**PENJELASAN SINGKAT MENGENAI GIT DAN GITHUB**

Git adalah sebuah sistem pengontrol versi (version control system) yang digunakan untuk mengelola dan mengatur perubahan-perubahan pada sekumpulan file dalam suatu proyek. Git memungkinkan kita untuk menyimpan versi-versi dari file-file tersebut, sehingga kita bisa dengan mudah kembali ke versi sebelumnya jika terjadi kesalahan atau perubahan yang tidak diinginkan.

GitHub adalah sebuah platform yang menyediakan layanan hosting untuk proyek-proyek yang menggunakan Git. GitHub memungkinkan kita untuk menyimpan dan mengelola proyek-proyek kita secara online, sehingga kita bisa bekerja secara bersama-sama dengan tim lainnya dari mana saja.

Berikut ini adalah beberapa istilah atau perintah yang sering digunakan dalam Git dan GitHub:

1. Clone: Perintah "git clone" digunakan untuk mengambil seluruh konten dari repository di GitHub dan menyimpannya ke dalam folder di komputer kita.
2. Commit: Perintah "git commit" digunakan untuk menyimpan perubahan yang telah kita lakukan ke dalam repository Git. Setiap commit harus dilengkapi dengan pesan commit yang menjelaskan perubahan apa yang telah kita lakukan.
3. Push: Perintah "git push" digunakan untuk mengirim perubahan yang telah kita commit ke repository di GitHub.
4. Pull: Perintah "git pull" digunakan untuk mengambil perubahan terbaru dari repository di GitHub ke proyek kita di komputer.
5. Branch: Git memungkinkan kita untuk membuat beberapa cabang (branch) dari proyek kita. Cabang ini bisa dijadikan sebagai tempat untuk mengerjakan fitur-fitur baru atau perbaikan tanpa mengganggu proyek utama (master). Perintah "git branch" digunakan untuk membuat, menghapus, atau mengelola cabang-cabang dari proyek kita.
6. Merge: Setelah selesai mengerjakan fitur atau perbaikan di cabang tertentu, kita bisa menggabungkannya kembali ke proyek utama (master) dengan proses merge. Perintah "git merge" digunakan untuk menggabungkan perubahan dari cabang ke proyek utama.
7. Fetch: Perintah "git fetch" digunakan untuk mengambil perubahan terbaru dari repository di GitHub ke proyek kita di komputer, namun tidak menggabungkannya ke dalam proyek kita. Ini bisa digunakan untuk melihat perubahan apa saja yang telah dibuat oleh orang lain sebelum kita melakukan pull.
8. Remote: Remote adalah sebuah repository yang tidak berada di komputer kita, tetapi di server atau hosting lain. Perintah "git remote" digunakan untuk mengelola remote repository yang terhubung dengan proyek kita.
9. Pull request: Pull request adalah sebuah permintaan untuk menggabungkan perubahan dari cabang tertentu ke proyek utama (master). Kita bisa membuat pull request di GitHub untuk meminta review dari rekan kerja sebelum perubahan tersebut diterima ke dalam proyek utama.

Berikut ini adalah beberapa contoh penggunaan dari istilah atau perintah yang sering digunakan dalam Git dan GitHub:

1. Clone:

|  |
| --- |
| Contoh: |
| $ git clone <https://github.com/username/proyek-git.git> |
| Penjelasan: |
| Perintah ini akan mengambil seluruh konten dari repository "proyek-git" di GitHub milik pengguna "username" dan menyimpannya ke dalam folder baru bernama "proyek-git" di komputer kita. |

1. Commit:

|  |
| --- |
| Contoh: |
| $ git commit -m "Tambahkan file index.html" |
| Penjelasan: |
| Perintah ini akan menyimpan perubahan yang telah kita lakukan ke dalam repository Git dengan pesan commit "Tambahkan file index.html". |

1. Push:

|  |
| --- |
| Contoh: |
| $ git push origin master |
| Penjelasan: |
| Perintah ini akan mengirim semua perubahan yang telah kita commit ke repository "proyek-git" di GitHub yang terhubung dengan proyek kita melalui remote "origin". |

1. Pull:

|  |
| --- |
| Contoh: |
| $ git pull origin master |
| Penjelasan: |
| Perintah ini akan mengambil semua perubahan terbaru dari repository "proyek git" di GitHub yang terhubung dengan proyek kita melalui remote "origin" dan menggabungkannya dengan proyek kita di komputer. |

1. Branch:

|  |
| --- |
| Contoh: |
| $ git branch fitur-login |
| Penjelasan: |
| Perintah ini akan membuat sebuah cabang baru bernama "fitur-login" dari proyek kita. |

1. Merge:

|  |
| --- |
| Contoh: |
| $ git merge fitur-login |
| Penjelasan: |
| Perintah ini akan menggabungkan perubahan dari cabang "fitur-login" ke proyek utama (master). |

1. Fetch:

|  |
| --- |
| Contoh: |
| $ git fetch origin |
| Penjelasan: |
| Perintah ini akan mengambil perubahan terbaru dari repository "proyek-git" di GitHub yang terhubung dengan proyek kita melalui remote "origin", namun tidak menggabungkannya ke dalam proyek kita. |

1. Remote:

|  |
| --- |
| Contoh: |
| $ git remote add origin https://github.com/username/proyek-git.git |
| Penjelasan: |
| Perintah ini akan menambahkan sebuah remote repository bernama "origin" yang terhubung dengan repository "proyek-git" di GitHub milik pengguna "username". |

