**JOBSHEET 2**

**Sistem Version Control dan Kanbar Board**

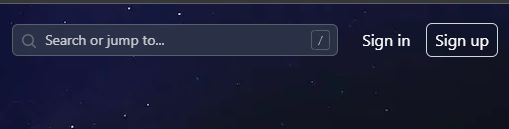
1. **Tujuan**

* Mahasiswa memiliki akun di repository
* Mahasiswa mampu menggunakan perintah-perintah dasar Github
* Mahasiswa mampu berkolaborasi dalam menggunakan Github
* Mahasiswa mampu menggunakan Kanban Board dalam kolaborasi

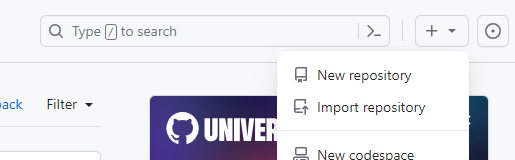
1. **Praktikum**
   1. **Percobaan 1: Menggunakan Github**

**Waktu percobaan : 120 menit**

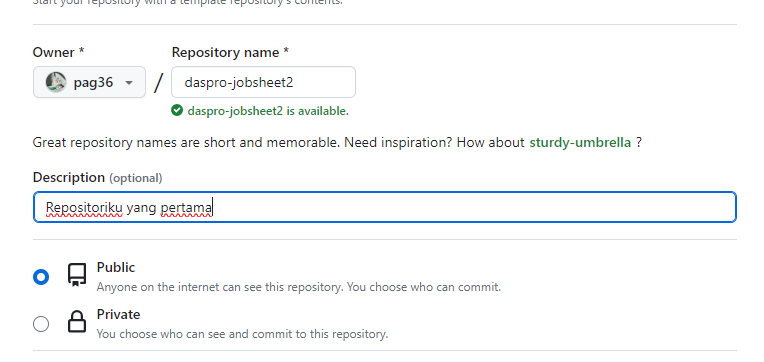
1. Buka situs web GitHub di **https://github.com**.
2. Klik tombol **"Sign up"** (Daftar) untuk membuat akun GitHub.



1. Ikuti petunjuk untuk mengisi informasi yang diperlukan dan verifikasi email Anda.
2. Setelah masuk ke akun GitHub Anda, klik tombol **"+"**, lalu pilih **"New repository"** (Repositori baru).



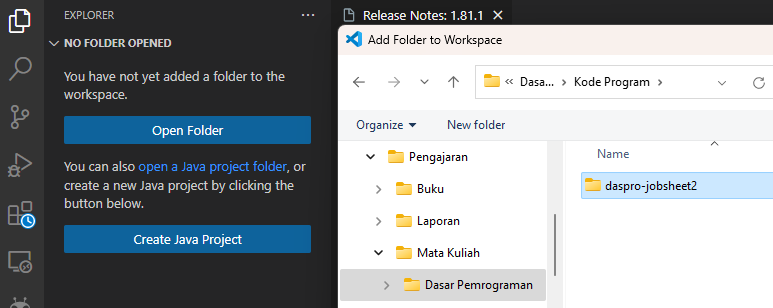
1. Isi **nama repositori, deskripsi (opsional), dan konfigurasi lainnya**.
2. Anda dapat membuat repositori publik atau pribadi sesuai kebutuhan. Jika sudah jangan lupa untuk klik tombol “**Create repository”**.



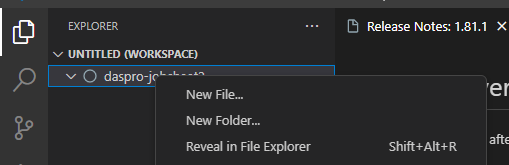
1. Untuk mengelola repositori secara lokal, Anda perlu mengklonanya ke komputer Anda. Tetapi, perlu melakukan installasi git client dulu di <https://git-scm.com/downloads>, setelah selesai download silakan melakukan installasi.
2. Gunakan perintah git clone dari terminal (command line) untuk mengklon repositori. Perintah umumnya **git clone** [**https://github.com/username/nama-repositori.git**](https://github.com/username/nama-repositori.git)



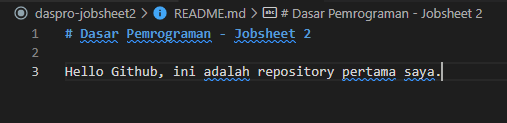
1. Buat atau **edit berkas-berkas di dalam repositori** sesuai kebutuhan. Buka folder repository menggunakan Visual Studio Code.



