

Advanced Git, GitHub, & Spring Boot

PENJELASAN

Halo teman-teman! Selamat datang di tutorial pertama APAP :D. Pada tutorial ini, Anda akan mempelajari penggunaan **Git** yang lebih *advanced* dan membuat *web* Is Palindrome menggunakan *framework* **Spring Boot**. Tutorial ini diharapkan dapat membantu Anda memahami teknik-teknik dasar untuk **bekerja dalam tim** dalam pengembangan suatu aplikasi, juga menjadi **dasar** untuk pengerjaan tutorial-tutorial berikutnya.

PRASYARAT

- Memahami operasi dasar direktori *folder* dan *file* melalui cmd/terminal (`dir`, `mv`, `mkdir`, `cd`). [DDP1]
- Memahami dasar-dasar pengoperasian Git (`init`, `add`, `commit`, `status`, `stash`). [PPW]
- Memahami bahasa pemrograman Java dan konsep OOP. [DDP2, SDA]
- Memiliki akun [GitHub](#).
- Melakukan setup.

LINGKUP PEMBAHASAN

- Git
 - Branch
 - Checkout
 - Revert
- GitHub
 - Development Flow
 - Pull Request (PR)
 - Issue Tracker
- Spring Boot
 - Setup

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan Semester Ganjil Tahun Ajaran 2021/2022 **Deadline: 8 September 2021, 23.55 waktu Scele**

- Maven Dependency
- Controller & View

EKSPEKTASI

- Memahami operasi-operasi penting dalam Git serta penerapannya.
- Memahami fungsi GitHub dan *development flow* menggunakan Git.
- Melakukan *setup* Spring Boot menggunakan *maven* di *local environment* Anda.

PENGUMPULAN

- Tenggat Waktu Pengumpulan (Due Date) :
Rabu, 8 September, sebelum 23.55 WIB untuk semua kelas. Setiap keterlambatan selama 24 jam akan mendapatkan pengurangan nilai sesuai peraturan yang berlaku.
- Pengumpulan tutorial dalam bentuk **Pull Request (PR)**. Waktu keterlambatan dilihat dari **aktivitas perubahan** di Pull Request.
- **Tidak ada demo** untuk tutorial ini, tetapi akan ada demo untuk tutorial-tutorial selanjutnya.
- Tambahkan akun Github **seluruh Asisten Dosen di kelas Anda** ([daftar asisten dosen](#)) ke repository GitHub yang Anda buat (tutorial-apap).
- Pastikan *repository* Anda **private**.

PENILAIAN

$\text{Nilai akhir} = \text{Nilai Tutorial} + \text{Nilai Catatan}$

Plagiarisme akan dikenakan sanksi nilai 0! ☹

- Indikator Penilaian Tutorial
 - Kesesuaian operasi Git
 - Keberhasilan proyek Is Palindrome
 - Kelengkapan dan Ketepatan Jawaban pada README.md

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

TUTORIAL

Git

Dasar-dasar

1. Anda akan **membuat sebuah repository** yang akan digunakan selama mengerjakan tutorial dan melakukan *versioning* pada *repository* tersebut.
2. Buka terminal Anda. Pastikan Anda sudah berada di dalam *folder* tempat Anda ingin membuat *repository*.
3. Inisialisasikan sebuah *repository* baru bernama `tutorial-apap` menggunakan perintah ini.

```
git init tutorial-apap
```

4. Setelah menjalankan *script* di atas, maka akan terbentuk sebuah *directory* baru.

Tips: nama *directory* sesuai dengan parameter yang diberikan saat menjalankan operasi “init”. Pada kasus ini, namanya adalah `tutorial-apap`.

5. Masuklah ke *directory* tersebut menggunakan perintah berikut ini.
Buka *folder* tersebut menggunakan IDE yang Anda gunakan.

```
cd tutorial-apap
```

6. Buatlah *file* bernama `README.md`.

Tips: *file* `README.md` berisi informasi mengenai aplikasi yang Anda kembangkan. *File* tersebut juga dapat digunakan untuk menulis dokumentasi. Pada mata kuliah APAP, `README.md` akan digunakan sebagai tempat menulis catatan dan jawaban Anda selama mengerjakan tutorial.

Isi *file* `README.md` sebagai berikut, lalu *save*.

```
# Tutorial APAP

## Authors

* **<Nama Lengkap di SIAK-NG>** - *<NPM>* - *<Kelas>*
```

7. **Commit** *file* tersebut dengan *commit message* “docs: add README.md file”.

```
git add .
```

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

```
git commit -m "docs: add README.md file"
```

Tips:

- *Commit message* yang baik adalah yang mudah dimengerti dan dapat di-track kembali sewaktu-waktu
- Format *commit message* yang baik: <type*>: <scope | fitur> - <subject*>. *wajib ada
- Untuk mempelajari lebih lanjut mengenai *commit message* yang baik, bacalah artikel [berikut ini](#).

8. Buka tautan berikut [.gitignore](#). Buatlah *file* baru bernama `.gitignore` di dalam *directory root project* Anda dan *copy-paste* isi dari tautan tersebut ke dalamnya menggunakan *code editor* (e.g. VSCode). Setelah itu, lakukan **commit** dengan *message* "chore: add gitignore".

Revert

1. Buatlah *file* baru bernama `Test.java`. Kemudian, **commit** *file* tersebut dengan *message* "feat: add Test.java file".
2. Setelah melakukan *commit* untuk *file* tersebut, Anda merasa bahwa penambahan/perubahan *file* tersebut merupakan hal yang tidak diinginkan sehingga Anda ingin mengembalikan keadaan seperti sebelum *commit* yang terbaru. Untuk mengembalikan ke kondisi tertentu, hal pertama yang perlu Anda lakukan adalah melihat *history commit* menggunakan perintah berikut.

```
git log
```

3. Perintah tersebut akan menunjukkan informasi sebagai berikut.

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

```
C:\Users\Fitri\Documents\tutorial-apap>git log
commit 5ad9d076fbcf9a955722e244c076bb054d580fa1 (HEAD -> master)
Author: Fitri Handayani <fitriihndynii@gmail.com>
Date: Sat Sep 4 21:34:56 2021 +0700

    feat: add Test.java file

commit 63f0b0aa15da2fb23bac6fb0e059c6b9168af5b1
Author: Fitri Handayani <fitriihndynii@gmail.com>
Date: Sat Sep 4 21:33:18 2021 +0700

    chore: add gitignore

commit a3b78698a2c363a54aade3471e8b772a78a0ad5
Author: Fitri Handayani <fitriihndynii@gmail.com>
Date: Sat Sep 4 21:30:43 2021 +0700

    docs: add README.md file
```

Karena Anda ingin mengembalikan ke kondisi sebelum adanya *file* Test.java, maka Anda perlu menjalankan perintah berikut.

```
git revert <commit hash pada commit yang ingin
di-revert>
```

Sesuai dengan informasi yang ada pada terminal Anda, maka akan digunakan:

```
git revert 5ad9d076fbcf9a955722e244c076bb054d580fa1
```

Tips: Jika muncul **vim** untuk menuliskan *commit message*, Anda bisa menambahkan *message*, lalu menekan tombol Esc, dan menulis *command* :wq.

