

Perancangan Web

IF2104

By:
Participant Handbook

Course Version: 2023

Universitas Mikroskil, Copyright ©2023

COURSE OVERVIEW

COURSE OVERVIEW

Mata kuliah ini membahas tentang bagaimana membangun aspek visual dan interaktif dari aplikasi web dengan mempelajari HTML, CSS, Javascript, dan Git untuk pembuatan sebuah web dari sisi front end.

COURSE GOALS

Capaian Pembelajaran Lulusan yang dibebankan:

- Mampu mengimplementasikan dan mengembangkan ilmu pengetahuan dan teknologi, menerapkan nilai humaniora serta menyelesaikan penugasan kerja sesuai bidang keahliannya.
- Mampu untuk memanfaatkan teknologi informasi digital yang tepat untuk membuat rencana dan mengubah operasi bisnis serta mengkonsep ide campaign marketing dalam pengembangan bisnis digital.
- Menguasai konsep teoritis yang mengkaji, menerapkan dan mengembangkan serta mampu memformulasikan dan mampu mengambil keputusan yang tepat dalam penyelesaian masalah.
- Mampu untuk menerapkan konsep dan teori dasar pemrograman komputer dan menggunakan berbagai pendekatan pemrograman untuk membangun dan mengembangkan aplikasi TIK.

COURSE OBJECTIVES

Capaian Pembelajaran Mata Kuliah:

- Mahasiswa mampu memahami dan menerapkan konsep-konsep ke dalam HTML.
- Mahasiswa mampu memilih element dan semantik yang sesuai pada HTML.
- Mahasiswa mampu menerapkan konsep tipografi dan multimedia pada halaman HTML.
- Mahasiswa mampu memilih CSS format yang sesuai untuk diterapkan ke dalam HTML.
- Mahasiswa mampu merencanakan penggunaan GIT untuk kolaborasi dalam proses pengembangan aplikasi.
- Mahasiswa mampu menerapkan konsep dari layout dan grid pada CSS.
- Mahasiswa mampu membangun sebuah halaman web yang responsive secara design dengan menggunakan media queries.
- Mahasiswa mampu menerapkan penggunaan dari list untuk pembuatan daftar pada halaman web.
- Mahasiswa mampu membangun tabel dan form untuk proses pendataan pada halaman web.
- Mahasiswa mampu menyusun web dengan menerapkan konsep javascript.



UNIT 6

PENGENALAN GIT

UNIT OVERVIEW

Pada minggu ke-6 membahas topik **Pengenalan GIT**. Git adalah sebuah sistem pengontrol versi (version control system) yang dirancang untuk mengelola perubahan dalam kode sumber (source code) dari sebuah proyek perangkat lunak. Git memungkinkan beberapa pengembang bekerja pada proyek yang sama secara bersamaan dan memungkinkan pengembang untuk memeriksa histori perubahan kode sumber dengan mudah. Git juga memungkinkan pengembang untuk membagikan kode sumber mereka dengan pengembang lain di seluruh dunia melalui penyedia layanan hosting seperti GitHub, GitLab, dan Bitbucket. Git dirancang oleh Linus Torvalds pada tahun 2005 untuk mengelola pengembangan kernel Linux.

UNIT OBJECTIVES

Adapun capaian yang akan didapatkan pada topik ini, adalah:

- Mengetahui Git sebagai Version Control
- Mengetahui Github sebagai layanan Git berbasis cloud

UNIT CONTENTS

Lesson 1: Pengenalan GIT 4 - 5

PRE LAB

Pre lab, berisi pertanyaan mendasar terkait teori materi yang sedang diajarkan.

QUESTION

1. Apa yang dimaksud dengan Version Control System?
2. Apa yang manfaat dari penggunaan Git?
3. Sebutkan Perintah dasar yang ada pada Git?

CONTENT LESSON

CASE STUDY / PROJECT

Saat ini perusahaan **PT ABC**, memiliki task project yang akan dikerjakan oleh tim pengembang web yang terdiri dari 5 developer dengan role yang berbeda. Project tersebut diharapkan selesai dalam 1 minggu untuk launching produk terbaru dari PT ABC. Tim developer membutuhkan teknologi yang mendukung proses cara kerja yang efektif untuk meletakkan base project pada secara cloud.

