produktivitas pengembangan. untuk dapat melakukan instalasi vagrant dapat dilakukan dengan cara berikut.

```

wget -O- https://apt.releases.hashicorp.com/gpg | gpg
--dearmor | sudo tee
/usr/share/keyrings/hashicorp-archive-keyring.gpg

echo "deb
[signed-by=/usr/share/keyrings/hashicorp-archive-keyring.
gpg] https://apt.releases.hashicorp.com $(lsb_release
-cs) main" | sudo tee
/etc/apt/sources.list.d/hashicorp.list

sudo apt update && sudo apt install vagrant

```

jika menggunakan Sistem Operasi lain dapat disesuaikan proses instalasi pada tautan berikut <https://developer.hashicorp.com/vagrant/downloads>.

4. install philo

philo ditulis menggunakan Go programming language yang dapat menghasilkan binari terkompilasi, sehingga kita cukup build binari dengan target arsitektur yang kita miliki. untuk mendapat pre-build binari dapat diakses melalui tautan berikut <https://github.com/Kyuubang/philo/releases>. penamaan binari dengan mengikuti format berikut

```
philo-<os>-<arch>
```

jika sudah berhasil terunduh bisa dapat dilakukan konfigurasi pada *variable path* dengan command berikut

```
chmod +x philo*
sudo cp philo* /usr/local/bin/philo
```

lalu test dengan menjalankan command `philo -version` / `philo.exe -version` pada terminal

```
➔ philo git:(master) ✕ philo --version
philo version dev
```

Gambar 2.4 cek versi philo

D. Panduan Pengguna

Philo adalah sebuah self grading system yang membantu guru dalam menilai tugas siswa secara otomatis. Aplikasi ini memiliki berbagai sub command yang memudahkan pengguna untuk mengelola sumber daya lokal dan menjalankan lab *environment*, serta mengecek dan melihat histori score siswa. Berikut adalah penjelasan fitur pada sub command philo server:

1. philo server create: Fitur ini memungkinkan pengguna untuk membuat sumber daya baru menggunakan Vagrant sebagai Infrastructure as Code. Pengguna dapat menentukan nama, tipe, dan konfigurasi lainnya untuk sumber daya yang dibuat.
2. philo server up: Fitur ini memungkinkan pengguna untuk menghidupkan sumber daya yang telah dibuat sebelumnya menggunakan Vagrant.

Dalam mode ini, aplikasi akan mem provision environment untuk menjalankan lab.

3. philo server status: Fitur ini memungkinkan pengguna untuk melihat status dari sumber daya yang telah dibuat dan diatur menggunakan Vagrant. Pengguna dapat melihat apakah sumber daya tersebut aktif atau tidak, dan informasi lainnya seperti IP address.
4. philo server halt: Fitur ini memungkinkan pengguna untuk mematikan sumber daya yang telah dihidupkan menggunakan Vagrant. Dalam mode ini, sumber daya akan dimatikan.
5. philo server destroy: Fitur ini memungkinkan pengguna untuk menghapus sumber daya yang telah dibuat menggunakan Vagrant. Pengguna harus memastikan bahwa sumber daya tidak lagi diperlukan sebelum menghapusnya, karena proses ini tidak dapat di-undo.
6. philo score check: Fitur ini memungkinkan pengguna untuk memeriksa pekerjaan yang telah dikerjakan oleh siswa dan memberikan nilai berdasarkan kriteria yang telah ditentukan. Dalam mode ini, aplikasi akan menampilkan hasil kerja siswa yang telah dinilai, dan pengguna dapat memberikan nilai berdasarkan kriteria yang telah ditentukan. Setelah pengguna memberikan nilai, sistem akan menyimpan nilai tersebut ke dalam basis data dan mengirimkan notifikasi ke siswa tentang hasil penilaian mereka.
7. philo score view: Fitur ini memungkinkan pengguna untuk melihat histori nilai yang telah diberikan kepada siswa. Dalam mode ini, aplikasi akan menampilkan histori nilai yang telah diberikan kepada siswa, termasuk nama siswa, tugas yang dinilai, nilai yang diberikan, dan waktu penilaian dilakukan.
8. philo lab start: Fitur ini memungkinkan pengguna untuk menjalankan preconfigured lab yang telah disiapkan sebelumnya. Dalam mode ini, aplikasi akan memeriksa dan menjalankan environment yang telah dikonfigurasi sebelumnya, termasuk *software*, *file* konfigurasi, dan sumber daya yang diperlukan.
9. philo lab clean: Fitur ini memungkinkan pengguna untuk membersihkan environment lab yang telah digunakan. Dalam mode ini, aplikasi akan menghapus semua *file*, *software*, dan sumber daya yang digunakan selama menjalankan environment lab. Ini memastikan bahwa tidak ada

siswa-siswa yang tersisa di dalam sistem dan mengurangi penggunaan sumber daya yang tidak perlu.

