

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 7

PENGENALAN GIT LANJUTAN

UNIT OVERVIEW

Pada minggu ke-7 kembali membahas topik **Pengenalan GIT**. Seperti yang telah kita ketahui bawah Git adalah sebuah sistem pengontrol versi (version control system) yang dirancang untuk mengelola perubahan dalam kode sumber (source code) dari sebuah proyek perangkat lunak. Pada GIT kita bisa melakukan beberapa hal yang tentunya dapat membantu kita untuk saling berkolaborasi dengan tim. Pada GIT juga kita bisa melakukan percabangan yang bertujuan untuk membuat versi-versi dari aplikasi yang kita buat, sehingga kita tidak perlu lagi melakukan copy dan paste terhadap file-file yang besar.

UNIT OBJECTIVES

Adapun capaian yang akan didapatkan pada topik ini, adalah:

- Melakukan proses kolaborasi dengan GIT
- Melakukan proses perubahan pada proyek
- Melakukan pengecekan history pada GIT
- Melakukan proses branching pada GIT

UNIT CONTENTS

Lesson 1: Kolaborasi GIT 4 - 5

PRE LAB

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

QUESTION

1. Perintah apa saja yang harus kita lakukan sebelum melakukan perubahan pada proyek GIT?
2. Sebutkan dan jelaskan tahapan-tahapan yang terjadi pada kolaborasi GIT?
3. Apa yang dilakukan pada perintah **git status**?
4. Setelah kita melakukan proses penambahan ke dalam file, apakah kita bisa langsung menambahkan ke dalam repository global? Jika tidak, jelaskan mengapa!

CONTENT LESSON

CASE STUDY / PROJECT

Pada perusahaan **PT ABC**, memiliki task project yang akan dikerjakan oleh tim pengembang web yang terdiri dari 5 developer dengan role yang berbeda. Setelah sebelumnya sudah menemukan satu buah solusi dengan menggunakan Version Control System, kembali dihadapi dengan dilema. Dimana, mereka diharuskan untuk menambahkan fitur baru ke dalam proyek. Namun, dikarenakan waktu yang semakin mepet dan mereka juga ingin menghindari terjadinya kesalahan pada saat proses konfigurasi fitur baru. Namun, dikarenakan keseringan melakukan proses backup, menyebabkan kapasitas dari sistem yang digunakan menjadi overload. Oleh karena itu, mereka ingin melakukan pencarian solusi baru untuk mengatasi masalah tersebut.

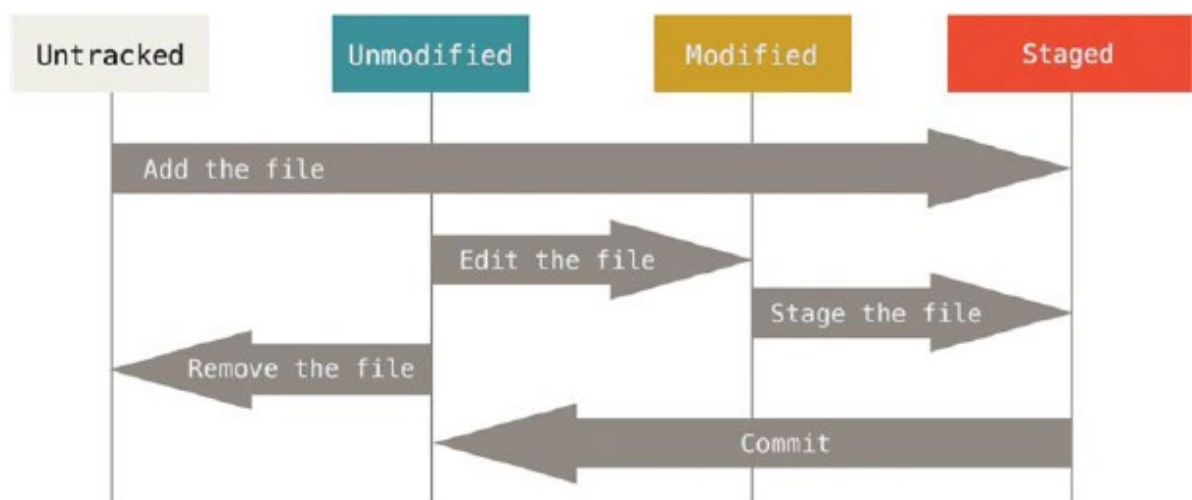
Mari kita bantu tim developer web PT ABC yuk ... !!!

IDENTIFICATION CONCEPT OF PROBLEM / PROJECT

Setelah membaca kembali persoalan yang diberikan, tentunya terjadi permasalahan terkait kapasitas penyimpanan yang habis (overload/kelebihan batas). Oleh karena itu, mungkin bisa menggunakan salah satu solusi yang ada pada Version Control System GIT, yaitu menggunakan proses **branching**.