1. Pull request:

|  |
| --- |
| Contoh: |
| Kita bisa membuat sebuah pull request di halaman repository di GitHub dengan mengklik tombol "New pull request". Kemudian pilih cabang yang ingin kita gabungkan ke proyek utama (master) dan ikuti langkah-langkah yang diperlukan untuk membuat pull request. Setelah itu, rekan kerja kita bisa melakukan review terhadap perubahan yang kita buat dan menyetujui atau menolak pull request tersebut. Jika pull request disetujui, perubahan tersebut akan diterima ke dalam proyek utama (master). |
| Penjelasan: |
| Pull request adalah sebuah permintaan untuk menggabungkan perubahan dari cabang tertentu ke proyek utama (master). Kita bisa membuat pull request di GitHub untuk meminta review dari rekan kerja sebelum perubahan tersebut diterima ke dalam proyek utama. |

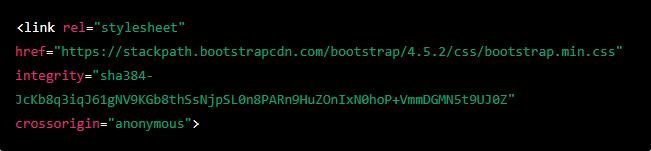
**PENJELASAN SINGKAT MENGENAI BOOTSTRAP**

Bootstrap adalah sebuah framework CSS yang populer digunakan untuk membuat website yang responsif. Ini berarti bahwa website yang dibuat menggunakan Bootstrap akan dapat ditampilkan dengan baik pada berbagai ukuran layar, seperti komputer, tablet, atau ponsel.

Framework ini menyediakan kumpulan class CSS yang dapat digunakan untuk mengaplikasikan gaya dan layout pada website. Beberapa class Bootstrap yang sering digunakan adalah:

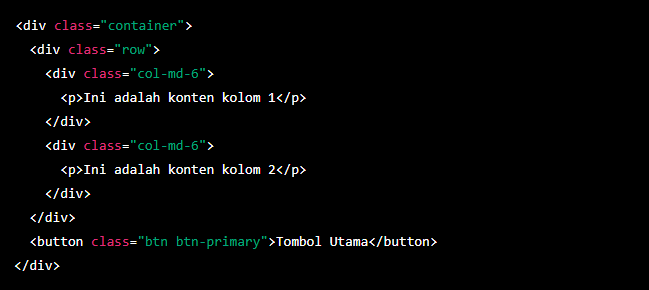
* container: digunakan untuk mengatur lebar konten website
* row: digunakan untuk membuat baris dalam grid Bootstrap
* col: digunakan untuk membuat kolom dalam grid Bootstrap, seperti col-md-6 yang akan membuat kolom dengan lebar 6 dari 12 pada layar medium
* btn: digunakan untuk membuat tombol dengan gaya Bootstrap, seperti btn-primary yang akan memberikan gaya tombol utama

Untuk memulai menggunakan Bootstrap, pertama-tama Anda perlu menambahkan link ke file CSS Bootstrap ke halaman HTML Anda. Anda dapat mengunduh file CSS Bootstrap dari situs web resmi Bootstrap atau menggunakan CDN (Content Delivery Network) seperti ini:



Setelah menambahkan link ke file CSS Bootstrap, Anda dapat mulai menggunakan class Bootstrap pada elemen HTML Anda untuk mengaplikasikan gaya Bootstrap.

Contoh penggunaan class Bootstrap pada elemen HTML:



Selain class untuk gaya, Bootstrap juga menyediakan beberapa komponen yang dapat digunakan seperti Navbar, Card, Modal, dll. Anda dapat menemukan dokumentasi lengkap tentang komponen Bootstrap di situs web resmi Bootstrap.

Selain itu, Bootstrap juga memiliki beberapa JavaScript plugin yang dapat digunakan untuk menambahkan fitur interaktif seperti modal, carousel, dan lain-lain.

**PENJELASAN SINGKAT MENGENAI MVC CODEIGNITER 3**

MVC adalah sebuah pola atau model yang digunakan dalam pemrograman untuk membantu kita mengelola kode program kita dengan lebih terstruktur dan terorganisir. MVC singkatan dari Model-View-Controller.

Model merupakan bagian dari program yang menangani data dan logika bisnis. Model bertugas mengambil data dari database atau sumber data lainnya, mengolah dan memproses data tersebut, lalu menyimpan kembali ke database atau sumber data yang sesuai.

View merupakan bagian dari program yang menangani tampilan atau user interface. View bertugas menampilkan data yang diperoleh dari model kepada pengguna melalui halaman web atau aplikasi.

Controller merupakan bagian dari program yang menjadi penghubung antara model dan view. Controller bertugas menerima permintaan dari pengguna, mengambil data yang diperlukan dari model, lalu menampilkan data tersebut ke view sehingga pengguna dapat melihatnya.

CodeIgniter adalah sebuah framework atau kerangka kerja yang digunakan untuk membantu kita dalam membangun aplikasi web dengan menggunakan bahasa pemrograman PHP. CodeIgniter mengikuti pola MVC sehingga kita bisa dengan mudah memisahkan bagian-bagian program kita sesuai dengan fungsinya masing-masing.

Jadi, MVC CodeIgniter adalah sebuah pola yang digunakan dalam pemrograman menggunakan framework CodeIgniter untuk membantu kita mengelola kode program dengan lebih terstruktur dan terorganisir. Bagian-bagian program tersebut terdiri dari model yang menangani data dan logika bisnis, view yang menangani tampilan, dan controller yang menjadi penghubung antara model dan view.