

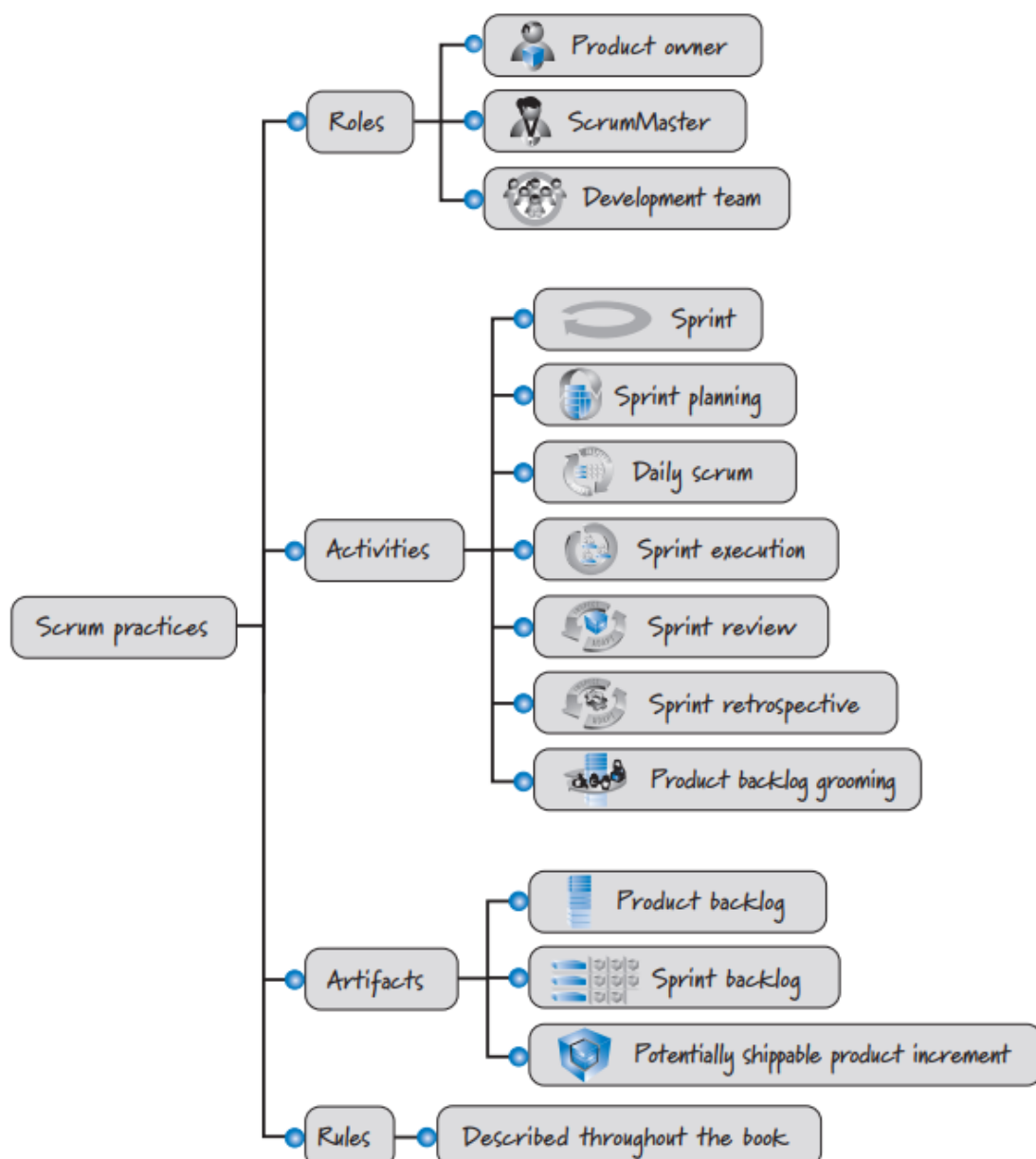
## MODUL 2

### ISSUES MANAGEMENT

Pada Modul 2 ini akan dijelaskan tentang metode pengembangan Scrum, pembuatan kanban pada GitHub, pembuatan *issue*, *commit* dengan *issue*, dan melakukan *resolve (close) issue* pada GitHub.

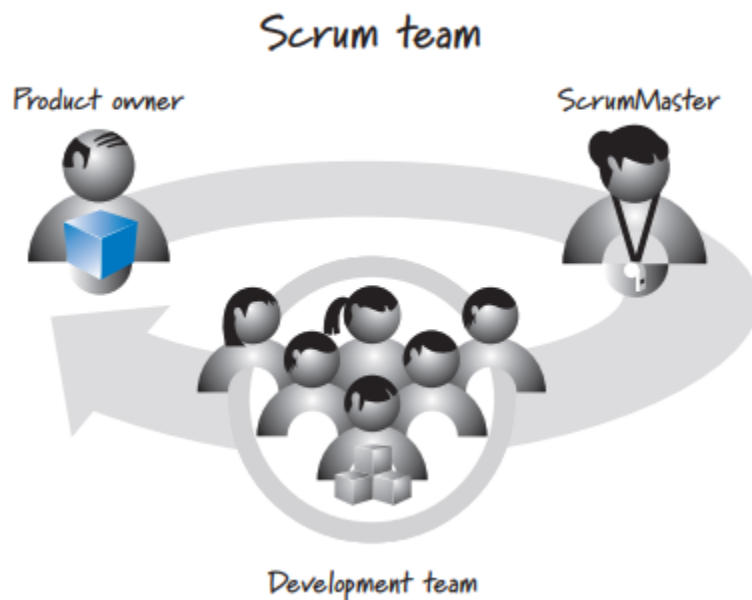
#### 1. Scrum

Scrum adalah metode pengembangan perangkat lunak yang menggunakan pendekatan atau prinsip - prinsip *Agile*. Dengan menerapkan pendekatan *agile*, kita dapat membuat sebuah *product backlog* atau daftar fitur yang diprioritaskan dalam mengembangkan sebuah produk, melalui *product backlog* ini kita dapat mengerjakan *item* yang paling penting atau prioritas tertinggi terlebih dahulu. Sehingga, jika kita kehabisan waktu, pekerjaan yang belum diselesaikan adalah pekerjaan dengan prioritas yang lebih rendah, karena prioritas utama telah dikerjakan lebih awal. Praktik metode scrum diwujudkan dalam peran, aktivitas, artefak, dan aturan terkaitnya, praktik tersebut dapat dilihat pada gambar berikut.



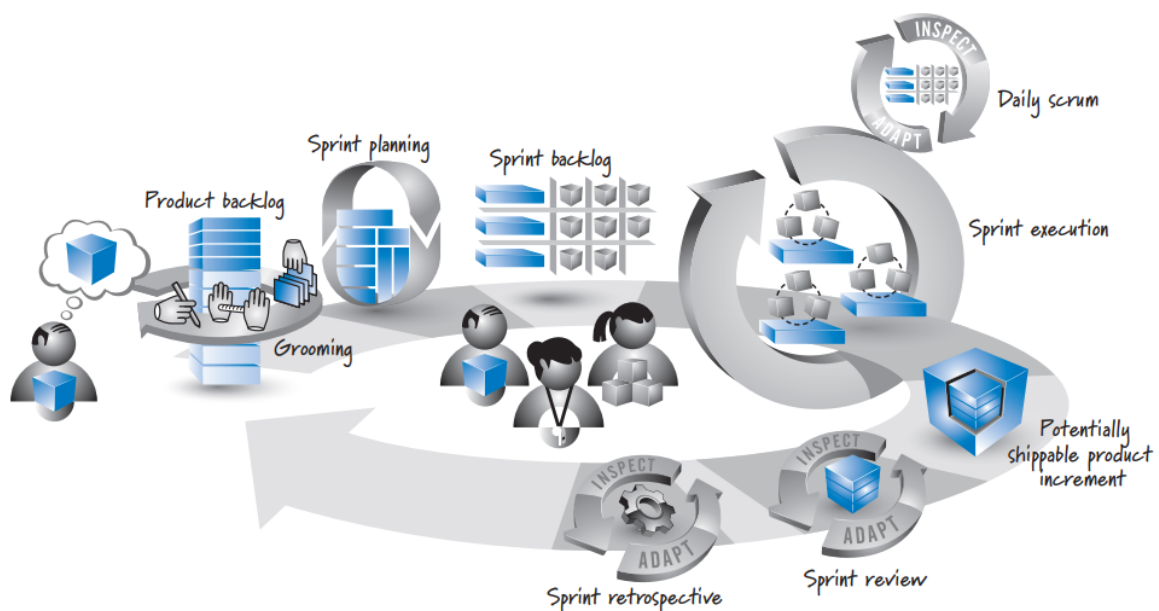
## 1.1 Peran Utama pada Metode Scrum

Berdasarkan gambar di atas, dapat dilihat bahwa pada metode Scrum terdapat tiga peran utama yaitu: *Product Owner* (Pemilik produk), *Scrum Master*, dan *Development Team* (Tim pengembang).



*Product Owner* bertanggung jawab atas produk yang dikembangkan beserta dengan urutannya. *Scrum Master* bertanggung jawab untuk memandu tim dalam membuat dan mengikuti prosesnya sendiri berdasarkan kerangka kerja Scrum. *Development team* bertanggung jawab untuk menentukan bagaimana menyampaikan apa yang diminta oleh pemilik produk.