4. Dapat dilihat pada *directory* Anda bahwa *file* 'Test.java' sudah tidak ada. Selain itu, jika dilihat pada git log, HEAD *repository* Anda sudah berada pada kondisi *revert*.

```
C:\Users\Fitri\Documents\tutorial-apap>git log
commit 88346288fb9563f921ad91473038cfb4b8d472c0 (HEAD -> master)
Author: Fitri Handayani <fitriihndynii@gmail.com>
Date: Sat Sep 4 21:39:04 2021 +0700

    Revert "feat: add Test.java file"

    This reverts commit 5ad9d076fbcf9a955722e244c076bb054d580fa1.
```

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

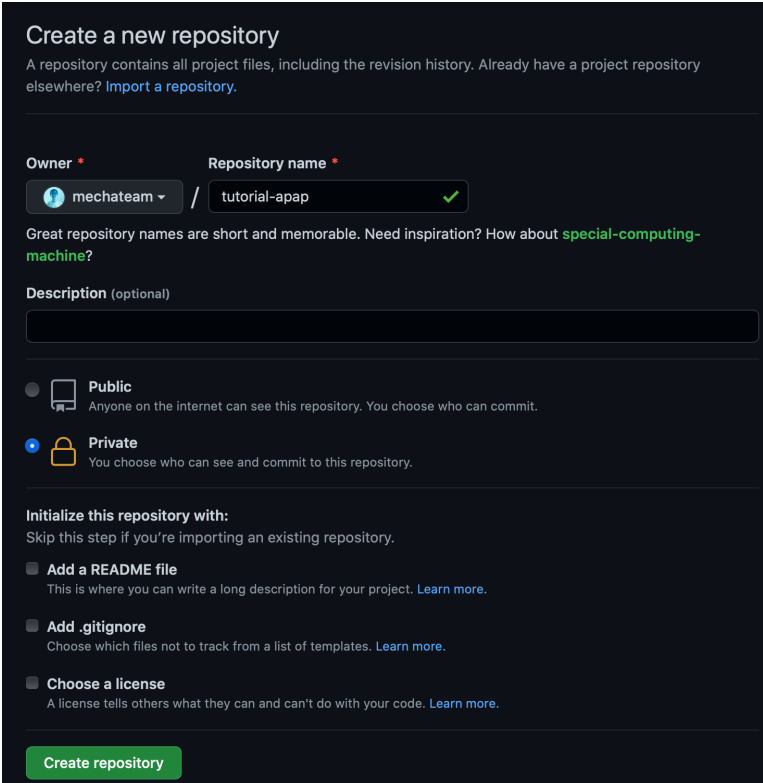
Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

GitHub

Sekarang, Anda akan membuat suatu *repository* di GitHub yang akan dihubungkan dengan *local repository* tutorial-apap yang Anda buat sebelumnya.


1. Login ke github.com, lalu klik tombol “New”.
2. Masukkan nama *repository* Anda `tutorial-apap`, lalu buat jadi *private repository*. Klik “Create repository”.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner ^{*} Repository name ^{*}

 mechateam / tutorial-apap ✓

Great repository names are short and memorable. Need inspiration? How about [special-computing-machine?](#)

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](#).

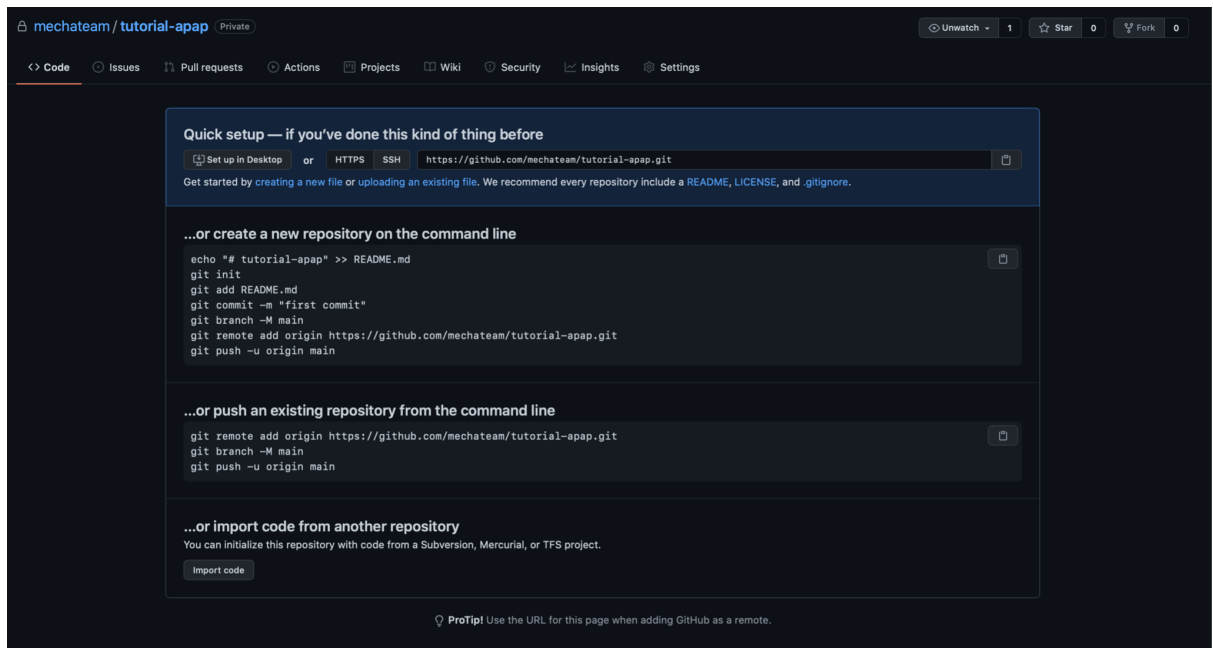
Create repository

3. Setelah *repository* sudah dibuat, maka akan muncul tampilan sebagai berikut.

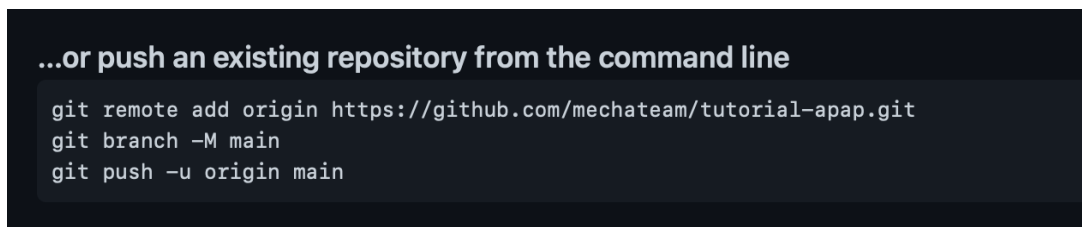
Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan


Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele



4. Push *repository* yang sudah Anda buat di *local* ke dalam *repository* GitHub yang baru saja dibuat. Untuk itu, lihat bagian **push an existing repository from the command line**. Eksekusi perintah-perintah tersebut saat *command line* Anda sudah berada pada direktori proyek Git yang telah Anda buat.