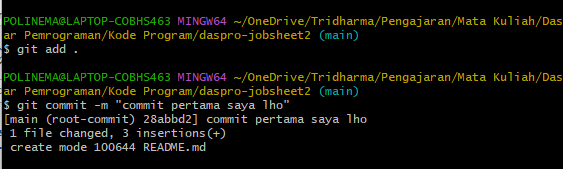
1. Tambahkan file dengan **klik kanan – New File**, berikana nama file tersebut dengan nama **“README.md”**



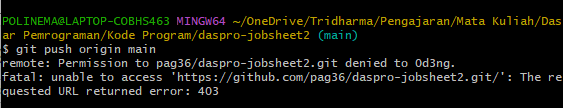
1. Isikan file **“README.md”** dengan isi seperti berikut



1. Setelah selesai, simpan perubahan Anda dan komit dengan perintah **git commit**. Anda akan diminta untuk **memberikan pesan komit** yang menjelaskan perubahan yang telah Anda lakukan.

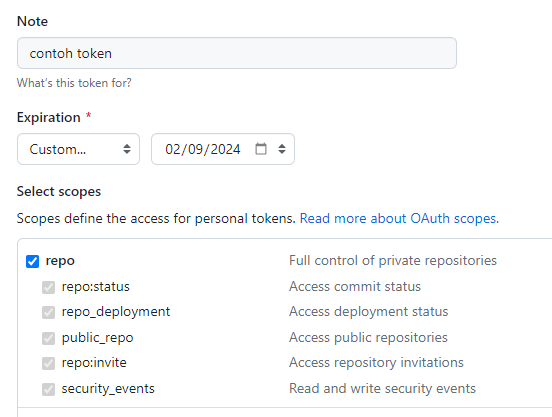


1. Untuk memperbarui repositori di GitHub dengan perubahan yang telah Anda lakukan secara lokal, gunakan perintah **git push**.
2. Misalnya, **git push origin nama-branch** akan mengirimkan perubahan ke branch di GitHub.



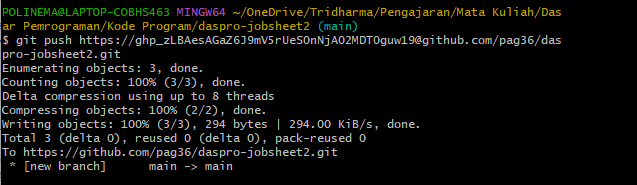
Normalnya untuk push ke repository akan meminta user atau password. Tetapi jika mengalami hal seperti di atas, yang perlu dilakukan adalah membuat token untuk push repository.

1. Klik akun Anda – Settings – Developer Settings – Tokens (classic) – Genereate new token (classic). Isikan bagian Note, Expiration, dan Select scopes. Jika sudah klik tombol **Generate token**.



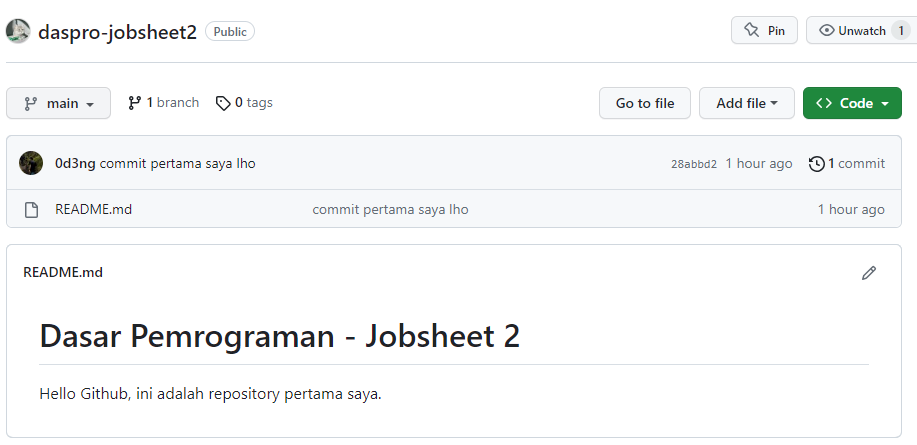
Simpan token tersebut karena tidak bisa dilihat kembali untuk digunakan push yang selanjutnya.

