

KAKAS PENGEMBANGAN PL

MODUL PRAKTIKUM



Prodi Sistem Informasi
Fakultas Sains & Teknologi
UIN Alauddin Makassar
2021

Kartu Kontrol

Nama Mahasiswa :

NIM :

Per temuan	Hari/Tanggal	Paraf Dosen / Asisten			
		Tugas Pendahuluan	Respon	Kehadiran	Laporan Praktikum
1	/				
2	/				
3	/				
4	/				
5	/				
6	/				
7	/				
8	/				
9	/				
10	/				



Daftar Isi

Kartu Kontrol	i
Daftar Isi	ii
Pendahuluan	1
Deskripsi Praktikum	1
Modul Praktikum 1: [Judul Percobaan]	2
Deskripsi Pertemuan	2
Syarat Kompetensi	2
Standar Kompetensi	3
Dasar Teori	3
Tugas Pendahuluan	3
Langkah-Langkah Praktikum	3
Tugas Laporan	25
Modul Praktikum 2: [Judul Percobaan]	26
Deskripsi Pertemuan	26
Syarat Kompetensi	26
Standar Kompetensi	26
Dasar Teori	26
Tugas Pendahuluan	26
Langkah-Langkah Praktikum	27
Tugas Laporan	27
Modul Praktikum 3: [Judul Percobaan]	35
Deskripsi Pertemuan	35
Syarat Kompetensi	35
Standar Kompetensi	35
Dasar Teori	35
Tugas Pendahuluan	35
Langkah-Langkah Praktikum	36
Tugas Laporan	36
Modul Praktikum 4: [Judul Percobaan]	51
Deskripsi Pertemuan	51
Syarat Kompetensi	51
Standar Kompetensi	51
Dasar Teori	51
Tugas Pendahuluan	51

Langkah-Langkah Praktikum.....	52
Tugas Laporan.....	63
Modul Praktikum 5: [Judul Percobaan]	1
Deskripsi Pertemuan	1
Syarat Kompetensi	1
Standar Kompetensi	1
Dasar Teori.....	1
Tugas Pendahuluan	1
Langkah-Langkah Praktikum.....	2
Tugas Laporan.....	7
Modul Praktikum 6: [Judul Percobaan]	8
Deskripsi Pertemuan	8
Syarat Kompetensi	8
Standar Kompetensi	8
Dasar Teori.....	8
Tugas Pendahuluan	8
Langkah-Langkah Praktikum.....	11
Tugas Laporan.....	22
Modul Praktikum 7: [Judul Percobaan]	23
Deskripsi Pertemuan	23
Syarat Kompetensi	23
Standar Kompetensi	23
Dasar Teori.....	24
Tugas Pendahuluan	24
Langkah-Langkah Praktikum.....	24
Tugas Laporan.....	27
Modul Praktikum 8: [Judul Percobaan]	1
Deskripsi Pertemuan	1
Syarat Kompetensi	1
Standar Kompetensi	1
Dasar Teori.....	1
Tugas Pendahuluan	2
Langkah-Langkah Praktikum.....	2
Tugas Laporan.....	5
Modul Praktikum 9: [Judul Percobaan]	1
Deskripsi Pertemuan	1
Syarat Kompetensi	1



Standar Kompetensi	1
Dasar Teori	1
Tugas Pendahuluan	3
Langkah-Langkah Praktikum	3
Tugas Laporan	6
Modul Praktikum 10: [Judul Percobaan]	1
Deskripsi Pertemuan	1
Syarat Kompetensi	1
Standar Kompetensi	1
Dasar Teori	1
Tugas Pendahuluan	5
Langkah-Langkah Praktikum	5
Tugas Laporan	5
Lampiran-Lampiran	1
Format Tugas Pendahuluan	2
Format Laporan Praktikum	3



Praktikum Kakas Perangkat Lunak

Nama Dosen: Firmansyah Ibrahim, S.Kom, M.Kom

Pendahuluan

Kelas ini akan mengajarkan Anda dalam mengelola *source code* yang digunakan untuk membangun sebuah software baik berupa aplikasi, ataupun website. Akan tetapi untuk memudahkan pembelajaran, hanya static-web development yang akan digunakan sebagai contoh. Contoh pembelajaran untuk Android, iOS, ataupun Game development akan disusun menyusul jika memang dibutuhkan.

Pengelolaan *source code* yang diajarkan, mencakup pembelajaran untuk:

1. Menelusuri perubahan kode (**tracing**).
2. Menelusuri siapa yang melakukan perubahan (**annotating**).
3. Mengelola *code versioning* (**branching management**).
4. Berkolaborasi dengan banyak developer.

Deskripsi Praktikum

Pertemuan 1: Membuat akun github,Menginstall dan setup sourcetree

Pertemuan 2: Create repository dan clone repository

Pertemuan 3: Commit dan push

Pertemuan 4: Atomic commit

Pertemuan 5: Branch

Pertemuan 6: Merge

Pertemuan 7: Latihan lanjutan Atomic Commit dan Branch Management

Pertemuan 8: Stash

Pertemuan 9: Collaboration

Pertemuan 10: SCM component and operations





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 1: Membuat akun github, menginstall dan setup sourcetree

Deskripsi Pertemuan

Sederhananya, GitHub adalah sebuah platform untuk menyimpan *source code* Anda di cloud. Selain GitHub, ada banyak platform lainnya yang bisa Anda gunakan. Akan tetapi hanya GitHub yang akan dibahas di dalam kelas ini.

Sourcetree adalah sebuah aplikasi untuk mengelola *source code*. Dengan aplikasi ini, Anda bisa melakukan semua hal yang dibutuhkan untuk mengelola kode, seperti mengambil kode dari cloud ke komputer Anda (clone), menyimpan perubahan kode dari komputer Anda ke cloud (push), bahkan sampai mengetahui siapa yang merusak kode Anda jika suatu hari terdapat masalah.

Terakhir kita membutuhkan sebuah *code editor*. Anda bisa menggunakan *code editor* favorit anda. Tetapi dalam kelas ini akan menggunakan Visual Studio Code karena merupakan *software open source*. Tidak sulit untuk men-download dan meng-install-nya.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Syarat Kompetensi

1. Memiliki E-Mail untuk membuat akun di www.github.com
2. Download aplikasi sourcetreeapp di www.sourcetreeapp.com
3. Download aplikasi Visual Studio Code di [www.code.visualstudiocode.com](https://www.code.visualstudio.com)

Standar Kompetensi

1. Mahasiswa dapat memiliki akun github untuk kebutuhan repository source kode.
2. Mahasiswa mampu memanfaatkan github untuk kebutuhan repository source kode dan dapat memanfaatkan sourcetreeapp untuk kebutuhan sinkronisasi dengan github.
3. Mahasiswa mampu mengkombinasikan dan menghubungkan framework visualstudiocode dalam membangun coding program dengan mengkombinasikan dengan sourcetreeapp dan github.

Dasar Teori

Sederhananya, GitHub adalah sebuah platform untuk menyimpan *source code* Anda di cloud. Selain GitHub, ada banyak platform lainnya yang bisa Anda gunakan. Akan tetapi hanya GitHub yang akan dibahas di dalam kelas ini.

Sourcetree adalah sebuah aplikasi untuk mengelola *source code*. Dengan aplikasi ini, Anda bisa melakukan semua hal yang dibutuhkan untuk mengelola kode, seperti mengambil kode dari cloud ke komputer Anda (clone), menyimpan perubahan kode dari komputer Anda ke cloud (push), bahkan sampai mengetahui siapa yang merusak kode Anda jika suatu hari terdapat masalah.

Aplikasi ini digunakan untuk menghindari penggunaan terminal ataupun command line. Aplikasi ini juga memberikan visualisasi dari aktivitas yang Anda lakukan dalam mengelola *source code*, sehingga cocok sekali untuk pemula atau non developer. Visualisasi ini akan membantu Anda dalam mengetahui apa saja yang sedang Anda kerjakan.

Terakhir kita membutuhkan sebuah *code editor*. Anda bisa menggunakan *code editor* favorit anda. Tetapi dalam kelas ini akan menggunakan Visual Studio Code karena merupakan *software open source*. Tidak sulit untuk men-download dan meng-install-nya.

Tugas Pendahuluan

1. Jelaskan apa yang dimaksud dengan Git-Hub, Sourcetree dan Visual studio code?

Langkah-Langkah Praktikum

GITHUB

1. Buka <https://github.com> di browser favorit anda.
2. Klik pada tombol **Sign up for Github** berwarna **hijau** di kanan bawah.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

The screenshot shows the GitHub sign-up interface. At the top, there's a navigation bar with links for "Why GitHub?", "Business", "Explore", "Marketplace", and "Pricing". A search bar and "Sign in" and "Sign up" buttons are also present. The main content area features a large heading "Built for developers" and a subtext explaining GitHub's purpose as a development platform for open source and business projects. To the right is a form for creating a new account, requiring a "Username", "Email", and "Password". A note below the password field specifies that it must be at least 15 characters long, containing a number and a lowercase letter. A green "Sign up for GitHub" button is at the bottom of the form, with a small disclaimer text below it.

3. Anda akan dibawa menuju halaman untuk mengisi data diri anda. Ada beberapa field yang harus Anda isi:

Username : Isi dengan yang Anda inginkan.

Email : Sebaiknya gunakanlah email Gmail untuk memudahkan ketika nanti menginstall Sourcetree.

Password : Isi dengan yang Anda inginkan.

Pastikan field **username** dan **email** menampilkan centang hijau. Ini adalah indikator bahwa **username** dan **email** yang Anda masukkan bisa digunakan sebagai identitas Anda (belum pernah digunakan oleh orang lain).

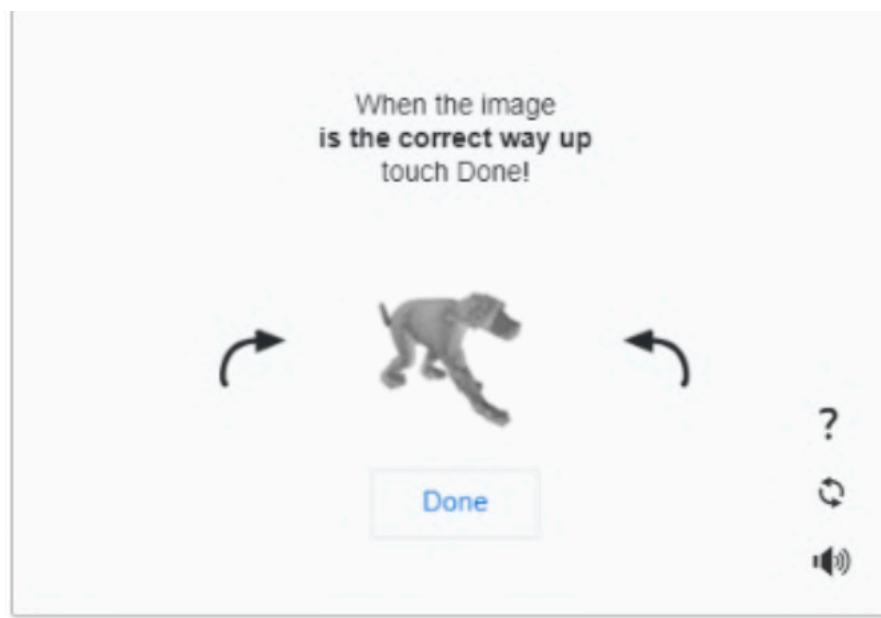
The screenshot shows the "Join GitHub" sign-up process. It consists of three steps: Step 1: Create personal account, Step 2: Choose your plan, and Step 3: Tailor your experience. The user is currently on Step 1. The "Create your personal account" section contains fields for "Username", "Email address", and "Password". A red error message box states "There were problems creating your account." Below the "Username" field, the value "arjuna135" is shown with a green checkmark. Below the "Email address" field, the value "aa.arjuna.pratama@gmail.com" is shown with a green checkmark. Below the "Password" field, the value "*****" is shown with a red error message box stating "Password can't be blank". A "Verify account" button is visible. To the right, a sidebar titled "You'll love GitHub" lists benefits such as "Unlimited collaborators", "Unlimited public repositories", "Great communication", "Frictionless development", and "Open source community", each accompanied by a green checkmark.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

4. Selanjutnya, Anda akan dihadapkan pada layout verifikasi. Lakukan saja apa yang diinstruksikan. Jika sudah terverifikasi maka akan muncul **centang hijau besar** pada layout tersebut. Sekarang Anda dipersilakan untuk klik tombol **Create an Account** berwarna hijau.



By clicking "Create an account" below, you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

[Create an account](#)

5. Selanjutnya, Anda akan diarahkan ke halaman plan. Halaman ini meminta Anda untuk memilih rencana penggunaan GitHub. Untuk pemula, kami sarankan pilih plan yang gratis, plan yang berbayar memungkinkan Anda untuk menyimpan repository kode secara privat, tapi untuk saat ini fitur tersebut tidak akan kita gunakan. Dua fitur lain di bawah plan adalah untuk set up organization dan menerima update lewat email, Anda bebas untuk mencentang atau melewatkannya. Silakan klik tombol **Continue** berwarna hijau untuk meneruskan.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

Choose your personal plan

Every plan comes with GitHub's most-loved features: Collaborative code review, issue tracking, the open source community, and the ability to join organizations.

Plan	Price	Includes
Free	\$0 per month	Personal account Unlimited public repositories Unlimited collaborators
Developer	\$7 per month	Personal account Unlimited public repositories Unlimited private repositories Unlimited collaborators

Free for students as part of the Student Developer Pack.

Help me set up an organization next

Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
[Learn more about organizations](#)

Send me updates on GitHub news, offers, and events

Unsubscribe anytime in your email preferences. [Learn more](#)

Continue

6. Selanjutnya Anda berada di halaman pengisian form experience. Anda bebas untuk mengisi atau melewatkannya. Klik tombol **Submit** jika Anda mengisi, klik **skip this step** jika Anda tidak mau mengisi.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

How would you describe your level of programming experience?

Very experienced Somewhat experienced Totally new to programming

What do you plan to use GitHub for? (check all that apply)

Development Design Project Management
 Research School projects Other (please specify)

Which is closest to how you would describe yourself?

I'm a professional I'm a hobbyist I'm a student
 Other (please specify)

What are you interested in?

game-development |

e.g. tutorials, android, ruby, web-development, machine-learning, open-source

Submit [skip this step](#)

7. Jika semuanya telah selesai, Anda akan dibawa kembali ke halaman dashboard. Selamat! Kini Anda telah memiliki akun GitHub. Mari sedikit eksplorasi ke bagian profile untuk mengupdate data diri. Klik icon avatar di kanan atas, lalu pilih **Your profile**, lalu klik **edit profile**.

The screenshot shows the GitHub dashboard with a dark theme. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. On the right, a user menu is open, showing the user is signed in as 'arjuna135' with options for Your profile, Your repositories, Your stars, Your gists, Help, Settings, and Sign out. The main content area features a banner for learning Git and GitHub with buttons to 'Read the guide' or 'Start a project'. Below this, there's a message about new Terms of Service and Privacy Statement. The 'Repositories' section shows a message: 'You don't have any repositories yet!'. The 'Discover activity' section encourages users to 'Discover interesting projects and people to populate your personal news feed.' with a 'Explore GitHub' button.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

- Seharusnya Anda sekarang berada di halaman edit profile. Jika bingung apa yang mau Anda update, setidaknya Anda bisa mengupload foto atau logo Anda menjadi avatar. Dengan begitu, Anda akan lebih percaya diri untuk membagikan url GitHub Anda ke programmer lainnya untuk saling bertukar *source code*.

The screenshot shows the 'Profile' tab selected in the left sidebar of the GitHub 'Personal settings' section. The main area is titled 'Public profile' and contains fields for 'Name', 'Public email' (with a note about setting it to private), 'Bio' (with a note about linking to others), 'URL', 'Company', and 'Location'. There is also a 'Profile picture' section with a placeholder image and a 'Upload new picture' button. At the bottom, there is a note about optional fields and privacy, followed by a green 'Update profile' button.

Selesai. Selamat! Anda kini telah memiliki akun GitHub dengan avatar yang keren. Jangan lupa untuk membuka email dan memverifikasi akun GitHub Anda dari email tersebut. Selanjutnya Anda bisa membuat *repository source code*. Akan tetapi, bagi Anda yang belum mengerti apa itu repository, akan dijelaskan lebih lanjut setelah Anda menginstall salah satu Git GUI, yaitu Sourcetree.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

Sourcetree

1. Buka browser favorit anda, lalu buka url berikut : <https://www.sourcetreeapp.com/>
2. Anda akan berhadapan dengan landing page dari Sourcetree

The screenshot shows the Sourcetree application interface. On the left, there's a sidebar with options like 'WORKSPACE', 'History' (which is selected), 'BRANCHES', 'TAGS', 'REMOTES', 'SHELVED', and 'SUBREPOSITORIES'. The main area displays a 'Graph' view of a Git commit history. A table lists commits with columns for 'Commit', 'Author', 'Description', and 'Date'. The commits listed are:

Commit	Author	Description	Date
b735b67	Rahul Chhab...	[r] master [r] origin/HEAD Removing old...	Mar 3, 2016, 11...
bbd8bf6f	Rahul Chhab...	Merged in update-google-verification (pull request #14)	Feb 18, 2016, 1:3...
dfe97bd	Tyler Tatej...	[r] origin/update-google-verification: Update google verification...	Feb 11, 2016, 2:2...
3bc3290	Tyler Tatej...	Replace outdated Atlassian logo in footer with base-64 en...	Feb 11, 2016, 2:1...
cba4719	Tyler Tatej...	Add .gitignore	Feb 11, 2016, 1:3...
f167645	Mike Minns...	Updated Mac min-spec to 10.10	Feb 18, 2016, 11...
72d32ab	Michael Min...	Merged in hero_images (pull request #13)	Feb 18, 2016, 10...
24d40ff	Joel Unger...	[r] origin/hero_images: Used Tinyipng to c...	Feb 11, 2016, 2:3...
948439e	Joel Unger...	Replacing hero images with new version of Sourcetree...	Feb 11, 2016, 2:2...
ce9f653	Michael Min...	Merged in bugfixes-https (pull request #12)	Feb 15, 2016, 10...
85387b0	Patrick Tho...	[r] origin/bugfixes-https: fixed date and https errors	Jan 17, 2016, 12...
4fb6557	Joel Unger...	New Favicon	Feb 8, 2016, 3:05...
384e6d9	Rahul Chhab...	[r] origin/reach-console-access: search console people var...	Feb 3, 2016, 2:09...
6fa47c9	Mike Minns...	updated to move supported version to OSX 10.8+	Dec 15, 2015, 2:0...
fd4979b	Mike Minns...	remove extra , when a line is skipped due to empty server	Nov 23, 2015, 21...
fae199e	Mike Minns...	Skip records with empty server/user id as gis rejects them	Nov 23, 2015, 21...
0cc9e98	Mike Minns...	corrected paths after merge	Nov 23, 2015, 21...
051a61b	Mike Minns...	corrected column counting	Nov 23, 2015, 11...
a723c2	Mike Minns...	Merge branch 'audgen'	Nov 23, 2015, 11...
6591d80	Mike Minns...	deal with invalid instanciods	Nov 23, 2015, 11...
500a892	Michael Min...	Merged in eu2gex (pull request #11)	Nov 23, 2015, 10...

A free Git client for Windows and Mac

Sourcetree simplifies how you interact with your Git repositories so you can focus on coding. Visualize and manage your repositories through Sourcetree's simple Git GUI.

A free Git client for Windows and Mac - Sourcetree simplifies how you interact with your Git repositories so you can focus on coding. Visualize and manage your repositories through Sourcetree's simple Git GUI.

Begitulah Sourcetree memperkenalkan diri. Sourcetree menyederhanakan interaksi dengan *source code*, sehingga developer dapat fokus dalam menuliskan kode dan tidak dibebankan dengan masalah yang seringkali muncul dalam mengelola *source code*.

Silakan Klik tombol download **berwarna biru**. Anda tak perlu takut salah download karena halaman ini sudah cukup pintar untuk mengetahui bahwa Anda pengguna Windows atau Mac. Walaupun memang akan ada sedikit perbedaan user interface (UI) antara Sourcetree Windows dengan Sourcetree Mac, tidak akan berpengaruh banyak pada kelas ini. Untuk sementara pada kelas ini akan menggunakan Sourcetree Windows sebagai contoh.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

3. Sambil menunggu proses download selesai, silakan scroll ke bawah pada halaman landing page Sourcetree.

The screenshot shows the Sourcetree application interface. On the left, there's a code editor window titled "docs/themes/sourcetree/templates/base.html". It displays two hunks of code: Hunk 1 (Lines 1-7) and Hunk 2 (Lines 15-24). The code is color-coded for syntax highlighting. On the right side of the application, there are two sections: "Simple for beginners" and "Powerful for experts".

Simple for beginners

Say goodbye to the command line - simplify distributed version control with a Git client and quickly bring everyone up to speed.

Powerful for experts

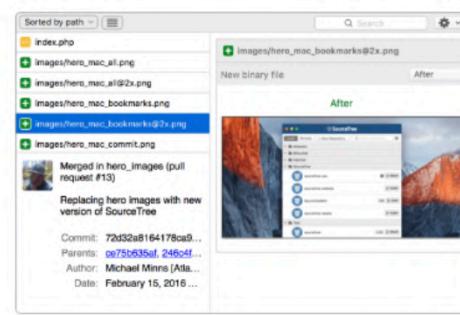
Perfect for making advanced users even more productive. Review changesets, stash, cherry-pick between branches and more.

Visualize your code

Seeing really is believing. Get information on any branch or commit with a single click.

Git and Hg on your desktop

A fully-featured GUI that offers an efficient, consistent development process right out of the box. Works with Git and Mercurial.

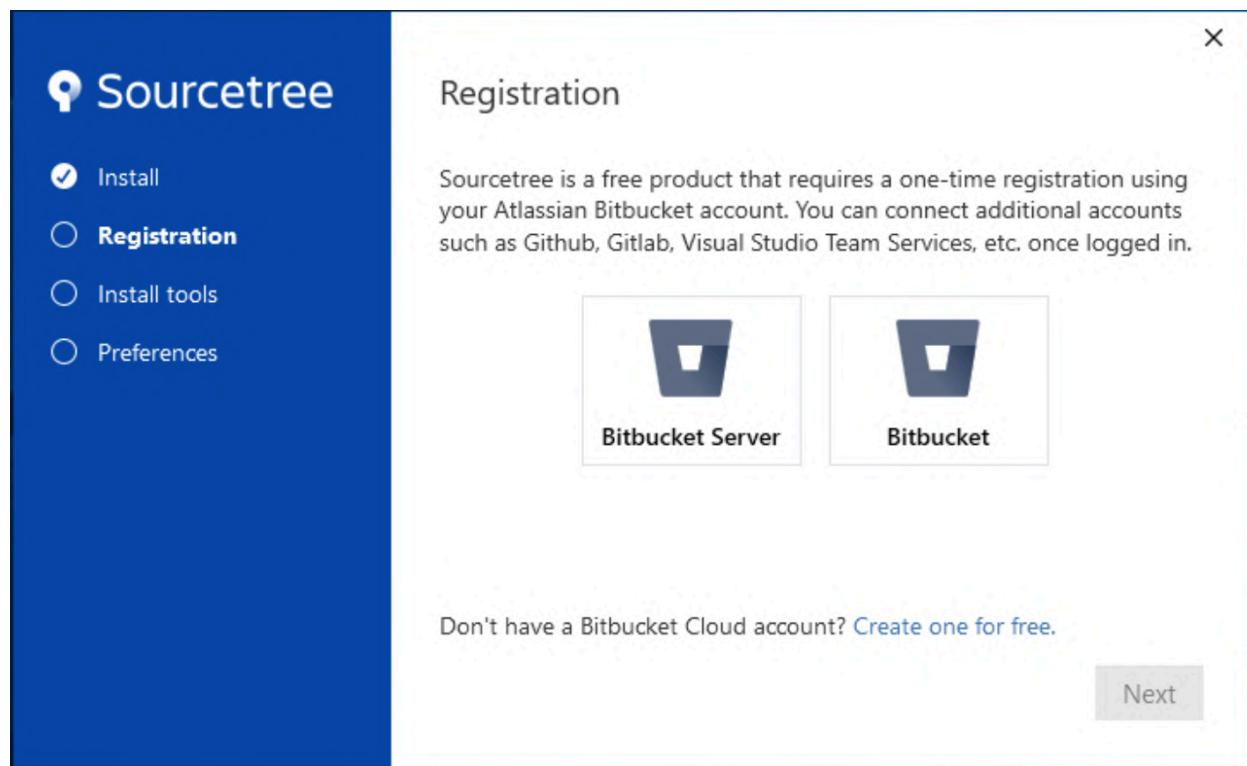


Anda akan melihat sedikit gambaran tentang bagaimana Sourcetree memberikan visualisasi terhadap yang sedang Anda lakukan saat mengelola source code. “Simple for beginners, Powerful for experts”, itulah sedikit tagline yang dibanggakan oleh Sourcetree. Sourcetree sangat mudah sekali digunakan oleh pemula, walaupun demikian Sourcetree merupakan powerful tool yang digunakan oleh para experts yang sudah bertahun-tahun mengembangkan software dengan jumlah puluhan ratusan developer.

Jika anda melihat tulisan “Git” atau “Hg” (Mercury - baca: merkuri), dan penasaran apa artinya, Anda dipersilakan untuk mencarinya sendiri. Kami akan menjelaskannya nanti di modul-modul berikutnya.

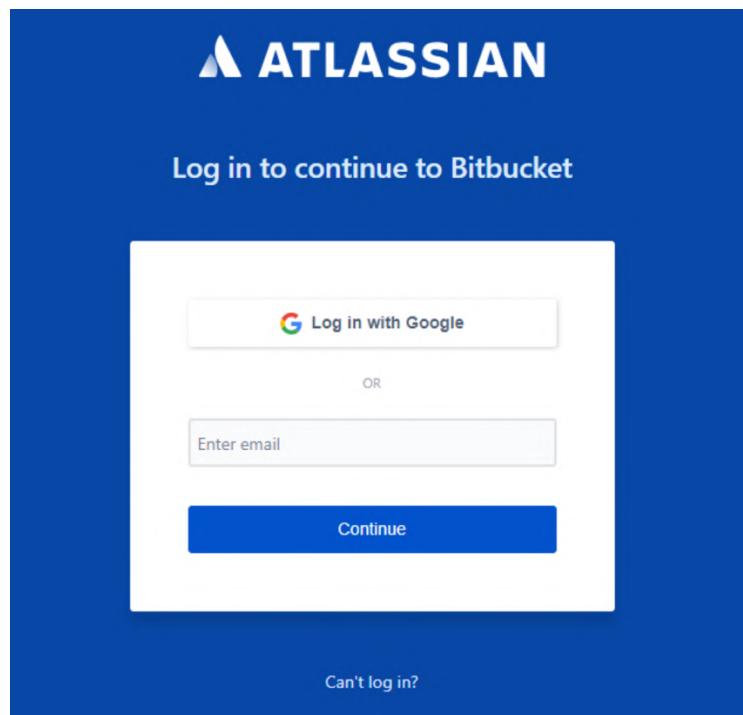
4. Jika proses download selesai, klik dua kali pada file yang terdownload untuk mulai menginstall. Proses installasi tidak akan membutuhkan waktu yang lama. Jika proses installasi sudah selesai, maka Anda dihadapkan dengan wizard seperti berikut :

Modul Praktikum Kakas Pengembangan Perangkat Lunak



Masih ada tiga langkah lagi yang harus dilakukan agar Sourcetree siap digunakan. Tahapan pertama yaitu registrasi.

5. Proses registrasi dilakukan dengan menghubungkan akun Bitbucket atau Bitbucket Server. Klik pada Bitbucket untuk registrasi. Maka Anda akan dibukakak halaman web Atlassian.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

Atlassian adalah sebuah perusahaan yang mengembangkan Sourcetree dan juga Bitbucket. Bitbucket adalah sebuah platform yang mirip dengan GitHub. Pada kelas ini kita tidak akan menggunakan Bitbucket, hanya saja kita membutuhkan akun Bitbucket untuk menggunakan Sourcetree.

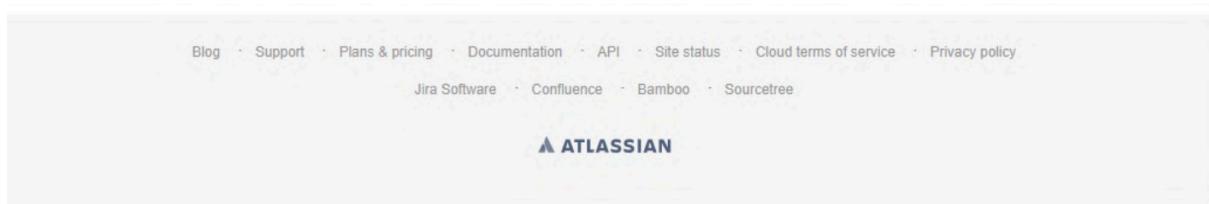
6. Klik **Log in with Google** untuk memudahkan proses login. Gunakanlah email Gmail yang sama pada saat membuat akun GitHub. Anda akan dibawa pada halaman untuk membuat akun Bitbucket.



Are you using the right account?

You are logged in with your Atlassian account for aa.arjuna.pratama@gmail.com, but you haven't yet signed up for Bitbucket Cloud with this account.

[Sign up for Bitbucket Cloud](#) using aa.arjuna.pratama@gmail.com or [log in with a different account](#).



Silakan klik pada **Sign up for Bitbucket Cloud**.

7. Sekarang Anda berada di halaman pengisian username untuk akun Bitbucket. Gunakan username yang sama dengan username GitHub Anda. Walaupun sebenarnya Anda bebas menggunakan username apapun.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

The screenshot shows the Bitbucket login page. At the top, there's a navigation bar with links for Features, Integrations, Server, Data Center, Pricing, Log in, and Get started. Below the navigation, the text "Almost done" is displayed. A form field contains "bitbucket.org / arjuna135". A green "Continue" button is centered below the input field. Below the button, a message states: "You're signing up with your Atlassian account for aa.arjuna.pratama@gmail.com. Not the account you want to use? Log out and enter the correct email address." The Bitbucket logo is visible at the bottom left of the page.

The screenshot shows the Atlassian footer navigation bar. It includes links for Blog, Support, Plans & pricing, Documentation, API, Site status, Cloud terms of service, Privacy policy, Jira Software, Confluence, Bamboo, and Sourcetree. The Atlassian logo is centered at the bottom.

Klik **continue**.

8. Seperti pada saat membuat akun Github, saat diminta untuk mengisi experience form. Anda bisa skip saja atau bisa juga mengisi form tersebut.



Before we drop you in Bitbucket, help us tailor your experience

The screenshot shows a user profile questionnaire. It includes three dropdown menus: "What describes your experience with source control?" (I have no experience), "How many people do you think you'll work with on Bitbucket?" (Just me), and "What best describes your role?" (Student). Below the dropdowns are two buttons: "Skip" and a blue "Submit" button.

