

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kegiatan Magang & Studi Independen Bersertifikat Everything Understand About Vue JS diadakan oleh Alterra Academy (PT. Marka Kreasi Persada). Alterra Academy (ALTA) *is a tech talent accelerator who helps potential tech talents to be the best version of themselves and help organization findin.*

MSIB Alterra Academy diajukan dengan harapan mencetak kualitas lulusan perguruan tinggi terbaik di bidang *tech* yang akan siap menghadapi dunia kerja dalam bidang teknologi. Pembelajaran di MSIB Alterra Academy dalam bentuk pembelajaran *synchronous* dan *asynchronous*. *Synchronous* yaitu peserta mengikuti *Live Session Expert Class* dimana peserta dapat berkonsultasi dengan *expert/mentor* secara langsung terkait materi yang dipelajari sebelumnya melalui video. *Asynchronous* yaitu peserta akan diwajibkan untuk melakukan *preparation learning* berupa menyimak video *recording* materi kemudian akan mendapatkan 3 tantangan beberapa *quiz*, membuat ringkasan serta mendapatkan tugas harian berupa praktek, dimana tantangan tersebut untuk melatih peserta dalam penguasaan materi.

1.2 Tujuan

1.2.1 Tujuan Pelaksanaan PKL/KP

- a. Melakukan magang sesuai ketentuan Institut Teknologi Telkom Purwokerto.
- b. Untuk memenuhi mata kuliah pada program studi Teknik Informatika sebagai salah satu syarat kelulusan dalam melaksanakan magang sebagai syarat kelulusan mata kuliah Kerja Praktek.
- c. Meningkatkan pengetahuan secara teori dan praktik tentang bahasa pemrograman dari dasar hingga tahap yang lebih mendalam.
- d. Menambah pengalaman secara praktik dan logika dalam menyelesaikan studi kasus.
- e. Melatih kemampuan diri baik intelektual maupun moral sebagai persiapan masuk ke dalam dunia kerja.
- f. Menggali ilmu dan pengetahuan mengenai coding dari para mentor yang sudah berpengalaman.

1.2.2 Tujuan Program

- a. Peserta menguasai dasar pemograman *Front-End* hingga *advace topic* untuk menggunakan bahasa pemograman *Javascript* khususnya dengan kerangka kerja *Vue*.
- b. Peserta mampu merancang dan membuat aplikasi yang dapat mengkonsumsi *RESTful API* ataupun *GraphQL*.
- c. Peserta mampu mengimplementasikan *Code Structure* yang efektif untuk mengatur agar perangkat lunak mudah untuk dipelihara dan dikembangkan.
- d. Peserta mampu mengimplementasikan *UI Framework* untuk responsivitas tampilan aplikasi.
- e. Peserta mampu melakukan *deployment*.
- f. Peserta mampu mengimplementasikan *Unit Testing* untuk menguji secara otomatis setiap fungsi dalam sistem.

1.3 Sistematika Penulisan

Untuk memudahkan serta mengarahkan dalam penulisan, maka dibuat sistematika penulisan, sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi pembahasan umum yang berhubungan dengan penyusunan Laporan Akhir MSIB, yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas tinjauan pustaka sebagai dasar ilmu pengetahuan dan acuan referensi yang diterpkan guna memperkuat kebaruan penelitian pada topik yang dibahas dalam laporan ini.

BAB III TINJAUAN UMUM

Bab ini membahas tentang profil perusahaan, visi perusahaan, misi perusahaan, dan sturktur organisasi perusahaan

BAB IV PEMBAHASAN DAN HASIL

Bab ini membahas tentang pembuatan aplikasi yang telah dibuat penulis dalam tugas *Mini Project* di MSIB Alterra Academy.

BAB V KESMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari kegiatan dari aplikasi dan saran untuk kegiatan MSIB kedepannya.