5. Sekarang *repository* Anda sudah terhubung dengan GitHub. *Refresh* halaman *repository* GitHub Anda.
6. Buka *tab* Issue Tracker pada halaman *repository* GitHub Anda ( Issues). Buatlah *issue* pertama dengan isi sebagai berikut:
- Judul: “Tutorial 1 is Palindrome”
 - Body: “Buka [link ini](<https://start.spring.io/>) untuk download project spring boot.”
 - Assignees: Anda sendiri
 - Label: enhancement

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

7. Anda dapat menggunakan *issue* tersebut untuk mencatat, atau membuat *issue* baru jika terjadi masalah. Dengan begitu, masalah atau permintaan perubahan dapat di-*track* dengan mudah.
8. Tambahkan semua asisten dosen di kelas Anda sebagai *collaborator* dalam *repository*. Informasi asisten dapat diakses [disini](#).

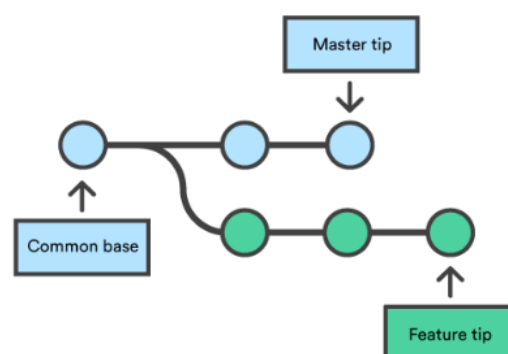
Tips: Apabila Anda hanya dapat menambahkan 3 *collaborator*, coba [apply](#) student package. Tambahkan *email* UI sebagai *secondary email* di akun Anda.

Branch & Checkout

Pada bagian ini, Anda akan mempelajari beberapa operasi Git yang lebih *advanced* untuk digunakan dalam *workflow development* tutorial & *project* Anda kedepannya.

1. Pertama, kita akan membuat sebuah *branch*. Apa itu *branch*? *Branch* adalah sebuah duplikat dari suatu objek. Saat Anda menduplikasi objek tersebut, Anda dapat melakukan perubahan secara paralel dalam suatu cabang yang berbeda tanpa mengubah objek utama dari duplikat tersebut.

Misal pada gambar dibawah (lihat dari kiri ke kanan) *branch* Master (biru) diduplikasi menjadi *branch* Feature (hijau). Isi dari Master dan Feature saat duplikasi terjadi tentu sama persis. Lalu tanpa harus menunggu *branch* Feature selesai, Master tetap dapat berjalan. Isi dari Master dapat berubah misal karena teman Anda juga memiliki fitur yang sedang dikerjakan dan sudah selesai terlebih dahulu, dan *branch* Feature tetap dapat berjalan tanpa harus bergantung pada Master (kecuali ada kasus khusus).

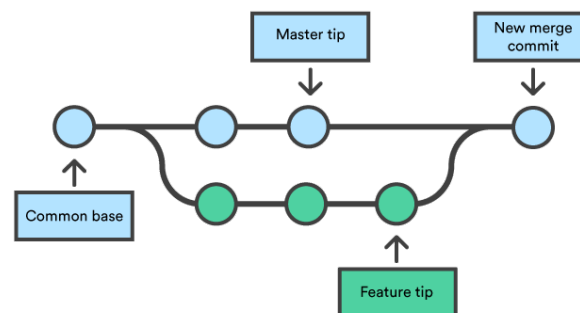


Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

Selanjutnya, ketika Anda selesai mengerjakan fitur Anda di *branch* Feature, Anda ingin fitur Anda dapat digunakan oleh orang lain. Untuk itu, Anda perlu menggabungkan perubahan yang Anda lakukan di *branch* Feature ke *branch* Master. Proses ini disebut dengan **Merging**. Anda dapat mempelajari lebih lanjut pada [tautan ini](#).



2. Karena Anda akan membuat suatu program bernama Is Palindrome, maka pertama-tama Anda perlu membuat *branch* untuk mengerjakan tutorial ini. Berikan *branch* tersebut nama ``feat/tutorial-1-is-palindrome``, dimana feat merupakan kependekan dari Feature. Untuk membuat *branch* baru, jalankan perintah berikut.

```
git branch feat/tutorial-1-is-palindrome
```

Tips:

- Saat Anda membuat *branch*, maka *branch* baru merupakan duplikat dari *branch* tempat Anda membuatnya. Misalnya jika Anda membuat *branch* baru dari Master, maka *branch* baru tersebut adalah duplikat dari Master.
- Penamaan *branch* adalah sebagai berikut:
`<type>/nama-scope-seperti-masalah-atau-fitur`
- Penamaan seperti ini membantu membedakan konteks dari setiap *branch* yang Anda buat, sehingga dapat memudahkan pekerjaan Anda.

3. Selanjutnya, kita perlu pindah ke *branch* tersebut. Untuk pindah *branch*, kita dapat melakukan operasi *checkout*.

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

```
git checkout feat/tutorial-1-is-palindrome
```

Tips:

- Anda dapat membuat *branch* sekaligus melakukan *checkout* dengan menambahkan argumen “b” di operasi *checkout* (contoh: “git checkout -b nama-branch”).