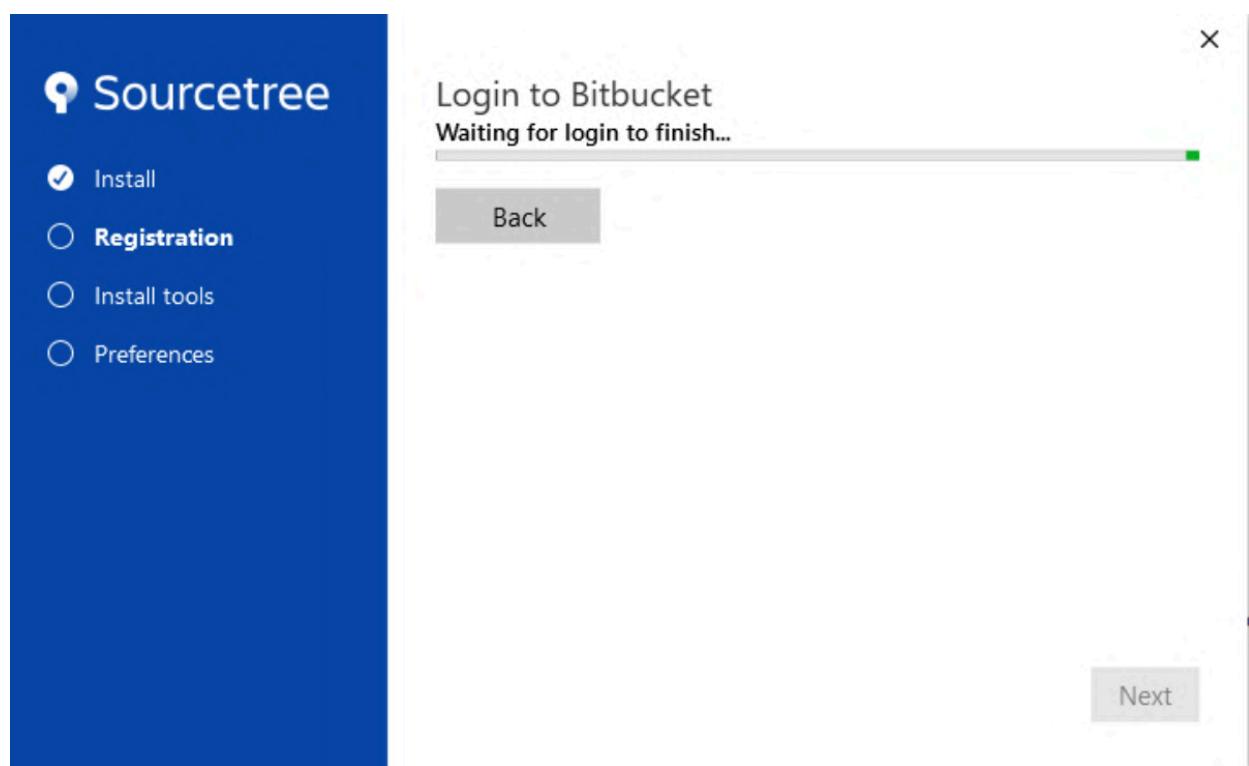
9. Anda kini berada di halaman dashboard atau overview dari Bitbucket. Selamat! Selain anda telah mempunyai akun GitHub, kini Anda juga memiliki akun Bitbucket.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

10. Kembali ke wizard registrasi Sourcetree. Anda akan menemukan bahwa wizard sedang menunggu Anda untuk login Bitbucket.



Sepertinya wizard ini kurang baik terhubung dengan browser anda. Klik **Back**, lalu klik lagi Bitbucket untuk menghubungkan Sourcetree dengan akun Bitbucket.

11. Kini halaman web baru akan muncul. Pada halaman ini Anda diminta untuk memberikan akses akun Bitbucket Anda agar dapat dibaca oleh Sourcetree.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Confirm access to your account

Sourcetree for Windows is requesting access to the following:

-  Read and modify your account information
-  Read and modify your repositories' issues
-  Read and modify your team's project settings, and read and transfer repositories within your team's projects
-  Read and modify your repositories and their pull requests
-  Administer your repositories
-  Read and modify your snippets
-  Read and modify your team membership information
-  Read and modify your repositories' webhooks
-  Read and modify your repositories' wikis

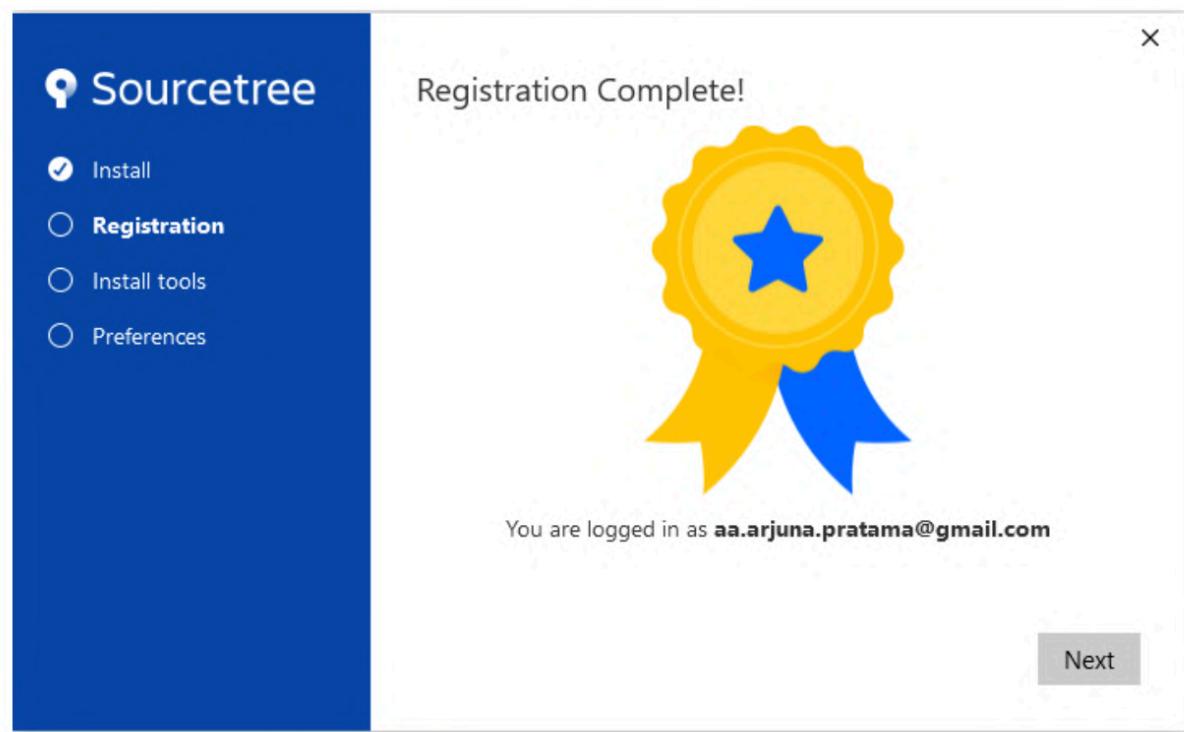
By installing the App you agree to the [privacy policy](#) and [terms of use](#) provided by Sourcetree for Windows.

Grant access **Cancel**

Klik tombol **Grant access**.

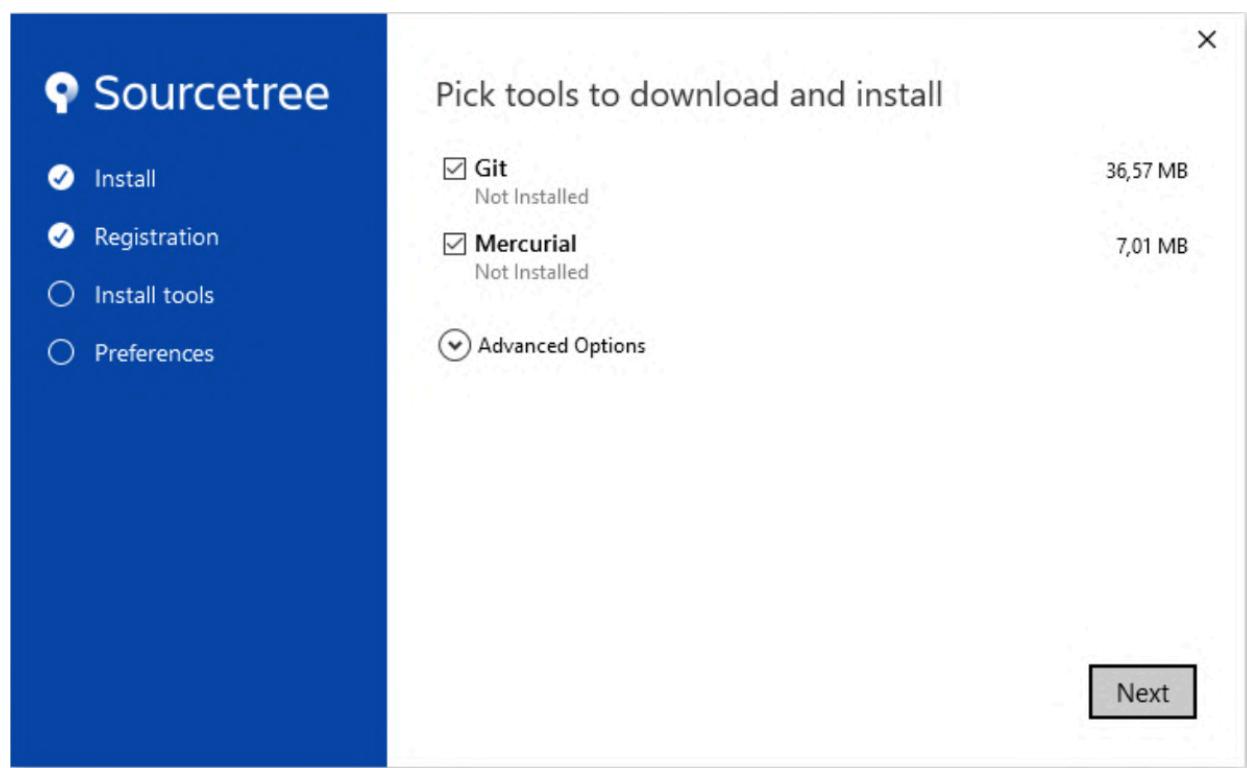
12. Jika Anda tetap berada di halaman web, maka hanya halaman kosong yang terlihat. Buka kembali jendela wizard registrasi Sourcetree, maka Anda akan melihat informasi bahwa Sourcetree kini telah berhasil teregistrasi menggunakan akun Bitbucket yang baru saja dibuat.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



Klik **Next** untuk melanjutkan.

13. Wizard selanjutnya adalah menginstall *version control tools*.

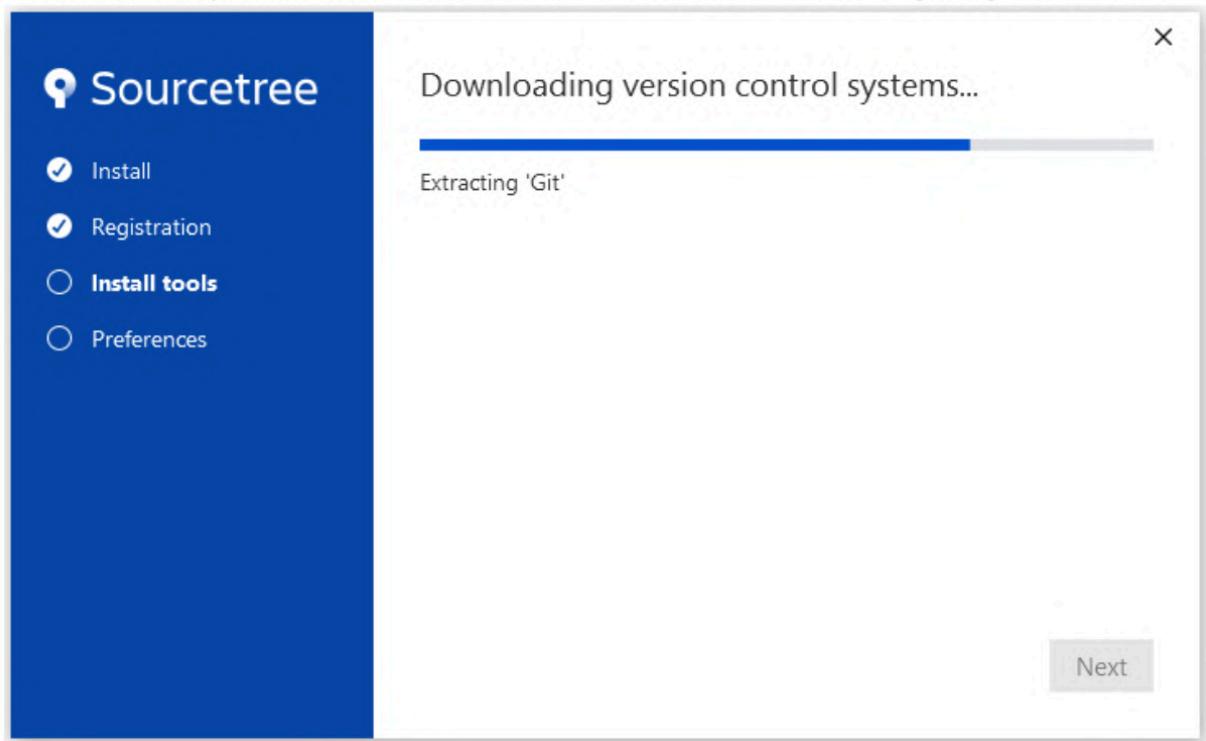




Modul Praktikum Kakas Pengembangan Perangkat Lunak

Ada dua pilihan *version control tools*, yaitu Git dan Mercurial. Pada kelas ini, Anda wajib menginstall Git. Menginstall Mercurial adalah optional, akan tetapi kami sarankan untuk menginstall Git dan juga Mercurial. Walaupun pada kelas ini tidak akan dibahas mengenai Mercurial, akan tetapi akan memudahkan nantinya untuk belajar Mercurial tanpa harus mengkonfigurasi ulang Sourcetree.

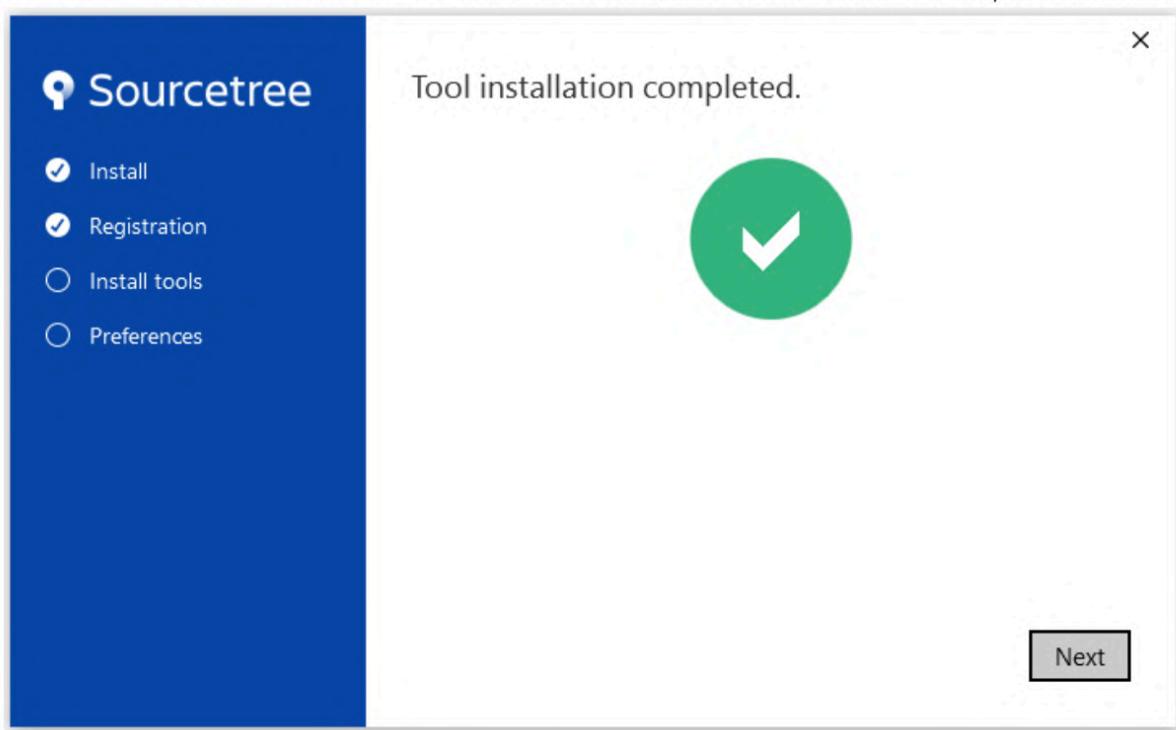
Klik **Next**, dan proses download dan instalasi Git akan berlangsung tidak lama.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

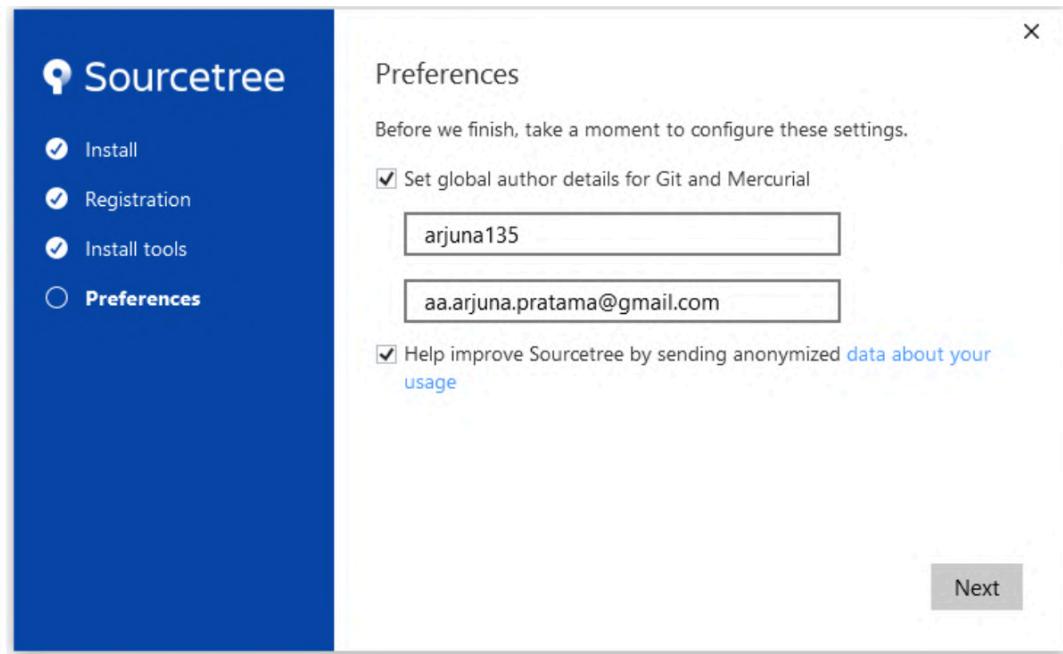
Jika download dan installasi selesai, maka akan muncul informasi seperti berikut:



Klik **Next** untuk meneruskan proses.

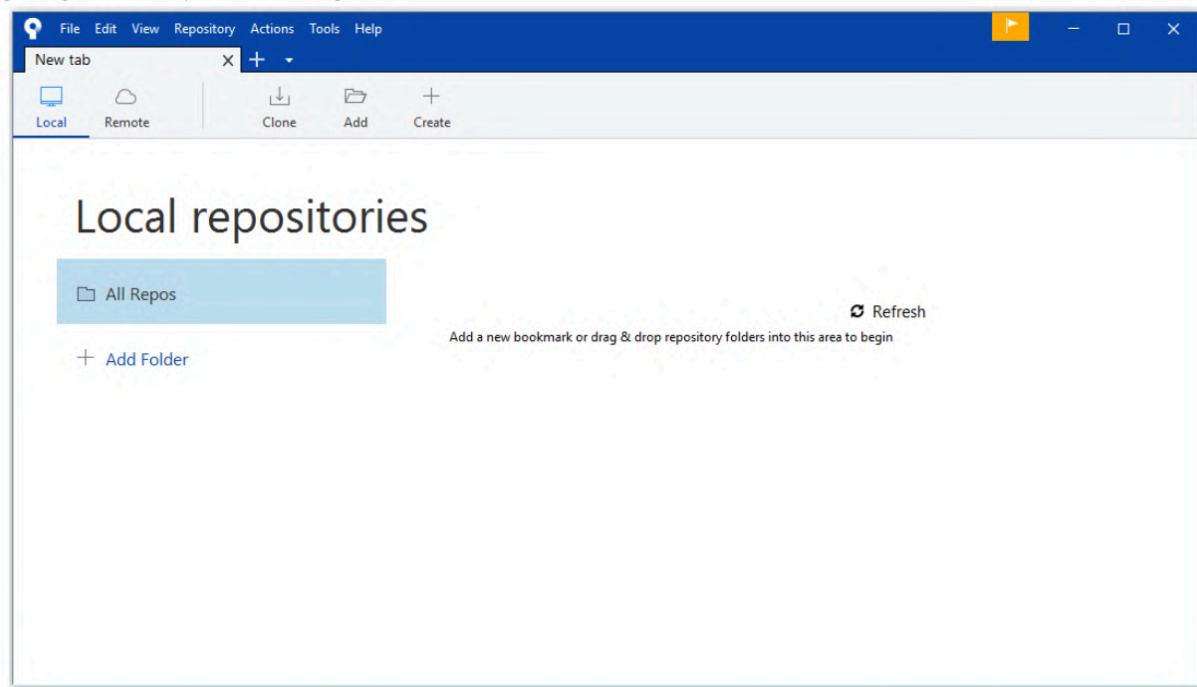
14. Ini proses yang terakhir, yaitu setup preferensi untuk username dan email untuk digunakan oleh Sourcetree sebagai tanda pengenal. Tanda pengenal ini akan melekat pada history perubahan kode yang Anda lakukan, sehingga Anda dan team dapat melacak perubahan kode pada *source code*, dan siapa yang melakukan perubahan tersebut.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



Masukan username dan email yang Anda inginkan. Sebaiknya gunakan username dan email yang sama dengan Akun GitHub anda.

15. Selesai. Proses installasi Sourcetree selesai! Anda akan melihat halaman dashboard Sourcetree yang kini siap untuk digunakan.



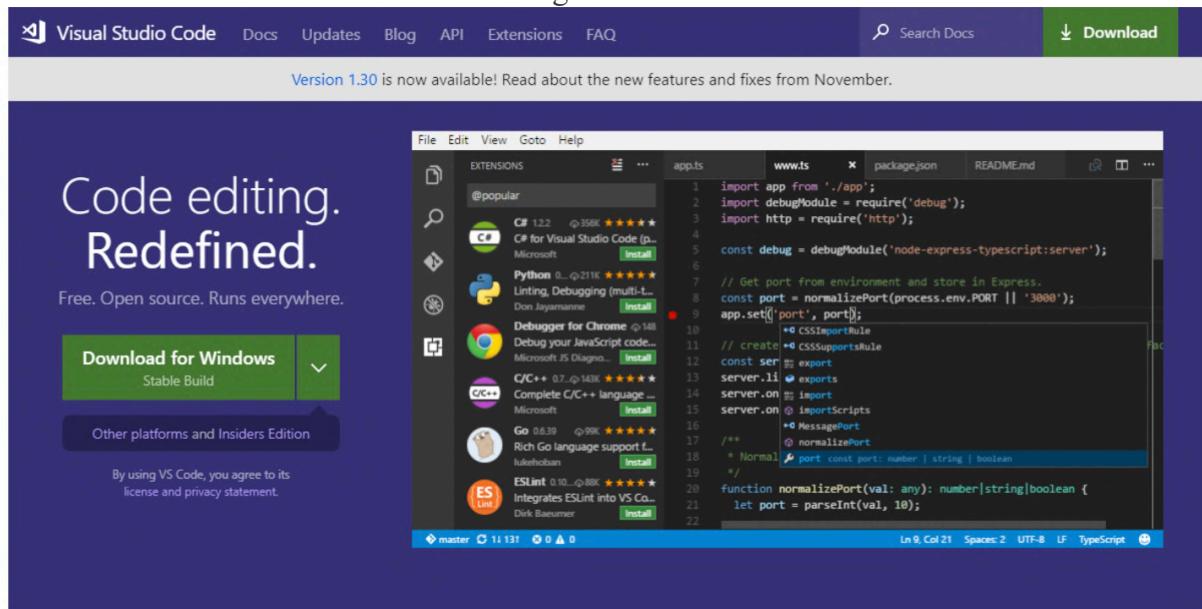
Silakan menuju modul berikutnya untuk menginstall Visual Studio Code.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

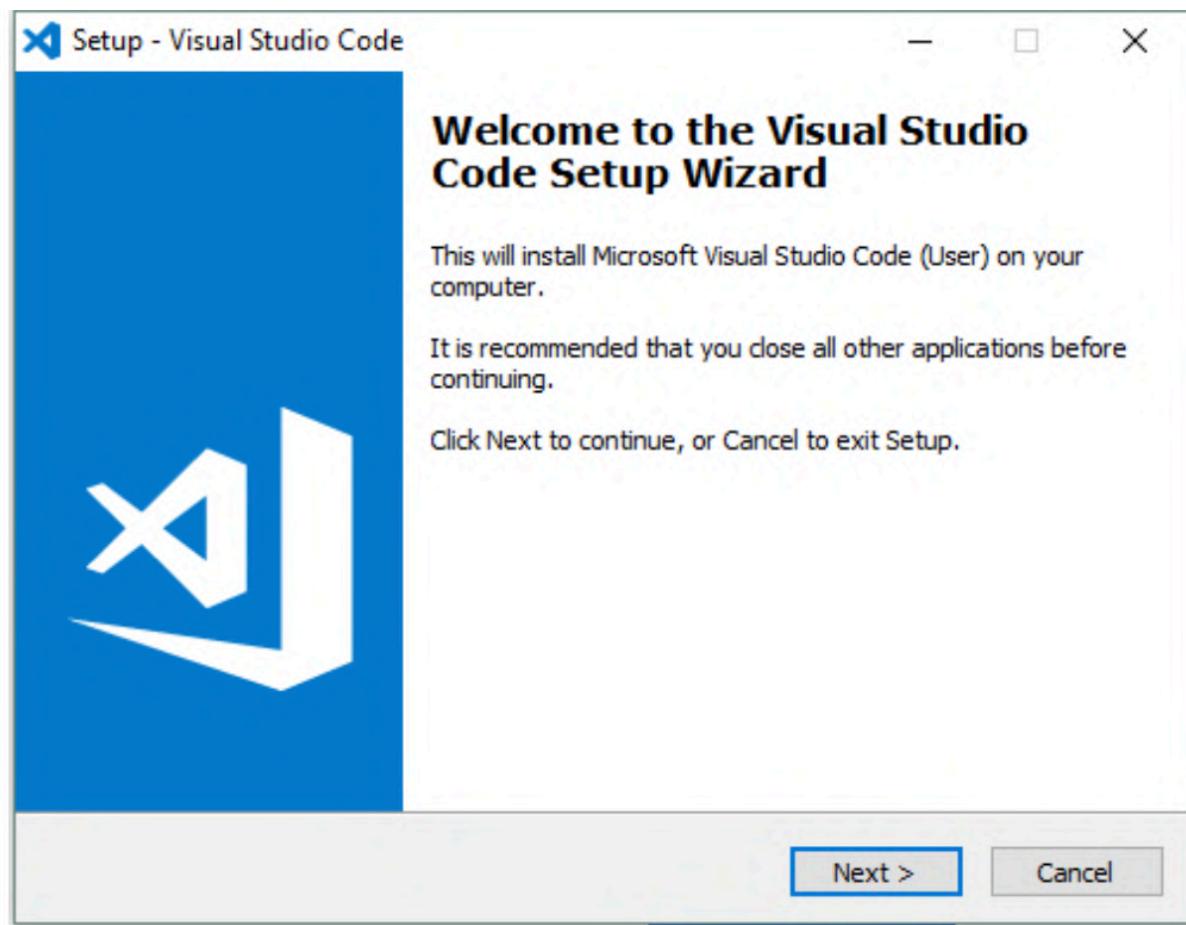
VISUAL STUDIO CODE

1. Buka browser favorit anda dan ketikkan URL berikut : <https://code.visualstudio.com/>
2. Anda akan berada di halaman landing untuk mendownload VS Code.



Klik **Download** atau **Download for Windows**. Halaman ini akan otomatis membaca OS yang digunakan sehingga Anda tidak akan salah men-download.

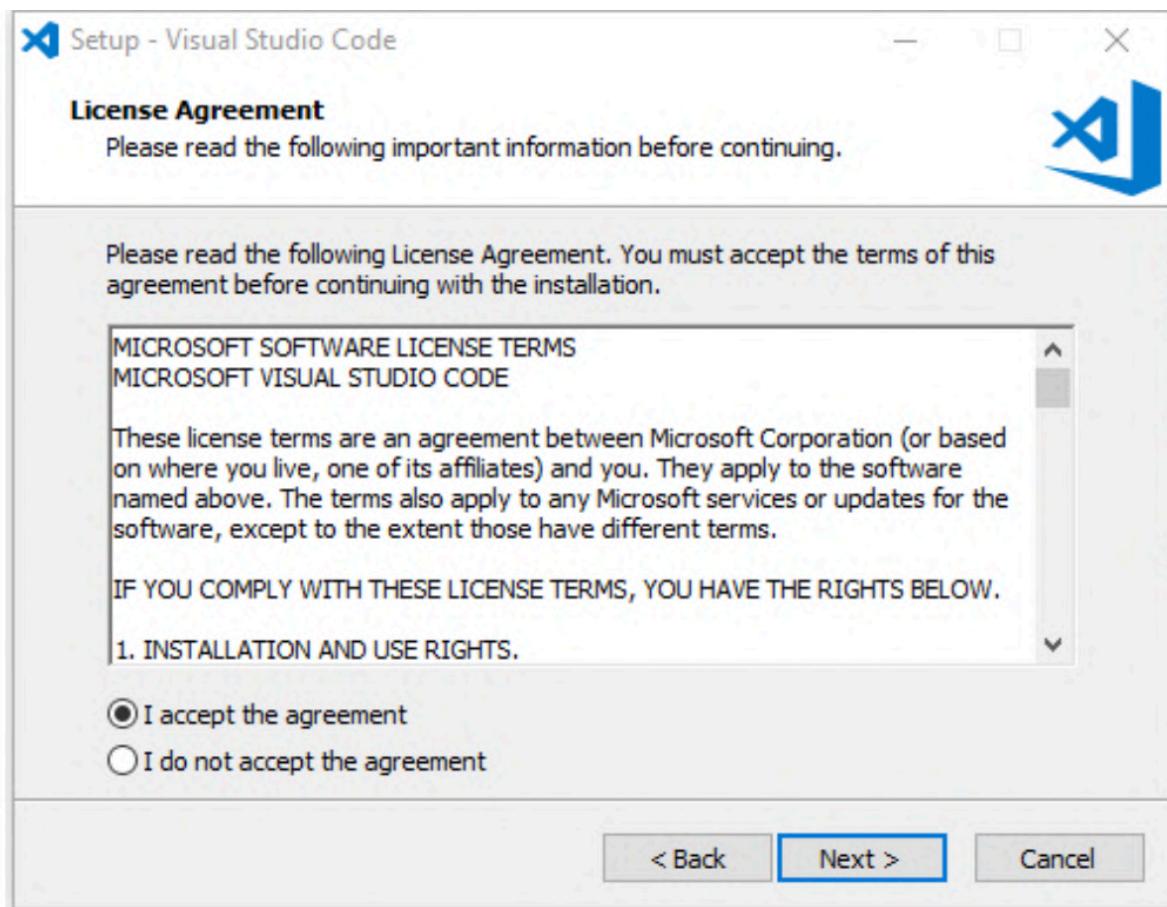
3. Klik dua kali pada file installer yang terdownload. Jendela Installation Wizard akan muncul



Klik **Next**.