## 1.2 Aktivitas dan Artefak pada Metode Scrum



Berdasarkan gambar di atas akan dijelaskan rangkaian aktivitas dan artefak yang dihasilkan pada metode Scrum mulai dari sisi kiri gambar dan bekerja searah jarum jam di sekitar panah perulangan utama (**sprint**).

1. Pemilik produk (*product owner*) memiliki visi tentang apa yang ingin dibuat (kubus besar) atau disebut ***grooming***, kubus itu dipecah menjadi sekumpulan fitur yang dikumpulkan ke dalam daftar prioritas yang disebut ***product backlog***.

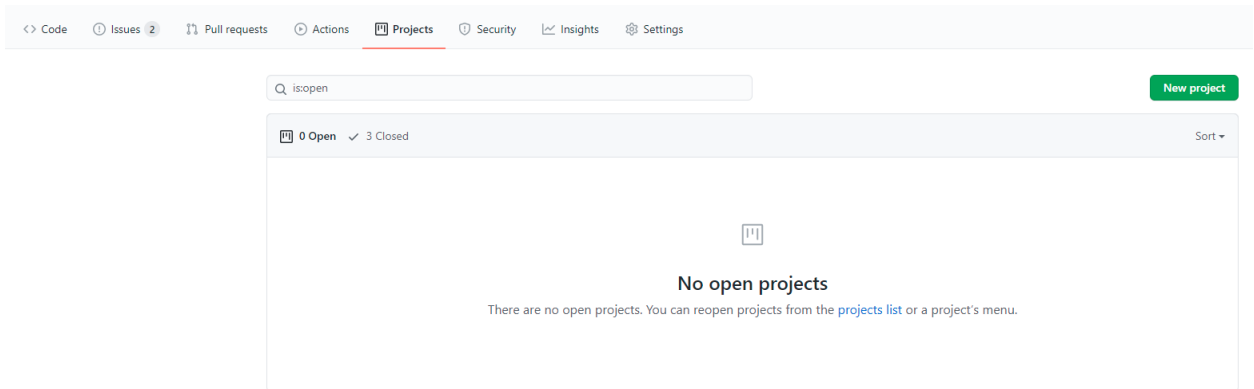
- 2. Sprint dimulai dengan perencanaan sprint (*sprint planning*), yang mencakup pekerjaan pengembangan selama sprint (disebut *sprint execution*), dan diakhiri dengan peninjauan (*sprint review*) dan retrospektif (*sprint retrospective*).
- 3. Untuk mendapatkan keyakinan bahwa tim pengembangan telah membuat komitmen yang sesuai, anggota tim membuat *backlog* kedua selama perencanaan sprint (*sprint planning*), yang disebut *sprint backlog*. *Sprint backlog* menjelaskan sekumpulan tugas terperinci, bagaimana tim berencana untuk merancang, membangun, mengintegrasikan, dan menguji subset fitur yang dipilih dari *product backlog* selama sprint tersebut.
- 4. Berikutnya adalah eksekusi sprint (*sprint execution*), dimana tim pengembangan melakukan tugas-tugas yang diperlukan untuk mewujudkan fitur yang dipilih. Setiap hari selama pelaksanaan *sprint*, anggota tim membantu mengatur alur kerja dengan melakukan sinkronisasi, inspeksi, dan kegiatan perencanaan adaptif yang dikenal dengan *daily scrum*. Di akhir eksekusi *sprint*, tim telah menghasilkan peningkatan produk yang berpotensi dapat dirilis yang mewakili beberapa visi pemilik produk (*product owner*).
- 5. Tim Scrum menyelesaikan *sprint* dengan melakukan dua aktivitas inspeksi dan adaptasi. Yang pertama, disebut *sprint review*, para pemangku kepentingan (*stakeholder*) dan tim Scrum memeriksa produk yang sedang dibangun. Yang kedua, disebut *sprint retrospective*, tim Scrum memeriksa proses Scrum yang digunakan untuk membuat produk. Hasil dari aktivitas ini mungkin adaptasi yang akan masuk ke dalam *product backlog* atau dimasukkan sebagai bagian dari proses pengembangan tim.

Pada titik ini, siklus *sprint* Scrum berulang, dimulai lagi dengan tim pengembangan menentukan set item backlog produk terpenting berikutnya yang dapat diselesaikannya. Setelah jumlah sprint yang sesuai diselesaikan, visi pemilik produk akan terwujud dan solusinya dapat dirilis.

## 2. Pembuatan Kanban pada GitHub

GitHub menyediakan *board* yang secara otomatis terintegrasi dengan isu yang dibuat. Berikut langkah-langkah pembuatan kanban melalui GitHub.

- a. Pertama, masuk ke dalam repository yang ingin dibuatkan *board*-nya. Kemudian pilih tab Projects.



- b. Selanjutnya memilih tombol New Project. Teman-teman dapat menuliskan judul dari project dan deskripsi project.

ProjectsSecurityInsightsSettings

Create a new project  
Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

Project board name

Nama Project ditulis disini

Description (optional)

Deskripsi dari project

Project template

Save yourself time with a pre-configured project board template.

Template: None

Create project

- c. Kemudian teman-teman dapat menentukan template dari projectnya. GitHub menyediakan kanban yang terintegrasi dengan isu yang akan dibuat. Kita pilih Automated Kanban untuk saat ini.

Project template

Save yourself time with a pre-configured project board template.

Template: None

Templates

None

Start from scratch with a completely blank project board. You can add columns and configure automation settings yourself.

Basic kanban

Basic kanban-style board with columns for To do, In progress and Done.

Automated kanban

Kanban-style board with built-in triggers to automatically move issues and pull requests across To do, In progress and Done columns.

Automated kanban with reviews

Everything included in the Automated kanban template with additional triggers for pull request reviews.

Bug triage

Triage and prioritize bugs with columns for To

d. Setelah memilih template, kita memilih tombol Create Project

Create a new project

Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

Project board name

Nama Project ditulis disini

Description (optional)

Deskripsi dari project

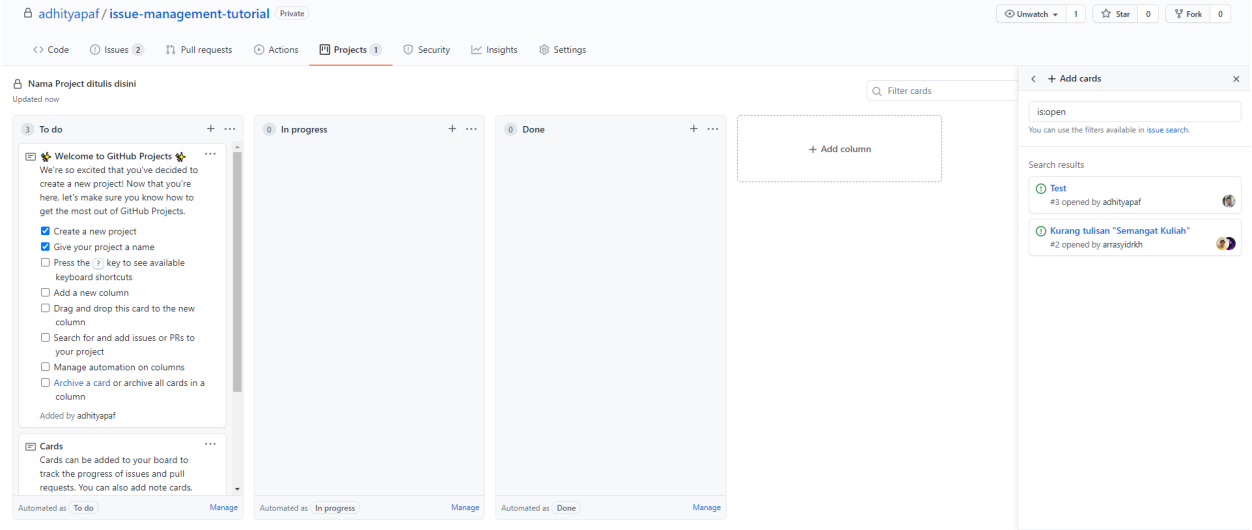
Project template

Save yourself time with a pre-configured project board template.