4. Anda sekarang sudah berada pada *branch* `feat/tutorial-1-is-palindrome`.

```
(base) Meldi-Hafizh:tutorial-apap user$ git checkout feat/tutorial-1-is-palindrome  
Switched to branch 'feat/tutorial-1-is-palindrome'
```

Spring Boot

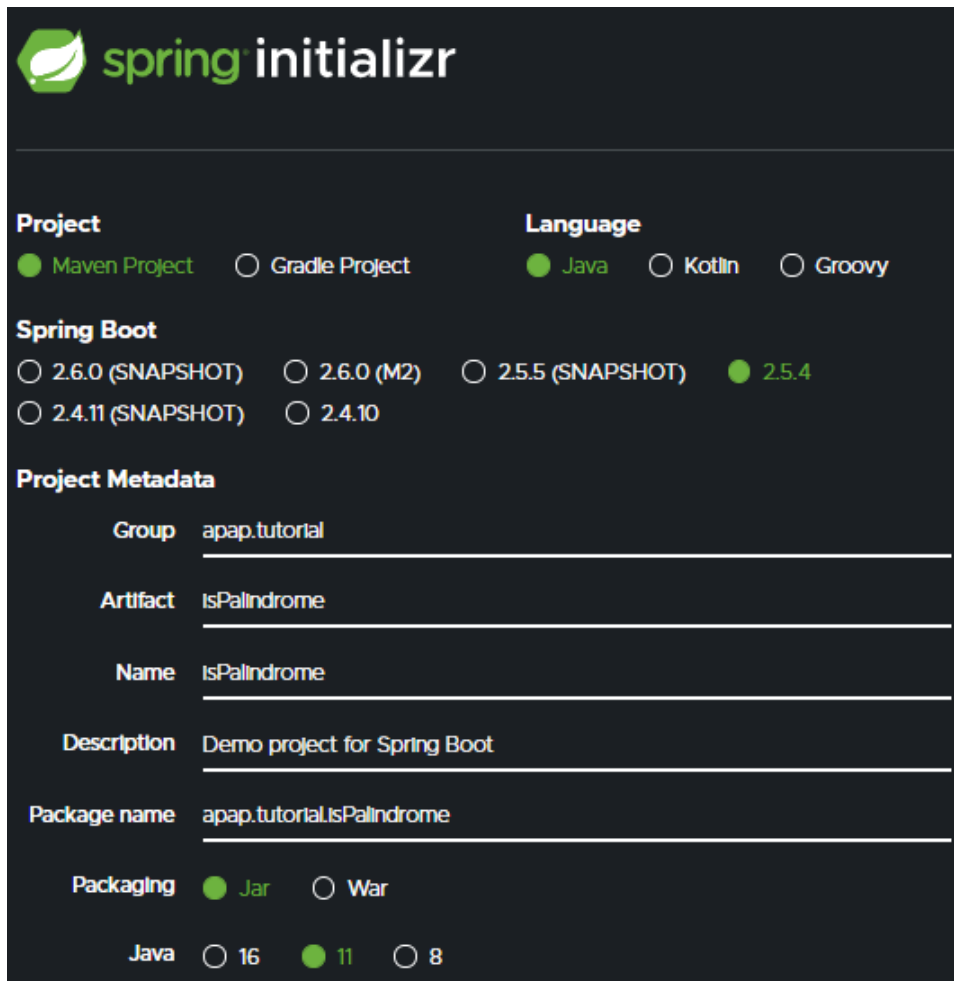
Pada bagian ini Anda akan berlatih menggunakan Spring Boot untuk mengerjakan tutorial. Spring Boot akan digunakan seterusnya pada tutorial, tugas, UTS, dan tugas akhir.

1. Buka issue yang Anda buat tadi dan klik Link di issue yang Anda buat sebelumnya. Selanjutnya, akan muncul window browser sebagai berikut. Isi form sesuai gambar berikut.

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele



The screenshot shows the Spring Initializr web application interface. It has a dark theme with green accents. The header features the Spring logo and the text 'spring initializr'. Below the header, there are two main sections: 'Project' and 'Language'. In the 'Project' section, 'Maven Project' is selected with a green radio button, while 'Gradle Project' is unselected. In the 'Language' section, 'Java' is selected with a green radio button, while 'Kotlin' and 'Groovy' are unselected. Below these sections is the 'Spring Boot' section, where '2.5.4' is selected with a green radio button, and other versions like '2.6.0 (SNAPSHOT)', '2.6.0 (M2)', '2.5.5 (SNAPSHOT)', and '2.4.11 (SNAPSHOT)' are unselected. The '2.4.10' version is also listed but unselected. Below the Spring Boot section is the 'Project Metadata' section, which contains several input fields: 'Group' with the value 'apap.tutorial', 'Artifact' with the value 'IsPalindrome', 'Name' with the value 'IsPalindrome', 'Description' with the value 'Demo project for Spring Boot', and 'Package name' with the value 'apap.tutorial.IsPalindrome'. At the bottom of the form, there is a 'Packaging' section with 'Jar' selected (green radio button) and 'War' unselected. Below that, there is a 'Java' section with '11' selected (green radio button) and '16' and '8' unselected.

2. Tambahkan dependencies berikut

a. Spring Web

Digunakan untuk membangun web dan web service berbasis MVC.

b. Thymeleaf

Template engine, disini HTML, CSS, JS akan diproses.

c. Spring Boot DevTools

Untuk live reload, restarts, dan konfigurasi.

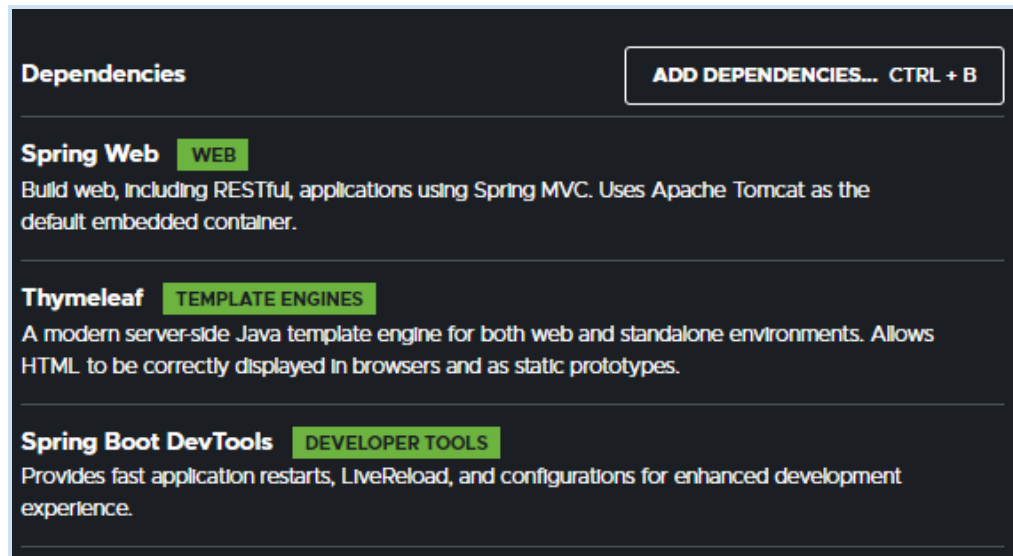
Tips: Package merupakan suatu cara pengelompokkan/pengorganisasian kelas-kelas dan interface

yang saling terkait menjadi suatu unit tunggal dalam library.