10. philo login: Proses login dilakukan dengan menggunakan credentials yang telah diberikan oleh administrator atau pengelola server. Dalam mode ini, pengguna akan diminta untuk memasukkan username dan password yang telah disediakan. Setelah proses login berhasil, pengguna akan mendapatkan akses ke server scoring dan dapat mulai melakukan proses penilaian dan penyimpanan score.
11. philo update: Pada aplikasi self grading system digunakan untuk melakukan proses *self-update* binari ke versi terbaru. Fitur ini memungkinkan pengguna untuk selalu menggunakan versi terbaru dari aplikasi self grading system tanpa harus mengunduh versi terbaru secara manual dari website atau repository.
12. philo admin: subcommand ini memungkinkan admin melakukan administrasi sumber daya seperti membuat, menghapus, mengubah, dan menampilkan sumber daya yang ada pada basis data. sumber daya yang tersedia antara lain, *users*, *class*, *labs*, dan *courses*

philo juga memberikan *manual page* yang mana memudahkan pengguna dalam mengetahui lebih dalam setiap fungsi *subcommand* serta contoh *command* yang harus dijalankan. dengan hanya menjalankan command philo tanpa argument apapun philo akan memberikan manual singkat penggunaan.

```

→ philo git:(master) × philo
Philo is a test case executor for your labs. It is designed to be used with
the shopeia server, which is used to manage your labs and their test cases.

Usage:
  philo [command]

Available Commands:
  completion  Generate the autocompletion script for the specified shell
  config      config is a command to manage config file
  course      course is a command to manage course
  help        Help about any command
  init        initialize your philo environment
  lab         view lab and list lab
  login       login to server
  score       manage score of lab
  server      manage your local server

Flags:
  -h, --help      help for philo
  -v, --version    version for philo

Use "philo [command] --help" for more information about a command.

```

Gambar 2.5 philo panduan pengguna

Philo juga membantu guru dalam melakukan rekapitulasi nilai dengan menggunakan *command* `philo admin export` dengan *parameter class id* dan *course id*. Philo akan mengkonversi response dari API menjadi file CSV.

	A	B	C	D	E	F	G	
1	Name	Username	docker-001-1	docker-001-2	docker-001-3	Total	Average	
2	Ahmad Bayhagi	Kyuubang	0	0	0	0	0	
3	Muhamad Jamil	fanreza	100	100	100	300	100	
4	Hermawan S	hermawans	80	0	80	160	53.33	
5								
6								

Gambar 2.6 hasil rekapitulasi nilai

BAB III

PENUTUP

A. Kesimpulan

Dengan demikian philo sebagai projek ini yang berfungsi self grading system terbukti efektif dalam membantu guru dan siswa untuk memberikan pengalaman belajar secara mandiri lebih berbeda dibanding sebelumnya. Ditambah pemberian *feedback* secara *realtime* menjadikan siswa lebih cepat dalam belajar dan melakukan improvisasi dalam mengerjakan tugas.

B. Saran

Dalam projek *self grading system* untuk tugas siswa, masalah utama adalah belum adanya platform tersentralisasi untuk menampung nilai siswa dalam satu dashboard. Ini dapat menyebabkan beberapa masalah seperti:

1. Tidak adanya *dashboard UI*: tidak adanya dashboard ui menjadikan kesulitan tersendiri bagi guru untuk melihat nilai siswa dalam visualisasi yang baik.
2. Sistem yang kemungkinan masih terdapat *vulnerability*: dikarenakan *self grading system* memberikan akses penuh kepada siswa untuk menilai secara mandiri pekerjaan yang telah dilakukan masih terdapat kemungkinan untuk siswa melakukan hal curang.

Untuk mengatasi masalah ini, perlu dibuat platform tersentralisasi untuk menampung nilai siswa dalam satu dashboard yang mudah diakses dan dikelola oleh guru. Platform tersebut dapat menampilkan nilai siswa dalam format yang mudah dibaca dan dianalisis, sehingga mempermudah bagi guru untuk menilai dan menentukan tindakan selanjutnya.

Solusi pengembangan lebih lanjut untuk pembuatan UI dashboard dapat menggunakan REST API yang sudah disediakan oleh proyek Shopiea. Dengan REST API ini, dapat dikembangkan aplikasi web yang dapat mengakses dan menampilkan informasi nilai siswa secara real-time.

Selain itu penerapan kriptografi untuk memastikan setiap nilai hanya di-*push* melalui agen yang terpercaya. setiap data push yang berkaitan dengan nilai serta generate token server untuk memastikan *binari* teregistrasi.

Menggunakan Linux sebagai sistem operasi *primary* dalam *project self grading system* menawarkan beberapa keuntungan seperti:

1. *Stabilitas*: Linux memiliki reputasi yang sangat baik sebagai sistem operasi yang stabil dan handal, yang memastikan bahwa aplikasi yang dijalankan di atasnya bekerja dengan baik dan tanpa masalah.
2. *Developer Support*: Secara default philo dikembangkan khusus dan dari Linux, ini akan mempermudah developer dalam memberikan bantuan teknis terhadap issue yang ada.
3. *Pemengembangan*: Selain itu siswa juga dapat berkontribusi langsung terhadap pengembangan dari sisi kode sumber maupun test case yang lebih variatif.

Meskipun demikian, Philo adalah aplikasi *open-source* yang didesain untuk mendukung *cross-platform*, sehingga siswa dapat menggunakan aplikasi *self grading system* ini pada sistem operasi apapun. Ini memastikan bahwa tugas siswa dapat dinilai dengan efisien dan akurat, tanpa terganggu oleh masalah kompatibilitas sistem operasi.

DAFTAR PUSTAKA

Massé Mark. 2012. "REST API Design Rulebook". United States of America: O'Reilly Media, Inc.

Vagrant 2023. "Install Vagrant on Linux". <https://developer.hashicorp.com/vagrant/downloads>. diakses pada 2 Maret 2023

Wikipedia. 2023. "Docker (Software)", [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)), diakses pada 2 Maret 2023.

Yellavula Naren. 2017. "Building RESTful Web Services with Go". Livery Place, UK: Packt Publishing.