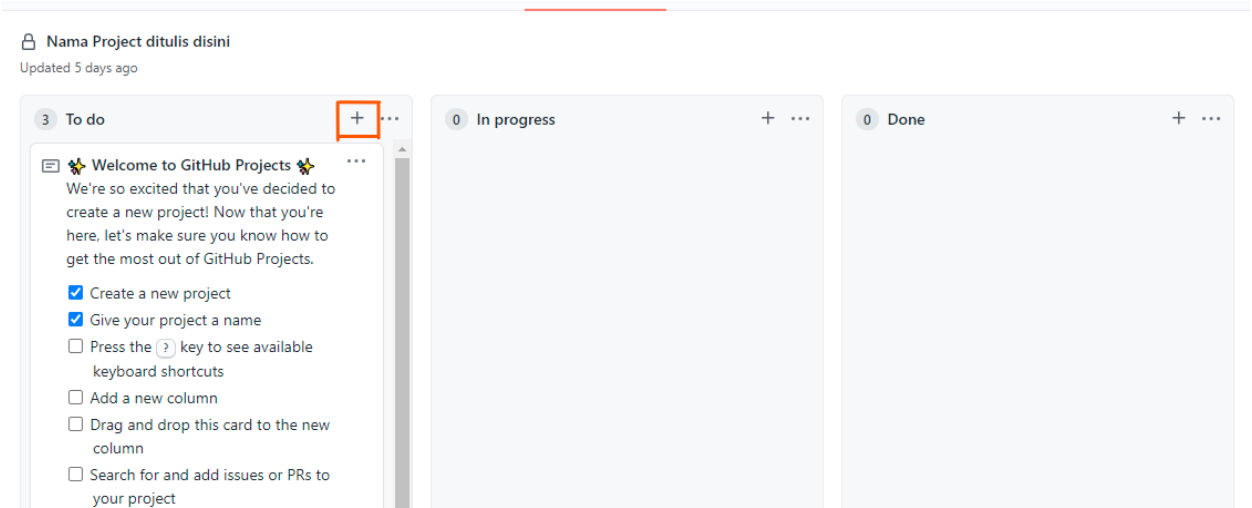
Template: Automated kanban

Create project

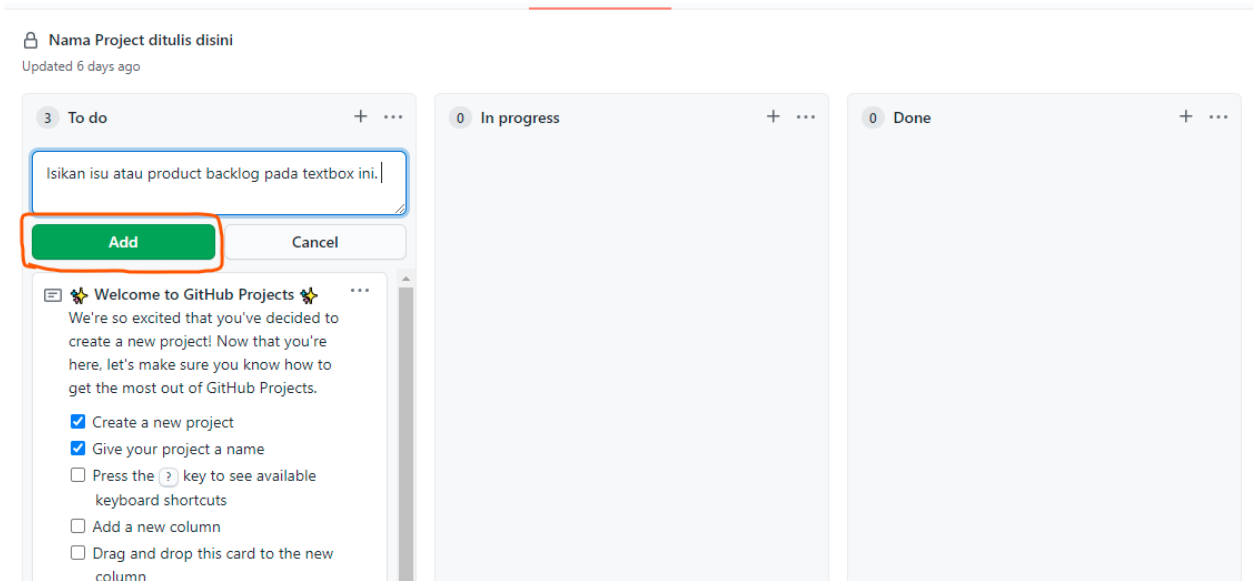
e. Tampilan project yang dibuat dapat dilihat seperti gambar berikut.



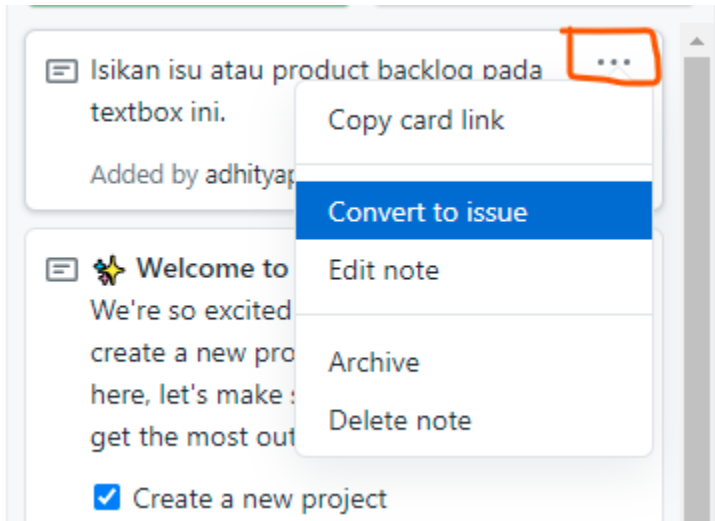
f. Teman-teman dapat langsung membuat isu melalui Card. Caranya, teman-teman menekan tanda + pada kolom To-Do.



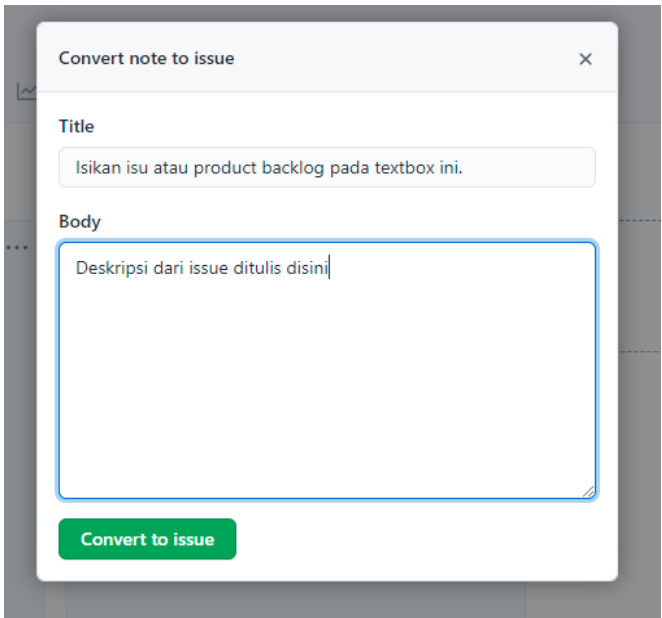
- g. Kemudian menuliskan isu atau product backlog item. Selanjutnya menekan tombol Add berwarna hijau.



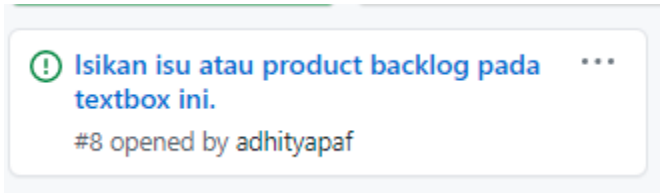
- h. Untuk mengubah ke dalam issue, teman-teman dapat menekan titik tiga pada tiap card,. Kemudian memilih Convert to Issue.



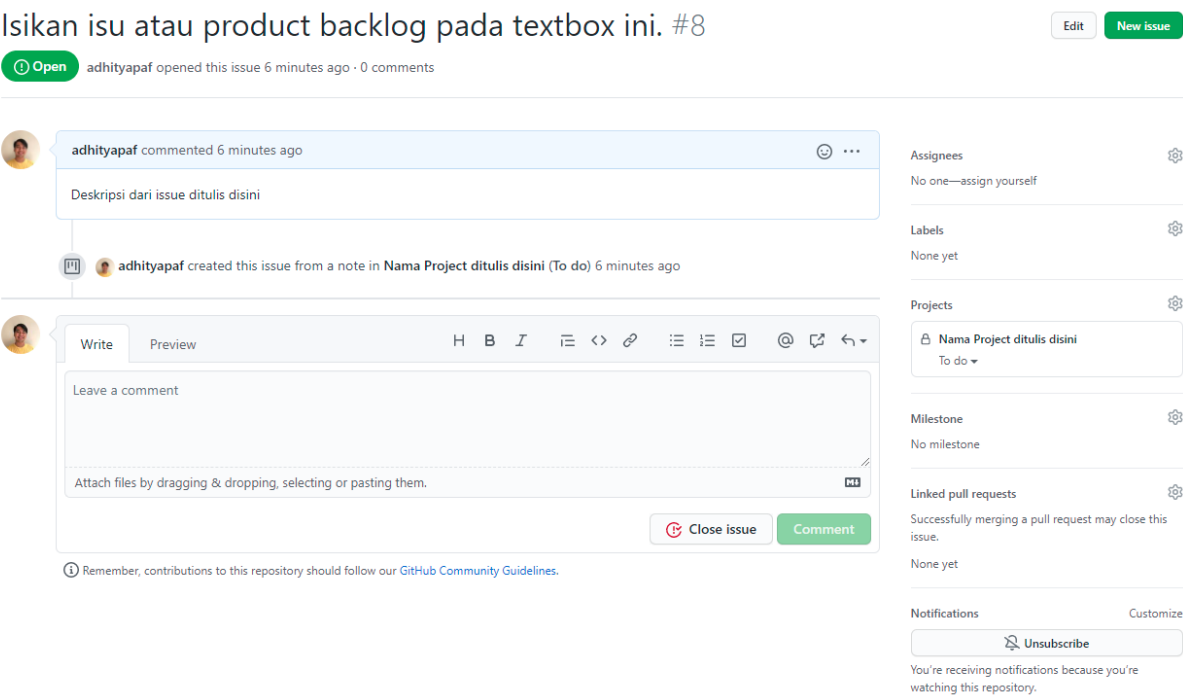
- i. Kemudian, teman-teman dapat menuliskan judul dari issue dan deskripsi issue. Setelah itu, teman-teman dapat menekan tombol Convert to Issue berwarna hijau.