4. Anda perlu meng-accept agreement untuk lanjut.

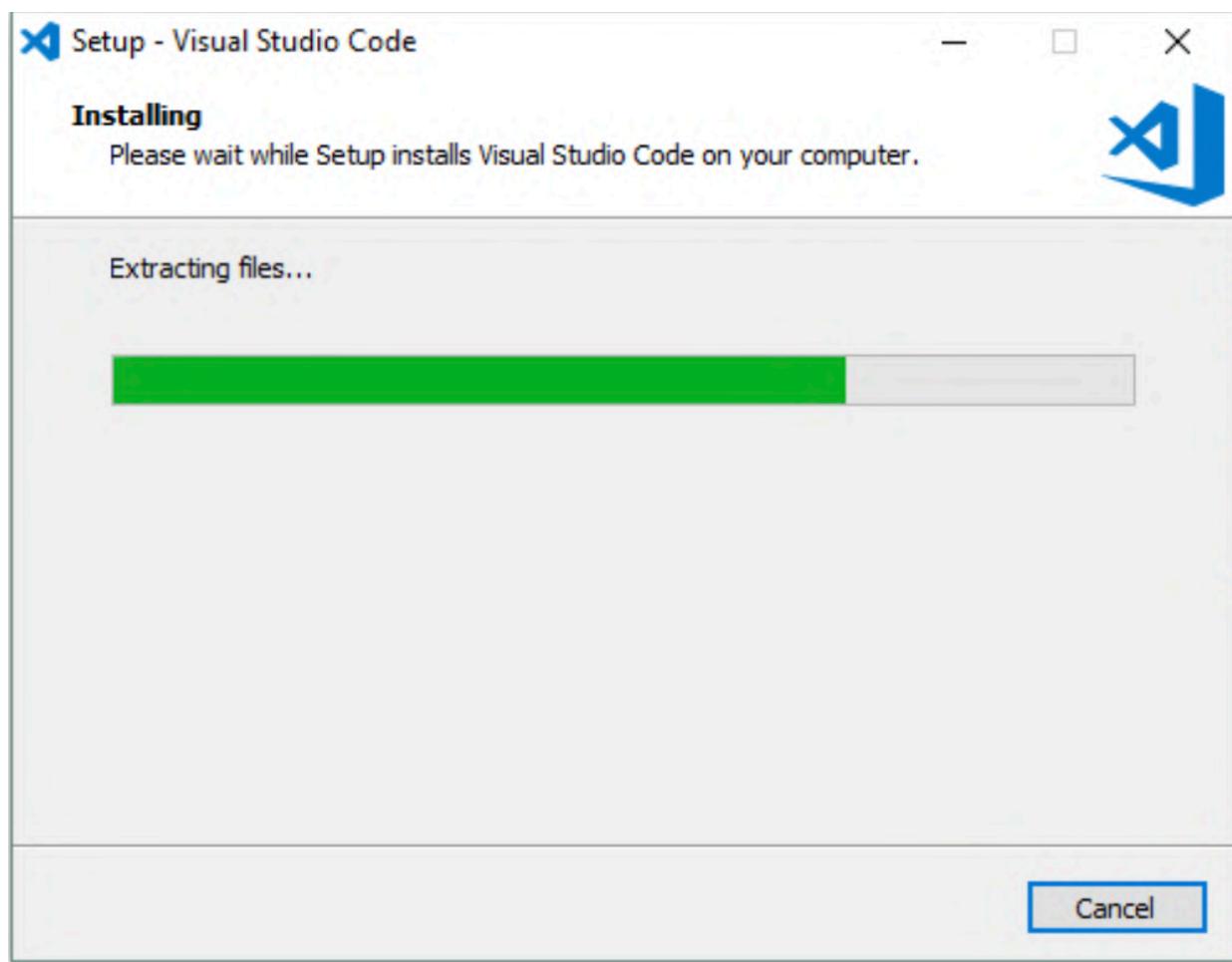
Modul Praktikum Kakas Pengembangan Perangkat Lunak



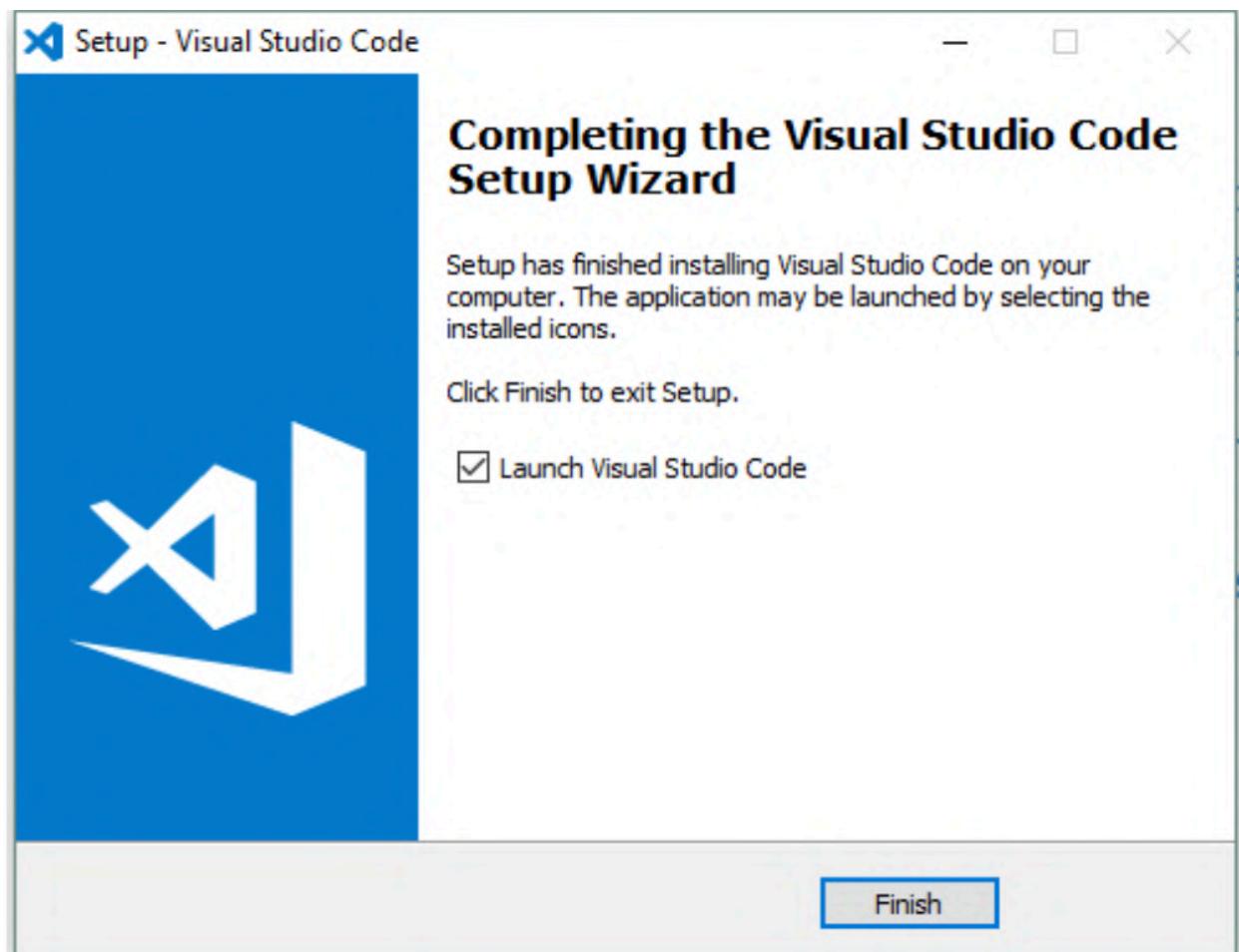
Centang **I accept the agreement**, lalu klik **Next**.

5. Proses instalasi berjalan.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



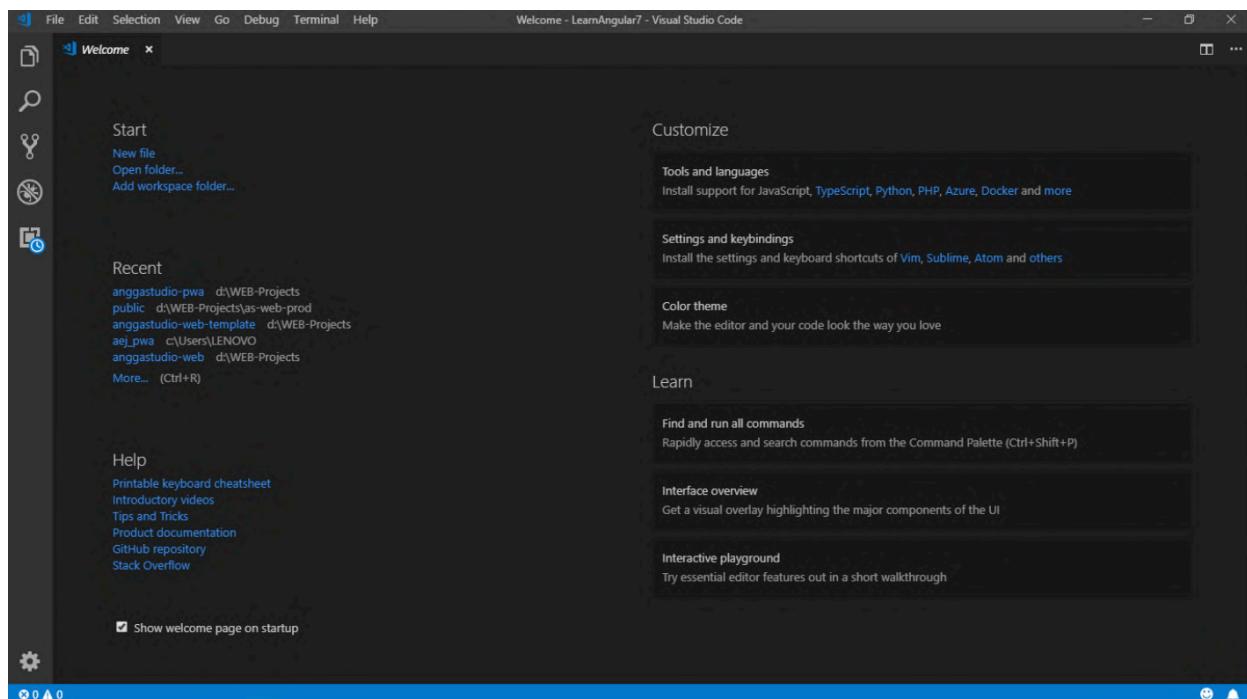
6. Tunggu sampai proses installasi selesai. Proses ini tidak akan memakan waktu yang lama.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Centang Launch Visual Studio Code, lalu klik **Finish**.

7. Visual Studio Code sudah siap untuk digunakan.



Tema gelap atau “dark theme” menjadi default dari tampilan VS Code. Jika Anda menginginkan tema lain, silakan mengurnya di halaman preference.

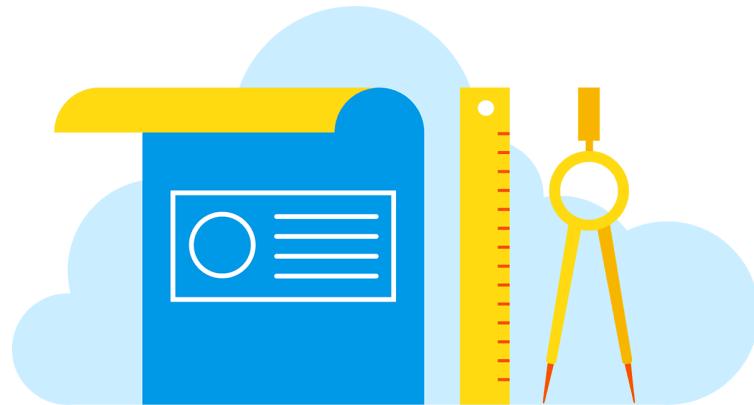
Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 2: Create Repository dan Clone Repository

Deskripsi Pertemuan

Di modul ini, akan diajarkan bagaimana **membuat repository** dan melakukan **clone repository** ke local. Lalu di modul berikutnya kita akan mulai belajar melakukan commit perubahan kode di local, kemudian melakukan push hasil commit ke remote repository. Wow, banyak sekali istilah-istilahnnya. Agar mudah dipahami, kita akan lakukan langkah demi langkah. Langkah pertama adalah dengan membuat remote repository dan meng-clone-nya menjadi local repository.

Syarat Kompetensi

1. Telah memiliki akun akun Git-hub
2. Telah memiliki akun Bitbucket
3. Telah menginstall Sourcetree dan mendaftarkannya di akun Atlassian
4. Telah menginstall Visual Studio Code sebagai code editor atau IDE

Standar Kompetensi

1. Mahasiswa mampu memanfaatkan Git-hub dan mengkombinasikan dengan bitbuclet dan sourcetree dengan menggunakan visual studio code dalam membuat repository
2. Mahasiswa mampu memanfaatkan Git-hub dan mengkombinasikan dengan bitbuclet dan sourcetree dengan menggunakan visual studio code dalam melakukan clone repository

Dasar Teori

Di modul ini, akan diajarkan bagaimana **membuat repository** dan melakukan **clone repository** ke local. Lalu di modul berikutnya kita akan mulai belajar melakukan commit perubahan kode di local, kemudian melakukan push hasil commit ke remote repository. Wow, banyak sekali istilah-istilahnnya. Agar mudah dipahami, kita akan lakukan langkah demi langkah. Langkah pertama adalah dengan membuat remote repository dan meng-clone-nya menjadi local repository.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

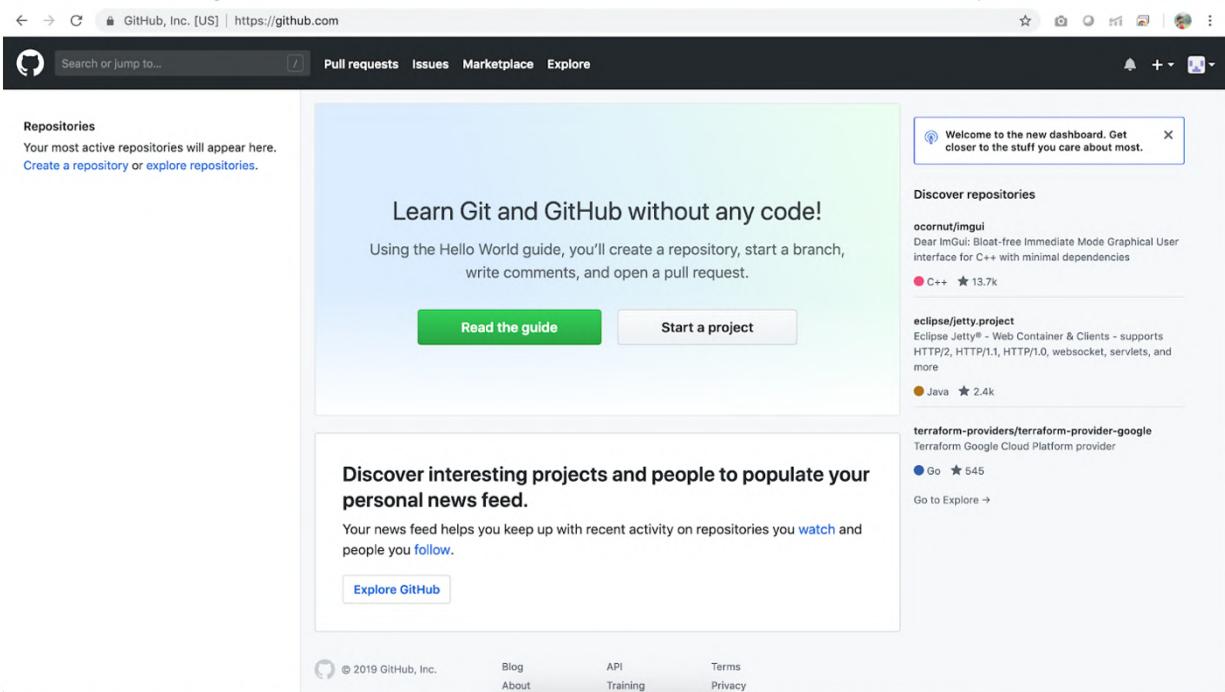
Tugas Pendahuluan

Jelaskan fungsi dari Create Repository dan Clone Repository

Membuat Remote Repository

Langkah-Langkah Praktikum

1. Buka browser favoritmu dan masuk ke halaman GitHub <https://github.com/>.
2. Pastikan untuk sign in dan Anda akan melihat halaman dashboard GitHub seperti ini:

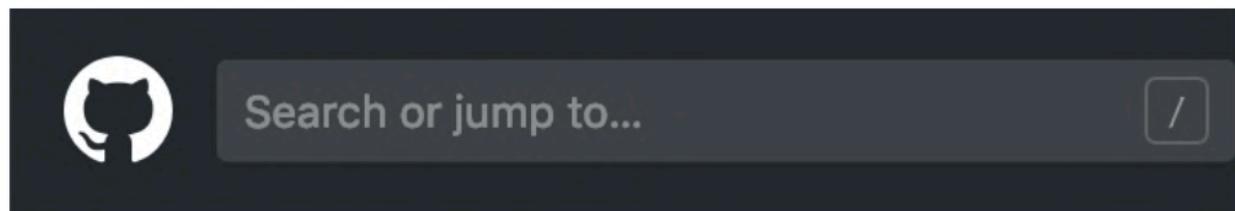


Seperti inilah tampilannya jika akun GitHub Anda belum memiliki repository.

3. Perhatikan kolom di kiri atas, lalu klik pada tulisan **Create a repository** yang berwarna biru.



Modul Praktikum Kakas Pengembangan Perangkat Lunak



Repositories

Your most active repositories will appear here.

[Create a repository](#) or [explore repositories](#).

4. Anda akan dibawa ke halaman pembuatan repository.

Create a new repository

A repository contains all project files, including the revision history.

Owner Repository name *

/

Great repository names are short and memorable. Need inspiration? How about [effective-octo-enigma](#)?

Description (optional)

this repository is created from [github](#)

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: Add a license:

Create repository

Silakan tuliskan “my-repo-from-remote” pada Repository name field. Isikan deskripsi seperti pada contoh, atau Anda bisa menuliskan sendiri deskripsinya semau Anda. Sebaiknya gunakan bahasa Inggris, karena GitHub juga dipakai sebagai social media untuk ajang pamer karya bagi para



Modul Praktikum Kakas Pengembangan Perangkat Lunak

developer.

Pilih public repository untuk saat ini. Private repository akan kita bahas di modul lain. Biarkan saja radio button pada Initialize README, tidak perlu dicentang untuk saat ini. Klik tombol **Create repository** jika semuanya sudah selesai.

5. Selamat! Anda sudah berhasil membuat remote repository pertama Anda.

```
Set up in Desktop or HTTPS SSH https://github.com/arjuna135/my-repo-from-remote.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line
echo "# my-repo-from-remote" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/arjuna135/my-repo-from-remote.git
git push -u origin master

...or push an existing repository from the command line
git remote add origin https://github.com/arjuna135/my-repo-from-remote.git
git push -u origin master
```

Apa

langkah

selanjutnya?

Walaupun Anda sudah memiliki project repository, tapi tidak banyak yang bisa dilakukan jika Anda tidak memiliki local repository. Untuk itu kita perlu melakukan clone remote repository ke local repository.

Di halaman GitHub Anda saat ini, terdapat beberapa dokumentasi yang menunjukkan beberapa perintah terminal untuk mengelola repository. Tapi kita sudah memiliki Sourcetree sehingga perintah tersebut tidak akan kita gunakan.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Membuat Remote Repository Ke Local

Untuk melakukan clone repository, silakan ikuti langkah-langkah berikut:

- Pastikan Anda masih berada di halaman repository yang baru dibuat. Temukanlah sebuah url dengan scheme https seperti berikut:

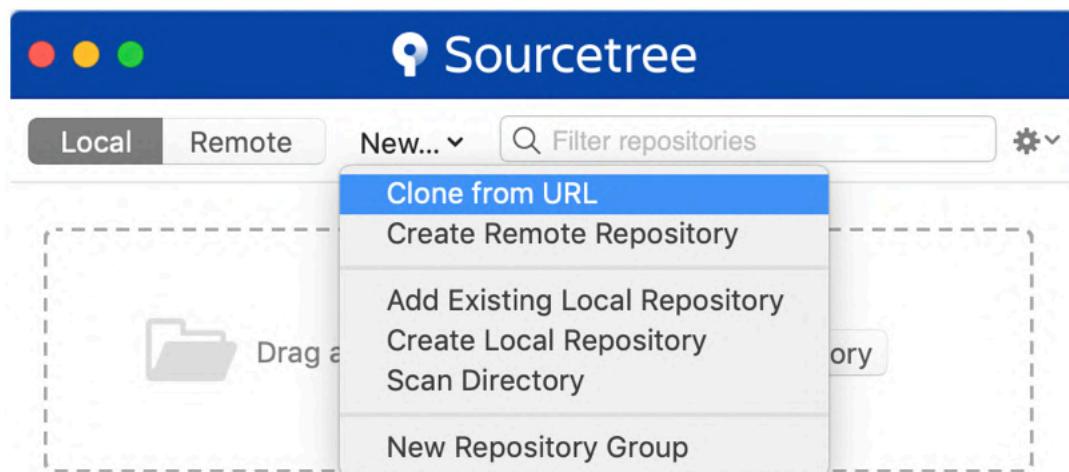
Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/arjuna135/my-repo-from-remote.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a `README`, `LICENSE`, and `.gitignore`.

Copy-kan url tersebut, pastikan tombol HTTPS dalam keadaan aktif, ditandai dengan warnanya yang lebih gelap.

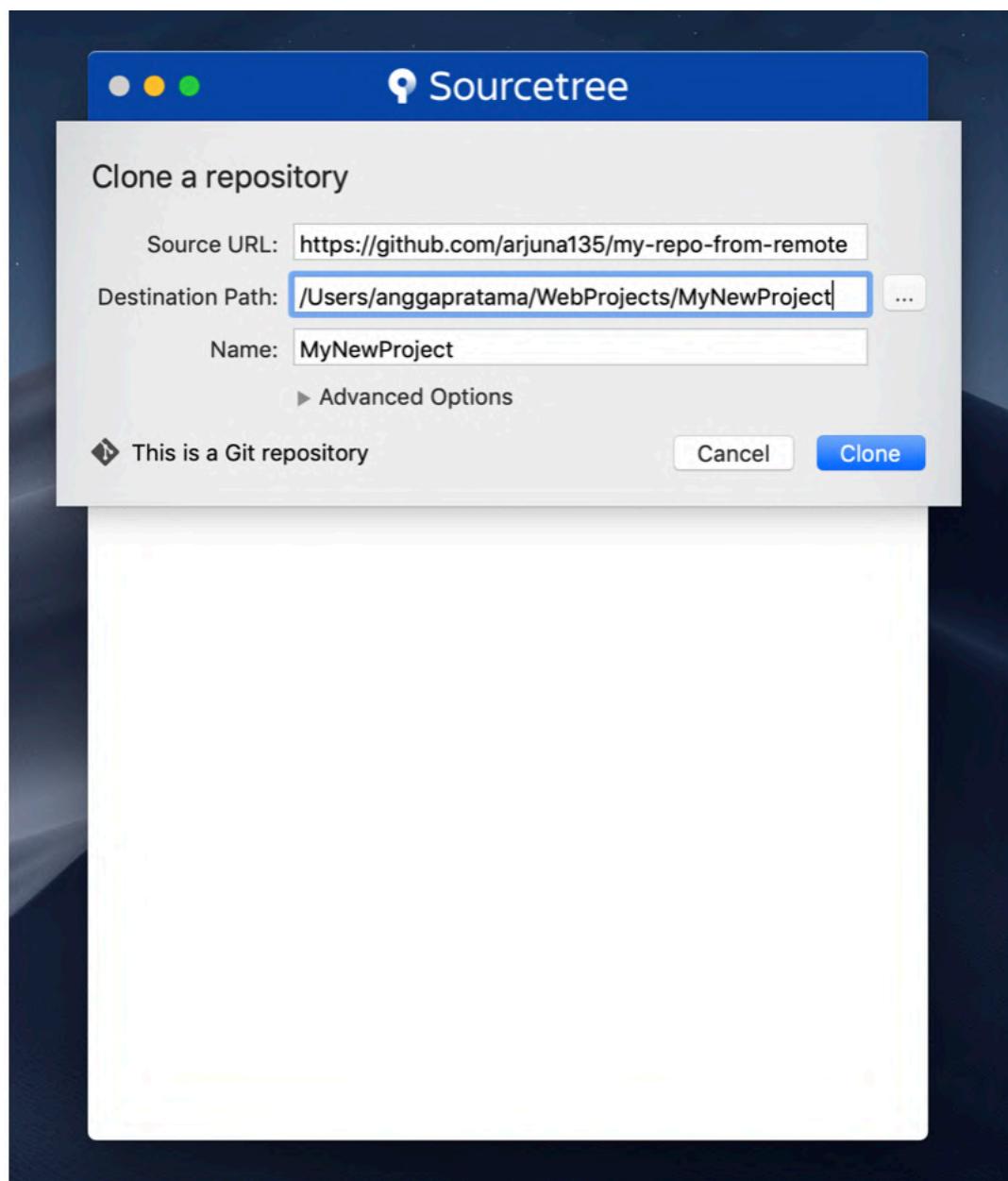
- Selanjutnya kita buka Sourcetree. Di modul ini saya akan menggunakan Sourcetree versi Mac.



Akan terbuka jendela kecil seperti di atas. Versi Windows akan menampilkan jendela lebih besar. Temukanlah Tombol **New** pada Mac, atau icon (+) pada Windows untuk membuat repository baru di local. Metode pembuatan local repository, kita lakukan dengan cara clone dari remote repository. Selanjutnya klik **New** lalu klik **Clone from Url**.

Modul Praktikum Kakas Pengembangan Perangkat Lunak

- Paste-kan url remote repository tadi ke dalam field url.



Pilih destinasi folder untuk menyimpan local repository. Jika melihat gambar di atas, terdapat WebProjects folder sebagai path, lalu garis miring, lalu MyNewProject. WebProjects adalah folder yang telah saya buat sebelumnya di dalam direktori komputer saya, sedangkan MyNewProject adalah nama folder yang belum saya buat, tetapi akan dibuat otomatis oleh Sourcetree, sehingga saya hanya menuliskan namanya saja. Begitu juga dengan field Name, akan otomatis di populate oleh Sourcetree sebagai nama local repository Anda.

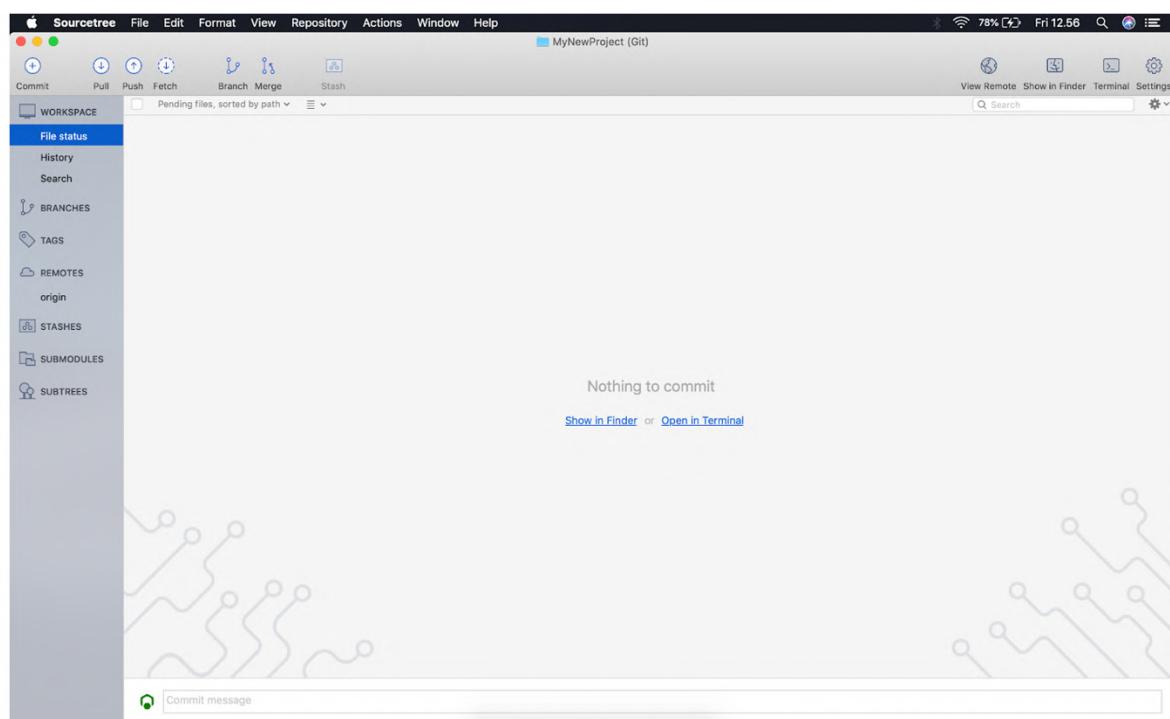


Modul Praktikum Kakas Pengembangan Perangkat Lunak

Kemudian perhatikan informasi yang ada di kiri bawah. Jika terdapat tulisan **This is a Git Repository**, artinya remote repository Anda sudah terlacak oleh Sourcetree dan proses clone siap dijalankan. Jika informasinya berwarna merah, mungkin terdapat kesalahan pada url sehingga Sourcetree tidak bisa melacak remote repository Anda.

Klik **Clone** jika semua sudah siap.