1. Silakan jalankan perintah git push <https://[token]@github.com/username/nama-repository.git>





1. Silakan cek halaman Github Anda



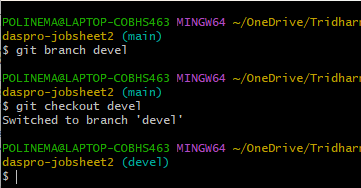
1. Jika langkah no.14 berhasil dilakukan, Anda bisa lompat ke langkah 17.
2. Agar tidak selalu meminta login ketika akan **push**, tambahkan perintah git config –**global user.email “**[**you@example.com**](mailto:you@example.com)**”** dan **git config –global user.name “Your Name”** pada git bash atau terminal Visual Studio Code. email dan username diisikan dengan email dan username yang sama ketika mendaftar di Github.

**Pertanyaan**

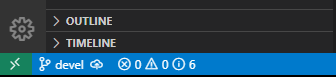
1. Jelaskan perbedaan perintah **git commit** dan **git push**?
2. Apakah bisa alurnya dibalik, membuat folder atau projek terlebih dahulu kemudian upload (push) ke Github? Jika bisa, buktikan!
   1. **Percobaan 2 : Dasar Kolaborasi di Github**

**Waktu percobaan : 60 menit**

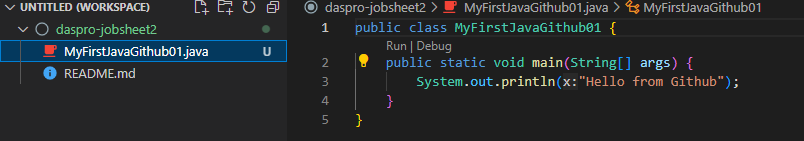
1. Sebelum memulai pekerjaan pada proyek, sebaiknya membuat branch terlebih dahulu untuk mengisolasi perubahan Anda dari cabang utama (biasanya "main" atau "master").
2. Gunakan perintah **git branch nama-branch** untuk membuat branch baru dan **git checkout nama-branch** untuk beralih ke branch tersebut.



Pada Visual Studio Code seharusnya akan berganti menjadi branch **“devel”**, jika belum silakan klik kemudian pilih branch **“devel”**.

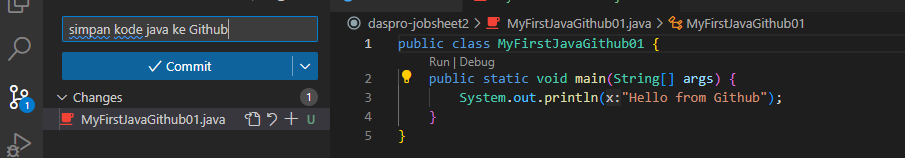


1. Buatlah file **MyFirstJavaGithubNoAbsen.java** (Gunakan No Absen Masing-masing).

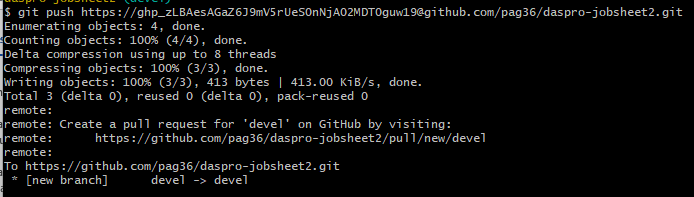


Jalankan kode program tersebut menggunakan langkah-langkah pada **Jobsheet 1**.