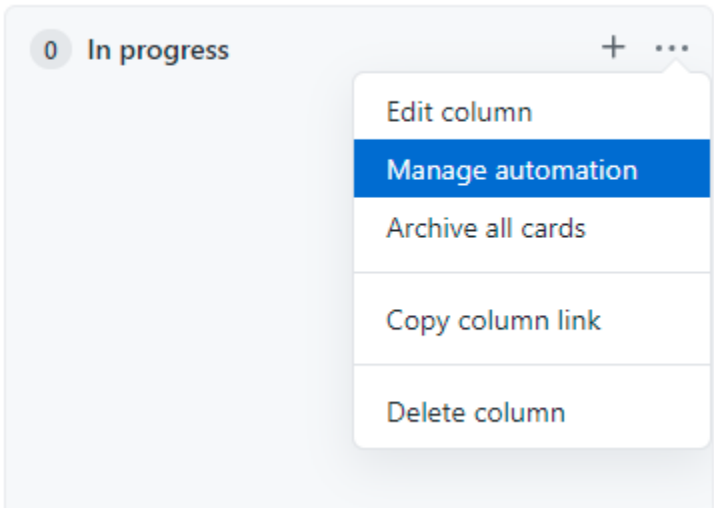
- j. Tampilan card akan berubah ketika card tersebut sudah di jadikan ke issue.



- k. Berikut adalah tampilan ketika teman-teman mengakses langsung dari tab Issue.



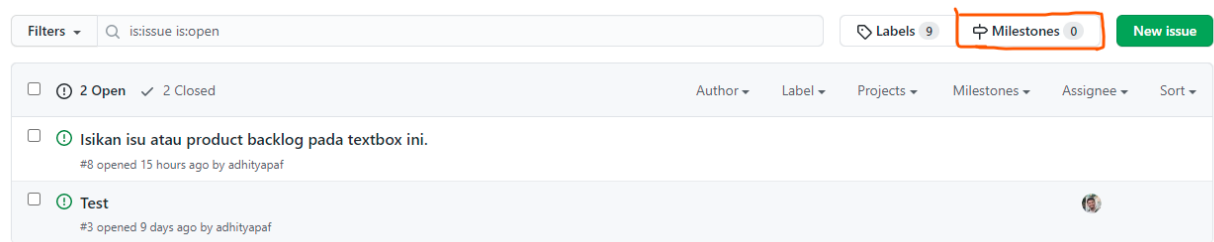
- l. Karena tadi kita membuat automatic kanban, isu yang telah di tutup, akan otomatis pindah ke kolom Done. Ketika isu kembali dibuka, card dari isu tersebut akan pindah ke kolom In Progress. Teman-teman dapat mengatur bagaimana otomatisasi dari masing-masing kolom dengan menekan tombol titik tiga pada tiap kolom dan menekan tombol Manage Automation.



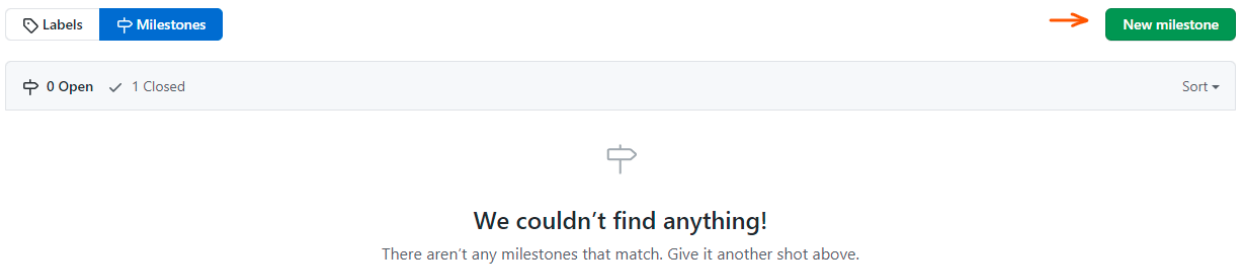
3. Pembuatan issue

Berikut adalah langkah-langkah dalam pembuatan issue pada GitHub.

- a. Jika teman-teman belum membuat milestone untuk Sprint, teman-teman dapat menekan tombol Milestones di sebelah kiri tombol New Issues.



- b. Kemudian menekan tombol New Milestones.



- c. Selanjutnya, mengisi judul dari milestone, tanggal berakhirnya milestone (bersifat opsional), dan deskripsi dari milestone. Jika sudah, teman-teman dapat menekan tombol Create Milestone berwarna hijau.

**New milestone**  
Create a new milestone to help organize your issues and pull requests. Learn more about [milestones and issues](#).

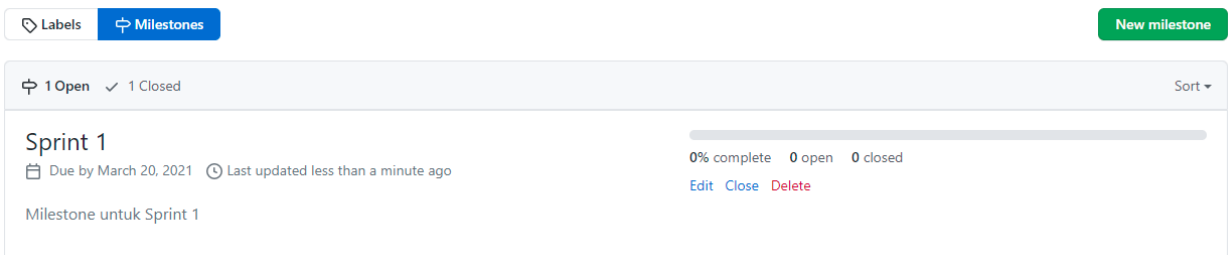
**Title**  
Sprint 1

**Due date (optional)**  
03/20/2021

**Description**  
Milestone untuk Sprint 1

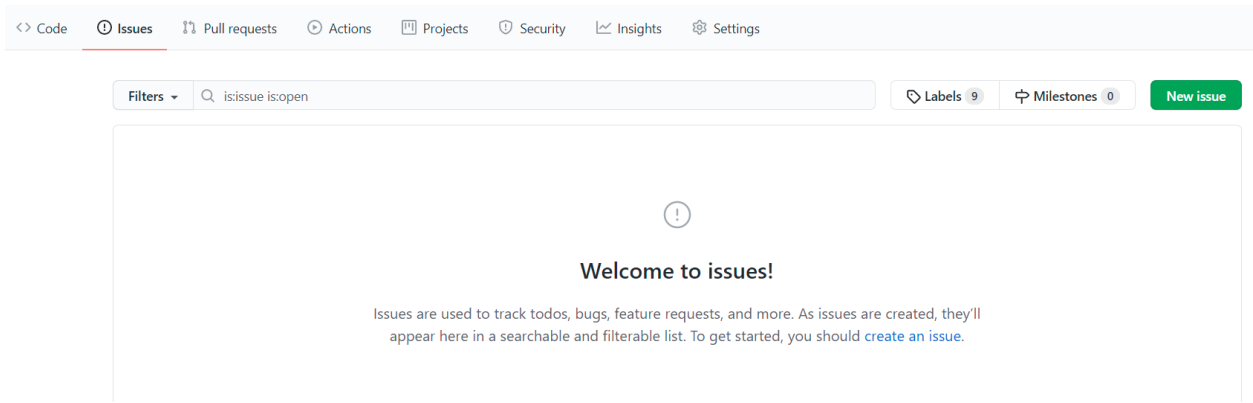
**Create milestone**

- d. Berikut adalah tampilan ketika milestone berhasil dibuat.





- e. Untuk membuat issue, teman-teman dapat menekan tab Issue pada repository teman-teman dan menekan tombol New Issue yang berwarna hijau seperti pada gambar.