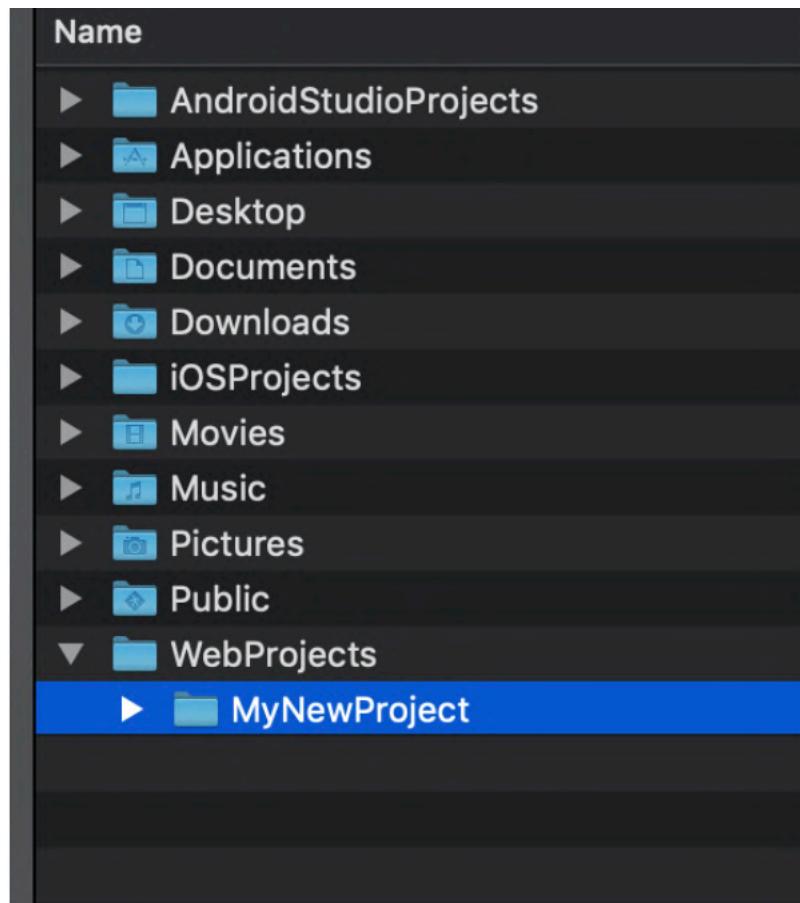
4. Akan muncul Jendela seperti ini pada Sourcetree:



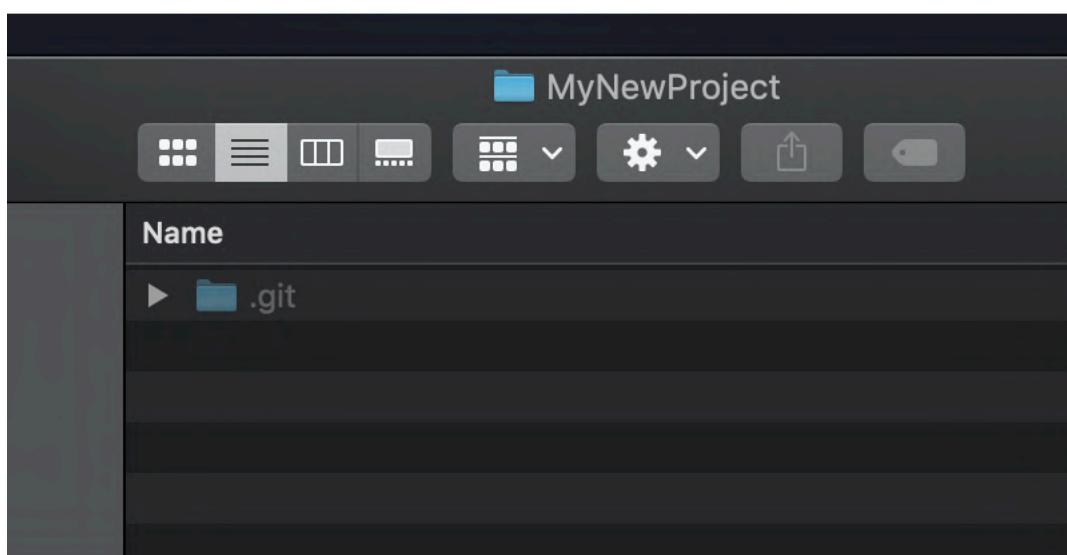
Berikut adalah tampilan jendela kerja pada Sourcetree versi Mac. Tampilan pada versi Windows akan sedikit berbeda. Perhatikan nama repository yang berada di tengah atas.

Selamat! Anda telah berhasil mengkloning remote repository ke local repository Anda. Untuk membuktikan hal tersebut, silakan buka finder di Mac atau explorer di Windows untuk melihat keberadaan folder local repositorynya.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



Folder MyNewProject telah berhasil dibuat di direktori folder WebProjects. Kemudian coba masuk ke dalam MyNewProject Folder, lalu tampilkan semua hidden folder.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

Akan terdapat hidden folder .git di dalam local repository. Hal ini menandakan bahwa folder MyNewProject adalah git repository. Semua aktivitas yang dilakukan di dalam folder ini akan terekam otomatis oleh git dan divisualisasikan oleh Sourcetree. Oleh karena itu, belum lengkap rasanya jika tidak langsung mencoba beberapa baris kode untuk kemudian disimpan ke remote repository. Baiklah saatnya kita menuliskan kode pertama kita lalu melakukan commit dan push.

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 3: Commit and Push

Deskripsi Pertemuan

Commit and Push merupakan suatu fitur yang terdapat dalam sourcetree yang memungkinkan dapat melakukan commit and push (upload source code) kedalam Git-Hub. Tujuan utamanya adalah dapat melakukan dokumentasi source code dalam Git-Hub.

Syarat Kompetensi

1. Sudah menginstall aplikasi Visual Studio Code
2. Sudah memiliki akun Git-Hub
3. Sudah menginstall Sourcetree

Standar Kompetensi

1. Diharapkan mampu mengimplementasikan fungsi commit
2. Diharapkan mampu mengimplementasikan fungsi push

Dasar Teori

- Git commit : untuk menyimpan perubahan yang dilakukan, tetapi tidak ada perubahan pada *remote repository*.
- Git push : untuk mengirimkan perubahan file setelah di commit ke *remote repository*.

Tugas Pendahuluan

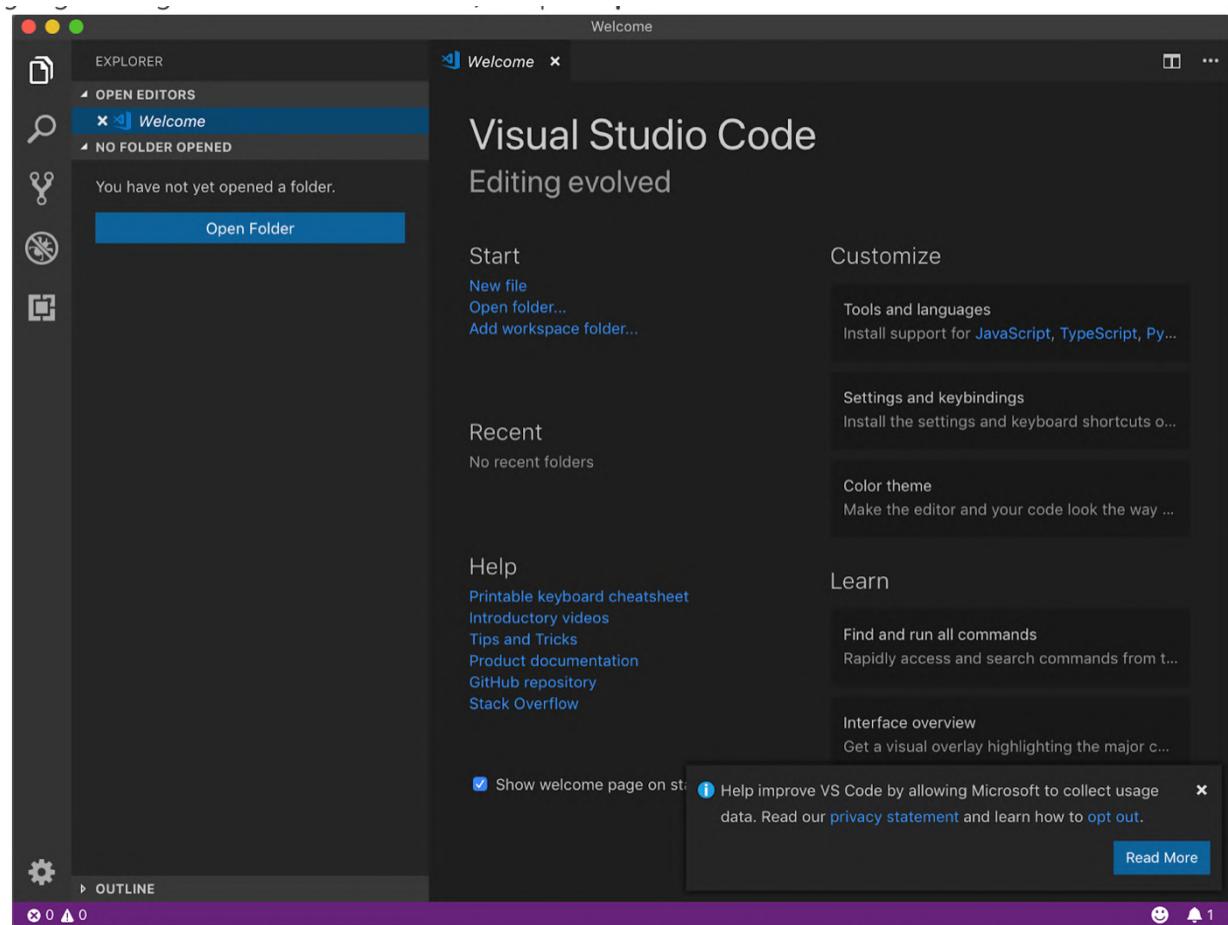
Jelaskan fungsi dari Commit dan Push



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Langkah-Langkah Praktikum

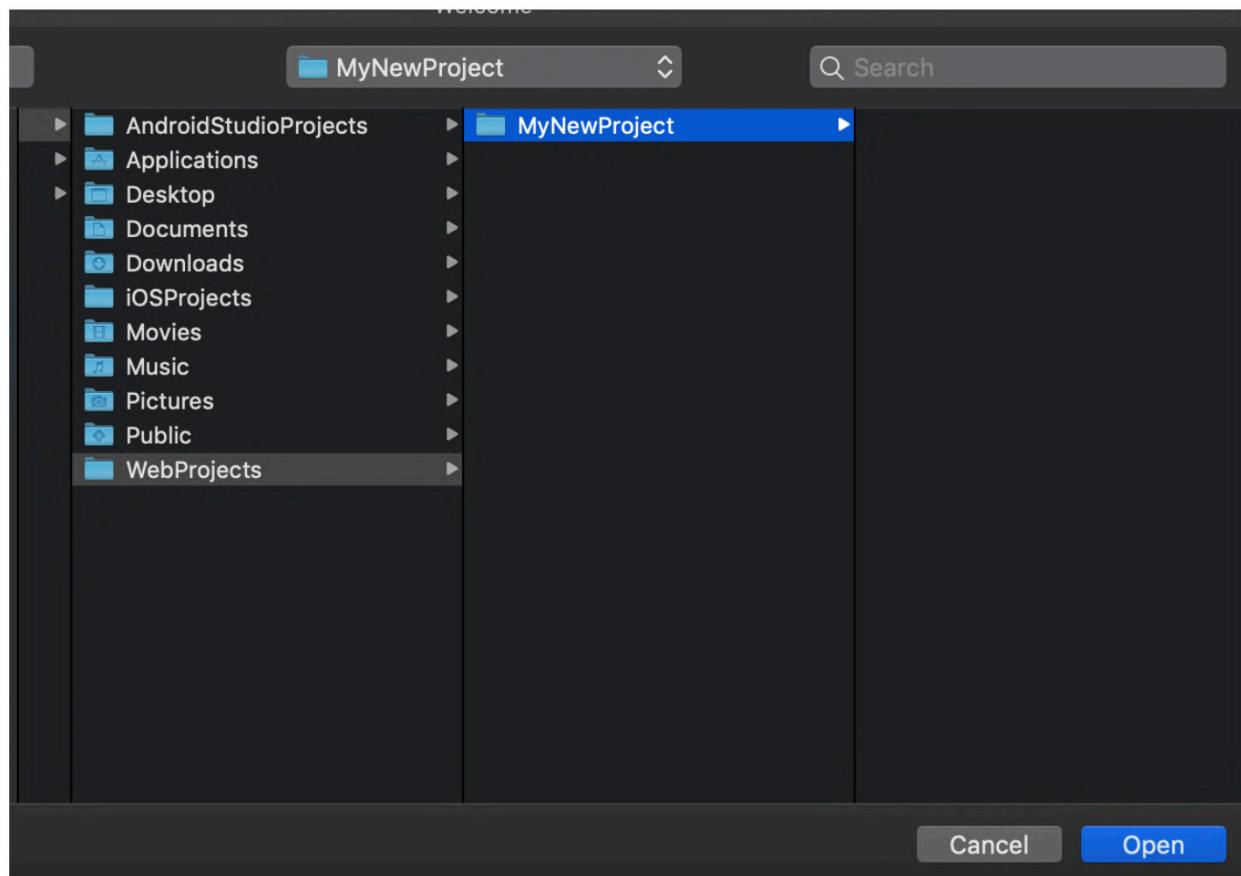
1. Buka Visual Studio Code.
2. Perhatikan **icon explorer**, yaitu icon kertas bertumpuk di paling **kiri atas** dari jendela VSCode yang sedang buka. Klik icon tersebut, lalu pilih **open folder**.



Atau Anda juga melakukannya dengan klik "**open folder**" pada tab **Welcome > Start**.

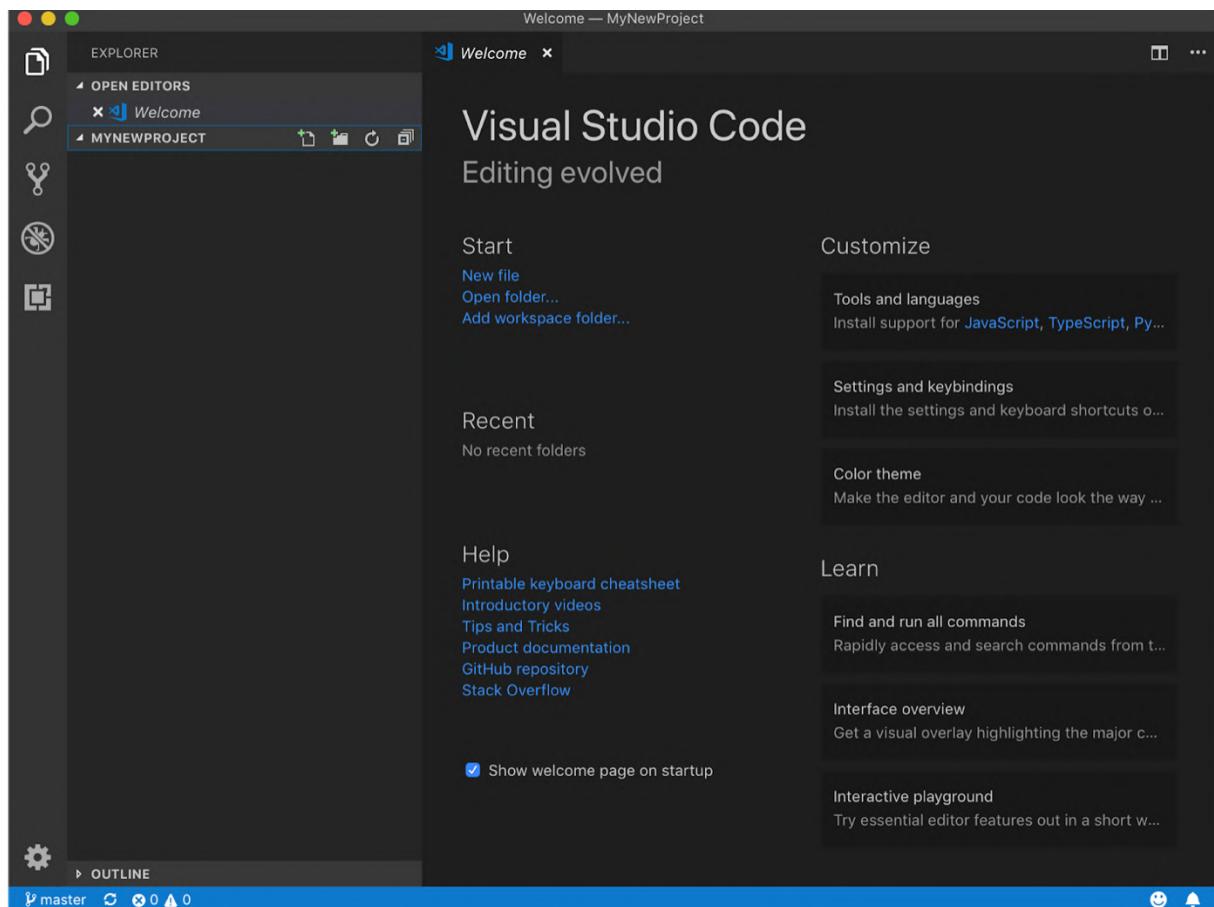
3. Cari dan pilih folder MyNewProject yang telah kita buat. Lalu klik **Open**.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



4. Penampilan Explorer akan tampak seperti ini. Folder MyNewProject kini terdaftar sebagai root directory pada VSCode. Folder tentu masih kosong karena hanya mengandung folder .git yang bersifat hidden folder.

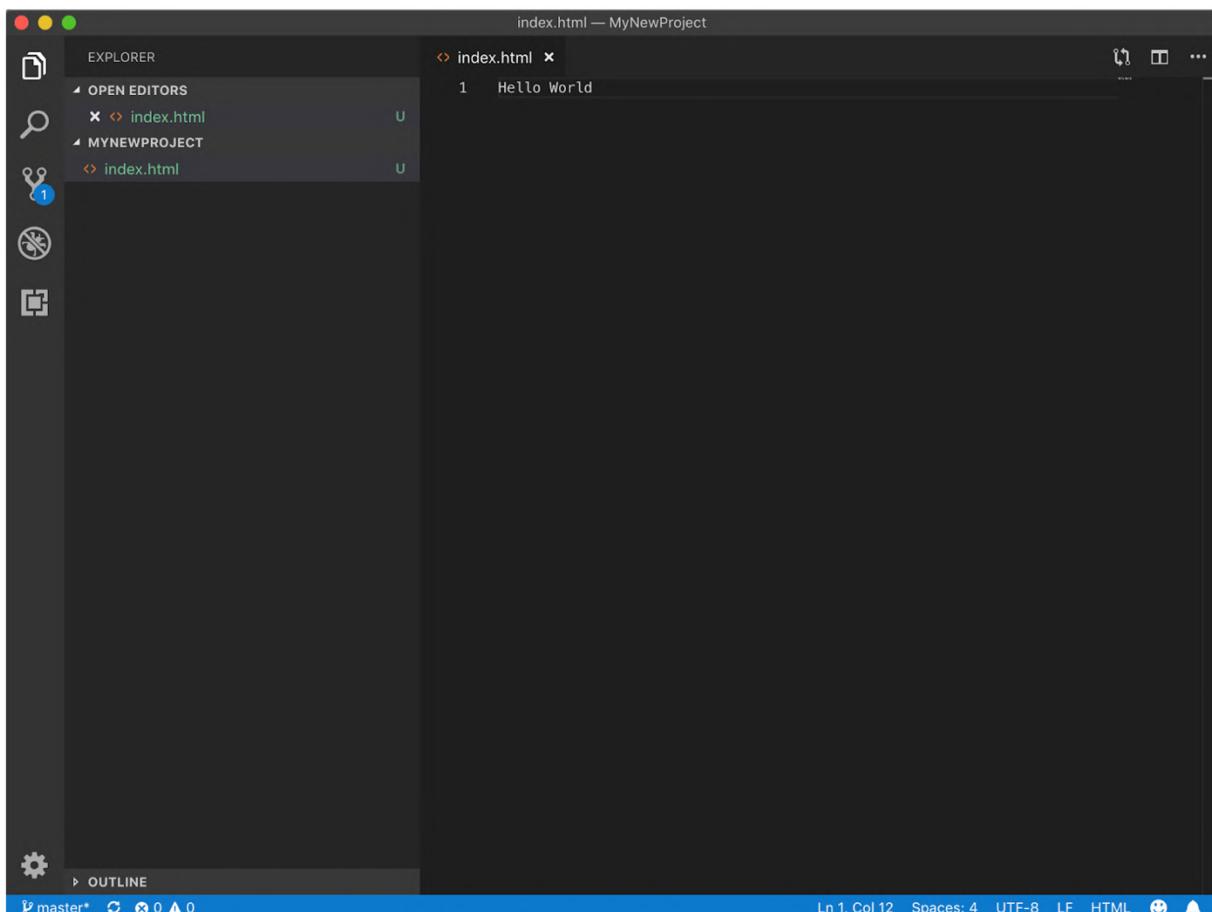
Modul Praktikum Kakas Pengembangan Perangkat Lunak



Anda juga bisa menutup tab Welcome jika dirasa mengganggu.

5. Kini kita akan membuat satu file baru di dalam folder MyNewProject. Perhatikan area explorer, temukan beberapa icon di **sebelah kanan** folder MyNewProject. Anda bisa langsung membuat file baru dengan klik pada icon kertas dengan tanda **plus(+) berwarna hijau**. Klik icon tersebut, tuliskan “index.html” sebagai nama file.

Modul Praktikum Kakas Pengembangan Perangkat Lunak

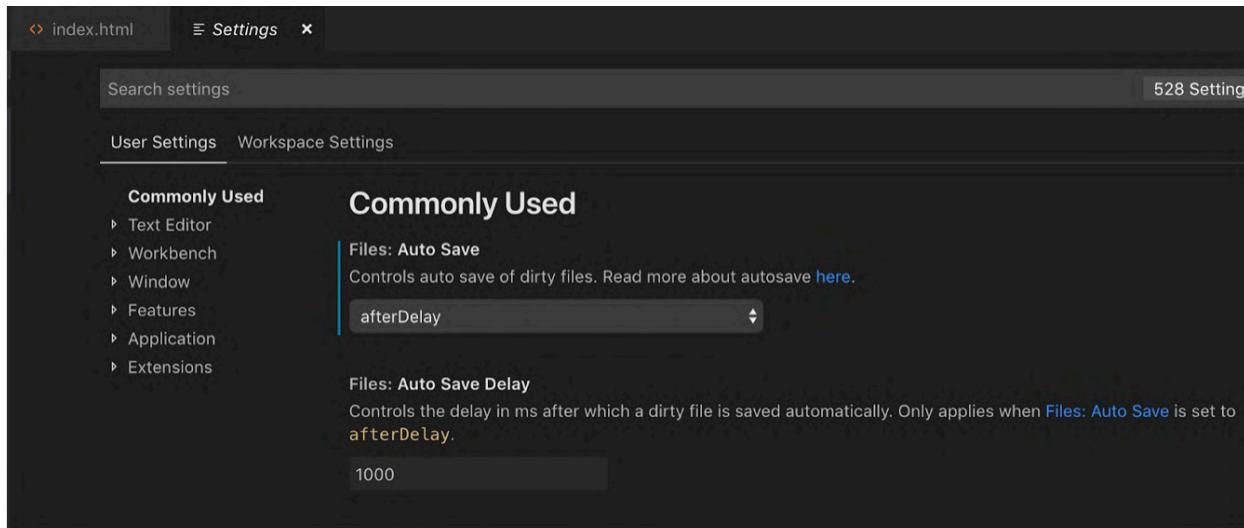


Tab baru akan muncul di area editor. Tuliskan “Hello World” lalu simpan perubahan dengan kombinasi **cmd + s** pada mac atau **ctrl + s** pada windows.

Ya, secara default VSCode menonaktifkan fitur auto save. Jika Anda ingin mengaktifkan fitur auto save, lakukan langkah berikut:

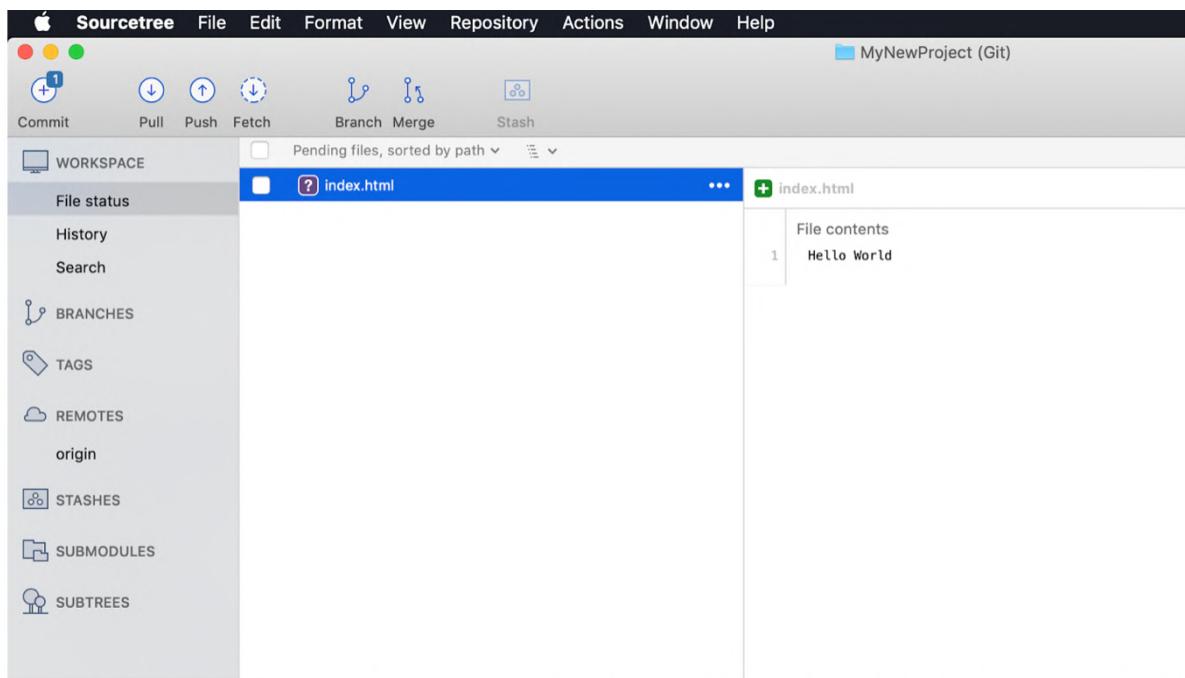
- Buka **Settings**. Mac dan Windows akan sedikit berbeda untuk menu ini.
- Temukan konfigurasi **autosave**. Secara default fitur autosave bernilai off. Ubah nilai menjadi after delay, dan pastikan delay tersebut memiliki nilai 1000 atau 1 detik. Anda bisa memilih konfigurasi lain sesuai selera.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



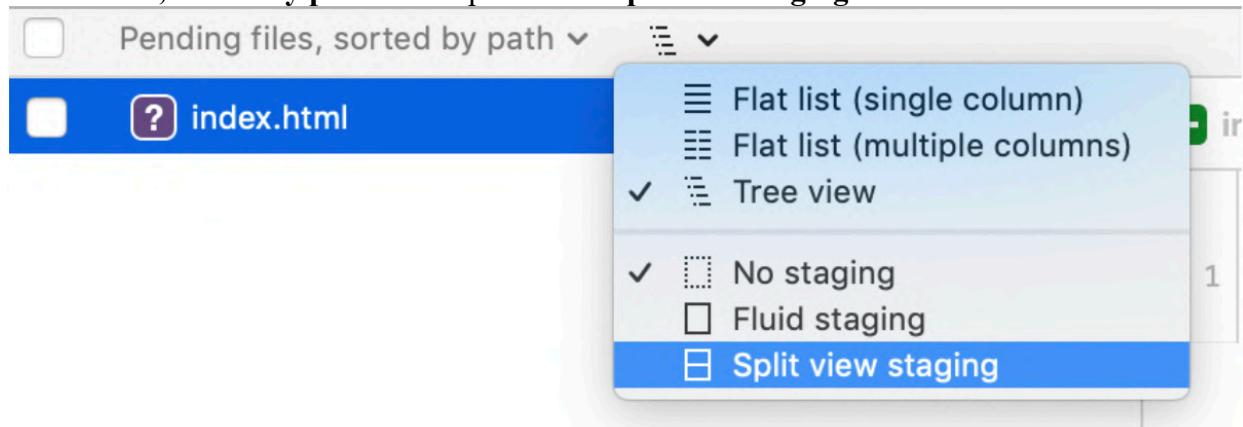
6. Pada tahap ini Anda telah membuat file baru bernama index.html pada folder MyNewProject. Jika Anda perhatikan area explorer pada VSCode, Anda akan menemukan file index.html berwarna hijau. Warna hijau menandakan bahwa file tersebut terdeteksi sebagai file baru oleh Git. Saat ini sudah banyak text editor ataupun IDE yang melakukan behavior yang sama. Berikut beberapa warna indikator file yang terekam oleh Git dan divisualisasikan oleh Text Editor atau IDE: Hijau, Putih/Hitam, Biru, Merah, dan Kuning. Lain tools lain pula warnanya, akan tetapi hal-hal yang diindikasikan tetaplah sama, di antaranya warna-warna tersebut mengindikasikan :
 1. File baru (biasanya hijau atau biru),
 2. Tidak terdapat perubahan pada file ini (biasanya putih atau hitam),
 3. Terdapat perubahan pada file ini (biasanya biru, atau hijau, atau kuning),
 4. File ini gagal di-compile karena terdapat satu atau beberapa line error (biasanya merah).
7. Visualisasi perekaman oleh Git juga dilakukan oleh sourcetree. Silakan buka Sourcetree.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



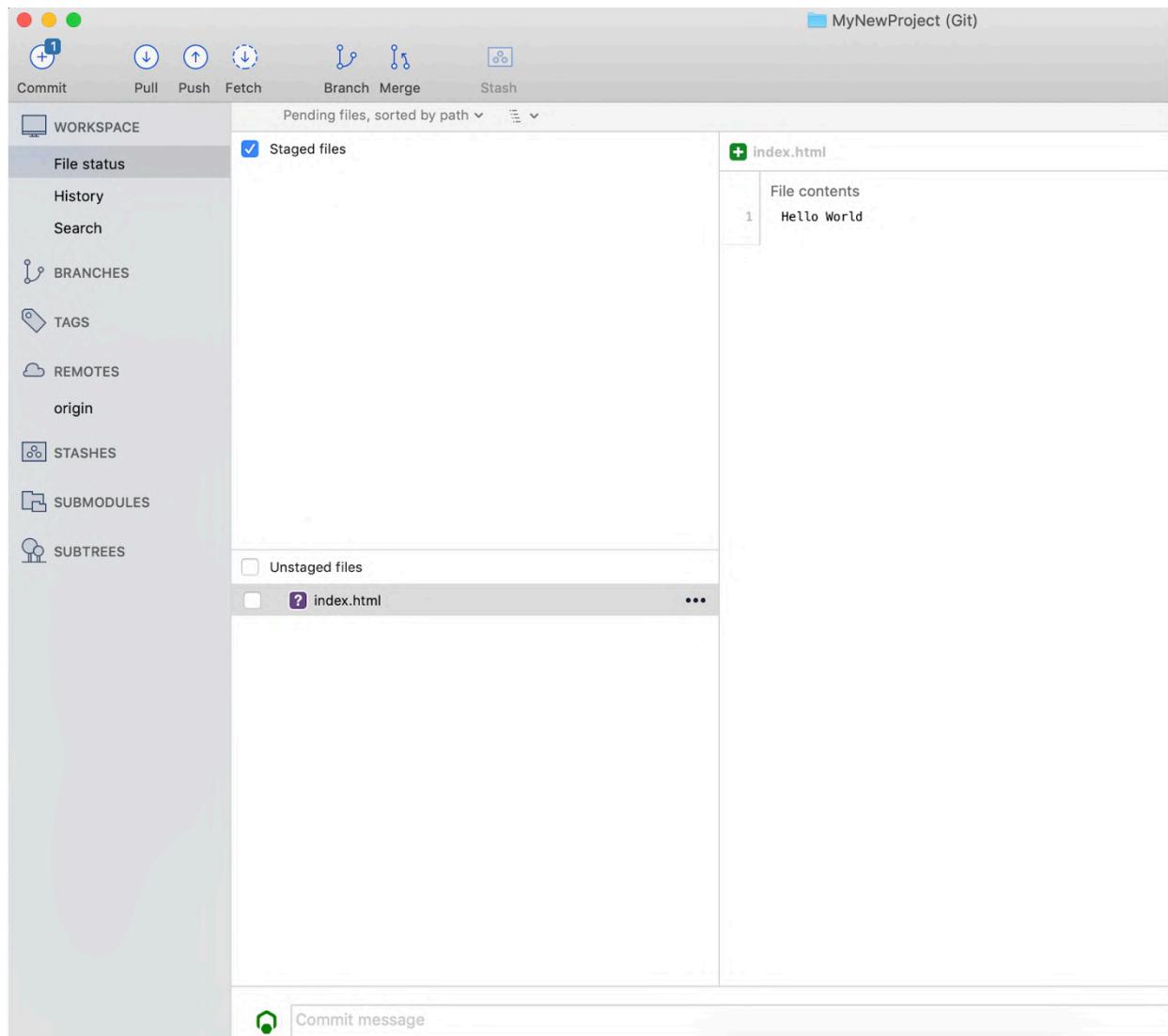
Perhatikan Gambar. File status pada sourcetree menunjukkan bahwa terdapat file baru bernama index.html dengan content “Hello World” pada line nomor 1.