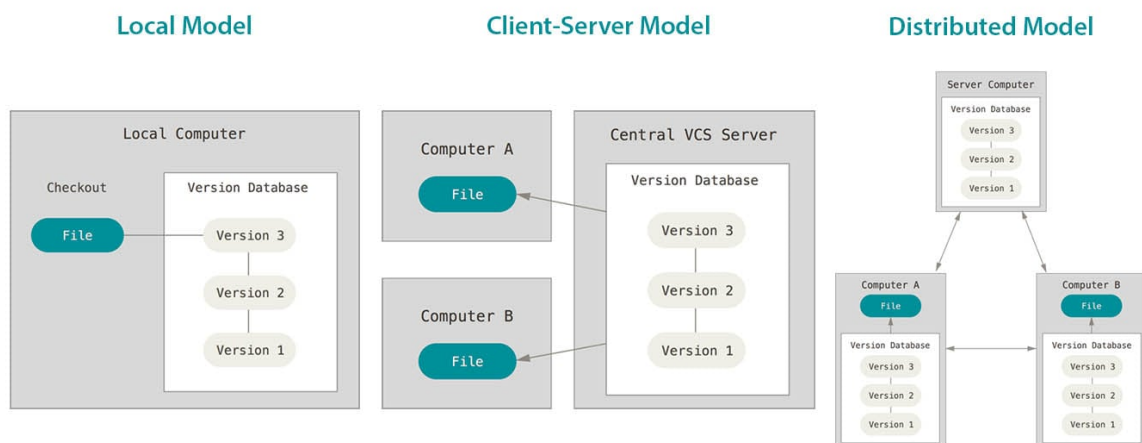
Mari kita bantu buat web PT ABC yuk ... !!!

IDENTIFICATION CONCEPT OF PROBLEM / PROJECT

Setelah membaca kembali persoalan yang diberikan, tentunya tujuan akhir dari studi kasus yang diberikan adalah menggunakan **Version Control System** yang ada pada git dan menggunakan **Github** yang menerapkan git berbasis cloud.

Sebelum kita mengerjakan soal tersebut, mari kita belajar terlebih dahulu caranya ... !!!

LESSON: PENGENALAN GIT SEBAGAI VERSION CONTROL SYSTEM



Terdapat 3 jenis dari Version Control System yaitu:

1. Local Model

Local version control systems merupakan jenis VCS paling sederhana dan memiliki database yang menyimpan semua perubahan pada file. LCS juga merupakan salah satu alat VCS yang paling umum. Sistem ini mampu membuat set patch, atau perbedaan antara file, dalam format khusus pada sebuah

disk. Dengan menambahkan semua patch, LCS dapat menciptakan bentuk orisinal dari semua file ketika seorang engineer atau developer membutuhkannya..

2. Centralized version control systems / Client-server Model

CVCS menjadi jembatan kolaborasi di antara para engineer dan developer. CVCS dapat memberikan informasi tentang apa yang dilakukan para engineer dan developer pada sebuah proyek. Hal ini memungkinkan database administrator untuk mengontrol pekerjaan yang harus dilakukan seluruh pihak dalam proyek tanpa terjadinya benturan dan miskomunikasi.

SOLUTION

Setelah memahami jenis dari VCS, berikut perlu menghadirkan solusi untuk menjawab permasalahan yang ada diatas.

INSTRUCTION

Silahkan ikuti langkah-langkah berikut untuk menyelesaikan permasalahan yang kalian buat:

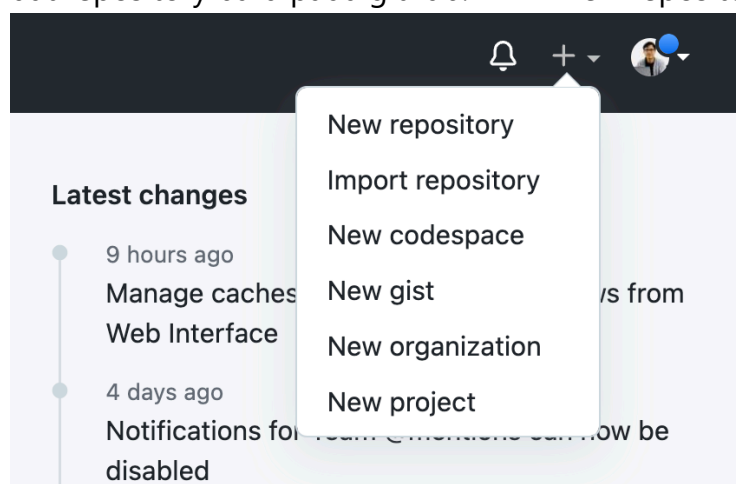
1. Buat **1 buah folder pada folder** kalian masin-masing dengan nama **M06**.
2. Gunakan **aplikasi Visual Studio Code dan Buka Folder M06** pada komputer lab masing-masing.
3. Pada bagian atas, silahkan kalian klik new file, dengan nama file **index.html**
4. Buat base html dasar seperti pada tahap 5.