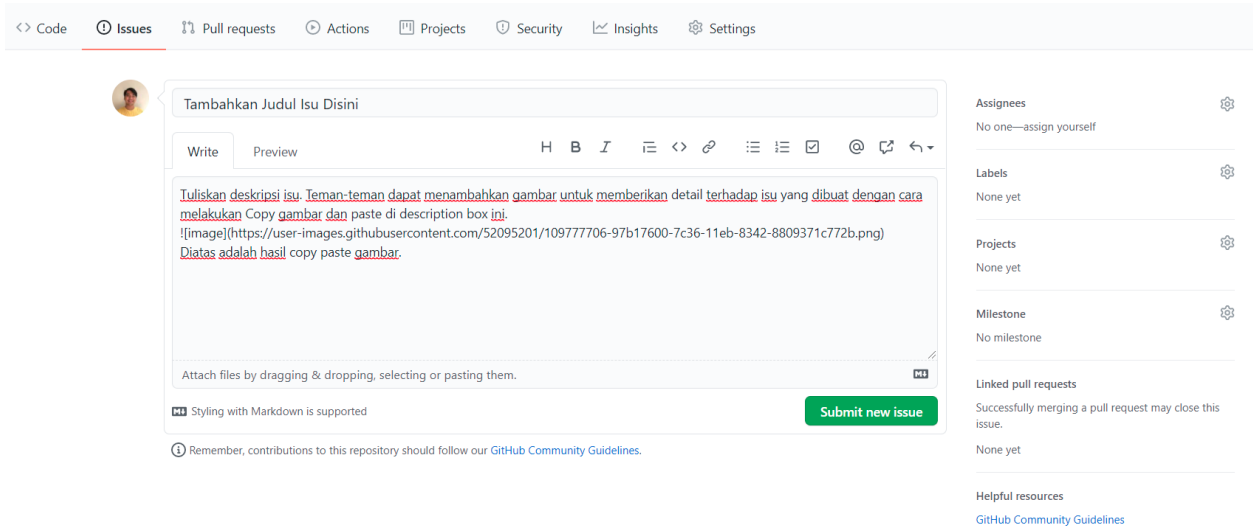


- f. Kemudian, teman-teman dapat menambahkan judul dan deskripsi isu. Jika isu tersebut merupakan PBI, judul isu bisa menggunakan format seperti berikut.

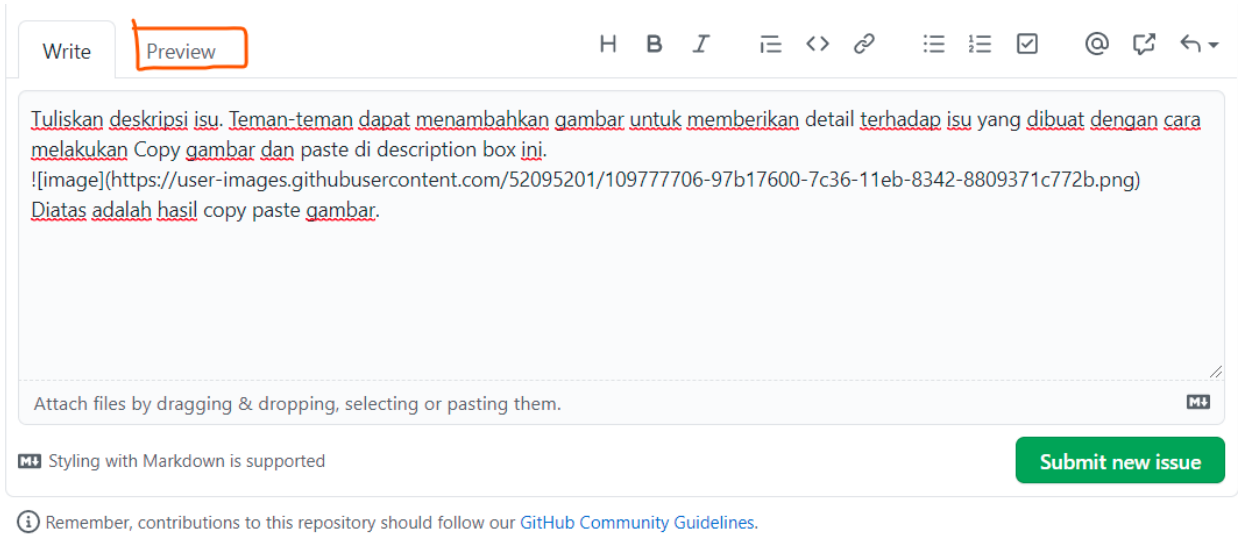
[SX - PBIX]

S = Sprint | PBI = Product Backlog Items | X = Angka

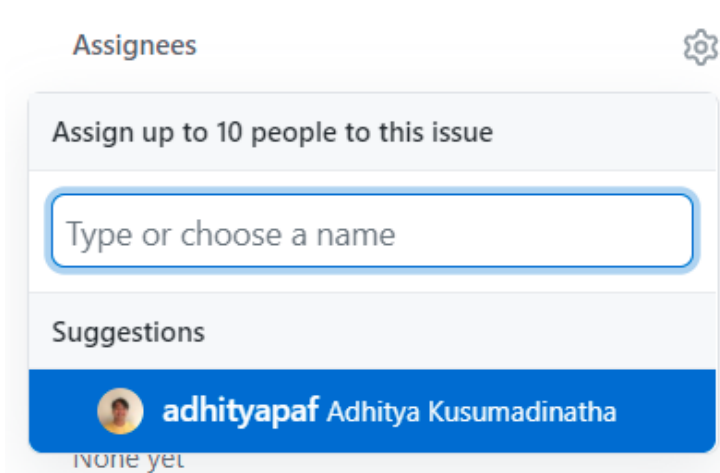
Contoh : Sprint 1 dengan nomor PBI 2 → [S1 - PBI2]



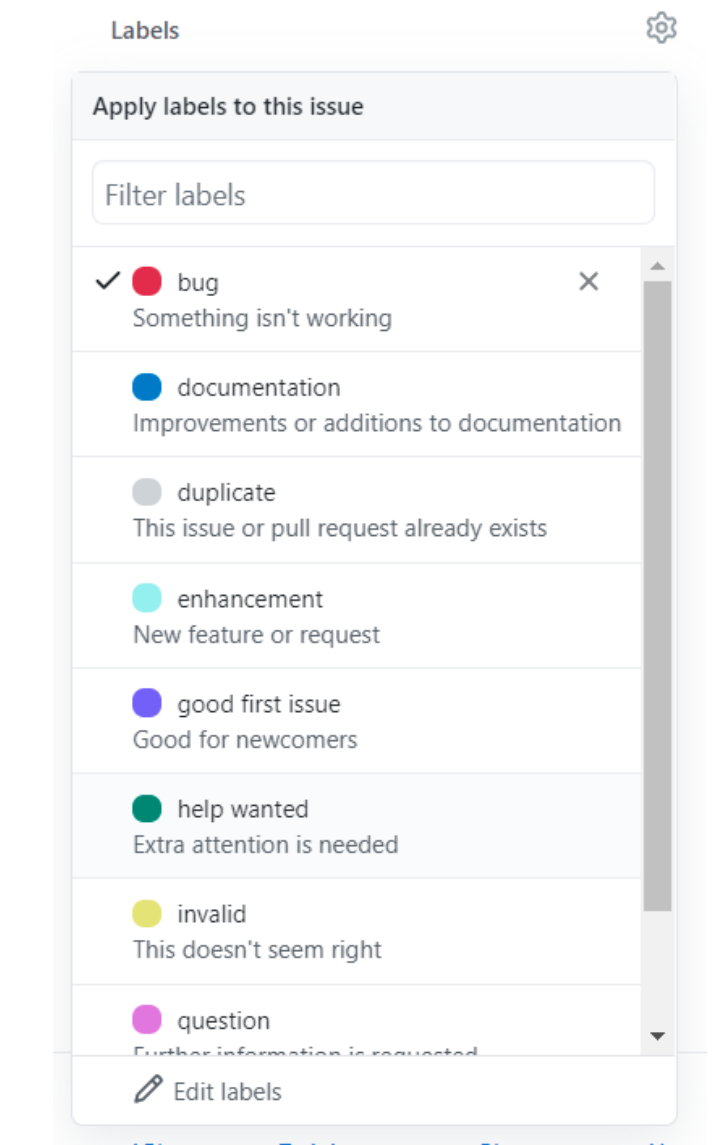
- g. Untuk melihat bagaimana tampilan isu setelah di submit, teman-teman dapat menekan tab Preview di sebelah kanan tab Write.



- h. Kemudian, dalam pembuatan isu, teman-teman dapat menentukan isu tersebut dilakukan oleh siapa. Cara menentukan siapa yang bertanggung jawab terhadap isu tersebut dengan menekan text “Assignees” dan memilih orang.

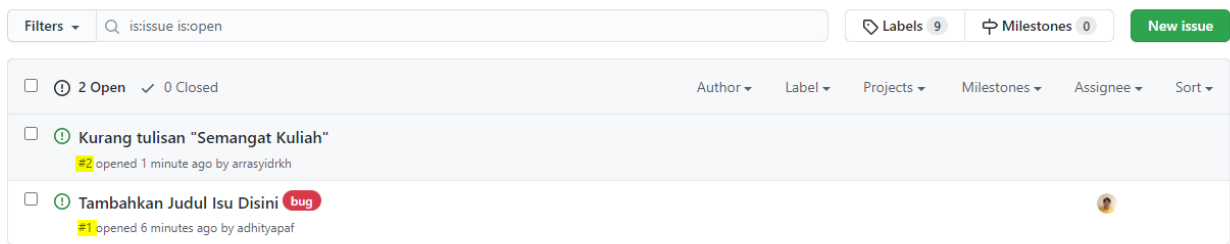


- i. Untuk mengorganisir isu agar lebih rapi, teman-teman dapat menambahkan label isu. Label ini menandai bahwa isu tersebut harus ditangani seperti apa. Secara umum, terdapat label “Bug” untuk sesuatu yang tidak berfungsi, “Duplicate” untuk isu yang sudah ada tetapi dibuat lagi, “Enhancement” untuk fitur baru, dan lain-lain. Penentuan label dapat dilakukan dengan menekan text “Labels” kemudian memilih tipe label seperti pada gambar dibawah.