8. Sekarang saatnya kita melakukan commit untuk file baru ini. Tapi sebelumnya, saya ingin memperkenalkan staging dan unstaging area. Klik icon di sebelah kanan tulisan “Pending files, sorted by path”. Lalu pilih mode **split view staging**.



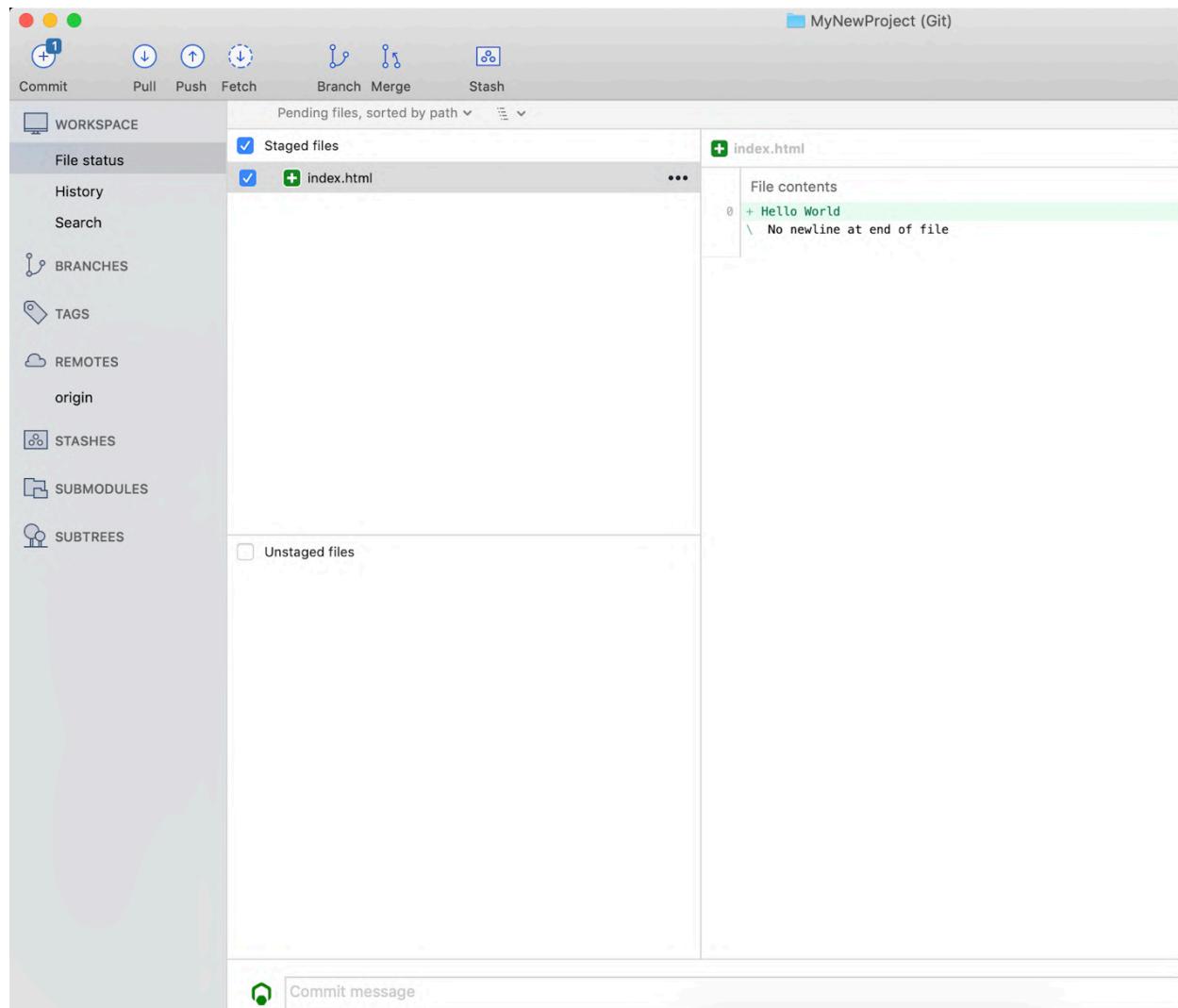
Area file status kini terpecah menjadi dua area, staging files, dan unstaging files. Apa itu staging dan unstaging files akan kita bahas di modul berikutnya. Untuk saat ini anda bisa perhatikan bahwa index.html yang dibuat berada di area unstaged files.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



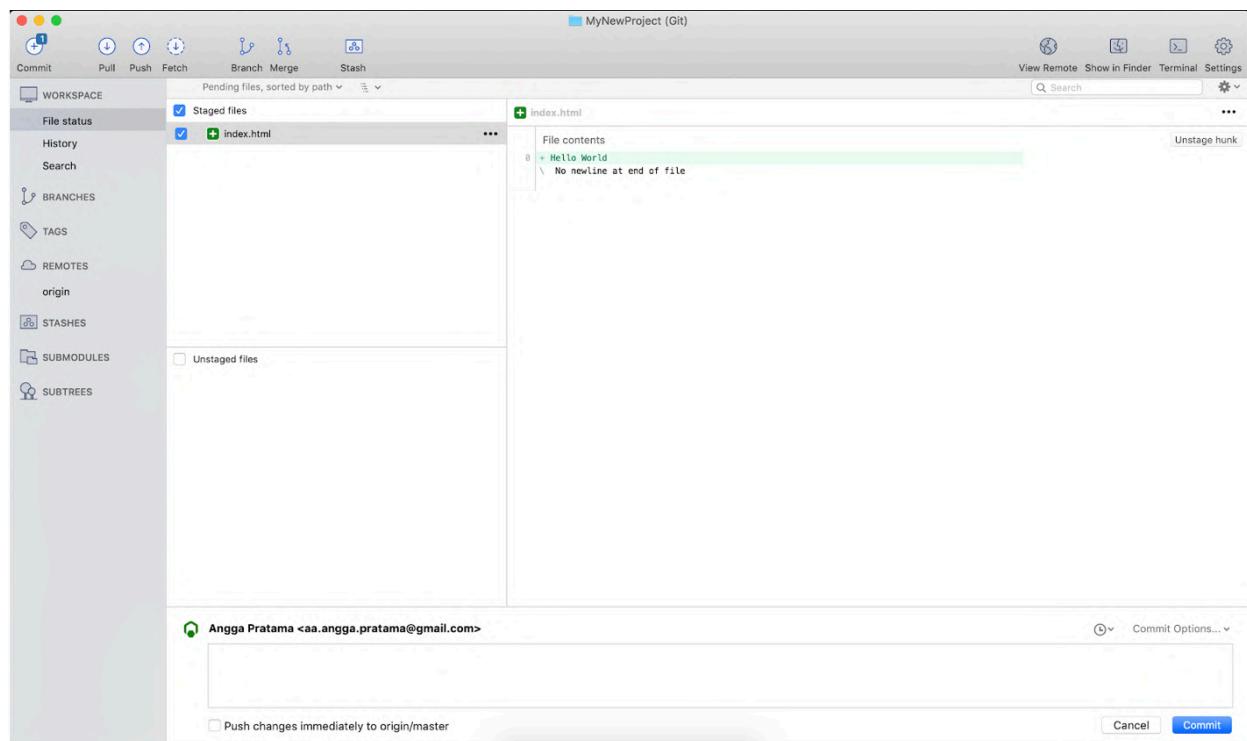
9. Agar bisa melakukan commit, file yang hendak di-commit harus terlebih dahulu masuk ke area staging. Silakan klik tombol checklist di sebelah kiri file index.html. Secara otomatis file index.html akan naik dan kini berada di area staging.

Modul Praktikum Kakas Pengembangan Perangkat Lunak

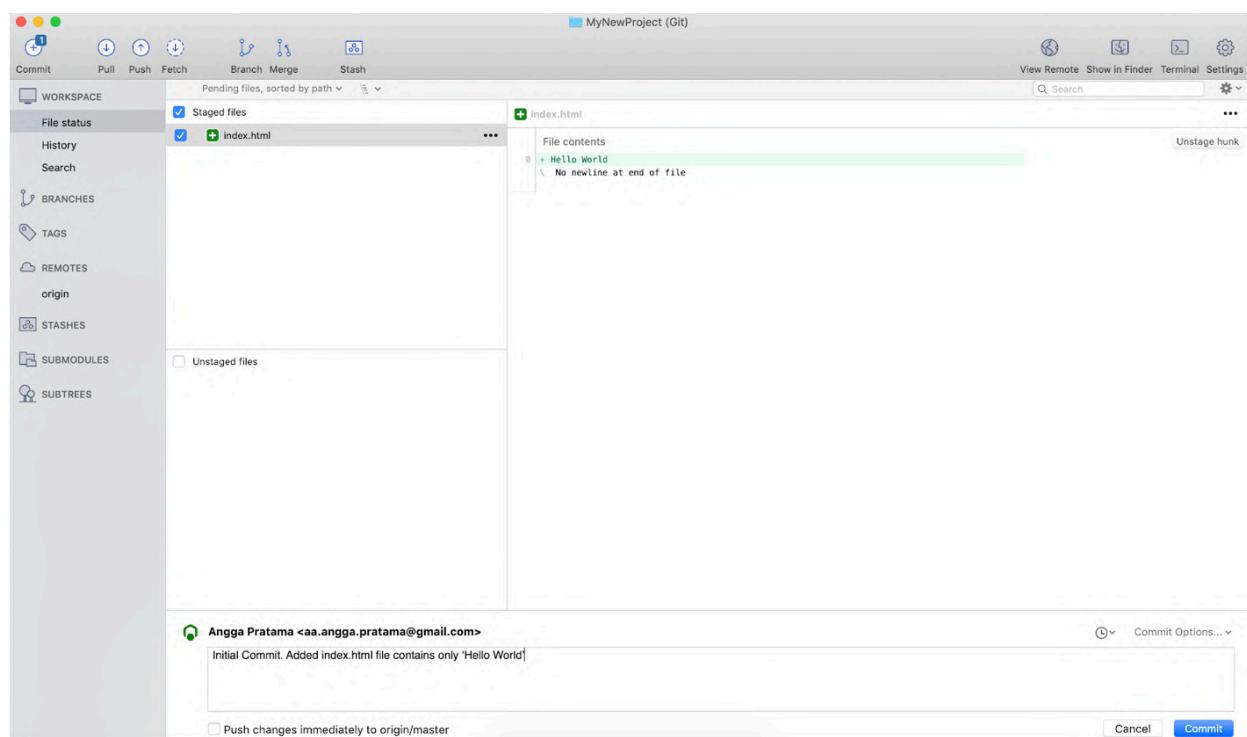


10. Pastikan file index.html di area staging ter-check. Kemudian silakan klik tombol + di **kiri atas**dengan commit. Setelah diklik, maka area commit message di bagian bawah akan melebar seperti gambar berikut :

Modul Praktikum Kakas Pengembangan Perangkat Lunak



11. Tuliskan commit message di text area. Message ini digunakan sebagai deskripsi peng-commit-an, agar perubahan yang dilakukan memiliki detail deskripsi sehingga jejak perubahan dapat dimengerti. Tuliskan message dalam bahasa inggris. Tujuannya adalah agar project Anda dapat juga dimengerti oleh orang-orang di seluruh dunia. Tuliskan message seperti ini : “Initial Commit. Added index.html file contains only ‘Hello World’”.

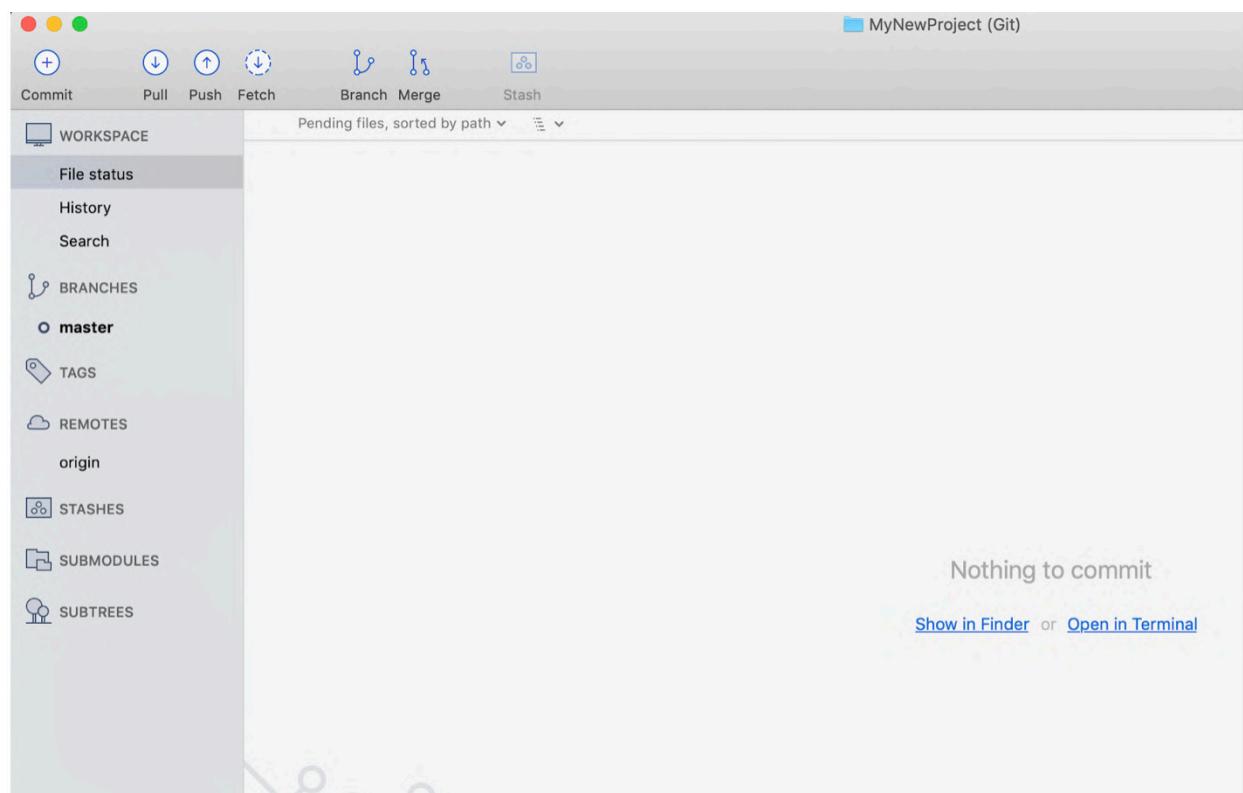




Modul Praktikum Kakas Pengembangan Perangkat Lunak

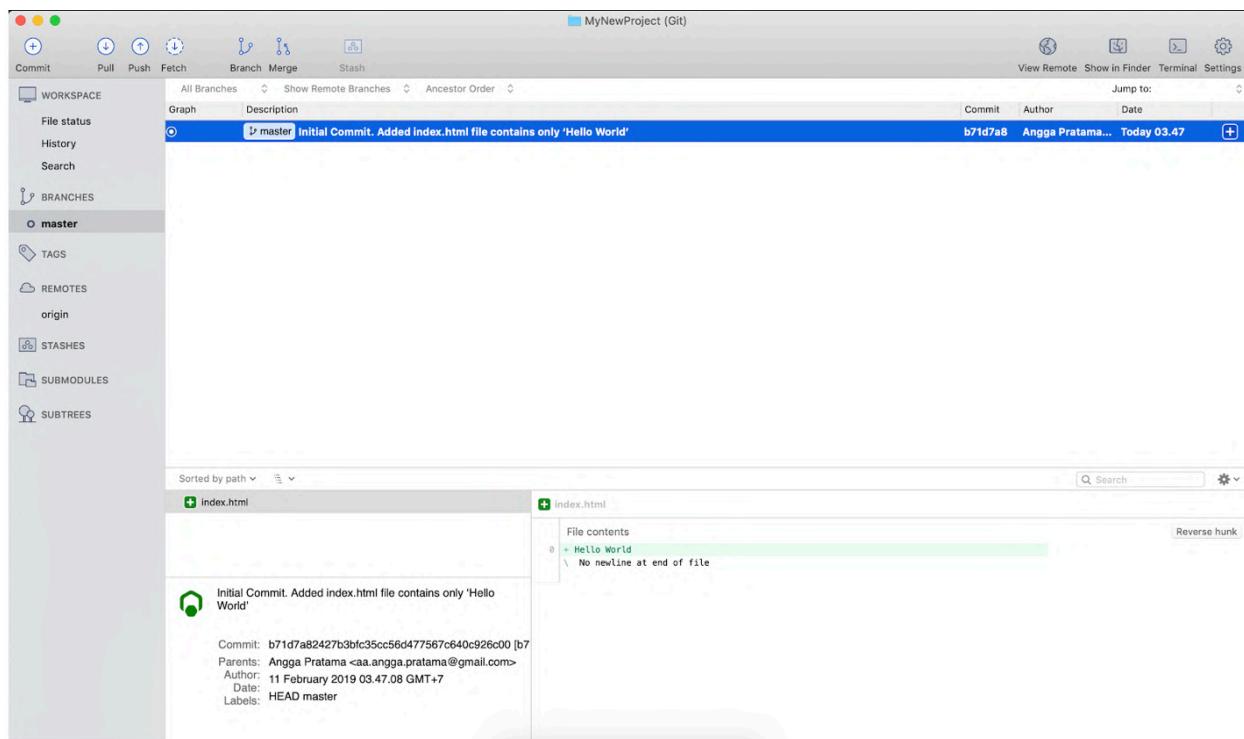
Perhatikan tulisan “**Push changes immediately ...**”, biarkan saja seperti itu, tidak perlu di-check. Jika Anda men-check ‘**push immediately**’, hasil commit akan segera langsung di-push ke remote repository. Bukan seperti itu yang kita inginkan sekarang. Cukup klik tombol commit di sebelah **kanan bawah**.

12. Jendela file status akan menghilang karena semua perubahan pada workspace area sudah di-commit. Perhatikan kolom sebelah kiri, temukan tulisan “**Branches**”, hover mouse pointer Anda ke tulisan tersebut, lalu tekan ‘show’. Seluruh branch pada repository local Anda kini terlihat. Akan tetapi saat ini hanya ada branch master saja.



13. Klik pada branch master untuk melihat seluruh history peng-commit-an yang terjadi pada branch ini. Saat ini, baru hanya satu saja aktivitas commit pada branch master.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



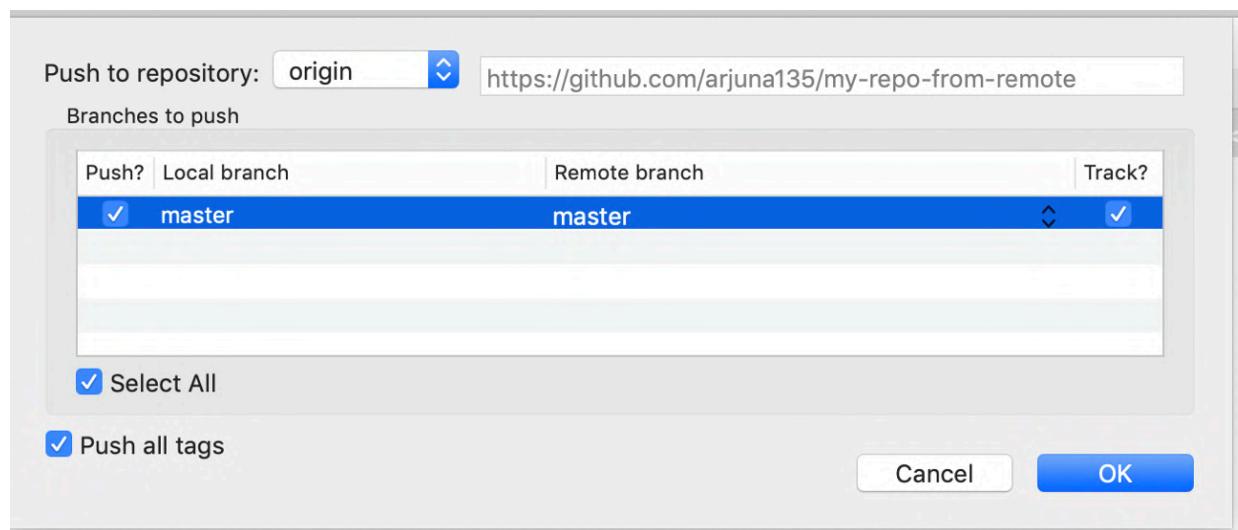
14. Sekarang perhatikan area commit history. Terdapat kolom Graph, Description, Commit Id, Author, Date, dan Commit Type. Karena baru hanya satu commit, tidak banyak yang bisa diceritakan oleh area ini. Untuk sekarang, kita fokuskan saja dahulu pada Description, Author, dan Date.



Pada kolom Description, terdapat nama branch dan commit message. Kolom ini menggambarkan bahwa aksi commit tersebut dilakukan pada branch ‘master’. Kolom Author akan menginformasikan nama dan email dari pelaku commit. Dan kolom Date menginformasikan tanggal dan waktu commit dilakukan. Langkah selanjutnya adalah Push. Perhatikan baik baik perubahan apa yang akan terjadi pada kolom Description.

15. Untuk melakukan Push, Anda hanya tinggal meng-klik tombol panah ke atas dengan tulisan push. Sejajar dengan tombol Commit. Jendela dialog akan muncul.

Modul Praktikum Kakas Pengembangan Perangkat Lunak

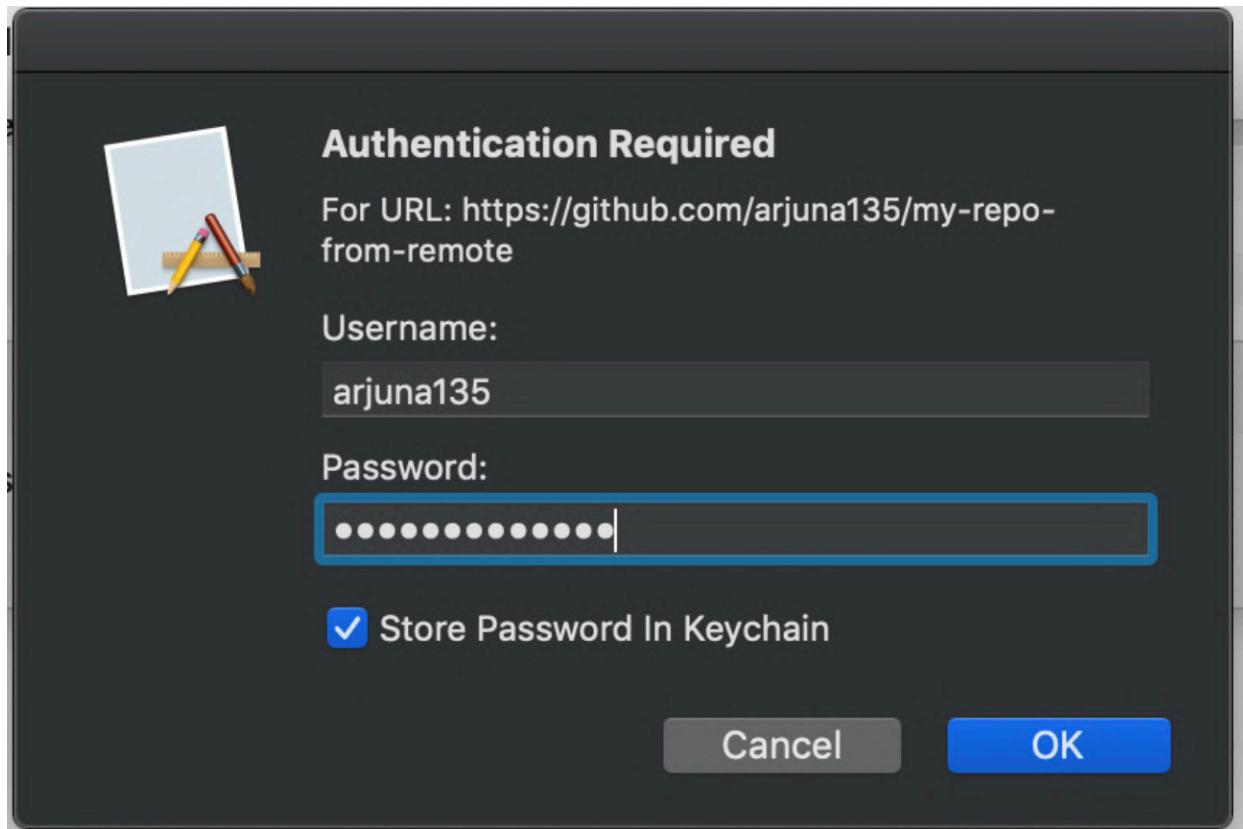


Akan ada empat kolom pada dialog ini yaitu **Push?**, **Local Branch**, **Remote Branch**, dan **Track?**. Silakan centang kolom **Push?** dengan **Local Branch master**. Dengan begitu, kira-kira seperti ini perintah yang disampaikan pada Sourcetree: “Lakukan Push pada Local Branch dengan nama ‘master’ ke Remote Branch dengan nama ‘master’. Pastikan kamu (Sourcetree) melakukan track pada kedua branch ini (local dan remote)”. Jika Anda sudah paham pada perintah yang akan sampaikan pada Sourcetree, silakan klik tombol **OK**.

Jika jendela Authentication muncul, isikan username dan password akun GitHub anda. Pastikan untuk **mencentang “Store Password ...** agar jendela ini tidak muncul kembali jika kita melakukan Push.

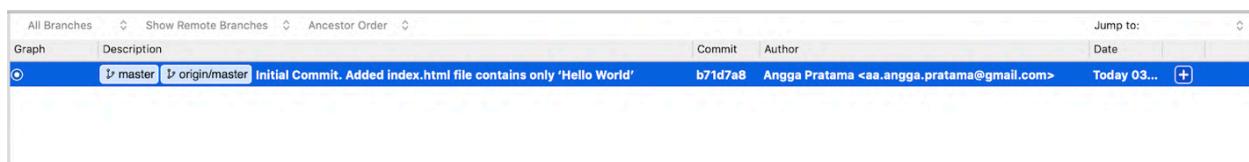


Modul Praktikum Kakas Pengembangan Perangkat Lunak



Akan muncul sebuah jendela progress dialog menginformasikan proses Push sedang berlangsung. Pastikan komputer anda terkoneksi dengan internet, karena proses Push sebenarnya adalah mengupload perubahan yang telah di-commit ke remote repository, dalam hal ini yang berada di akun GitHub Anda.

16. Selesai. Semua dialog akan hilang, lalu akan terjadi sedikit perubahan pada kolom Description di area commit history pada branch master:



Bandingkan dengan gambar pada langkah nomor 14. Awalnya pada Description hanya terdapat icon branch ‘master’ saja, tapi kini ditambah dengan icon branch ‘origin/master’. Begini penjelasannya. Origin adalah nama default dari remote repository anda. Anda bisa saja mengganti dengan nama lain, tapi hal ini tidak lazim dilakukan. Icon ‘master’ dan icon ‘origin/master’ berada berdampingan mengindikasikan bahwa aktivitas commit yang dilakukan telah terekam, baik oleh branch ‘master’ pada local repository anda juga oleh branch ‘master’ pada remote repository anda. Jika anda melihat nama branch dengan pre-path ‘origin’, tanda branch tersebut adalah branch master. Kita akan membahas mengenai branch lebih lanjut pada modul berikutnya.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Anda baru saja melakukan Commit untuk perubahan di local repository Anda, dan Push untuk mengupload hasil commit ke remote repository. Sourcetree juga telah memberikan informasi di area commit history bahwa remote repository Anda telah merekam perubahan tersebut dengan tanda branch master dan origin/master yang terletak berdampingan di Kolom Description.

Tapi apakah betul demikian? Apakah betul perubahan yang terjadi sudah ada di GitHub Anda?

Untuk membuktikan silakan buka halaman remote repository Anda di browser.

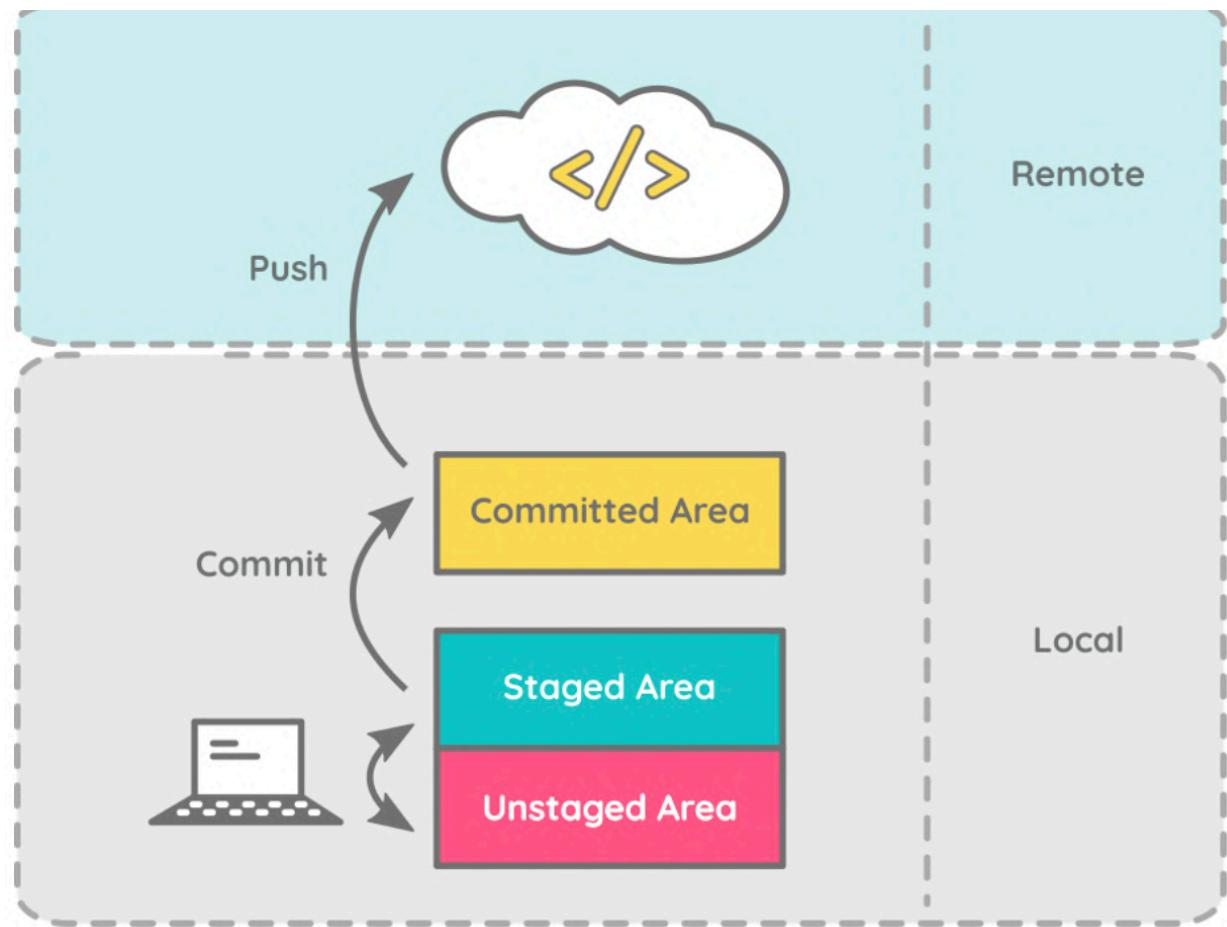
The screenshot shows a GitHub repository page for 'arjuna135 / my-repo-from-remote'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The latest commit was made by 'anggastudio' 40 minutes ago, with the message: 'Initial Commit. Added index.html file contains only 'Hello World''. There is a button to 'Add a README'.