BAB II

LANDASAN TEORI

2.1 *Version Control and Branch Management (Git)*

2.1.1 *Versioning*

Versioning adalah mengatur versi dari *source code* program, yaitu mengupgrade file revisi *source code* program kedalam *file* yang berbeda dan ter-*update*. *Tools* yang dipakai dalam *versioning* adalah sebagai berikut:

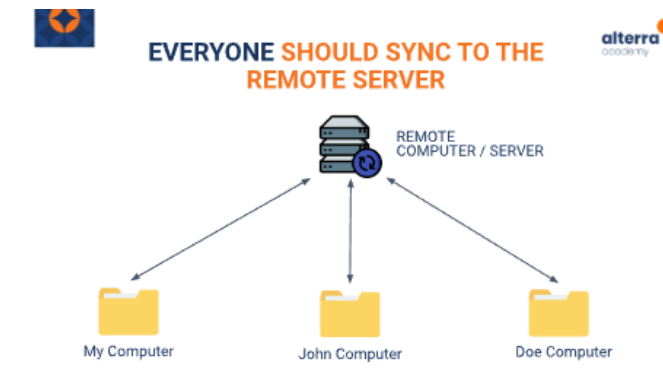
1. Git
2. Visual Studio Code
3. Github

Sejarah dari *versioning* :

1. Version Control System (*VCS*)
 - a. *Single User* :
 - SCSS – 1972 *Unix only*.
 - RCS – 1982 *Cross platform, text only*.
 - b. *Centralized* :
 - CVS – 1986 File focus
 - *Perforce* – 1995
 - *Subversion* – 2000 – *track directory structure*
 - *Microsoft Team Foundation Server* – 2005.
 - c. *Distributed* :
 - *Git* – 2005
 - *Mercurial* – 2005
 - *Bazaar* – 2005

2. GIT

Salah satu *version control system* populer yang digunakan para *developer* untuk mengembangkan *software* secara bersama-sama. Dengan adanya GIT kita dapat melakukan *programming* dengan kolaborasi di dunia nyata dan dimana lokasinya yang harus sudah terhubung dengan internet. Pemanfaat *GIT* ini terdistribusi bukan tersentralisasi. *GIT* dibuat oleh Linus Torvalds (2005) – *Linux Kernel* <https://github.com/torvalds/linux>.



Gambar 2. 1 Remote Computer/Server

file telah tersinkronkan oleh GIT semua

remote server, dengan masing – masing *file* pengguna mempunyai *backup file* tersendiri yang telah di clone dari *GIT Repository* dan tidak akan berpengaruh ketika *server* mendapatkan sebuah masalah. *GIT Repository* terbagi kedalam dua bagian yaitu *Folders* and *Files* dan *History*.

3. Github

Github adalah situs *web* dan layanan *cloud* yang bisa membantu para pengguna untuk menyimpan, mengelola dan mengembangkan. Didalam *Github* kita bisa meng-*Upload file*, membuat *file* yang mana *file*-nya bisa dikelola dengan *version control system* punya *github*. *Github* adalah inti *hosting platform* untuk kontrol versi kolaborasi dan berintegrasi dengan layanan seperti *Amazon*, *Google Cloud* dan *Code Climate*, yang mana dapat menampung sintaks dari 200 bahasa pemrograman yang tidak sama.[1]

2.2 UI/UX.

2.2.1 User Experience (UX)

Adalah hal yang pengguna rasakan ketika menggunakan produk. Berikut adalah 4 hal dari *User Experience* :

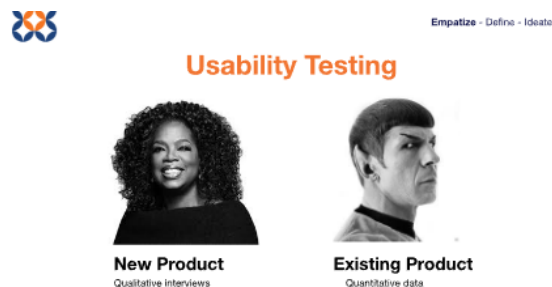
1. *Emphatize*

Yaitu seolah kita melihat dan memperhatikan dunia user kita, berikut adalah tahapan dari *Emphatize*:



Gambar 2. 2 *Emphatize*

- a. User Persona, untuk melihat karakter dari setiap produk. Yang terdiri dari profile user yang didasari ekspektasi, kebutuhan, tujuan dan kebiasaan dari user.
- b. User Job Stories Based on Hooked Model, yang memiliki 3 hal inti yaitu :
 - *Situation* dari user
 - *Motivation* user
 - *Outcome* yang diinginkan user
- c. *Usability Testing* merupakan salah satu cara untuk mengetahui seberapa efisien user menggunakan aplikasi secara efektif, mengetahui tujuan *user* dan kepuasan *user*. Tahapan ini bisa dilakukan dengan menggunakan produk baru dengan pengambilan data secara *qualitative*, dan juga bisa menggunakan produk yang sudah ada dengan pengambilan data secara *quantitative*.