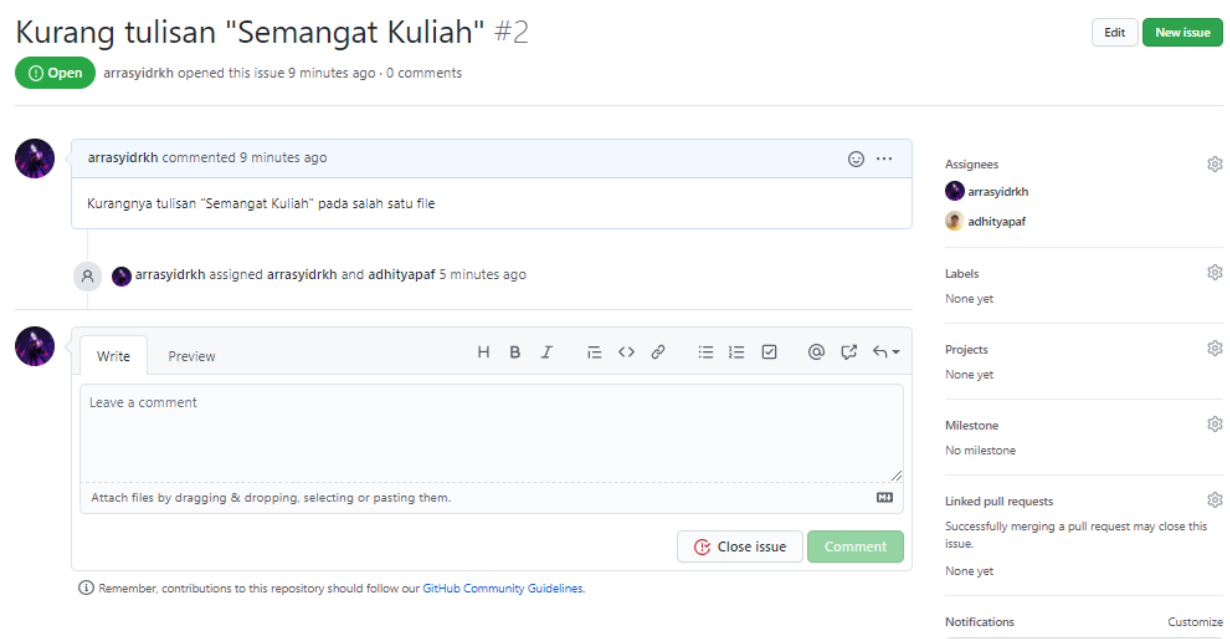


4. Commit dengan issue

Kita juga dapat melakukan commit dengan isu tertentu sesuai dengan nomor dari isu yang sudah dibuat. Pada gambar dibawah ini, akan terlihat nomor isu yang diawali tanda pagar (#) dan diikuti oleh nomor dari isu tersebut.



Dalam tampilan tersebut dapat dijelaskan bahwa ada 2 isu yang berstatus “Open” yang berarti bahwa isu tersebut belum dikerjakan atau diperbaiki sehingga orang dapat melakukan perbaikan tersebut sesuai dengan Assignee yang telah ditentukan saat membuat isu. Setelah mengklik salah satu isu, maka tampilan yang akan muncul adalah sebagai berikut.

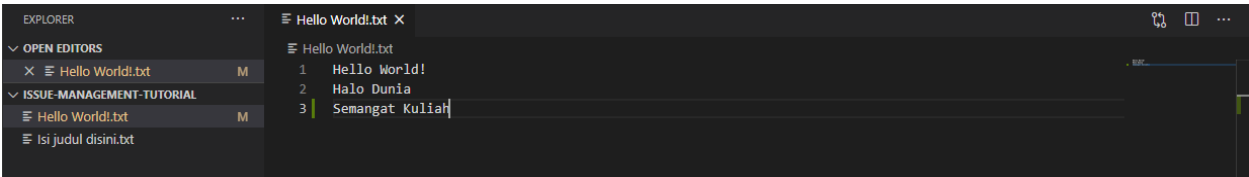


Selanjutnya, kita dapat melakukan commit dengan mereferensikan isu terkait yang ada dalam daftar isu tersebut, berikut adalah cara untuk melakukan commit dengan isu di dalam github.

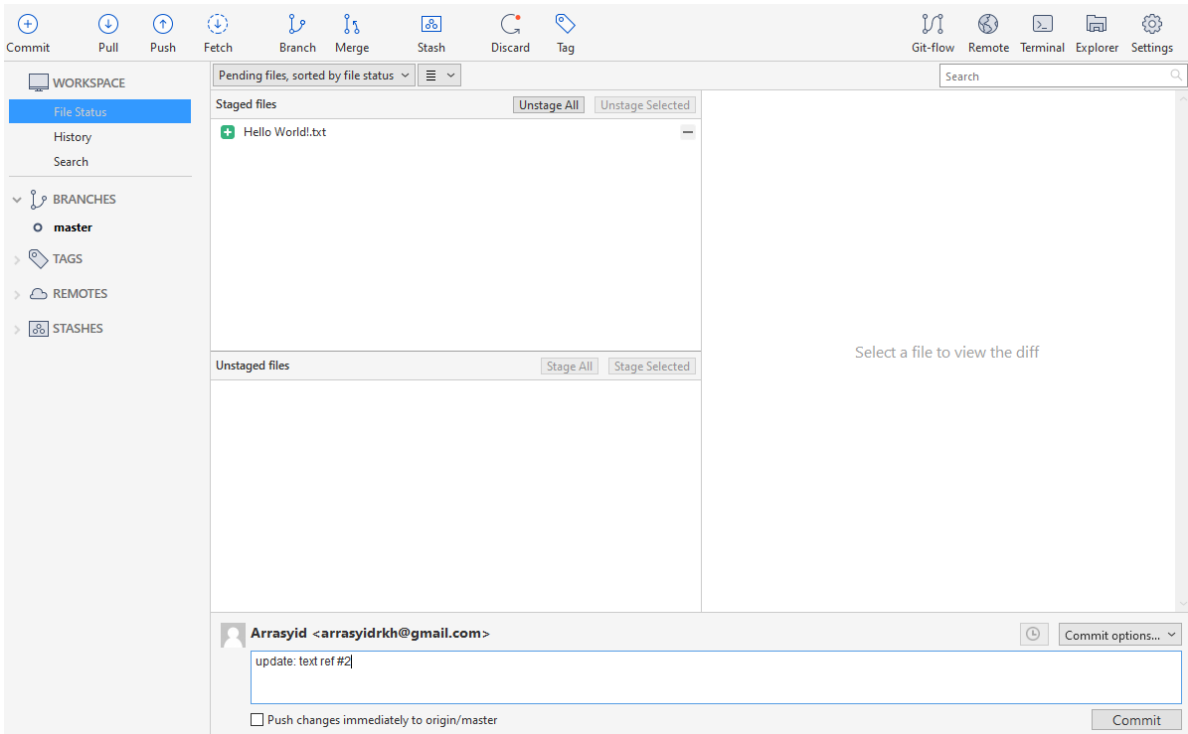
- a. Memastikan bahwa repository telah di clone di local

Name	Date modified	Type	Size
.git	03/03/2021 16:29	File folder	
Hello World!.txt	03/03/2021 16:31	Text Document	1 KB
Isi judul disini.txt	03/03/2021 16:29	Text Document	1 KB

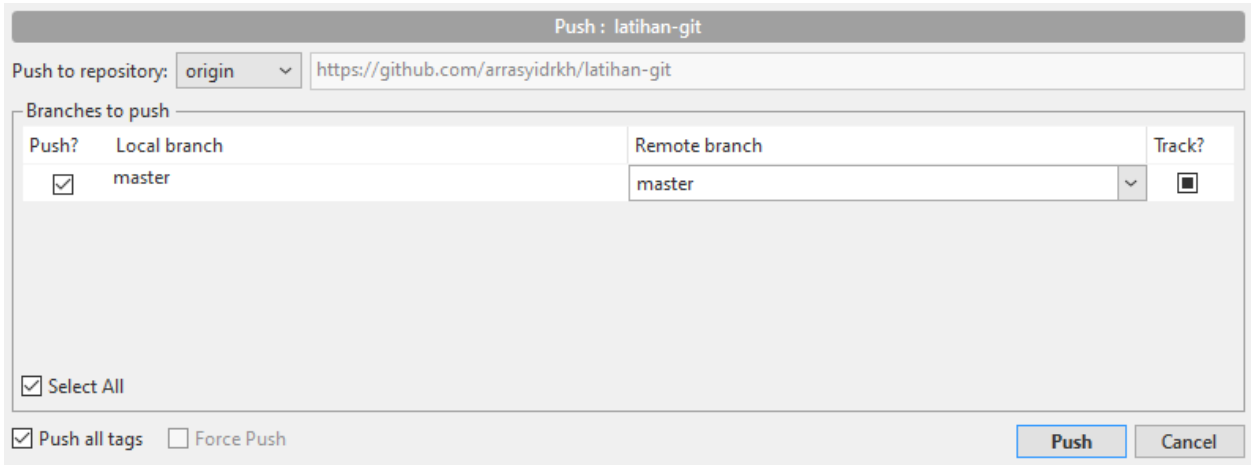
- b. Lakukan perbaikan sesuai dengan isu terkait. Contohnya adalah menambahkan kalimat “Semangat Kuliah” ke dalam file Hello World!.txt