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

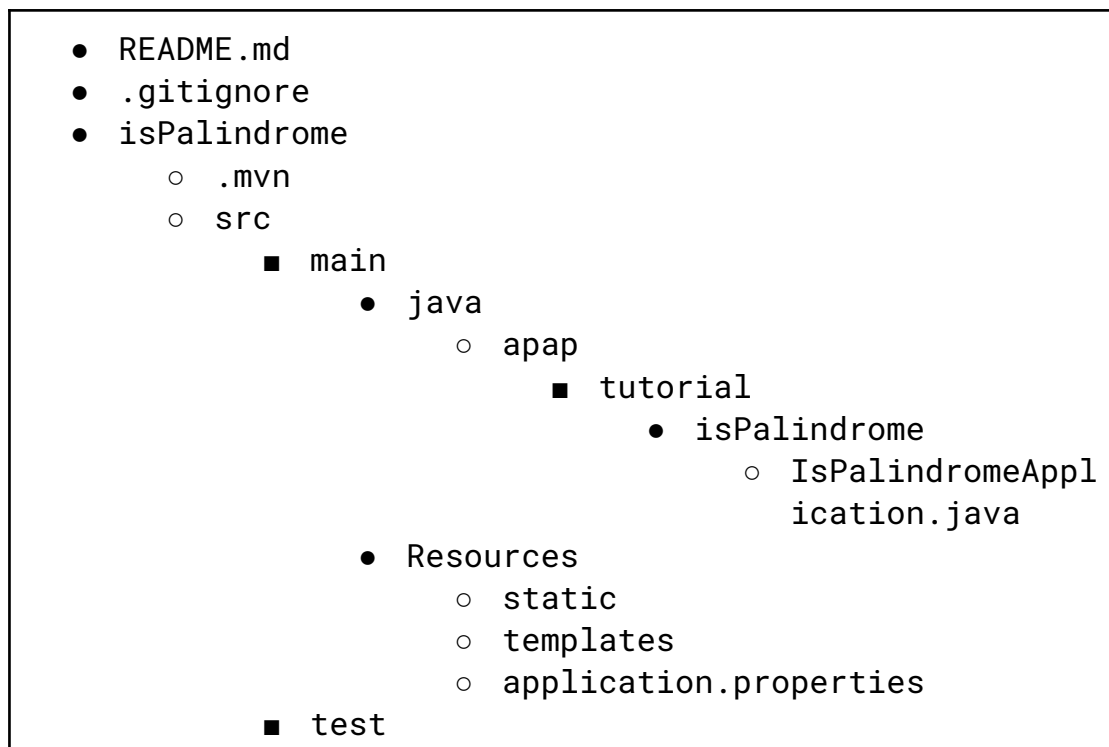
Deadline: 8 September 2021, 23.55 waktu Scele



3. Klik “Generate”. Suatu file zip terdownload. Ekstrak isi zip ke dalam root directory project.



4. Directory project Anda seharusnya akan berisi tree sebagai berikut.



Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

- mvnw
- mvnw.cmd
- pom.xml
- HELP.md
- .gitignore

5. Buka folder `isPalindrome` menggunakan IDE yang Anda gunakan dengan fitur `import`. Pastikan program Anda berjalan.
6. IntelliJ:
 - a. Pilih menu Open or Import



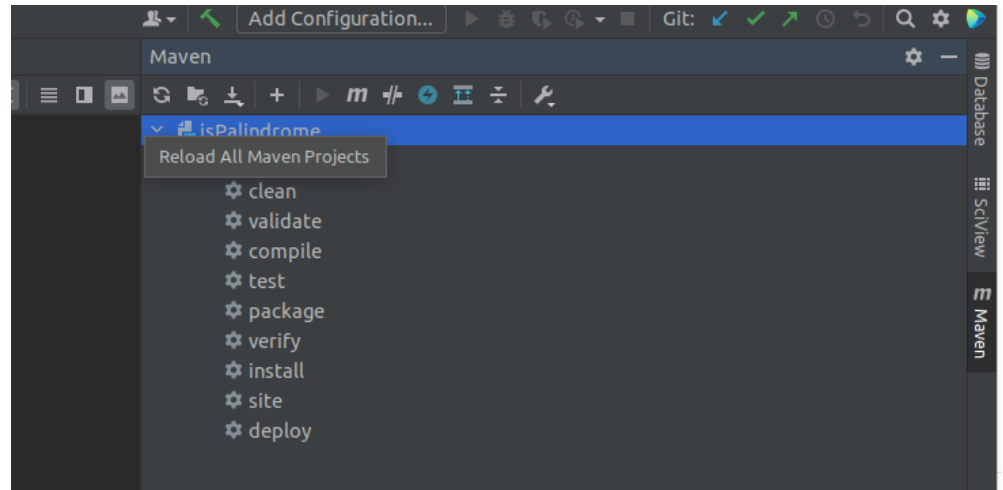
- b. Pilih folder `isPalindrome`
- c. Tunggu IntelliJ indexing & resolve dependencies yang digunakan.
 - i. Anda juga dapat melakukan indexing manual dengan menekan tombol Maven dibagian kanan IntelliJ dan klik pada tombol "Reload All

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

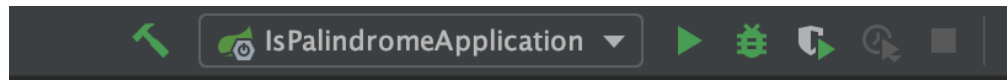
Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

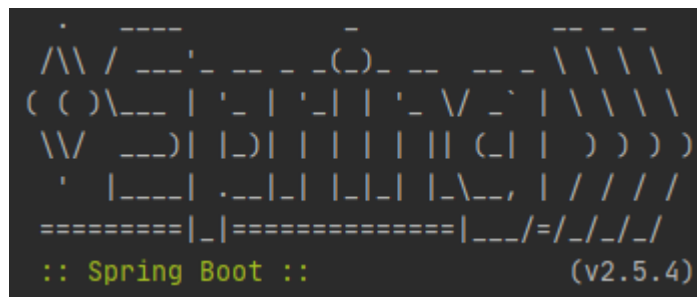
Maven Projects”



d. Jalankan program



e. Jika muncul tampilan seperti berikut, Selamat! Program Spring pertama Anda berhasil berjalan



f. Commit perubahan pada git repository Anda dengan message sebagai berikut
 `feat: tutorial 1 - initialize spring boot project`.