Sebelum kita menyelesaikan permasalahan tersebut, mari kita belajar terlebih dahulu caranya ... !!!

LESSON: KOLABORASI PROYEK PADA GIT



Terdapat 4 tahapan yang harus kita ketahui jika kita ingin melakukan proses kolaborasi pada GIT, yaitu :

1. Tahapan Untracked

Tahapan ini merupakan tahapan awal. Dimana jika file yang termasuk ke dalam tahapan ini, menandakan bahwa file tersebut tidak masuk ke dalam repository local maupun global. File yang berada pada tahapan ini biasanya harus dilakukan proses penambahan terlebih dahulu dengan menggunakan perintah `git add`, supaya bisa dimasukkan ke dalam repository git.

2. Tahapan Unmodified

Pada tahapan ini biasanya file yang sebelumnya sudah masuk ke dalam repository atau berasal dari proses **cloning**. Pada tahapan ini biasanya tidak ada file yang bisa dilakukan proses **git add**, sehingga kita tidak perlu khawatir terhadap data yang hilang atau rusak.

3. Tahapan Modified

Pada tahapan ini biasanya file yang berasal dari proses cloning sebelumnya atau file yang baru ditambahkan namun belum dilakukan proses **git add**, maka akan masuk ke dalam tahapan ini. Tahapan ini biasanya akan terjadi jika terjadi perubahan pada file atau adanya penambahan file baru. Oleh karena itu, jika kita ingin melakukan penyimpanan pada repository local, kita harus melakukan proses penambahan terlebih dahulu.

4. Tahapan Staged

Pada tahapan ini biasanya file telah dilakukan proses `git add`. Pada tahapan ini juga file yang telah ditambahkan bisa kita masukkan ke dalam repository local dengan menggunakan perintah `commit`. Setelah file yang telah dilakukan proses `commit` nantinya akan masuk kembali ke dalam proses tahapan unmodified.

Untuk dapat melihat hasil kolaborasi pada GIT, kita juga bisa melakukan pengecekan terhadap sejarah yang telah terjadi pada git dengan menggunakan perintah : **git log**. Perintah ini nantinya akan memberitahukan kepada kita terkait history commit yang pernah terjadi, siapa yang melakukan dan apa pesan commit yang dilakukan.

Pada melakukan kolaborasi dengan menggunakan GIT kita bisa juga melakukan proses branching. Hal ini bertujuan supaya kita bisa menambahkan lebih banyak fitur pada proyek yang kita buat. Biasanya dalam proyek yang dibuat, pastinya dibutuhkan beberapa fitur dan tentunya ada proses backup yang harus dilakukan supaya jika terjadi kesalahan pada konfigurasi fitur, kita tetap bisa kembali ke proses sebelumnya. Namun, seiring berjalannya waktu, proses backup yang terus-terus dilakukan dapat mengakibatkan terjadi kekurangan space. Sehingga, kita harus menghapus proyek-proyek sebelumnya.

Untuk mencegah itu terjadi kita mungkin bisa memanfaatkan proses branching untuk mengatasi masalah tersebut. Dimana proses branching ini nantinya akan membuat sebuah penunjuk baru, yang nantinya akan digunakan untuk menunjuk terhadap fitur baru. Sehingga, pengguna tidak merasa khawatir, apabila terjadi kesalahan dalam konfigurasi pada proyek untuk fitur yang dibuat, bisa langsung berganti ke cabang awal tanpa perlu melakukan restore terlebih dahulu.

SOLUTION

Setelah memahami proses pembelajaran terhadap kolaborasi GIT, 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 baru pada folder** kalian masin-masing dengan nama **M07**.
2. Silahkan kalian clone proyek M06 pada minggu lalu ke dalam folder M07.
3. Tambahahkan 1 buah percabangan baru berupa **keterangan**.
4. Lakukan perpindahan ke cabang **keterangan**.
5. Tambahkan 1 buah file baru dengan nama file_keterangan.
6. Isi file tersebut dengan NIM, Nama, Kelas dan No PC yang digunakan.
7. Silahkan add file tersebut ke dalam repository local.
8. Lakukan proses commit dan berikan pesan "Ini penambahan file berupa keterangan data pribadi".
9. Lakukan proses push ke dalam Github masing-masing.

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 MENGUMPULKAN SELURUH TASK YANG PERNAH ANDA KERJAKAN DI LAB KE DALAM GITHUB MASING-MASING. ANDA BISA MENERAPKAN KONSEP PERCABANGAN UNTUK SETIAP MINGGUNYA ATAU MENGGUNAKAN FOLDER YANG BERBEDA. SETELAH SEMUANYA SIAP, LINK GITHUB KALIAN AKAN DISUBMIT KE DALAM TEAMS YANG DISEDIAKAN OLEH DOSEN.



UNIVERSITAS
MIKROSKIL