1. Simpan perubahan tersebut di local dengan cara **commit** kemudian **push** ke Github menggunakan Visual Studio Code. Jangan lupa **memberikan pesan** ketika akan melakukan **commit**. Caranya **klik icon ranting** – **isikan pesan commit** – klik **tombol Commit – klik tombol Publish Branch**.

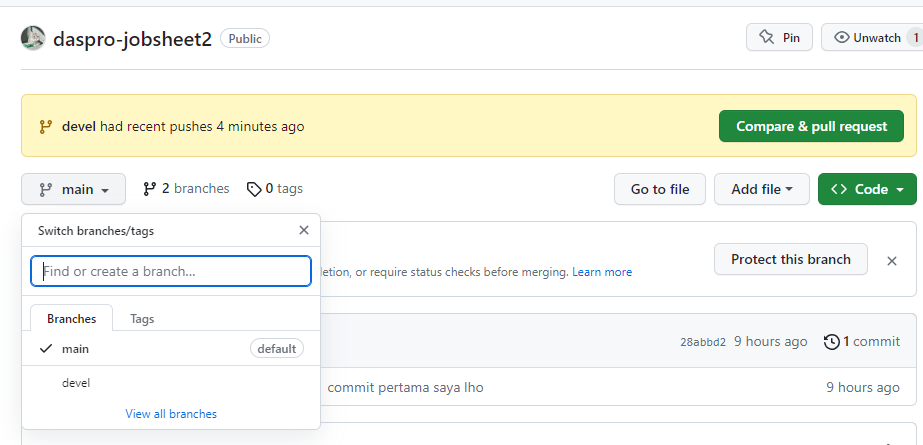


Atau Anda juga bisa menggunakan perintah di **git bash** yang sebelumnya, hasilnya seperti di bawah ini

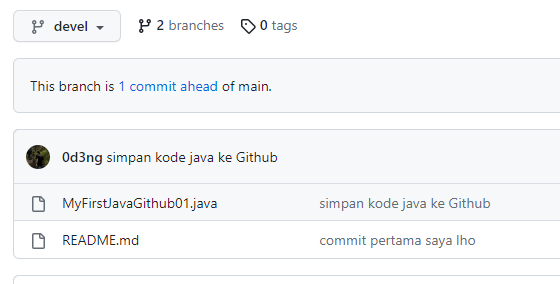




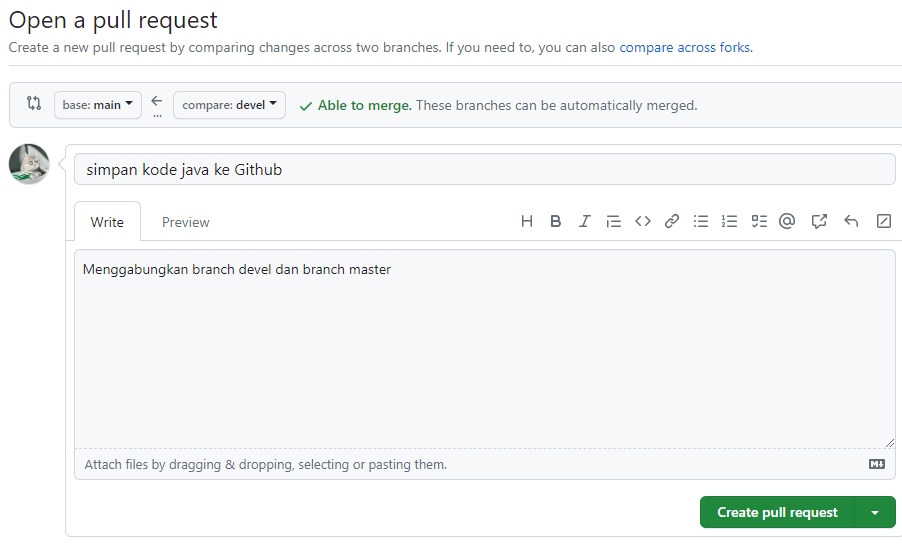
1. Silakan menuju ke halaman Github, seharusnya akan muncul **branch devel** yang beberapa waktu telah di-push.