Program Is Palindrome

**Program Pengecek
Palindrome**

Kata anda adalah: aku

**Kata anda bukanlah
Palindrome**

Pada bagian ini kita akan membuat halaman cek Palindrome. Palindrom adalah sebuah kata, frasa, angka maupun susunan lainnya yang dapat dibaca dengan sama baik dari depan maupun belakang. Program yang akan dibuat adalah yang dapat mengecek suatu kata apakah termasuk kata palindrome atau bukan. Program akan mengembalikan suatu html berisikan kategori kata tersebut apakah termasuk kata palindrome atau bukan. Kembali ke IDE Anda.

1. Buat dua packages baru bernama `model` dan `controller` pada directory/package `apap.tutorial.isPalindrome`

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

2. Buat class baru bernama `Palindrome` di package `model`.

```
1 package apap.tutorial.isPalindrome.model;
2
3 import java.io.Serializable;
4
5 public class Palindrome implements Serializable {
6
7     private String kalimat;
8
9     @
10    public Palindrome(String kalimat){
11        this.kalimat = kalimat;
12    }
13
14    public boolean isPalindrome(){
15        int n = kalimat.length();
16
17        for (int i = 0; i < (n/2); ++i){
18            if (kalimat.charAt(i) != kalimat.charAt(n - i - 1)){
19                return false;
20            }
21        }
22
23        return true;
24    }
25
26    public String getKalimat() {
27        return kalimat;
28    }
29 }
```

3. Buat class baru bernama `PalindromeController` di package `controller`. Kita perlu membuat satu fungsi untuk setiap satu URI. Pada kasus ini, diberikan sebuah kata yang akan di cek apakah kata tersebut sebuah palindrome di sebuah page.
4. Pertama tambahkan import berikut.

```
package apap.tutorial.isPalindrome.controller;

import apap.tutorial.isPalindrome.model.Palindrome;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestParam;
```


Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

5. Pertama Anda akan membuat satu halaman yang menerima query kata menggunakan RequestParam. Dengan menggunakan RequestParam, Anda dapat mengekstrak data yang diberikan melalui query dengan parameter yang sudah ditentukan.

```
1 package apap.tutorial.isPalindrome.controller;
2
3
4 import apap.tutorial.isPalindrome.model.Palindrome;
5 import org.springframework.stereotype.Controller;
6 import org.springframework.ui.Model;
7 import org.springframework.web.bind.annotation.GetMapping;
8 import org.springframework.web.bind.annotation.RequestParam;
9
10
11 @Controller
12 public class PalindromeController {
13
14     @ private String getPalindromePage (String kalimat, Model model){
15
16         if (kalimat.isEmpty()){
17             model.addAttribute("msg", "Tolong masukan sebuah kata");
18         }
19         else{
20             final Palindrome palindrome = new Palindrome(kalimat);
21             model.addAttribute("palindrome", palindrome);
22         }
23
24         return "isPalindromePage.html";
25     }
26
27     @GetMapping(value="/is-palindrome")
28     public String isPalindromeRequestParam(
29         @RequestParam(value = "kalimat") String kalimat,
30         Model model){
31
32         return getPalindromePage(kalimat, model);
33     }
34 }
35 }
```

6. Tambahkan file HTML5 bernama `isPalindromePage` sebagai View pada directory `resources/templates`.

Tips: Nama file HTML5 sesuai dengan String yang di return pada fungsi di Controller.

```
<!DOCTYPE html>
<!-- attribute xmlns:th harus ada jika ingin menggunakan
template engine milik Thymeleaf -->
<html lang="en" xmlns:th="http://www.thymeleaf.org">
```

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

```
<head>
  <meta charset="UTF-8">
  <title th:text="'Is Palindrome'"></title>
  <style>
    .card {
      width: 30%;
      display: flex;
      flex-direction: column;
      border: 1px blue solid;
      position: absolute;
      left: 50%;
      top: 50%;
      -webkit-transform: translate(-50%, -50%);
      transform: translate(-50%, -50%);
    }
    .header {
      height: 30%;
      background: blue;
      color: white;
      text-align: center;
    }
    .container {
      padding: 2px 16px;
    }
  </style>
</head>
<body>
  <!-- kerjakan body disini -->
</body>
</html>
```

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

```
1 <!DOCTYPE html>
2 <!-- attribute xmlns:th harus ada jika ingin menggunakan template engine milik Thymeleaf -->
3 <html lang="en" xmlns:th="http://www.thymeleaf.org">
4 <head>
5 <meta charset="UTF-8">
6 <title th:text="Is Palindrome"></title>
7 <style>
8     .card {
9         width: 30%;
10        display: flex;
11        flex-direction: column;
12        border: 1px blue solid;
13        position: absolute;
14        left: 50%;
15        top: 50%;
16        -webkit-transform: translate(-50%, -50%);
17        transform: translate(-50%, -50%);
18    }
19    .header {
20        height: 30%;
21        background: blue;
22        color: white;
23        text-align: center;
24    }
25    .container {
26        padding: 2px 16px;
27    }
28 </style>
```

```
<body>
<div class="content">
<div class="card">
<div class="header">
<p>Program Pengecek Palindrome</p>
</div>
<div class="container">
<div th:if = "${palindrome != null}">
<p>Kata anda adalah: <span th:text="${palindrome.getKata()}"></span></p>
<p><b th:if="${palindrome.isPalindrome()}" th:text="Kata anda merupakan Palindrome"></b></p>
<p><b th:if="${palindrome.isPalindrome()} == false" th:text="Kata anda bukanlah Palindrome"></b></p>
</div>
<div th:unless="${palindrome != null}">
<h3 th:text = "${msg}"></h3>
</div>
</div>
</div>
</body>
```

7. Jalankan App, lalu buka browser dan masukkan url berikut.

[http://localhost:8080/is-palindrome?kalimat=\(\(kata anda\)\)](http://localhost:8080/is-palindrome?kalimat=((kata anda)))
Contoh: <http://localhost:8080/is-palindrome?kalimat=isi>

8. Selamat! Anda telah berhasil menyelesaikan fitur Is Palindrome menggunakan RequestParam ☺ .

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

- Selanjutnya, terdapat cara lain untuk melemparkan suatu data dengan method GET, yaitu dari URI dengan menggunakan **Path Variable**.
- Buatlah fungsi berikut pada controller Anda.

```
@GetMapping(value="/is-palindrome/{kalimat}")  
public String isPalindromeWithPathVariable(  
    @PathVariable(value = "kalimat") String kalimat,  
    Model model){  
    return getPalindromePage(kalimat, model);  
}
```

- Restart App, lalu buka browser dan masukkan url berikut.

[http://localhost:8080/is-palindrome/\(\(kata anda\)\)](http://localhost:8080/is-palindrome/((kata anda)))
Contoh: <http://localhost:8080/is-palindrome/isi>

- Selamat! Anda telah berhasil menyelesaikan fitur Is Palindrome menggunakan PathVariable ☺ .