Berikut Ilustrasi proses commit dan push yang telah anda lakukan:





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Selamat! Pada tahap ini Anda telah melakukan create repository, clone repository, commit changes, dan push commit. Di modul berikutnya kita akan melakukannya lagi. Terutama Commit dan Push, akan lebih sering kita lakukan. Silakan langsung menuju modul Atomic Commit, atau intermezzo dahulu di beberapa modul setelah ini.

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 4: Atomic Commit

Deskripsi Pertemuan

Apa itu atomic commit? Sesuai namanya, atomic commit berarti melakukan commit pada perubahan-perubahan sekecil atom. Maksudnya adalah Anda tidak direkomendasikan untuk membuat perubahan yang banyak, lalu kemudian melakukan commit untuk perubahan yang banyak itu. Anda direkomendasikan untuk langsung melakukan commit pada perubahan kecil yang Anda lakukan.

Syarat Kompetensi

1. Mahasiswa memahami fungsi commit di sourcetree
2. Mahasiswa mampu menggunakan visual studio code dan mengintegrasikan dengan sourcetree dan melakukan commit

Standar Kompetensi

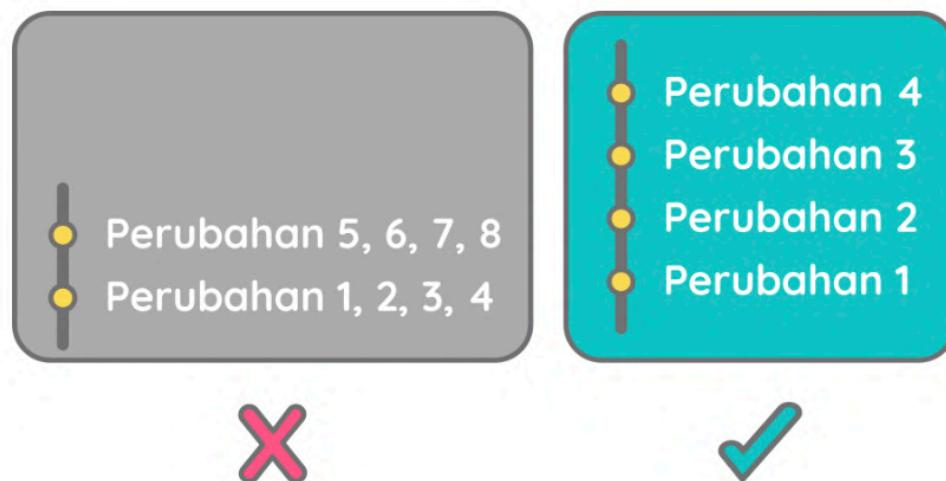
1. Mahasiswa mampu melakukan commit terhadap perubahan-perubahan kecil dari setiap source code yang dilakukan
2. Mahasiswa tidak disarankan melakukan commit terhadap perubahan yang begitu banyak

Dasar Teori

Pada modul ini kita akan belajar atomic commit. Apa itu atomic commit? Sesuai namanya, atomic commit berarti melakukan commit pada perubahan-perubahan sekecil atom. Maksudnya adalah Anda tidak direkomendasikan untuk membuat perubahan yang banyak, lalu kemudian melakukan commit untuk perubahan yang banyak itu. Anda direkomendasikan untuk langsung melakukan commit pada perubahan kecil yang Anda lakukan



Modul Praktikum Kakas Pengembangan Perangkat Lunak

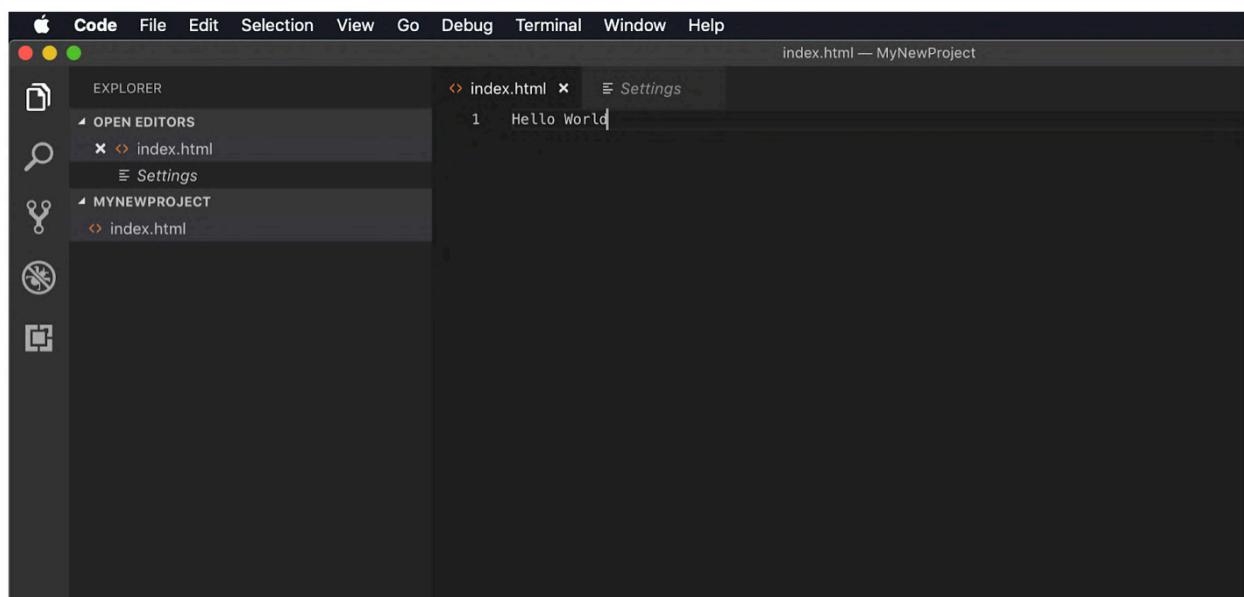


Tugas Pendahuluan

[soal-soal TP yang berhubungan dengan materi pertemuan, berbasis masalah rill]

Langkah-Langkah Praktikum

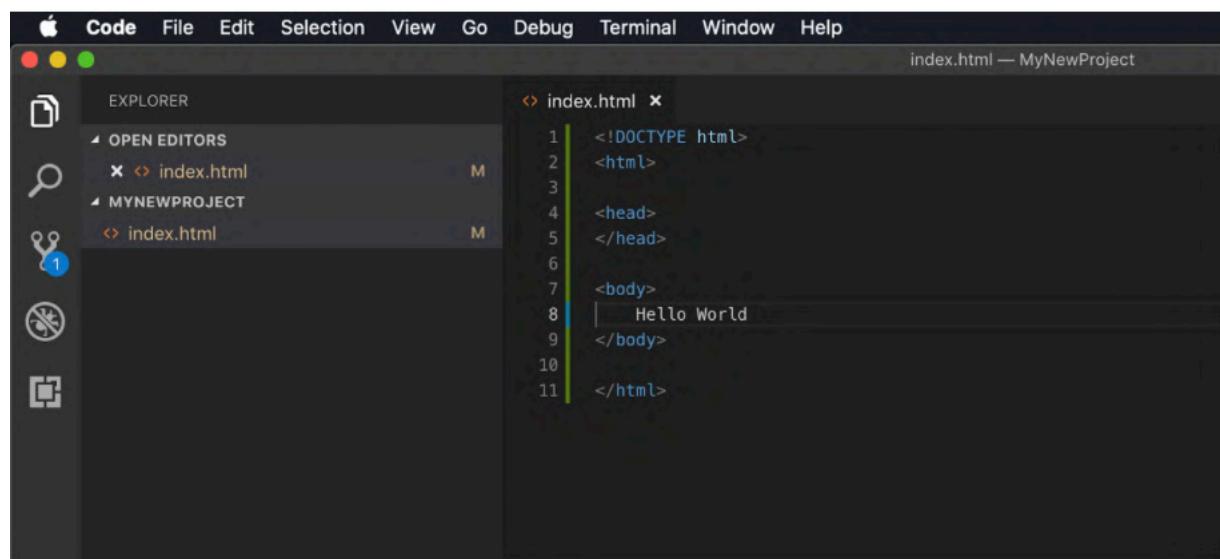
1. Pastikan Anda mengerjakannya di repository yang sudah kita buat di GitHub. Yaitu <https://github.com/arjuna135/my-repo-from-remote>. Url akan berbeda sesuai dengan username akun GitHub Anda.
2. Buka Visual Studio Code, dan pastikan Anda bekerja di dalam folder repository yang dimaksud pada poin 1. Anda akan menemukan file index.html yang berisi Hello World pada line 1.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

3. Sekarang kita akan mulai menuliskan markup untuk membuat konten website. Jika Anda sudah pernah membuat website, tuliskan saja markup basic seperti Doctype declaration, element html beserta head dan body-nya. Lalu masukkan Hello World tadi ke dalam element body. Hasilnya akan menjadi seperti ini:

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.   </head>
5.
6.   <body>
7.     Hello World
8.   </body>
9.
10. </html>
```



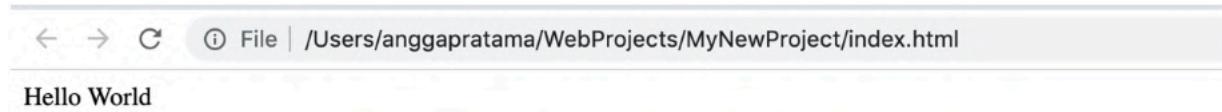
Warna file index.html di area explorer akan berubah warna menjadi kuning, tanda bahwa ada perubahan di dalam file ini.

Sedikit tambahan, coba perhatikan icon yang mirip ketapel di sebelah kiri. Icon tertimpa angka 1 berwarna biru. Silakan klik icon tersebut, Anda akan melihat jendela baru dan dialog untuk memasukkan message commit. Ya benar, VS Code juga memiliki fitur Git GUI. Begitu juga dengan IDE lain, seperti Android Studio, X Code, dan IDE modern lainnya. Anda bisa mencobanya nanti. Untuk saat ini kita akan fokus menggunakan Git GUI Sourcetree.



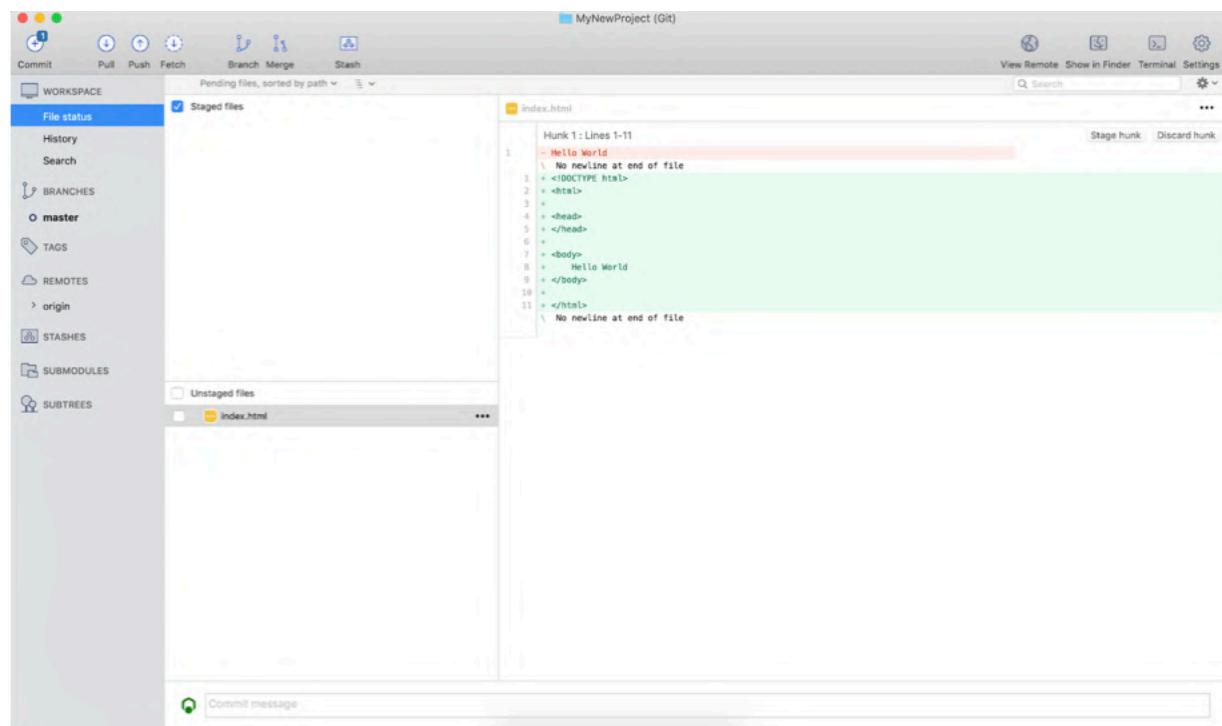
Modul Praktikum Kakas Pengembangan Perangkat Lunak

4. Silakan run index.html di browser favorit anda untuk mengecek apakah terdapat error di perubahan yang dilakukan.



Jika tampilan di browser Anda seperti pada gambar di atas. Artinya file index.html aman, tidak terdapat error. Anda bisa langsung melakukan commit.

5. Buka Sourcetree, lalu klik File Status.



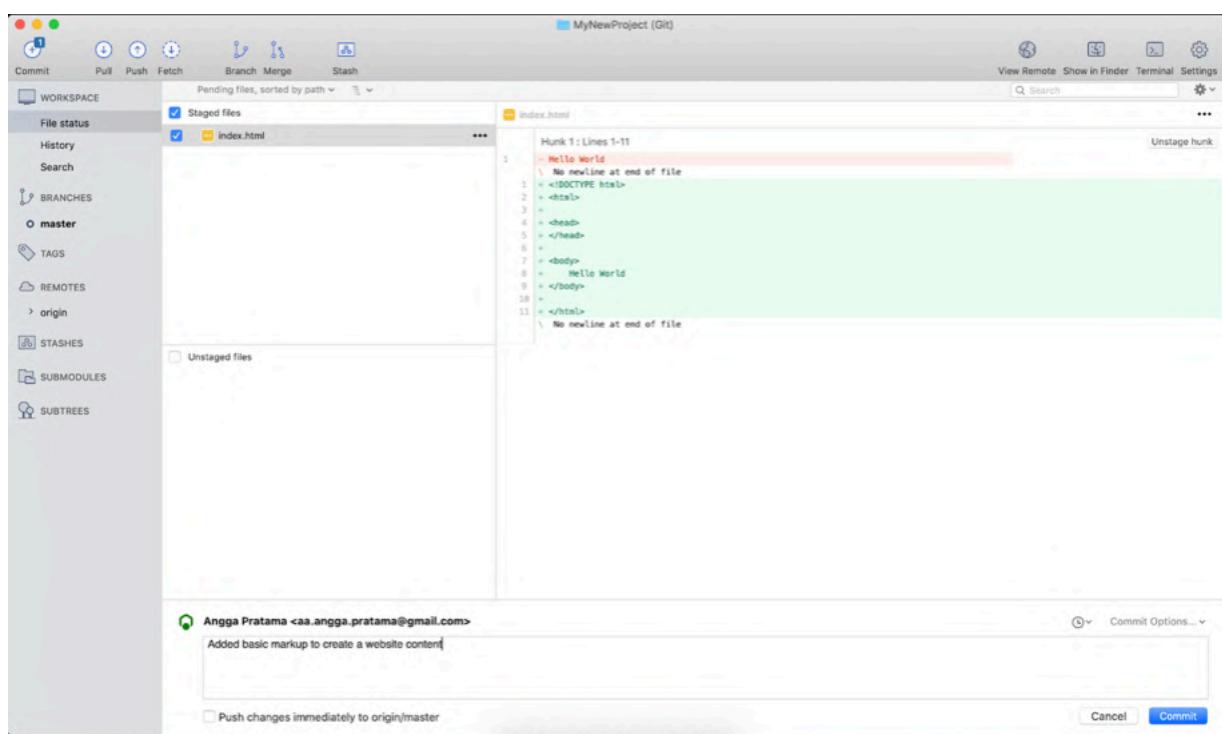
Yang perlu Anda perhatikan pertama kali adalah pada area unstaged files. Di area ini Anda akan menemukan file-file yang terjadi perubahan di dalamnya. Jika Anda klik salah satu filenya, saat ini hanya ada satu yaitu html.index, maka selanjutnya Anda perlu memperhatikan area perubahan konten **file** **di sebelah** **kanan**. Akan nampak area berwarna putih, merah, dan hijau. Area putih menunjukkan bahwa tidak terjadi



Modul Praktikum Kakas Pengembangan Perangkat Lunak

perubahan pada area tersebut. Area merah menunjukkan bahwa terjadi penghapusan di area tersebut. Dan area hijau menunjukkan ada penambahan di area tersebut. Anda juga bisa melihat nomor baris di sebelah kiri. Perubahan tersebut menggambarkan bahwa terjadi penghapusan Hello World di line 1, dan ditambahkan code baru dari line 1 sampai line 11.

6. Masukkan index.html di unstaged files area ke staged files area. Lalu klik **commit** sehingga muncul area message **di bawah**. Tuliskanlah seperti ini untuk message:
“Added basic markup to create a website content”,
Hasilnya akan nampak seperti ini:



The screenshot shows the SourceTree application interface. On the left is the workspace sidebar with options like Commit, Pull, Push, Fetch, Branch, Merge, and Stash. The main area shows a file named 'index.html' in the 'Staged files' section. The code editor displays the following content:

```
1 Hello World
2 <!DOCTYPE html>
3 <html>
4 <head>
5 </head>
6 <body>
7   Hello World
8 </body>
9 </html>
```

Below the code editor, the commit message field contains:

Angga Pratama <aa.angga.pratama@gmail.com>
Added basic markup to create a website content

At the bottom of the commit dialog, there is a checkbox labeled "Push changes immediately to origin/master" and two buttons: "Cancel" and "Commit".

7. Klik **Commit**, dan klik icon **Push** pada toolbar. Silakan cek GitHub untuk memastikan perubahan terakhir sudah berhasil di-push.

Modul Praktikum Kakas Pengembangan Perangkat Lunak

this repository is created from github

Manage topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

anggastudio Added basic markup to create a website content Latest commit 8218f14 a minute ago

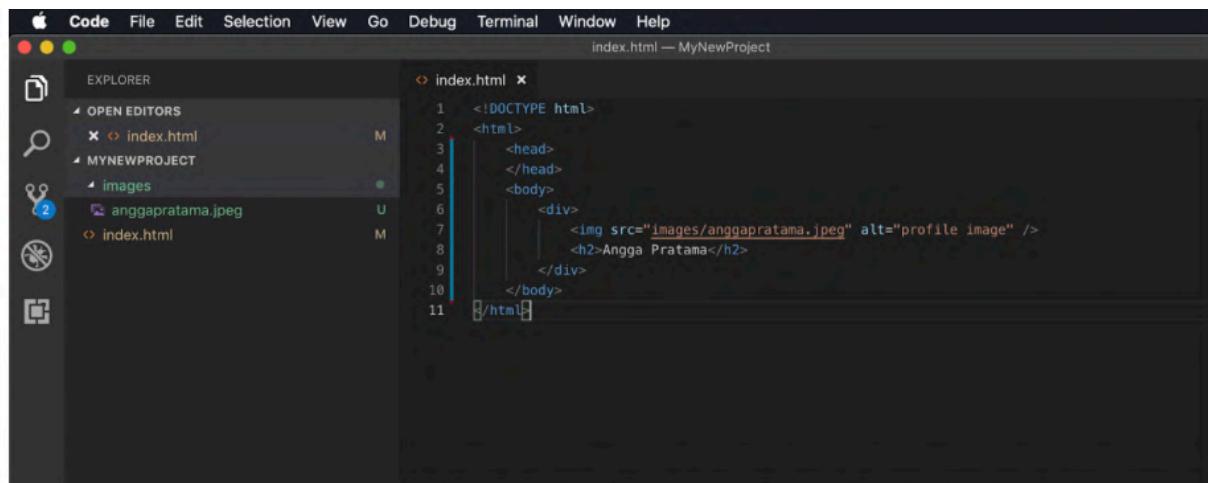
index.html Added basic markup to create a website content a minute ago

Add a README

8. Selanjutnya kita akan melakukan perubahan yang lain. Buka kembali Visual Studio Code. Hapus Hello World dan ganti dengan markup seperti ini:

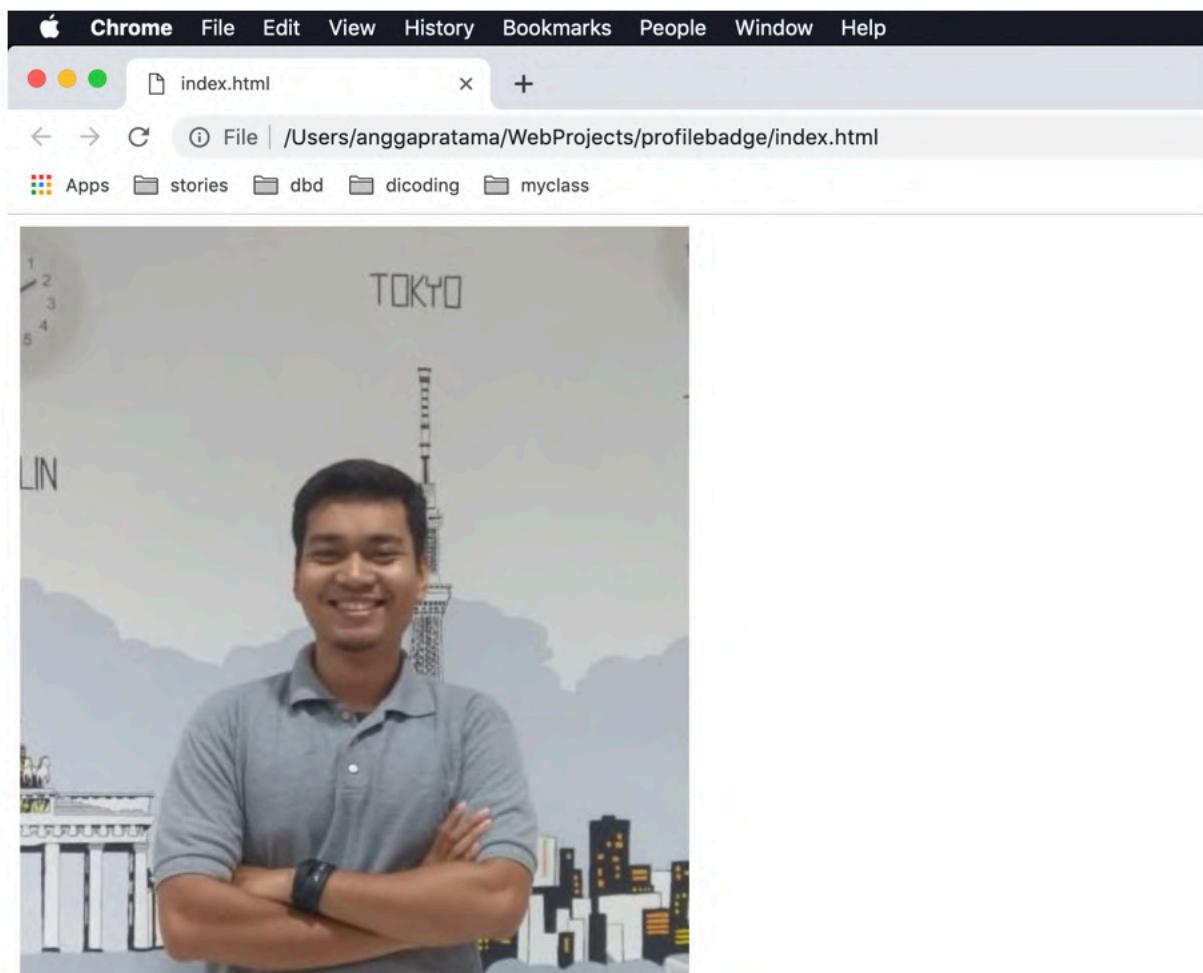
```
<div>
  Angga Pratama</h2>
</div>
```

9. Buatlah folder baru dengan nama images sejajar dengan file index.html.
10. Masukkan suatu gambar dengan format file .jpg, .jpeg, atau .png ke dalam folder images.
11. Sesuaikan nama file gambar dengan value pada atribut src di file index.html. Sehingga keseluruhan kode akan nampak seperti berikut:



Dan jika index.html dijalankan akan nampak seperti ini pada browser:

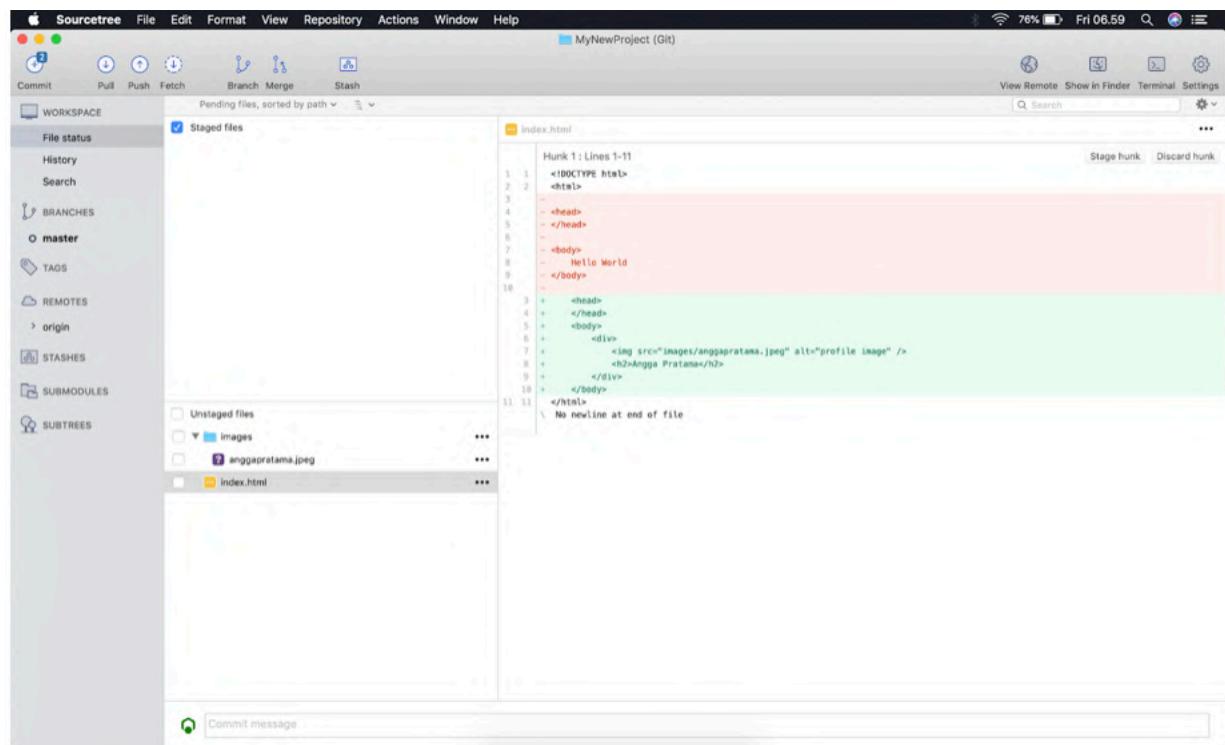
Modul Praktikum Kakas Pengembangan Perangkat Lunak



Angga Pratama

12. Buka Sourcetree, klik **File Status**, dan perhatikan area workspace, maka akan nampak bahwa terdapat file baru, yaitu gambar yang Anda masukkan, dan terdapat perubahan pada file index.html:

Modul Praktikum Kakas Pengembangan Perangkat Lunak



13. Centang **semua file** dan folder pada unstaged files sampai semuanya naik ke area staged files.
14. Lakukan Commit dengan message “Changed Hello world with img and h2 to display simple profile data. Image file added in the folder images and used as resource in the img element in index.html file”.
15. Lakukan **push**.
16. Coba istirahat sejenak dan renungkan, apakah kini Anda mulai paham dengan atomic commit? Jika belum, lanjutkan langkah-langkah berikut:
17. Pada element div yang menampung element img dan h2, tambahkan id “profile-container”.
18. Pada element img tambahkan id “profile-image”.
19. Pada element h2 tambahkan id “profile-name”.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



```
index.html x anggapratama.jpeg
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <div id="profile-container">
7       
8       <h2 id="profile-name">Angga Pratama</h2>
9     </div>
10   </body>
11 </html>
```

20. Lakukan **commit** dengan message “Added ids for each profile element”.
21. Lakukan **push**.
22. Coba istirahat sejenak dan renungkan, apakah kini Anda mulai paham dengan atomic commit? Jika belum, lanjutkan langkah-langkah berikut:
23. Buat file baru dengan nama style.css bersebelahan dengan file index.html, lalu isikan kode berikut:



Modul Praktikum Kakas Pengembangan Perangkat Lunak