Gambar 2. 3 *Usability Testing*

2. *Define*

Tahap untuk mendapatkan informasi dari *empathize*, berikut adalah tahapannya :

a. *Prioristing Pain Points*



Gambar 2. 4 Prioristing Pain Points

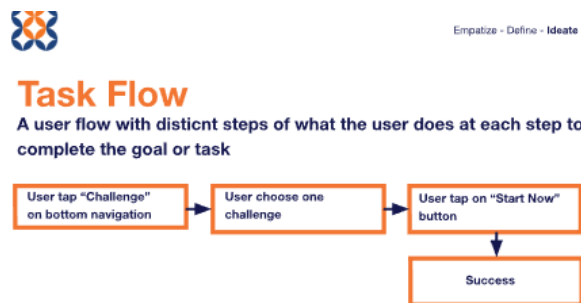
b. *Defining the Core Problem*, untuk mencari inti dari permasalahan yang telah di analisis untuk kebutuhan produk kita.

3. *Ideate*

Tahapannya :

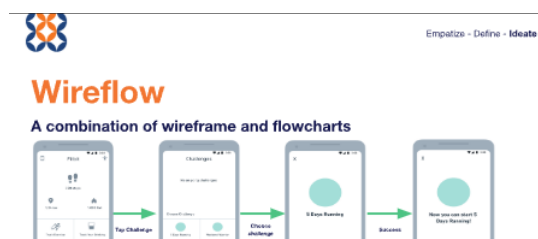
a. *User Goal*, berisi tentang alur *user*, persyaratan, journey dari pengguna, tahapan, cerita untuk mencapai tujuan.

b. *Task Flow*, berisikan langkah – langkah yang harus dikerjakan oleh pengguna.



Gambar 2. 5 Task Flow

c. *Wireflow*, kombinasi dari *wirefram* dan *flowchart*. *Wirefram* adalah kerangka untuk menata suatu item didalam proyek, *Flowchart* adalah diagram untuk menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses.



4. *Prototype dan Validate*

Tahapan yang paling berhubungan untuk mengevaluasi desain – desain yang di buat.[2]

2.2.2 *User Interface (UI)*

Adalah visual pengguna untuk berinteraksi dengan produk/aplikasi. Ada 4 pilar yang didasari tentang bagaimana penampilan dari desain, sebagai berikut :

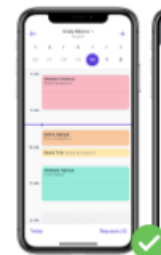
1. *Consistency*

Yaitu membuat desain dengan pola yang sama pada setiap halaman desain.



2. *Keep the interface simple* Gambar 2. 7 Consistency

Tidak menggunakan desain yang berantakan dan menggunakan bahasa yang menarik.



Gambar 2. 8 *Keep the interface simple*

3. *Good typography*

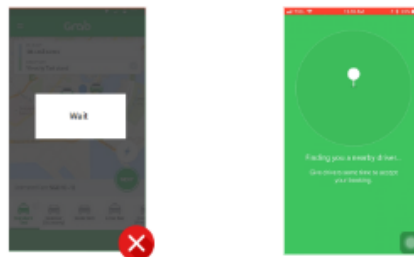
Dengan mengedepankan penggunaan *typeface* yaitu menggunakan kumpulan huruf yang berpola sama dengan memperhatikan jenis *font*, ukuran dan pengaturannya.