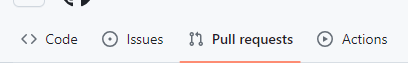
Selanjutnya Anda bisa membedakan antara **branch main** dan **branch devel**.



1. Klik tombol **Compare & pull request**, Anda dapat memilih branch mana yang akan digabungkan (devel ke master). **Isikan pesan** dan klik tombol **Create pull request**, tunggu beberapa saat kemudian klik tombol **Merge pull request**. Terakhir, klik tombol **Confirm merge**.



1. Pindah ke **tab Code**, kemudian amati hasil antara **branch main** dan **branch devel**.

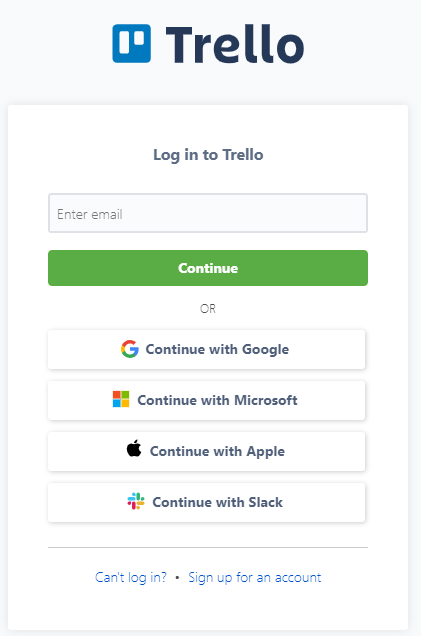


**Pertanyaan!**

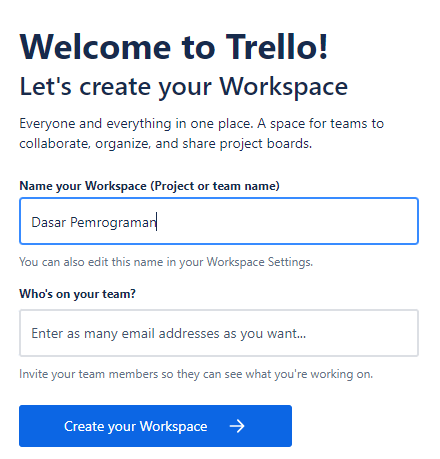
1. Jelaskan fungsi dari **Pull requests**!
2. Mengapa kita perlu membuat sebuah **branch**, manfaatnya apa?
   1. **Percobaan 3 : Menggunakan Trello**

**Waktu percobaan : 45 menit**

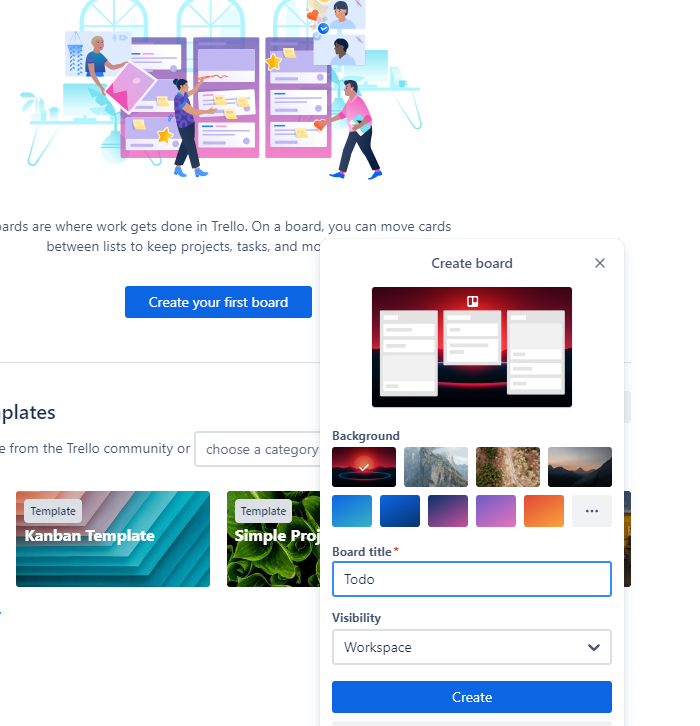
1. Buka situs web **Trello (https://trello.com/)** dan klik tombol **"Sign Up"** untuk membuat akun baru.
2. Anda dapat mendaftar menggunakan alamat email atau menghubungkannya dengan akun Google Anda.