```
1. body {  
2.     width: 100%;  
3.     margin: 30px;  
4.     background-color: antiquewhite;  
5. }  
6.  
7. #profile-container{  
8.     max-width: 300px;  
9.     background-color: white;  
10.    border: 1px solid gray;  
11.    border-radius: 6px;  
12. }  
13.  
14. #profile-image{  
15.     width: 100%;  
16.     height: auto;  
17. }  
18.  
19. #profile-name{  
20.     text-align: center;  
21. }
```

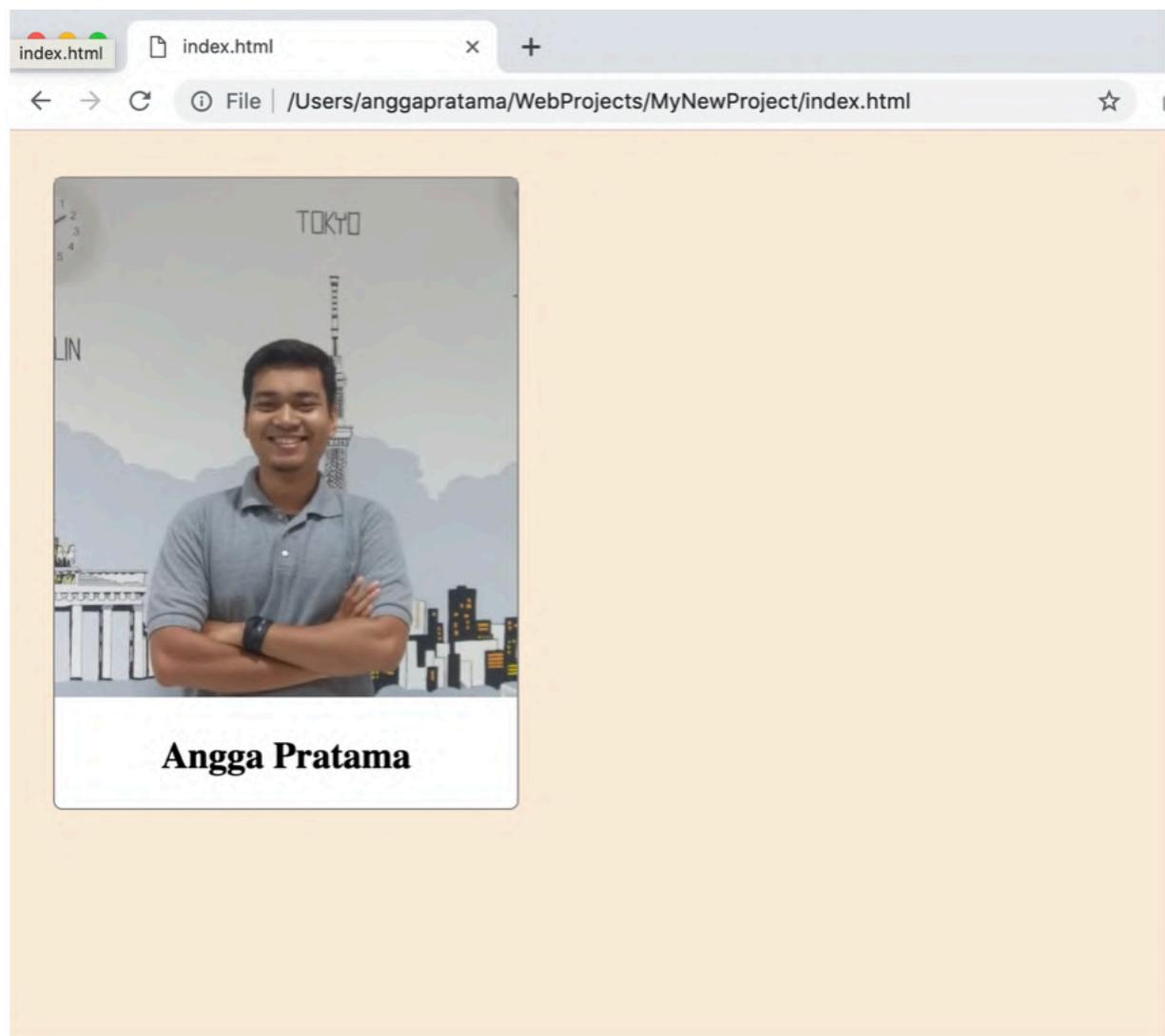
24. Pada file index.html, tambahkan markup berikut di dalam element head:

```
1. <link rel="stylesheet" href="style.css">
```

25. Lalu coba buka browser Anda. Seharusnya tampilan web saat ini akan seperti berikut :



Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Dengan kodennya sebagai berikut :

The screenshot shows a code editor with two files open. On the left is `index.html` containing:

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div id="profile-container">
        Angga Pratama</h2>
    </div>
</body>
</html>
```

On the right is `style.css` containing:

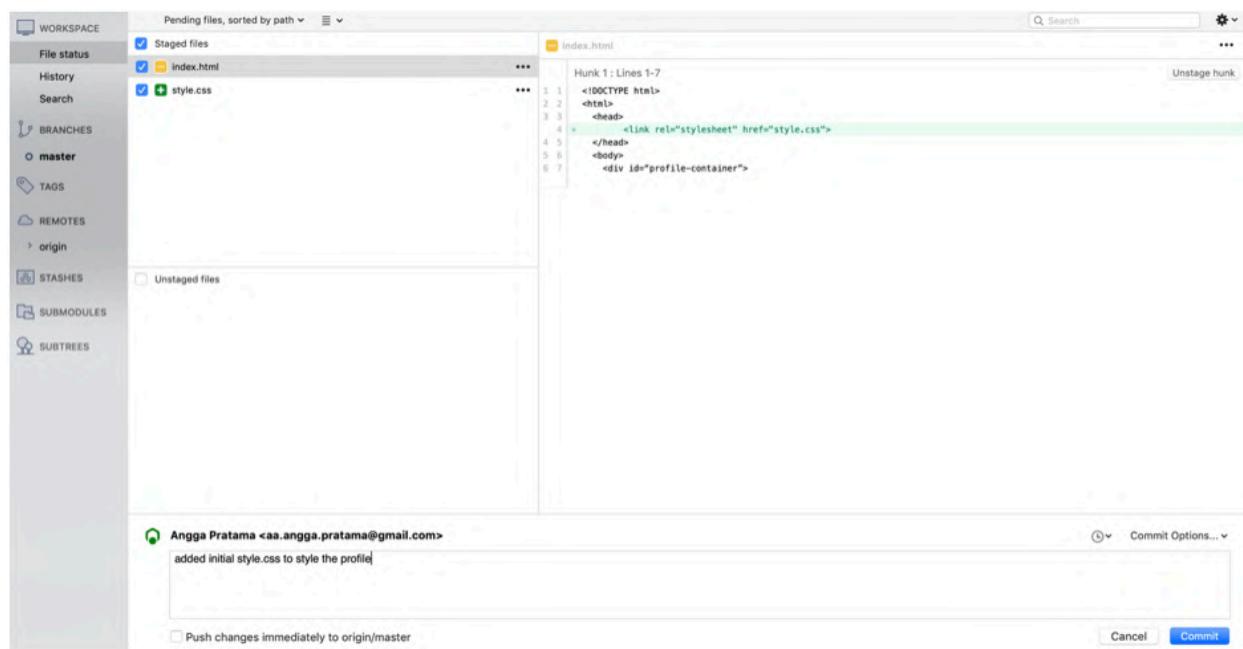
```
body {
    width: 100%;
    margin: 30px;
    background-color: #antiquewhite;
}

#profile-container{
    max-width: 300px;
    background-color: #white;
    border: 1px solid #gray;
    border-radius: 6px;
}

#profile-image{
    width: 100%;
    height: auto;
}

#profile-name{
    text-align: center;
}
```

26. Buka Sourcetree, **Commit** dengan message “added initial style.css to style the profile”. Lanjutkan dengan **Push**.



The screenshot shows the Sourcetree interface with the `master` branch selected. The commit history table is displayed:

Date	Author	Description
Today 10...	Angga Pratama <aa.angga.pratama@gmail.com>	4493d90 D master > origin/master added Initial style.css to style the profile
Today 10.15	Angga Pratama <aa.angga.pratama@gmail.com>	10c53d8 Added Ids for each profile element
Today 10.10	Angga Pratama <aa.angga.pratama@gmail.com>	6f14e8b Changed Hello world with img and h2 to display simple profile data. Image file added in the folder images and use...
Today 10....	Angga Pratama <aa.angga.pratama@gmail.com>	f616f15 Added basic markup to create a website content



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Baiklah, sampai titik ini, istirahat dulu sejenak. Coba tarik nafas dan bayangkan kembali apa yang telah Anda lakukan. Apa yang Anda lakukan sudah bisa disebut dengan atomic commit. Yaitu melakukan commit untuk satu perubahan kecil. Perubahan kecil yang dimaksud cukup mewakili satu konteks perubahan. Sejauh ini perubahan perubahan yang kita lakukan adalah:

1. Inisiasi markup html. Lihat poin 6.
2. Menambahkan element dasar untuk konten profile. Lihat poin 11.
3. Menambahkan id pada masing-masing element profile. Lihat poin 17.
4. Menambahkan file style.css untuk mengatur layout profile.

Atomic commit adalah sebuah *best practice* dalam mengelola source code. Jika Anda tidak menerapkannya, maka tidak salah, yang terpenting adalah software yang Anda bangun *deliverable* dengan baik. Akan tetapi, jika Anda menerapkan atomic commit, berarti Anda telah membawa software anda ke level yang lebih tinggi. Level software dengan manajemen yang baik, yaitu software yang memiliki predikat *maintainable, scalable, and well developed*.

Teruslah berlatih untuk melakukan atomic commit sehingga Anda terbiasa dan dengan mudah menentukan besar kecilnya suatu konteks perubahan untuk kemudian melakukan commit dengan perubahan tersebut.

Kini Anda sudah siap untuk mulai belajar Branch dan cara mengelolanya. Silakan lanjut ke modul berikutnya.

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 5: Branch

Deskripsi Pertemuan

Pada pertemuan kali ini kita akan membahas salah satu fungsi dari Version Control yang terdapat dalam code versioning atau source code management.

Syarat Kompetensi

1. Sourcetree sudah terinstall dengan baik
2. Mempunyai akun Git-Hub
3. Memiliki Visual Studio Code

Standar Kompetensi

1. [standar kompetensi, capaian minimal setelah praktikum]
2. [standar kompetensi, capaian minimal]

Dasar Teori

Branch. Apa itu Branch? Jika diterjemahkan ke dalam bahasa Indonesia, maka artinya adalah cabang. Tapi terjemahan tersebut kurang menggambarkan Branch yang akan kita lakukan. Singkatnya, Branch inilah yang menjadi komponen utama dalam code versioning atau Source Code Management.





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Tugas Pendahuluan

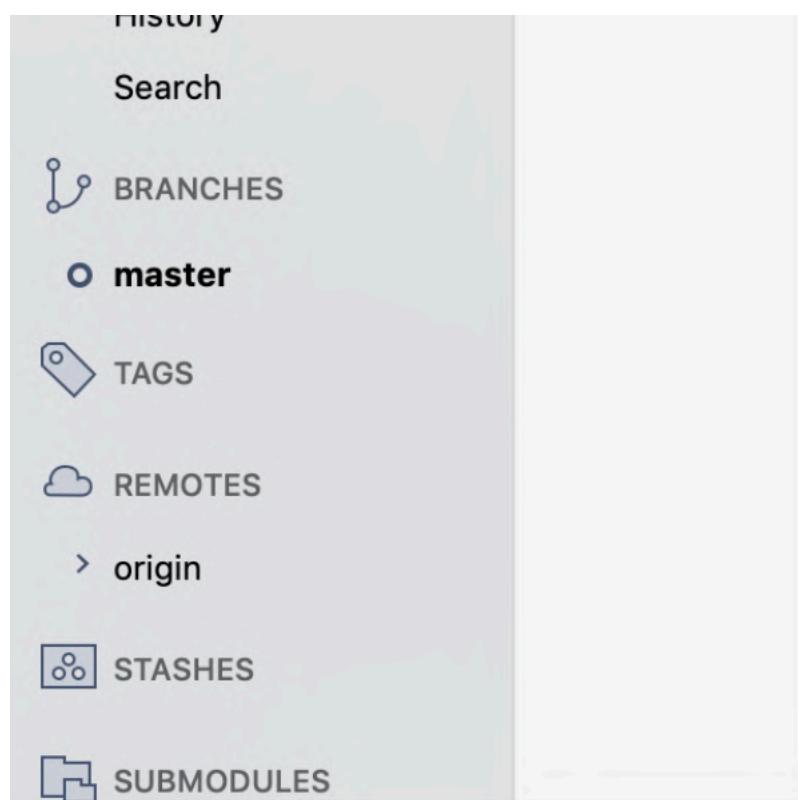
[soal-soal TP yang berhubungan dengan materi pertemuan, berbasis masalah rill]

Langkah-Langkah Praktikum

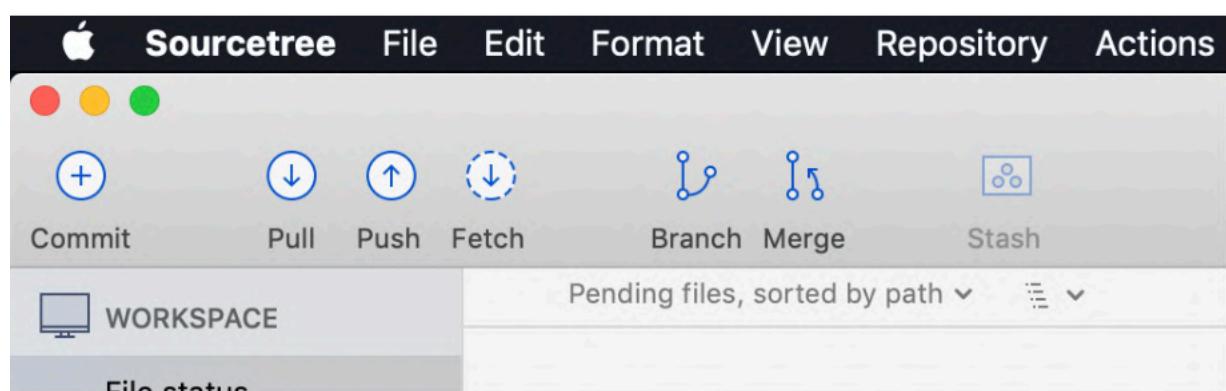
Membuat Branch Baru

1. Buka Sourcetree. Pastikan Anda berada di repository yang selama kita ini gunakan. Pastikan juga Anda masih berada di Branch Master. Perhatikan area di sebelah kiri pada Sourcetree, temukan Branches, dan pastikan master dalam keadaan aktif, ditandai dengan tulisannya yang tercetak tebal.

Modul Praktikum Kakas Pengembangan Perangkat Lunak



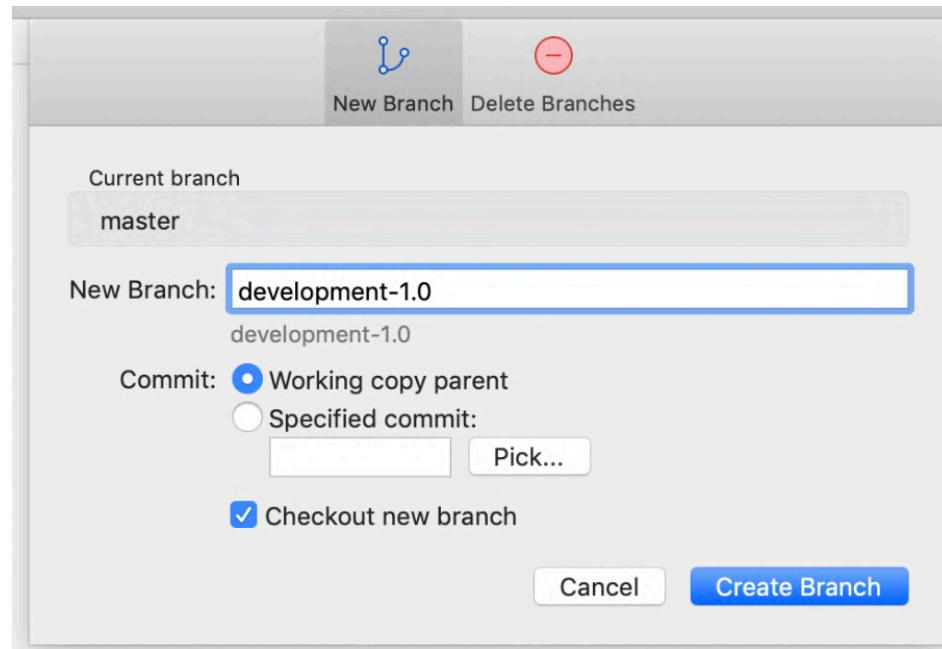
2. Sekarang kita akan membuat sebuah branch baru dengan branch master dijadikan sebagai parent dari new branch yang akan kita buat. Perhatikan icon branch pada toolbar Sourcetree.



3. Untuk membuat branch baru, klik **icon branch** tersebut maka akan muncul sebuah dialog.

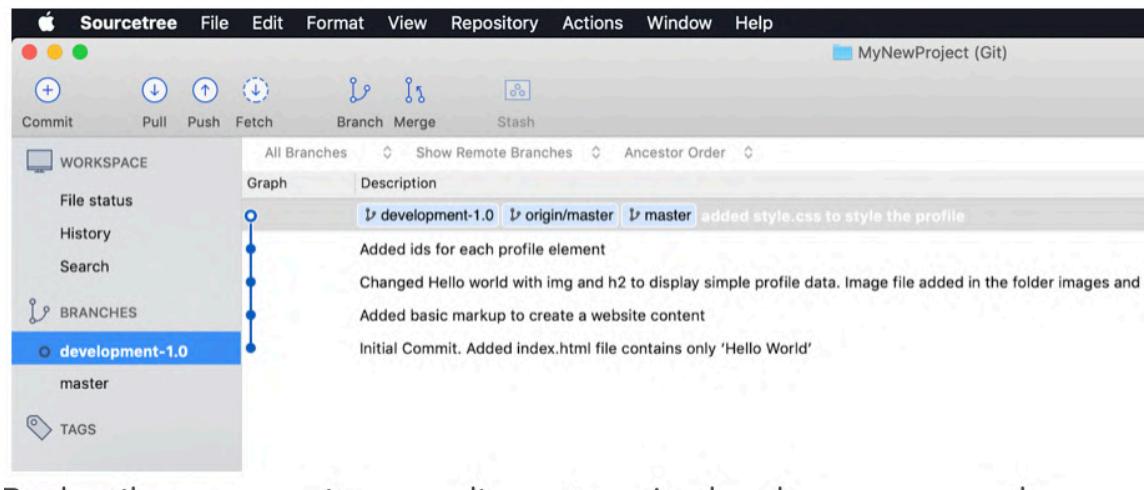


Modul Praktikum Kakas Pengembangan Perangkat Lunak

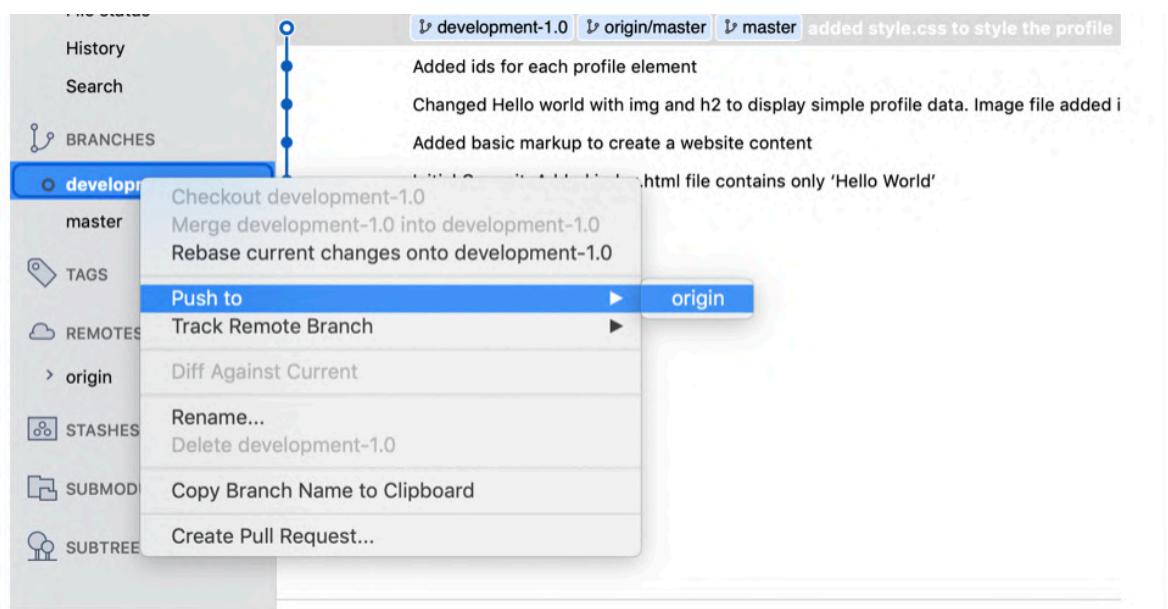


- Pertama, perhatikan barisan icon paling atas, pastikan icon yang aktif adalah **New Branch**.
 - Kedua, perhatikan **Current Branch**, branch saat ini yang aktif adalah master.
 - Ketiga, tuliskan nama branch baru anda dengan “development-1.0”. Anda bebas memasukkan nama apa saja, tetapi pastikan nama yang buat cukup mendeskripsikan tujuan dari branch tersebut. Nama “development-1.0” mendeskripsikan bahwa pada branch ini mengandung kode-kode untuk development versi 1.0.
 - Keempat, perhatikan Commit, untuk saat ini kita akan pilih Working copy parent, artinya branch baru ini akan memuat semua kode yang saat ini dimiliki oleh branch yang sedang aktif, yaitu master. Jika Anda tidak ingin mengambil semua kode dari master, Anda bisa memilih specified commit, klik **Pick...**, lalu memilih commit-an yang mana akan diambil kodennya.
 - Kelima. Perhatikan **Chekout new branch**. Checkout adalah aktivitas perpindahan working area, dari satu branch ke branch lain. Jika pilihan ini tercentang, maka working area Anda akan langsung berpindah ke branch baru. Biarkan saja tercentang.
 - Keenam, klik **Create Branch**, maka branch baru akan terbentuk.
4. Anda sudah membuat branch baru dan branch baru tersebut saat ini aktif menjadi working area Anda. Silakan klik pada branch baru tersebut, dan area di **sebelah kanan** Sourcetree akan berubah menjadi history area.

Modul Praktikum Kakas Pengembangan Perangkat Lunak

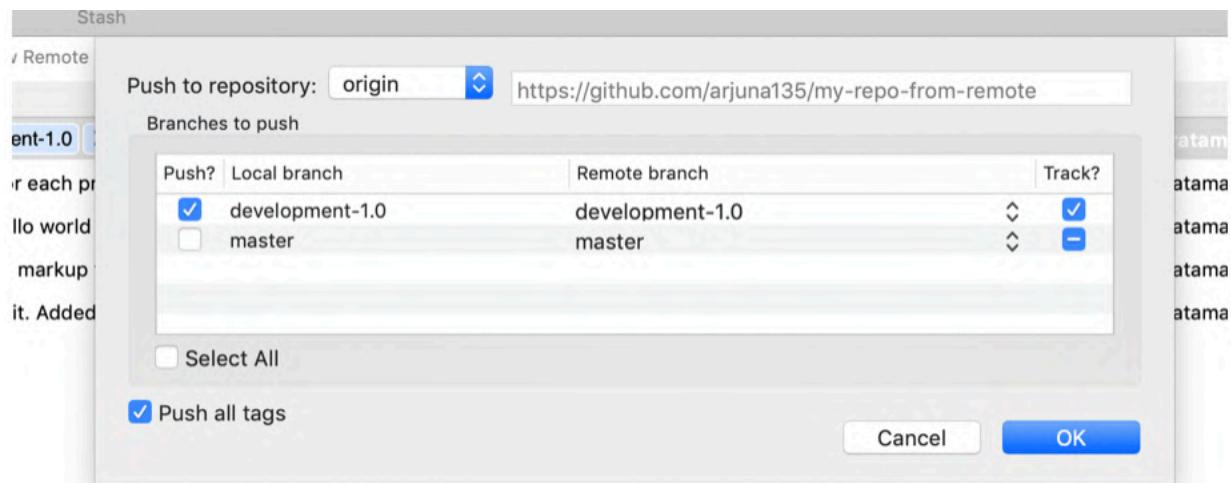


Perhatikan commit-an paling atas, Anda akan menemukan tiga branch berada pada commit-an yang sama, yaitu development-1.0, origin/master, dan master. Hal ini dikarenakan kita mengambil semua kode dari commit-an teratas pada branch master untuk membuat branch development-1.0. Tapi mengapa hanya development-1.0? Mengapa tidak ada origin/development-1.0 seperti pada branch master. Silakan ingat-ingat kembali pelajaran kita tentang remote repository. Betul sekali! Branch development-1.0 yang kita buat saat ini masih berada di local repository, kita belum mengupload branch baru kita ini ke remote repository. Klik kanan pada development-1.0, pilih **Push To**, pilih **origin**.



Maka akan muncul dialog seperti berikut:

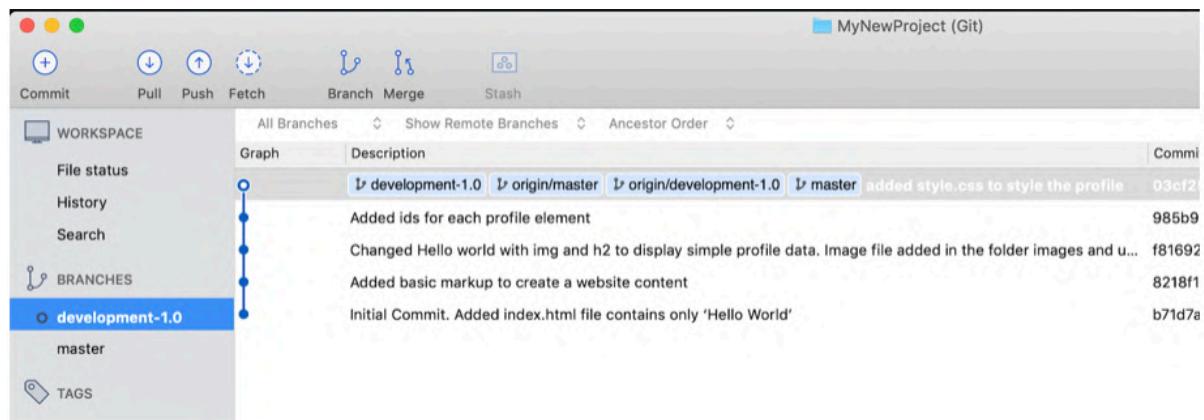
Modul Praktikum Kakas Pengembangan Perangkat Lunak



Baris pada development-1.0 secara default tercentang. Kira kira seperti ini perintah yang akan dijlankan:

Lakukan Push untuk local branch development-1.0 ke remote branch dengan nama development-1.0 dan track semua perubahan di remote branch ini jika ada perubahan. Klik **OK** untuk mem-Push branch baru kita ini.

5. Anda telah mengupload branch baru ke remote repository. Di Sourcetree, Anda bisa memastikannya dengan klik pada branch local, lalu temukan nama branch yang sama dengan path origin sebelum namanya. Perhatikan gambar berikut:



Pada commit-an terakhir, kini kita bisa melihat branch development-1.0 dan origin/development-1.0 berada berdampingan. Hal tersebut berarti branch local development-1.0 dan branch remote origin/development-1.0 berisikan kode-kode yang sama.

6. Untuk memastikan bahwa branch baru sudah terupload bisa dilakukan juga di cloud tempat Anda menyimpan repository, dalam hal ini cloud yang kita gunakan adalah GitHub. Buka halaman GitHub untuk repository Anda, dan perhatikan tulisan branches di sebelah commits.

Modul Praktikum Kakas Pengembangan Perangkat Lunak

this repository is created from github

Manage topics

5 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

anggastudio added style.css to style the profile Latest commit 03cf25f 2 days ago

images Changed Hello world with img and h2 to display simple profile data. I... 2 days ago

index.html added style.css to style the profile 2 days ago

style.css added style.css to style the profile 2 days ago

Add a README

Kini terdapat 2 branch pada repository Anda. Untuk melihat detail branch apa saja yang ada di repository, klik tulisan branches tersebut dan Anda akan menemukan branch list.

Selamat, Anda kita sudah bisa membuat branch baru di local dan meng-upload-nya ke remote. Kita tidak akan melakukan perubahan di branch ini, karena branch development-1.0 ini akan kita gunakan sebagai wadah saja. Hmm apa maksudnya? Silakan lanjut ke modul berikutnya.

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 6: Merge

Deskripsi Pertemuan

Baiklah, kita sudah sampai di modul Merge. Jika Branch adalah komponen utama code versioning, maka bisa dikatakan bahwa Merge adalah aktivitas utamanya. Merge adalah aktivitas untuk menggabungkan semua perubahan pada satu branch ke branch lainnya. Sebelum memulai praktik seperti biasanya, kita akan bahas dulu sedikit penggambaran tentang merge.

Syarat Kompetensi

1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
2. Sed mollis aliquam nibh.
3. Pellentesque pellentesque dictum urna.

Standar Kompetensi

1. [standar kompetensi, capaian minimal setelah praktikum]
2. [standar kompetensi, capaian minimal]

Dasar Teori

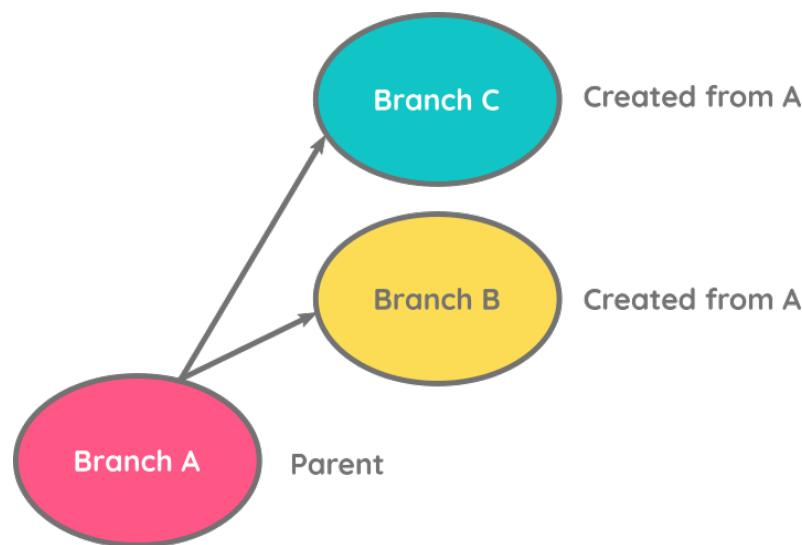
Baiklah, kita sudah sampai di modul Merge. Jika Branch adalah komponen utama code versioning, maka bisa dikatakan bahwa Merge adalah aktivitas utamanya. Merge adalah aktivitas untuk menggabungkan semua perubahan pada satu branch ke branch lainnya. Sebelum memulai praktik seperti biasanya, kita akan bahas dulu sedikit penggambaran tentang merge.

Misalnya anda adalah seorang Technical Leader, dan di dalam projek anda saat ini memiliki branch dengan nama A, B, dan C. Branch A akan kita gunakan sebagai wadah saja, sedangkan branch B dan C dibuat dari Branch A sebagai parent, dan akan kita gunakan untuk mengembangkan feature dari software yang kita bangun. Maka team developer akan bekerja di branch B dan C. Aktivitas Commit dan Push akan banyak terjadi di branch B, juga di branch C. Masing-masing memiliki perubahan yang berbeda karena feature yang digunakan juga berbeda. Sampai tiba saatnya feature-feature tersebut sudah siapkan untuk digabungkan ke dalam Branch A.





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Branch A dibuat untuk menampung seluruh feature-feature yang sudah dikembangkan. Karena itulah Branch A hanya digunakan sebagai wadah, tidak ada aktivitas commit dan push di branch ini, yang ada hanyalah aktivitas Merge. Pastikan yang memiliki wewenang melakukan Merge hanyalah Technical Leader saja, yaitu Anda.

Selanjutnya seluruh perubahan pada branch B akan di-merge ke Branch A, juga seluruh perubahan pada branch C, akan di-merge ke branch A. Aktivitas merge hanya terjadi satu arah. Artinya jika branch B di-merge ke Branch A, maka branch B berperan sebagai donor, dan branch A sebagai resipien. Branch A akan memiliki semua kode yang dari branch B, tetapi Branch B belum tentu memiliki semua kode dari Branch A. Perhatikan diagram berikut:

B ----> A

B di merge ke A. Sehingga A memiliki semua perubahan yang terjadi di B. Akan tetapi B tidak punya perubahan yang terjadi di A.