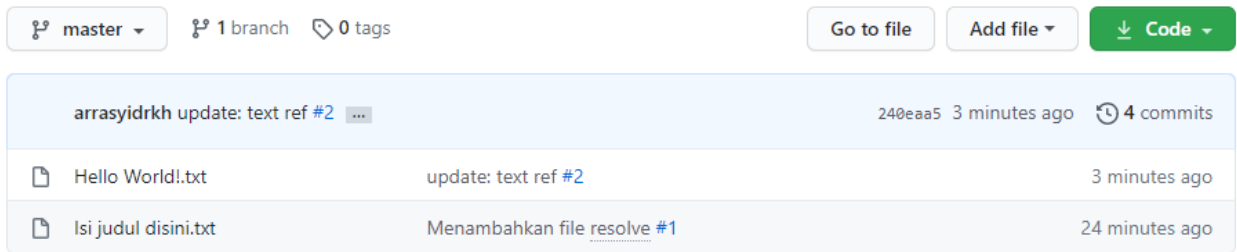
- c. Memastikan bahwa perubahan tersebut sudah dilakukan dan melakukan commit dengan ada kata-kata “ref #2” (ref sebagai reference dan #2 sebagai isu nomor 2) dan lakukan commit dengan mengklik “Commit”.



- d. Setelah itu melakukan push dengan mengklik “Push” untuk melakukan push ke master




- e. Melihat bahwa commit berhasil dan telah di push




- f. Pada tampilan isu tersebut, terdapat deskripsi bahwa isu tersebut diperbaiki pada commit yang sudah dilakukan

## Kurang tulisan "Semangat Kuliah" #2


 **Open**

arraysidrkh opened this issue 1 hour ago · 0 comments




arraysidrkh commented 1 hour ago


Kurangnya tulisan "Semangat Kuliah" pada salah satu file



arraysidrkh assigned arraysidrkh and adhityapaf 43 minutes ago



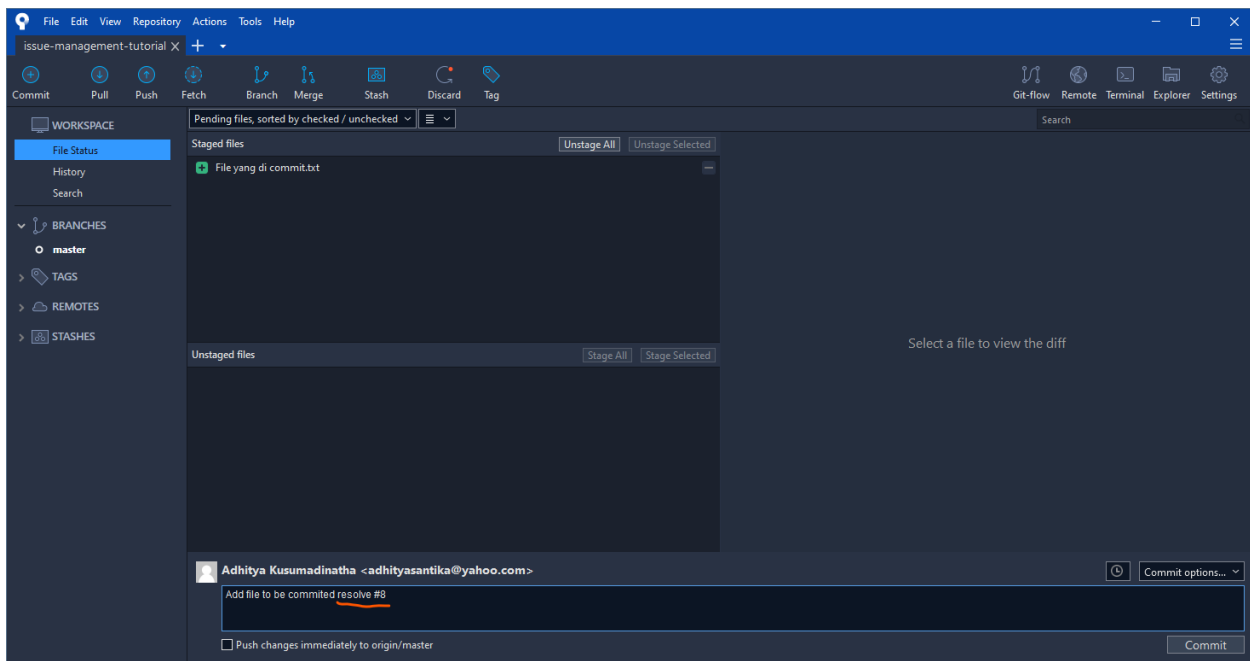
arraysidrkh pushed a commit that referenced this issue 2 minutes ago

 update: text ref #2

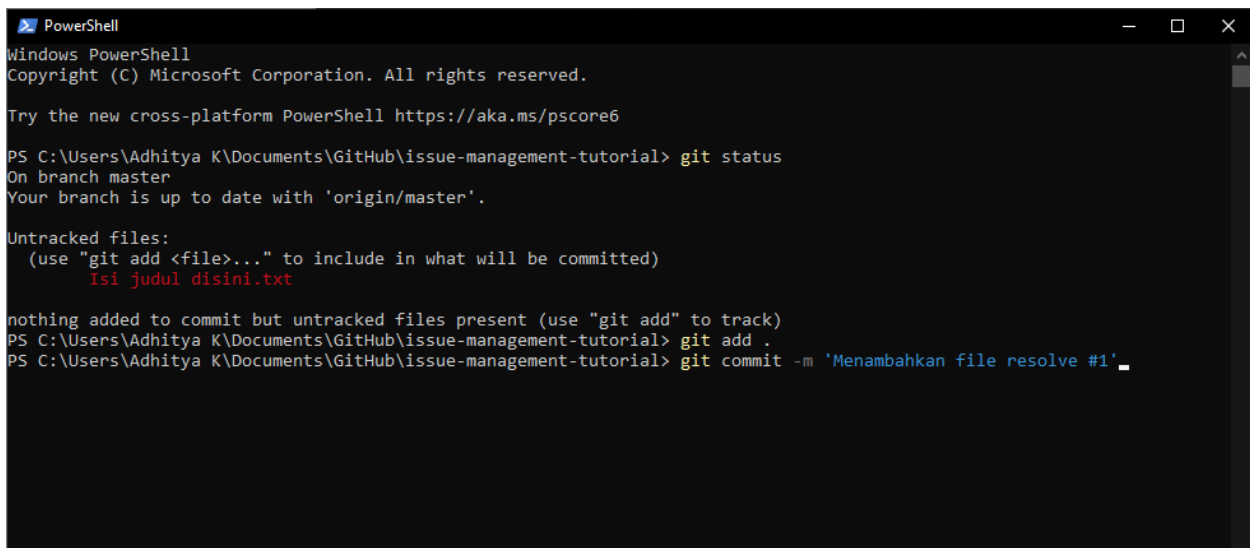
240eaa5

### 5. Resolve (Close) issue

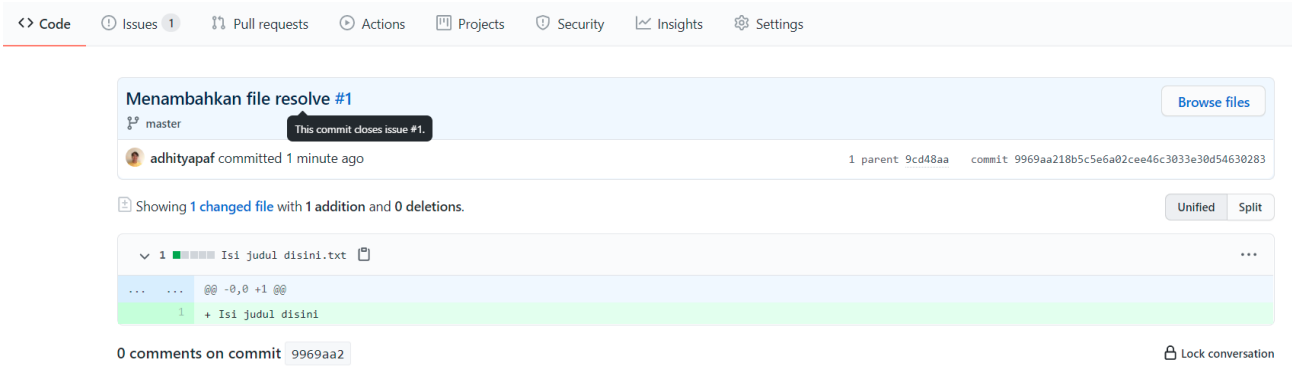
Isu dapat kita tutup bersamaan dengan commit. Caranya dengan menambahkan kata resolve #angka. Contoh : memperbaiki bug A resolve #1 (artinya kita tutup isu nomer 1 dengan commit ini). Gambar dibawah adalah resolve isu dengan Sourcetree.