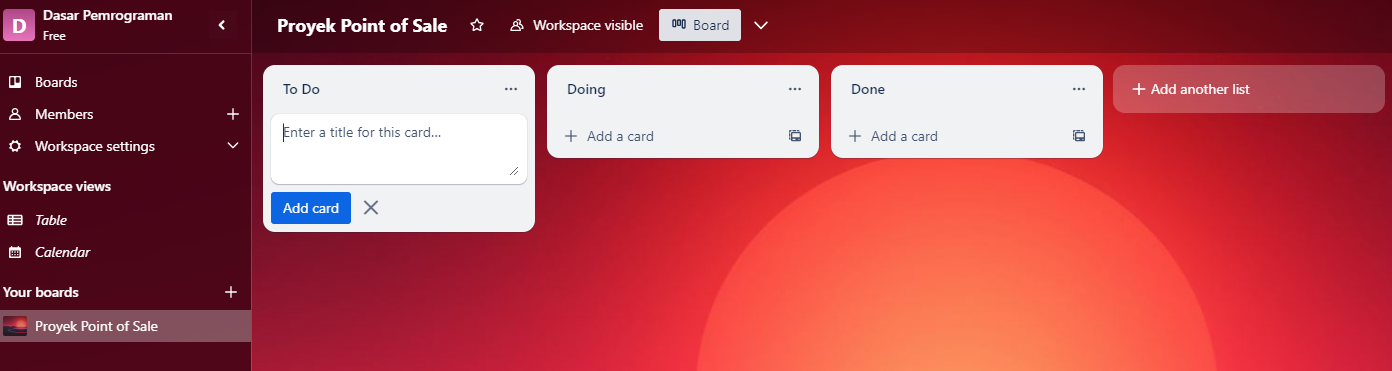
Jika pendaftaran sudah selesai dilakukan, silakan masukkan nama **Workspaces** dan anggota tim (optional), kemudian klik **Create your Workspace**.



1. Setelah Anda masuk, Anda akan berada di beranda Trello. Untuk membuat papan Kanban baru, klik tombol **"Create your first board"**.
2. Berikan judul untuk papan Anda dan atur visibilitasnya (Publik, Privat, atau Tim).



**Board title** bisa sebagai nama proyek yang akan dikerjakan, Anda bisa mengganti Board title sesuai dengan kebutuhan.



1. Di dalam papan Anda, Anda akan memiliki satu list awal yang disebut "To Do". Anda dapat menambahkan list tambahan sesuai dengan tahapan alur kerja Anda. Misalnya, "In Progress", "Review", "Testing", dan "Done".
2. Klik "Add a list" di samping list yang sudah ada untuk menambahkan list baru.

**Pertanyaan!**

1. Bagaimana langkah yang dilakukan untuk mengundang anggota tim untuk bergabung ke dalam board Anda?
2. Apakah memungkinkan untuk membuat Board tidak dari awal (menambahkan list satu per satu), jika mungkin bagaimana caranya?
3. **Tugas**

**Waktu pengerjaan : 60 menit**

1. Buatlah kelompok yang terdiri dari 3 orang!
2. Buatlah repository berdasarkan topik yang dipilih oleh kelompok Anda, undang semua anggota tim ke dalam repository tersebut.
3. Buatlah sebuah Kanban Board dengan judul board lengkap dengan card sesuai dengan topik proyek yang kelompok Anda pilih, undang semua anggota tim ke dalam board tersebut.