Pull Request

Pada bagian ini Anda akan belajar bagaimana membuat pull request. Kenapa dinamakan pull request? Pull request sendiri berarti Anda me-request target repository anda untuk menarik perubahan yang telah Anda buat. Dengan adanya pull request, Anda dapat menginformasikan ke kontributor lain perubahan yang Anda push ke branch pada suatu repository. Sehingga, Anda dapat berdiskusi dan review perubahan tersebut dengan kontributor lainnya sebelum melakukan merge.

Pull request ini akan digunakan sebagai pengumpulan tutorial yang sudah dikerjakan, dimana asisten

dosen Anda akan approve pekerjaan Anda setelah selesai review.

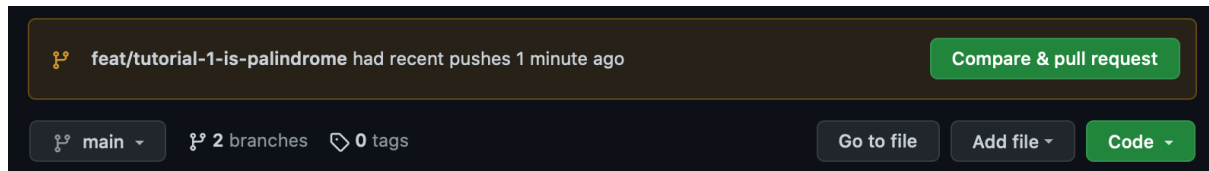
- Setelah Anda selesai mengerjakan fitur is Palindrome, commit fitur is Palindrome milik Anda di branch `feat/tutorial-1-is-palindrome`.
- Push branch `feat/tutorial-1-is-palindrome` ke Github.

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

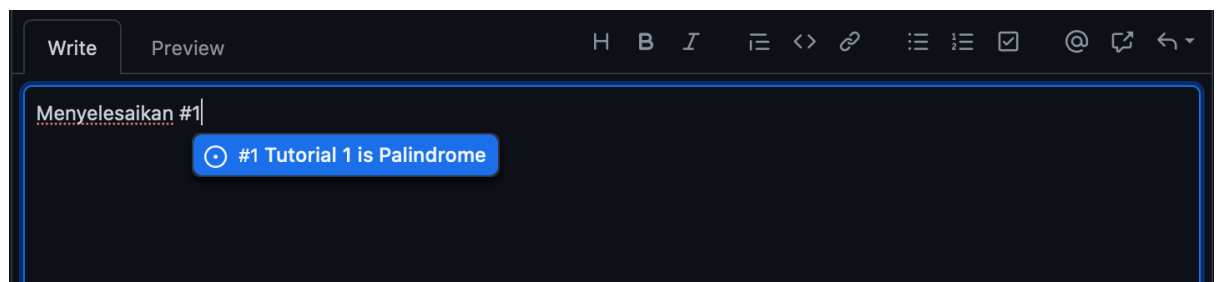
Deadline: 8 September 2021, 23.55 waktu Scele

3. Bukalah Github dan buatlah Pull Request untuk branch `feat/tutorial-1-is-palindrome` tertuju ke `master`. Anda dapat membuat Pull Request dengan `New pull request` atau `Compare & pull request` seperti pada gambar di bawah.

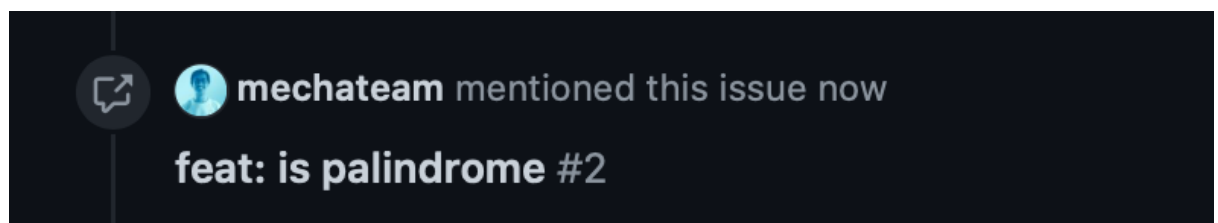


4. Tambahkan asisten dosen mingguan Anda sebagai reviewer.
5. Tambahkan label yang berkaitan dan assign Anda sebagai yang mengerjakan.
6. Mention issue yang berkaitan dengan pull request ini, yaitu issue yang sebelumnya Anda buat.

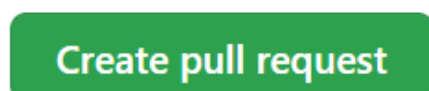
Ketik `#` dan akan keluar pilihan issue.



Jika Anda kembali ke issue yang tadi Anda buat akan tampil activity baru. Tutup issue ketika pull request telah masuk ke master.



7. Kemudian, `Create pull request`.



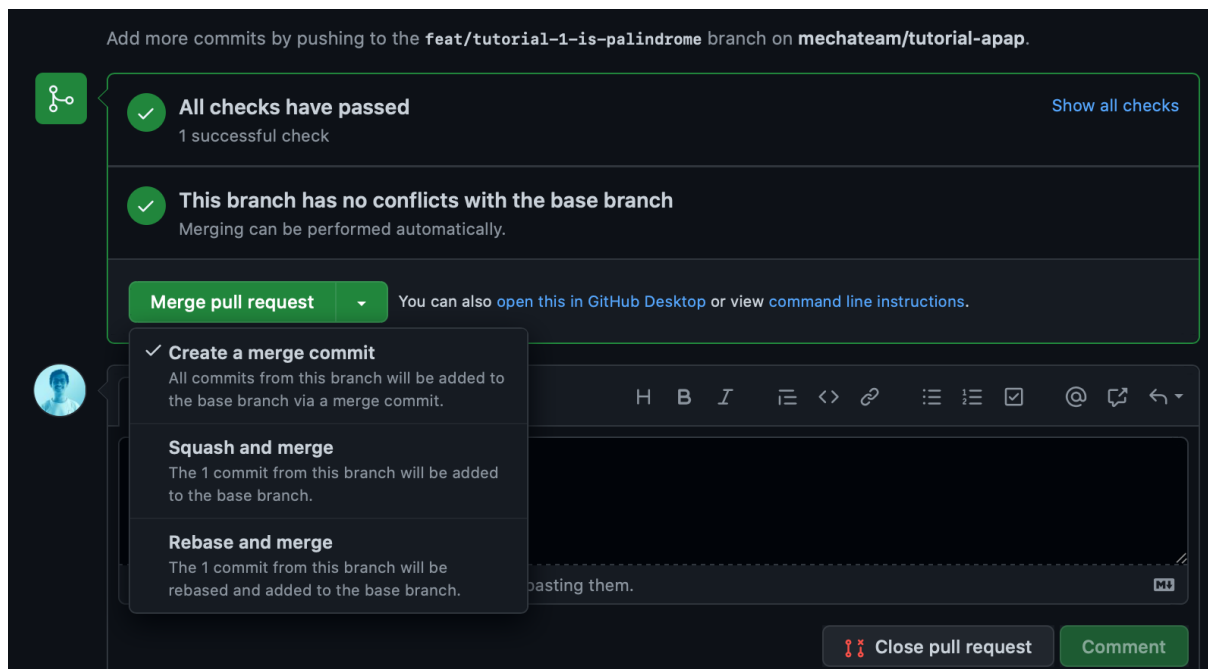
8. *Explore* apa saja yang bisa dilakukan di halaman pull request yang sudah Anda buat.