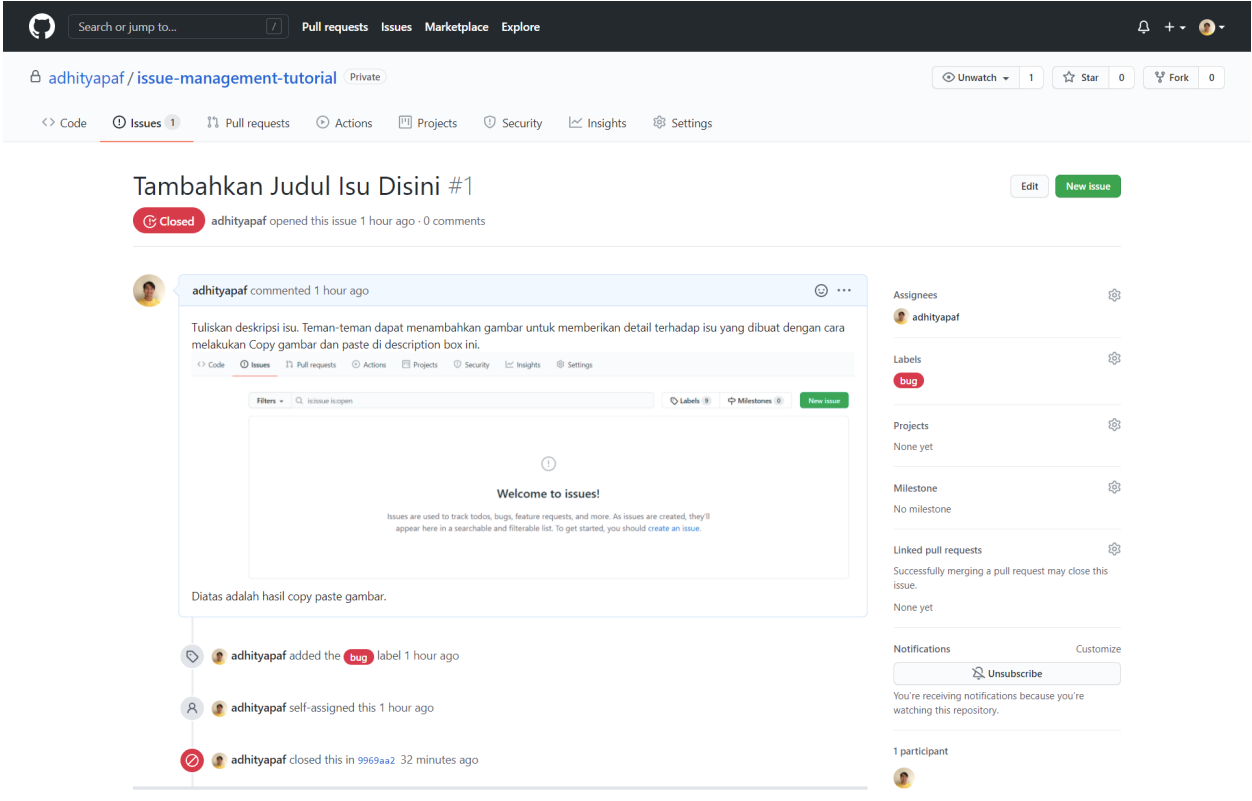
Dan gambar berikut adalah menggunakan Command Line Interface.



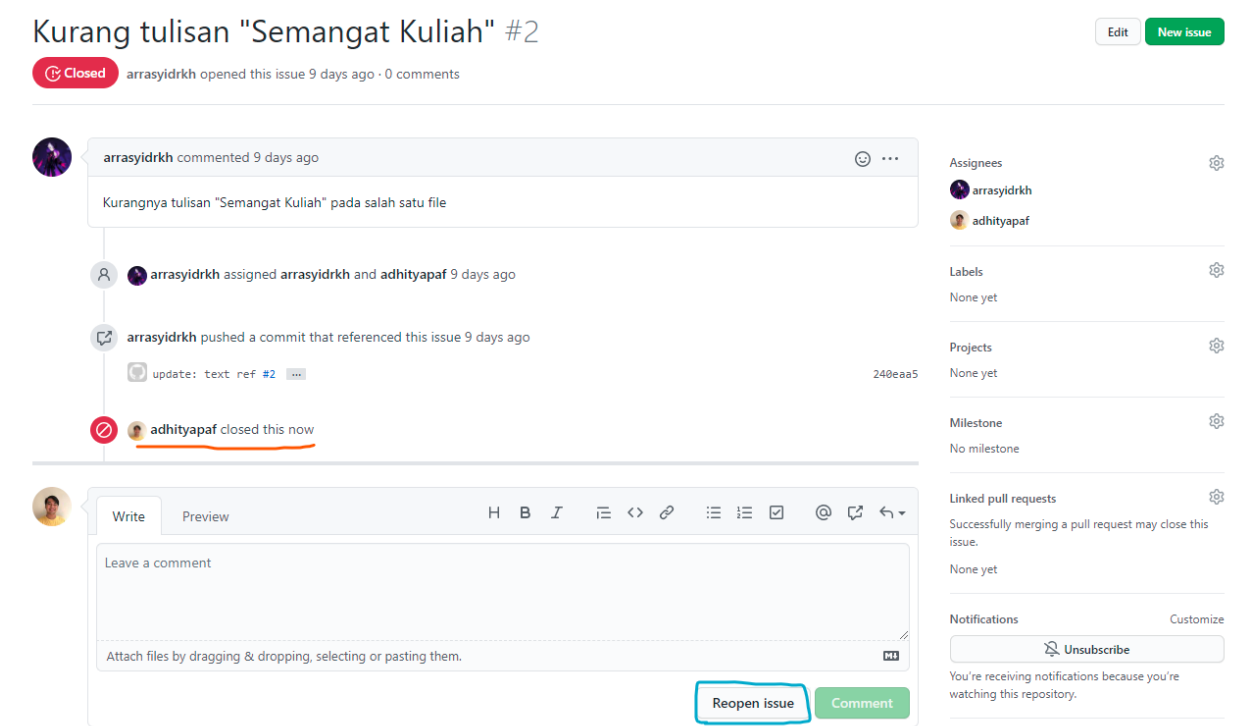
Hasilnya seperti gambar dibawah ini.



Isu akan secara otomatis ditutup bersamaan dengan commit yang telah dilakukan.



Berikut adalah tampilan ketika isu telah di tutup. Jika teman-teman ingin membuka kembali isu tersebut, teman-teman dapat menekan tombol Reopen Issue.



## STUDI KASUS WEEK 5

Dalam pengerjaan studi kasus ini, setiap tim akan membuat *issue* pada Git Hub berdasarkan *product backlog item* yang telah dibuat sebelumnya. Berikut adalah langkah - langkah yang harus dilakukan.

1. Membuat *repository* baru pada *organization* RPL2021 dengan format  
<< Kelas - Nama\_Project >>  
Contoh : SI4200 - Gojek
2. Invite seluruh anggota tim pada *repository* bersama
3. Membuat kanban pada GitHub
  - Template : Automated Kanban
  - Kolom pada Kanban : to do (*automated as to do*), in progress (*automated as in progress*), resolved (*automated as done*), dan done (*default*)
4. Menambahkan seluruh PBI pada kolom to do
5. Convert card menjadi issue sesuai dengan template *good issue*
  - Nama card / issue : [SX - PBIX] Fitur  
Keterangan : S (*Sprint*), PBI (*Product Backlog Item*), X (Nomor)  
Contoh : [S1 - PBI2] Membuat dashboard
  - Deskripsi card : Saya sebagai (*role* aplikasi) ingin (*requirement*/kebutuhan) sehingga (tujuan)  
Contoh : Saya sebagai **admin** ingin **melihat grafik kehadiran karyawan** sehingga **saya dapat memastikan siapa saja karyawan yang hadir.**
6. Membuat milestone sesuai dengan sprint yang ada
7. Mengedit issue yang telah dibuat sebelumnya
  - Lengkapi deskripsi issue dengan menambahkan gambar atau link mockup
  - Tambahkan Assignees
  - Tambahkan Labels
  - Tambahkan Milestones

## **STUDI KASUS WEEK 6**

Dalam pengerjaan studi kasus ini, setiap tim akan melakukan commit pada github sesuai dengan issue yang telah dibuat. Terdapat 2 studi kasus yang harus dilakukan.

### **CASE 1 : Melakukan Commit Sesuai Issue**

1. Masuk ke repository bersama.
2. Buka issue yang telah dibuat sebelumnya.
3. Buka folder repository di local, pastikan sudah melakukan kontribusi atau perubahan pada file di direktori lokal.
4. Lakukan commit sesuai dengan issue yang dikerjakan dengan menggunakan ref #nomer\_issue
5. Lakukan push ke branch masing masing.
6. Cek ke GitHub apakah commit tadi sudah mereferensikan issue yang dipilih.

### **CASE 2 : Latihan Melakukan Resolve Issue**

1. Salah satu perwakilan tim membuat satu issue random di repository bersama.
2. Buat file txt di local repository.
3. Isi file txt dengan nama anggota kelompok.
4. Lakukan resolve issue yang telah dibuat sebelumnya melalui commit resolved #nomer\_issue.
5. Cek ke GitHub apakah issue tersebut sudah di close.
6. Cek ke Kanban apakah card sudah pindah ke kolom resolved.
7. Reopen issue tersebut dan lakukan kembali langkah 4-6 sampai seluruh anggota tim melakukannya.