Gambar 2. 9 Good Typography

4. Offer Informative Feedback

Informasi balikan kepada user dengan berarti, relevan dan jelas.[2]



Gambar 2. 10 Offer Informative Feedback

2.2.3 Prototype

Yaitu sebuah simulasi yang menggambarkan hasil dari produk yang bertujuan untuk mengetes produk yang dibuat sebelum menghasbiskan biaya yang akan dibutuhkan.[3]

2.3 Figma

Figma adalah desain digital dan alata *prototyping*. Aplikasi desain UI dan UX yang digunakan untuk membuat situs *web*, aplikasi atau komponen antarmuka pengguna yang dapat diintegrasikan ke dalam proyek lain.[4]

Figma didukung oleh komunitas desainer dan pengembang yang kuat yang berbagi plugin untuk meningkatkan fungsionalitas dan mempercepat alur kerja.

Figma adalah alat atau *tool* yang solid termasuk dalam kemampuannya untuk membantu para deainer dan penggunanya dalam kemudahan pemasaran produk atau desain.[5]

2.4 Node.JS

Node.js adalah sistem perangkat lunak yang di desain untuk pengembangan aplikasi *web*. Menggunakan bahasa *JavaScript*, menggunakan basis *event* dan *asynchronous I/O*. *Node.js* dieksekusi sebagai aplikasi *server*. Yang terdiri dari *V8 JavaScript Engine* buatan Google dan beberapa modul bawaan yang terintegrasi.

Dengan menggunakan *Node.js* kita dapat membuat programming model yang mudah dengan skala *server* yang cukup besar dengan mudah. Berikut adalah 3 kelebihan *Node.js*. Yaitu :

1. ***Node is Easy*** – *Node* membuat I/O pemrograman menjadi lebih mudah dan dimengerti dari sebelumnya.
2. ***Node is Lean*** – *Node* tidak mencoba untuk menyelesaikan semua masalah, tetapi ini bergantung dengan pondasi *internet protocol* yang menggunakan fungsi API.
3. ***Node does not compromise*** - *Node* tidak mencoba jalan dengan *software* yang sudah *out of date*. Karena *Node.js* memberikan tampilan baru yang *fresh*.^[6]

2.5 *Vue Js*

Vue Js adalah suatu *library javascript* yang digunakan untuk membangun antarmuka *website* yang interaktif. *Vue* difokuskan hanya pada *view layer* sehingga sangat mudah diimplementasikan dan diintegrasikan dengan *library* lain ataupun juga dengan *project* yang sudah ada sebelumnya. Mempelajari *Vue* ini setidaknya harus sudah mengerti dasar HTML, CSS dan *Javascript*.^[7]

2.6 *Bootstrap*

Bootstrap merupakan *framework* HTML, CSS, *Javascript* populer untuk membangun situs *website* yang *responsive*. RWD (*Responsive Web Design*) adalah desain situs yang otomatis akan menyesuaikan diri agar tampil baik disemua perangkat dari tampilan *mobile* dan *desktop*. *Bootstrap* terdiri dari satu set *file css, javascript, dan jquery*.^[8]

2.7 *Bootstrap-Vue*

Bootstrap-Vue adalah *framework* dari *bootstrap* yang digunakan dalam pembuatan aplikasi menggunakan *library Vue Js*. Dengan menggunakan *Bootstrap-Vue* dapat mempermudah *styling* dalam pembuatan aplikasi yang *responsive* dalam tampilan *mobile* ataupun *desktop*.^[9]

2.8 MySQL

MySQL atau dibaca My Sequel merupakan sebuah Database Management System atau sering disingkat DBMS yang dijalankan menggunakan perintah SQL (Structured Query Language) yang populer digunakan untuk pembuatan aplikasi berbasis website. Selain itu SQL dipuji karena kesederhanaan sintaks yang pendek dan mudah dipahami. Sehingga SQL cocok dipilih sebagai bahasa terbaik untuk memulai untuk belajar data science bagi pemula sebelum belajar bahasa pemrograman. MySQL juga termasuk ke dalam RDBMS atau Relational Database Management System, dimana di dalam struktur databasenya sehingga ketika proses pengambilan data menggunakan metode relational database. Yang juga menjadi penghubung antara perangkat lunak dan database server^[10]