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

9. Setelah asisten Anda Approve perubahan yang Anda lakukan (otomatis dianggap disetujui pada hari Rabu di tutorial selanjutnya), Anda akan melakukan proses merging dengan operasi Merge Squash ke Master di Github. Anda dapat melakukannya dengan menggunakan pilihan 'Squash and merge' seperti gambar dibawah.



atau ketika memerlukan secara manual, Anda dapat menggunakan perintah berikut di branch master pada terminal Anda.

```
# contoh dan jangan lakukan sekarang, semua melalui Github Pull Request  
git merge --squash <nama branch>
```

10. Jika Anda belum selesai hingga akhir deadline tutorial, maka di hari Selasa sebelum tutorial selanjutnya akan diterbitkan solusi yang dapat dipelajari. Anda boleh memperbaikinya agar dapat mengerjakan tutorial selanjutnya.
11. Jika ada perbaikan dan aktivitas commit yang terjadi di Pull Request setelah deadline, maka **Pull Request berlaku nilai keterlambatan.**
12. Perbaikan dan aktivitas commit yang terjadi di Pull Request pada waktu setelah diterbitkannya solusi tidak dianggap sebagai nilai atau keterlambatan.

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

CATATAN

Tambahkan *template* ini di README.md. *Template* ini dapat digunakan sebagai contoh untuk tutorial-tutorial selanjutnya.

```
---
## Tutorial 1
### What I have learned today
(Masukkan pertanyaan yang diikuti jawaban di setiap nomor, contoh seperti dibawah. Anda juga boleh menambahkan catatan apapun di bagian ini)
### Github
1. Apa itu Issue Tracker? Apa saja masalah yang dapat diselesaikan dengan Issue Tracker?
   (Tuliskan jawabanmu)
2. Apa perbedaan dari git merge dan git merge --squash?
3. Apa keunggulan menggunakan Version Control System seperti Git dalam pengembangan suatu aplikasi?
### Spring
4. Apa itu library & dependency?
5. Apa itu Maven? Mengapa kita menggunakan Maven? Apakah ada alternatif dari Maven?
6. Selain untuk pengembangan web, apa saja yang bisa dikembangkan dengan Spring framework?
7. Apa perbedaan dari @RequestParam dan @PathVariable? Kapan sebaiknya menggunakan @RequestParam atau @PathVariable?

### What I did not understand
(tuliskan apa saja yang kurang Anda mengerti, Anda dapat men-_check_ apabila Anda sudah mengerti dikemudian hari, dan tambahkan tulisan yang membuat Anda mengerti)
- [ ] Kenapa saya harus belajar APAP?
- [x] Kenapa?
    Karena ...

(Anda dapat membuat tampilan code dalam README.md menjadi lebih baik. Cari tahu lebih dalam tentang penulisan README.md di GitHub pada link [berikut](https://help.github.com/en/articles/basic-writing-and-formatting-syntax))
```

GitHub

1. [Note] Apa itu Issue Tracker? Apa saja masalah yang dapat diselesaikan dengan Issue Tracker?
2. [Note] Apa perbedaan dari git merge dan git merge --squash?

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

3. [Note] Apa keunggulan menggunakan Version Control System seperti Git dalam pengembangan suatu aplikasi?

Spring

1. [Note] Apa itu library & dependency?
2. [Note] Mengapa kita menggunakan Maven? Apakah ada alternatif dari Maven?
3. [Note] Selain untuk pengembangan *web*, apa saja yang bisa dikembangkan dengan Spring *framework*?
4. [Note] Apa perbedaan dari `@RequestParam` dan `@PathVariable`? Kapan sebaiknya menggunakan `@RequestParam` atau `@PathVariable`?

GLOSSARY

Model-View-Controller

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan Semester Ganjil Tahun Ajaran 2021/2022 **Deadline: 8 September 2021, 23.55 waktu Scele**

Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

Dependency Injection

Dependency injection is a technique whereby one object (or static method) supplies the dependencies of another object. A dependency is an object that can be used (a service).

Why should we use dependency injection?

Let's say we have a car class which contains various objects such as wheels, engine, etc.

Here the car class is responsible for creating all the dependency objects. Now what if we decide to ditch **MRFWheels** in the future and want to use **Yokohama** Wheels?

We will need to recreate the car object with a new Yokohama dependency. But when using dependency injection (DI), we can change the Wheels at runtime (because dependencies can be injected at runtime rather than at compile time).

You can think of DI as the middleman in our code who does all the work of creating the preferred wheels object and providing it to the Car class.

It makes out Car class independent from creating the objects of Wheels, Battery, etc.

For more information about Dependency Injection, open this [link](#).

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan

Semester Ganjil Tahun Ajaran 2021/2022

Deadline: 8 September 2021, 23.55 waktu Scele

Git

Git is a Version Control. Version Control is a system that records changes to a file or set of files over time so that you can recall specific versions later. For more information about Git, you can open this [link](#).

Spring Boot

Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade Spring application that you can just run. You can get started with minimum configurations without the need for an entire Spring configuration setup.

Advantages

Spring Boot offers the following advantages to its developers:

- Easy to understand and develop Spring applications
- Increases productivity
- Reduces development time

Goals

Spring Boot is designed with the following goals:

- To avoid complex XML configuration in Spring
- To develop a production ready Spring application in an easier way
- To reduce the development time and run the application independently
- Offer an easier way of getting started with the application

Tutorial 1 Arsitektur & Pemrograman Aplikasi Perusahaan Semester Ganjil Tahun Ajaran 2021/2022 **Deadline: 8 September 2021, 23.55 waktu Scele**

References

Git:

<https://git-scm.com/doc>

<https://git-scm.com/book/en/v2>

<https://www.atlassian.com/git>

Spring Boot

https://www.tutorialspoint.com/spring_boot/index.htm

<https://www.baeldung.com/spring-boot>