5. Silahkan kalian ketikkan kode pada Gambar berikut ini:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Contoh Artikel Tanpa CSS</title>
9 </head>
10
11 <body>
12   <h1>Judul Artikel</h1>
13   <p>Ini adalah paragraf pertama dari artikel.</p>
14   <p>Ini adalah paragraf kedua dari artikel.</p>
15   <p>Ini adalah paragraf ketiga dari artikel.</p>
16   <ul>
17     <li>Item daftar pertama</li>
18     <li>Item daftar kedua</li>
19     <li>Item daftar ketiga</li>
20   </ul>
21   <p>Ini adalah paragraf keempat dari artikel.</p>
22   
23 </body>
24
25 </html>
```

Kemudian, bekerja dengan **git** dan **github**.

1. Buat repository baru pada github. Pilih "New repository"



2. Pada bagian create repository, tambahkan informasi terkait nama repository dan deskripsi. Kemudian klik tombol **create repository**

Owner ^{*} siopipin / Repository name ^{*} web-pt-abc ✓

Great repository names are short and memorable. Need inspiration? How about [crispy-potato](#)?

Description (optional)

☒ **Public**
 Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
 You choose who can see and commit to this repository.

Initialize this repository with:
 Skip this step if you're importing an existing repository.

☐ **Add a README file**
 This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
 Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
 A license tells others what they can and can't do with your code. [Learn more.](#)

① You are creating a public repository in your personal account.

[Create repository](#)

3. Tambahkan repository local dan push ke remote repository.

Pada tampilan berikutnya terdapat informasi untuk dapat bekerja pada git dan github. Pilih opsi **"... or create a new repository on the command line"** sebagai latihan.

Quick setup — if you've done this kind of thing before

or

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# web-pt-abc" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/siopipin/web-pt-abc.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/siopipin/web-pt-abc.git
git branch -M main
git push -u origin main
```

Ketik perintah berikut pada terminal / cmd (windows) base project.

- **git init** (berfungsi untuk inialisasi base project pada local data.)
- **git add .** (menambahkan semua file pada stage)
- **git commit -m "landing page"** (menambahkan ke stage commit dengan catatan)
- **git branch -M main** (menambahkan cabang main. --disarankan oleh github)
- **git remote add origin <url-git>** (untuk remote github ke git local)

- **git push -u origin main** (Push / upload ke github pada branch main)

4. Tampilan jika telah berhasil

Pada terminal:

```
perancangan_web_ti_a git:(master) git branch -M main
perancangan_web_ti_a git:(main) git remote add origin https://github.com/siopipin/web-pt-abc.git
perancangan_web_ti_a git:(main) git push -u origin main
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 8 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (30/30), 785.17 KiB | 34.14 MiB/s, done.
Total 30 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/siopipin/web-pt-abc.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
perancangan_web_ti_a git:(main)
```

pada github:

The screenshot shows the GitHub interface for a repository named 'web-pt-abc' by user 'siopipin'. The repository is public. At the top, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the repository name, it shows 'main' as the selected branch, '1 branch', and '0 tags'. There are buttons for 'Go to file', 'Add file', and 'Code'. The main content area displays a list of files and folders: 'siopipin landing page' (commit 92fd58a, 1 minute ago, 1 commit), 'M01' (landing page, 1 minute ago), 'M02' (landing page, 1 minute ago), 'M04' (landing page, 1 minute ago), '.DS_Store' (landing page, 1 minute ago), and 'html.png' (landing page, 1 minute ago). At the bottom, there is a prompt to 'Add a README'.

EXERCISE

EXERCISE OBJECTIVES

Pada latihan ini, mahasiswa diharapkan mampu untuk:

- Melakukan perubahan dan mencatat perubahan pada git.
- Menggunakan github untuk menyimpan perubahan secara cloud.

TASK 1: ANDA DIMINTA UNTUK MELENGKAPI SOURCE CODE LATIHAN DI ATAS. DI SETIAP PERUBAHAN ATAU PENAMBAHAN ELEMENT TAMBAHKAN PADA GITHUB.



UNIVERSITAS
MIKROSKIL