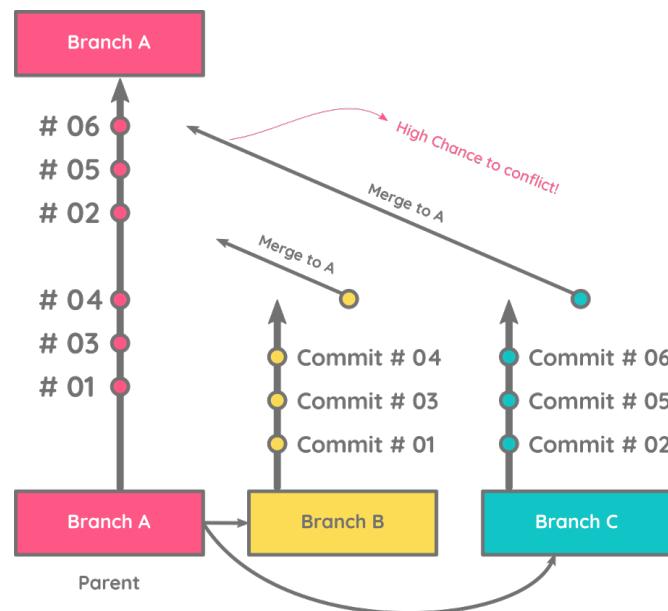
C ----> A (with B changes)

C di merge ke A. Sehingga A memiliki semua perubahan yang terjadi di C dan B. Akan tetapi C tidak punya perubahan yang terjadi di A maupun B.

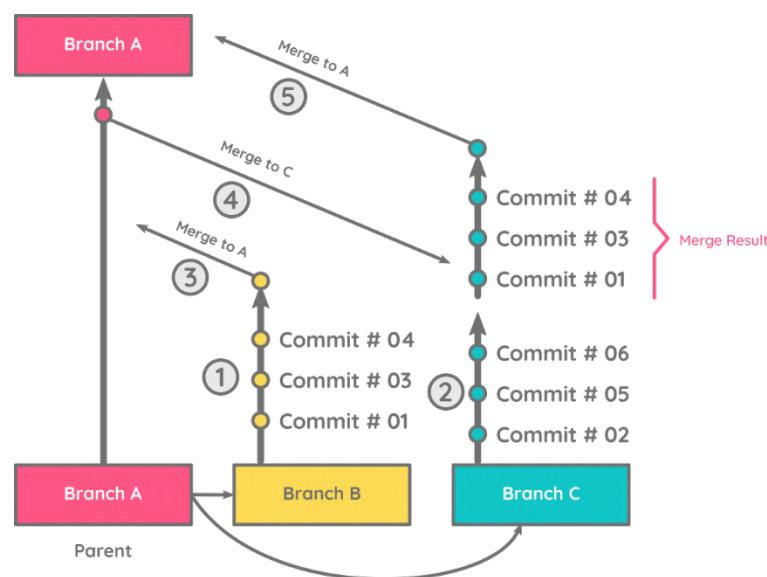
Pada saat branch B di-merge ke branch A, mungkin saja tidak akan terdapat banyak masalah. Karena sebelumnya pada branch A belum ada perubahan sejak Branch A digunakan sebagai parent untuk membuat branch B. dengan kata lain Branch B masih mengenal branch A. Akan tetapi ketika kemudian branch C di-merge ke Branch A, mungkin akan terdapat beberapa konflik. Sejak branch B di-merge ke Branch A, sekarang Branch A sudah banyak berubah. Sudah jauh berbeda jika dibanding sebelumnya sejak Branch A digunakan sebagai parent untuk membuat Branch C. Branch C kini sudah tidak begitu mengenali Branch A karena Branch A sudah sedikit berbeda. Begitulah drama yang terjadi dari proses merge.



Modul Praktikum Kakas Pengembangan Perangkat Lunak



Jika branch A dipaksa langsung menerima perubahan dari branch C, bisa terjadi banyak konflik. Konflik bisa saja dibetulkan, tapi kita tidak ingin melakukan commit dan push di branch A, karena branch ini hanya berperan sebagai wadah saja. Untuk itu perlu dilakukan merge ulang dari Branch A ke Branch C sebelum melakukan merge branch C ke branch A. Hal ini dilakukan agar branch C semakin mengenal perubahan yang terjadi di branch A, sehingga ketika branch C di-merge ke branch A, tidak akan ada masalah.



Perhatikan skema proses Merging di atas:

1. Developer melakukan beberapa perubahan di Branch B.
2. Developer melakukan beberapa perubahan di Branch C.
3. Branch B di Merge ke Branch A.
4. Branch A (yang kini sudah memiliki perubahan dari Branch B) di Merge ke Branch C.
5. Branch C (yang kini sudah memiliki perubahan dari Branch A, hasil Merge dari Branch B) di Merge ke Branch A.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Untuk memahami lebih dalam, mari kita praktikkan.

Kita akan membuat branch A, B dan C seperti pada kasus di atas. Branch A sudah kita buat sebelumnya, yaitu branch development-1.0. Kita juga akan membuat Branch header sebagai branch B, dan branch footer sebagai branch C.

Jadi seperti ini diagramnya:

Branch A = development-1.0

Branch B = header

Branch C = footer

Tugas Pendahuluan

[soal-soal TP yang berhubungan dengan materi pertemuan, berbasis masalah rill]

Langkah-Langkah Praktikum

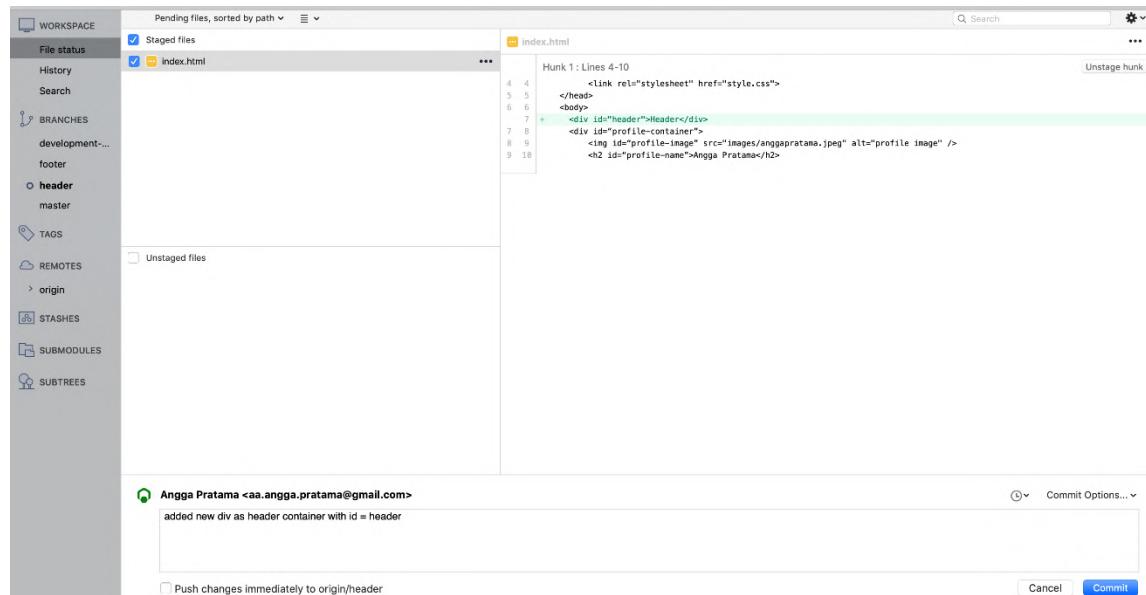
1. Buatlah branch baru dengan nama “header” dengan branch development-1.0 sebagai parent. Checkout ke branch header, lakukan upload branch header ke remote dengan cara Push to Origin.
2. Checkout kembali ke branch development-1.0 dan pastikan branch development-1.0 dalam keadaan aktif sebelum lanjut ke poin 3.
Langkah ini penting karena banyak dari peserta mengalami konflik di poin 12 akibat mengabaikan langkah ini.
3. Buatlah branch baru dengan nama “footer” dengan branch development-1.0 sebagai parent. Checkout ke branch footer, lakukan upload branch footer ke remote dengan cara Push to Origin.
4. **Checkout kembali ke branch header.** Pastikan working area yang aktif saat ini adalah branch header. Masukkan kode berikut di index.html line 7. Yaitu setelah opening tag body, dan sebelum opening tag div dengan id = profile-container:

```
1. <div id="header">Header</div>
```

```
index.html x
1  <!DOCTYPE html>
2  <html>
3  |  <head>
4  |  |  <link rel="stylesheet" href="style.css">
5  |  </head>
6  |  <body>
7  |  |  <div id="header">Header</div>
8  |  |  <div id="profile-container">
9  |  |  |  
10 |  |  |  <h2 id="profile-name">Angga Pratama</h2>
11 |  |  </div>
12 |  </body>
13 </html>
```

Commit dengan message “added new div as header container with id = header” lalu **Push**.

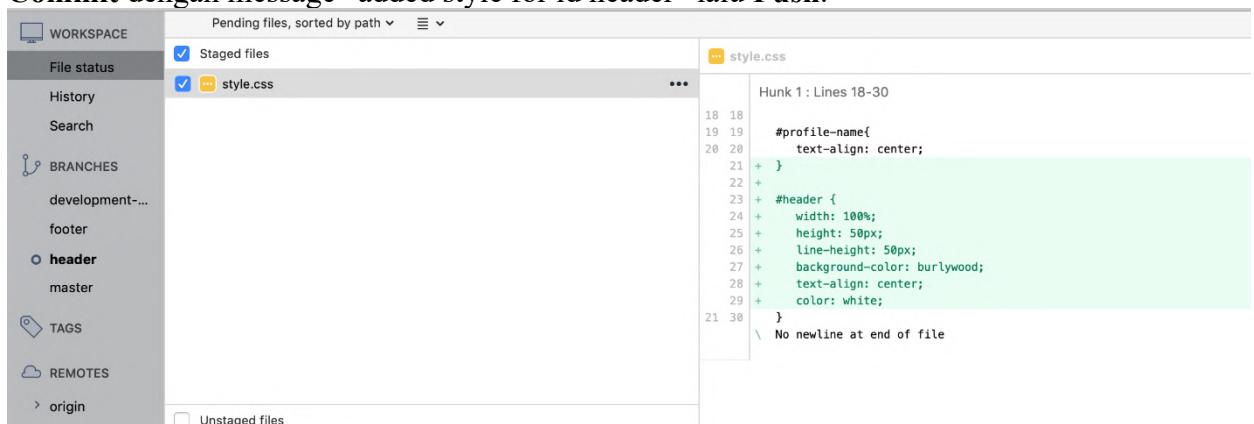
Modul Praktikum Kakas Pengembangan Perangkat Lunak



5. **Masih di branch header.** Masukkan kode berikut di style.css di line paling bawah:

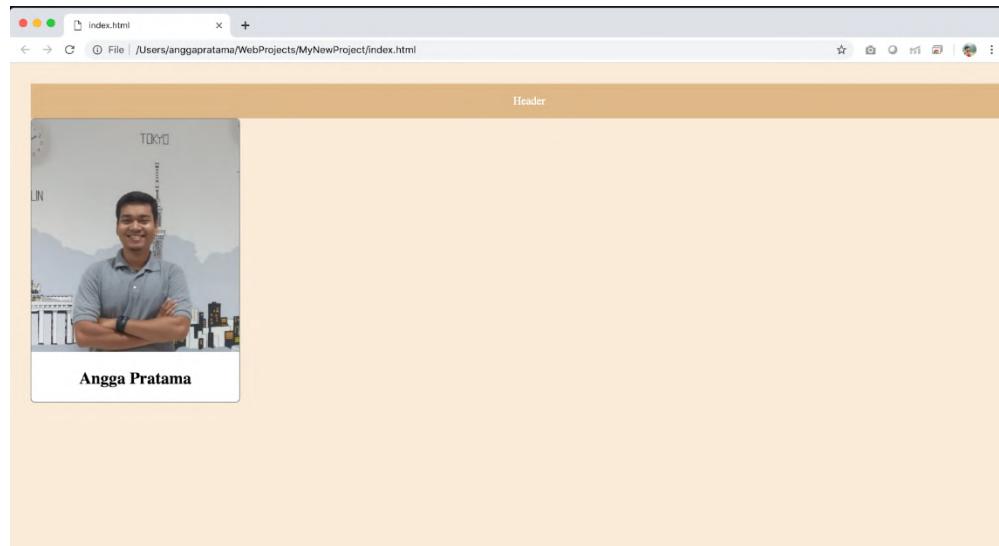
```
1. #header {
2.   width: 100%;
3.   height: 50px;
4.   line-height: 50px;
5.   background-color: burlywood;
6.   text-align: center;
7.   color: white;
8. }
```

Commit dengan message “added style for id header” lalu Push.



Run index.html di browser, maka akan tampil seperti ini:

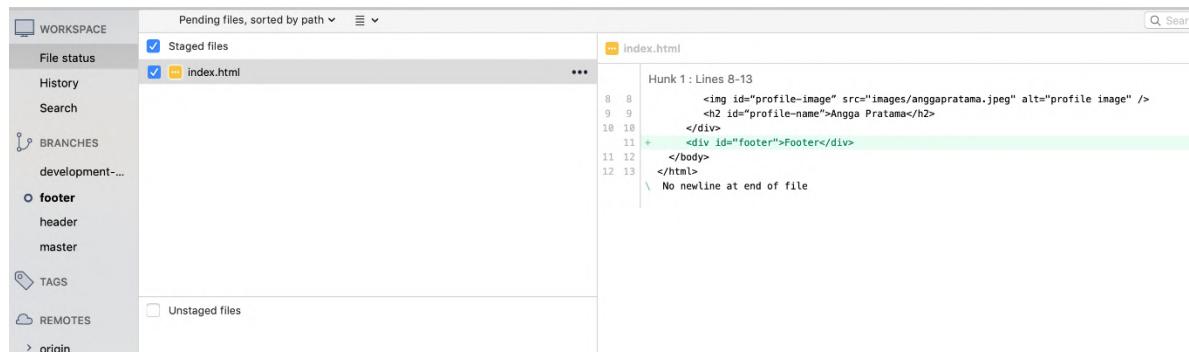
Modul Praktikum Kakas Pengembangan Perangkat Lunak



6. Selesai di branch header, selanjutnya lakukan **checkout ke branch footer**.
7. Pastikan working area yang aktif saat ini adalah branch footer. Masukkan kode berikut di index.html line 11, yaitu setelah closing tag div dengan id profile-container dan sebelum closing tag body:

```
1. <div id="footer">Footer</div>
```

Commit dengan message “added new div as footer container with id = footer” lalu Push.



8. Masih di branch footer. Masukkan kode berikut di style.css di line paling bawah:

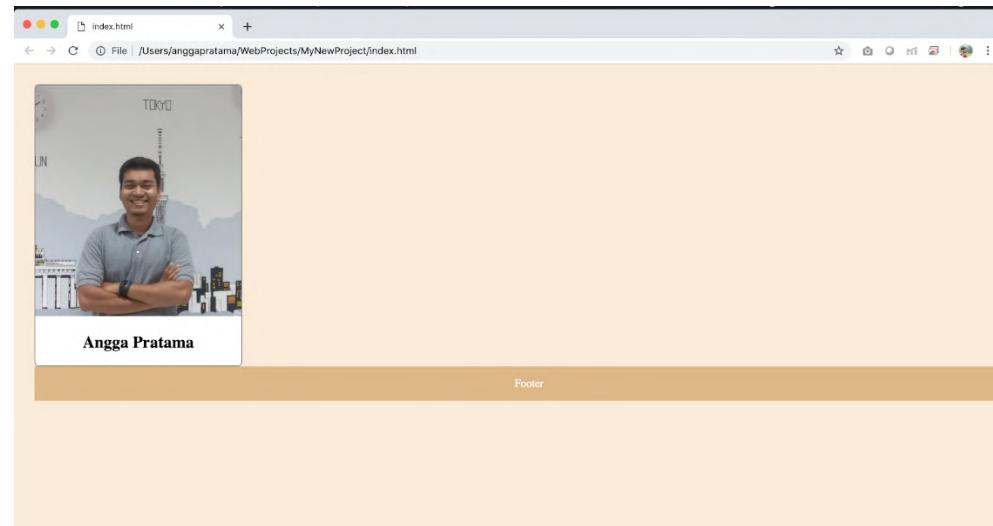
```
1. #footer {  
2.     width: 100%;  
3.     height: 50px;  
4.     line-height: 50px;  
5.     background-color: burlywood;  
6.     text-align: center;  
7.     color: white;  
8. }
```

Commit dengan message “added style for id footer” lalu Push.

Run index.html di browser, maka akan tampil seperti ini:

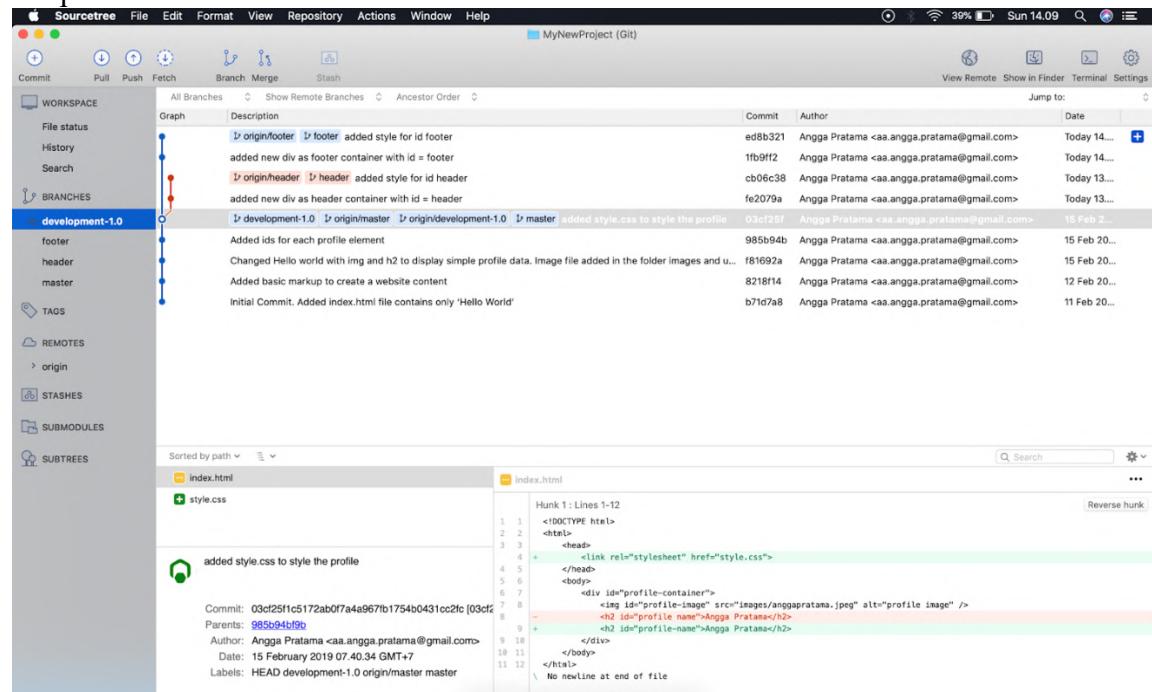


Modul Praktikum Kakas Pengembangan Perangkat Lunak



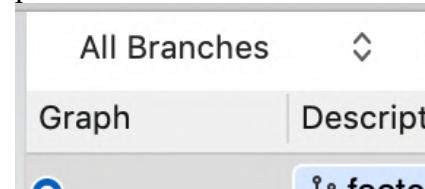
Tentu saja tanpa header. Karena pada saat ini kita berada di branch footer yang tidak memiliki code dari branch header.

9. Saatnya kita melakukan merge. **Checkout ke Branch development-1.0**. Seperti ini sekarang tampilan di Sourcetree.



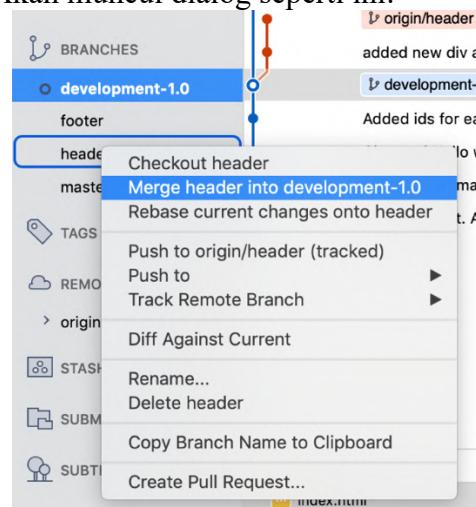
Perhatikan history area. Anda bisa melihat branch development-1.0 bercabang ke dua branch, header dan footer. Apakah saat ini anda mulai paham dengan Branch?

Pastikan Graph sceme yang dipilih adalah dari **All Branches**.



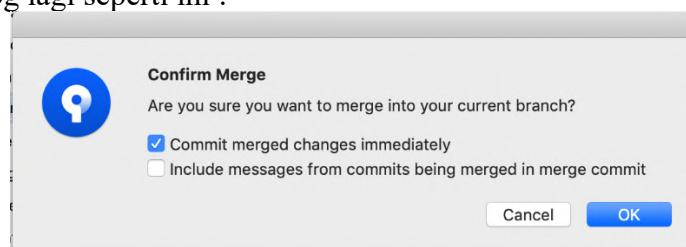
Modul Praktikum Kakas Pengembangan Perangkat Lunak

10. Arahkan pointer mouse Anda ke branch header, ingat, hanya arahkan (hover), bukan untuk checkout, lalu klik kanan. Akan muncul dialog seperti ini:

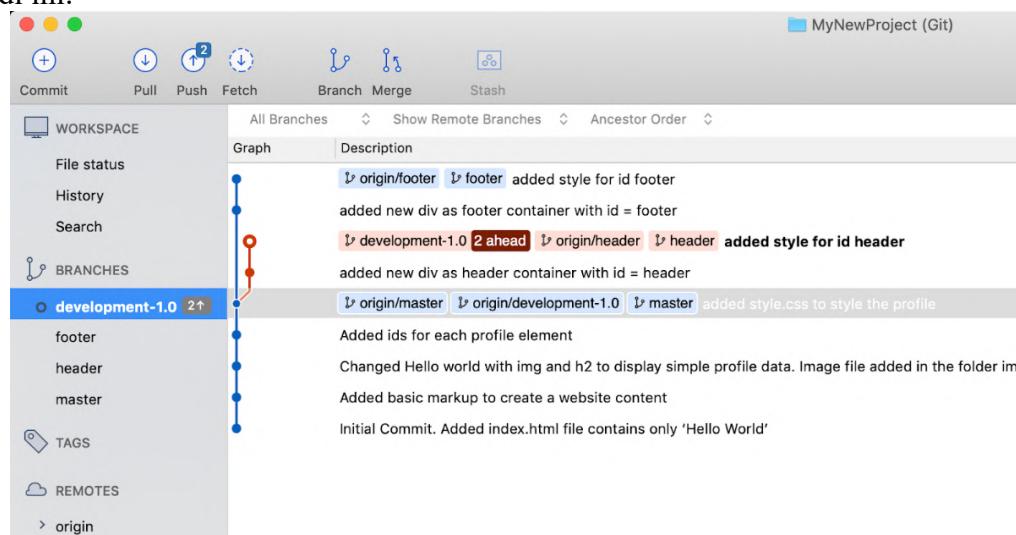


Pilih Merge header into development-1.0. Sesuai perintahnya, kita akan melakukan merge branch header ke dalam branch development-1.0, agar semua perubahan code yang ada di branch header dimiliki juga oleh branch development-1.0.

Akan muncul dialog lagi seperti ini :



Centang **Commit merged changes immediately**. Hal ini dikarenakan kita yakin bahwa tidak akan terjadi konflik pada proses merge ini. Klik **OK**, maka layar Sourcetree Anda akan menjadi ini:

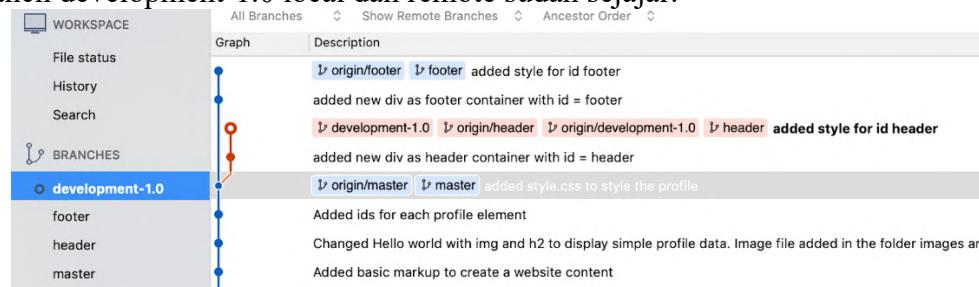


Perhatikan panel history area. Branch local development-1.0 kini naik sejajar dengan branch header, akan tetapi branch remote development-1.0 masih tertinggal 2 commit-an. Oleh karena itu jika Anda perhatikan lagi label development-1.0 local, terdapat tambahan label "2 ahead".

Modul Praktikum Kakas Pengembangan Perangkat Lunak

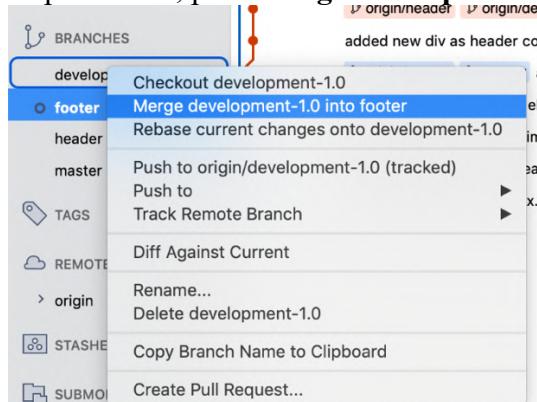
Artinya, branch local development-1.0 berada 2 commit-an di depan branch remote-nya. Label ini juga muncul di area Branches di panel sebelah kiri, juga pada icon Push di panel bagian atas.

Klik icon **Push** di panel atas agar hasil merge juga diterima oleh remote branch. Kini branch development-1.0 local dan remote sudah sejajar.

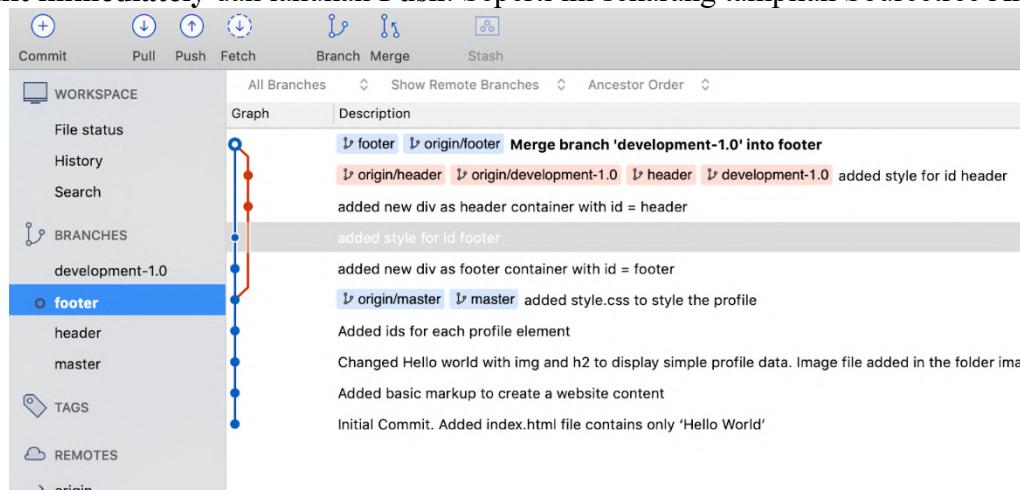


11. Selanjutnya kita akan melakukan merge dari branch footer ke branch development-1.0. Tapi langkah yang akan kita lakukan sedikit berbeda dengan pada saat kita melakukan merge dari branch header ke development-1.0. Perbedaannya adalah, sebelum kita melakukan merge dari footer ke development-1.0, kita akan melakukan merge dari development-1.0 ke footer. Untuk itu, **silakan checkout ke branch footer**, sehingga working area yang aktif saat ini adalah branch footer.

12. Klik kanan di branch development-1.0, pilih **Merge development-1.0 into footer**.



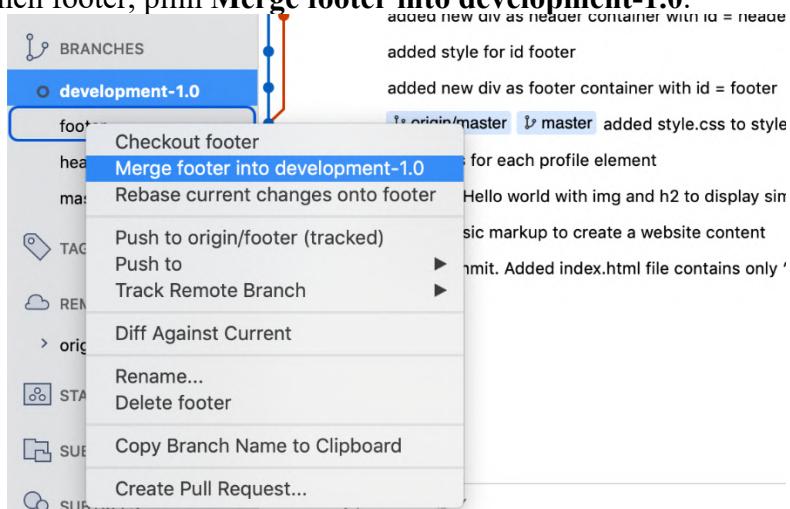
13. Commit immediately dan lakukan **Push**. Seperti ini sekarang tampilan Sourcetree Anda:



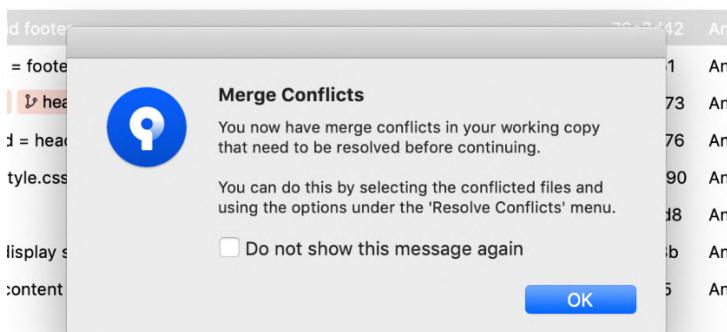
Modul Praktikum Kakas Pengembangan Perangkat Lunak

Branch footer berada satu commit-an di atas development-1.0 dan header. Kondisinya saat ini yaitu, perubahan pada branch development-1.0 dan header kini dimiliki juga oleh footer. Akan tetapi perubahan yang di footer, tidak dimiliki oleh development-1.0. Saat kita melakukan merge balik dari footer ke development-1.0.

14. Checkout ke branch development-1.0, dan pastikan working area yang aktif saat ini adalah branch development-1.0.
15. Klik kanan di branch footer, pilih **Merge footer into development-1.0**.



16. Commit Immediately, maka akan terjadi konflik.

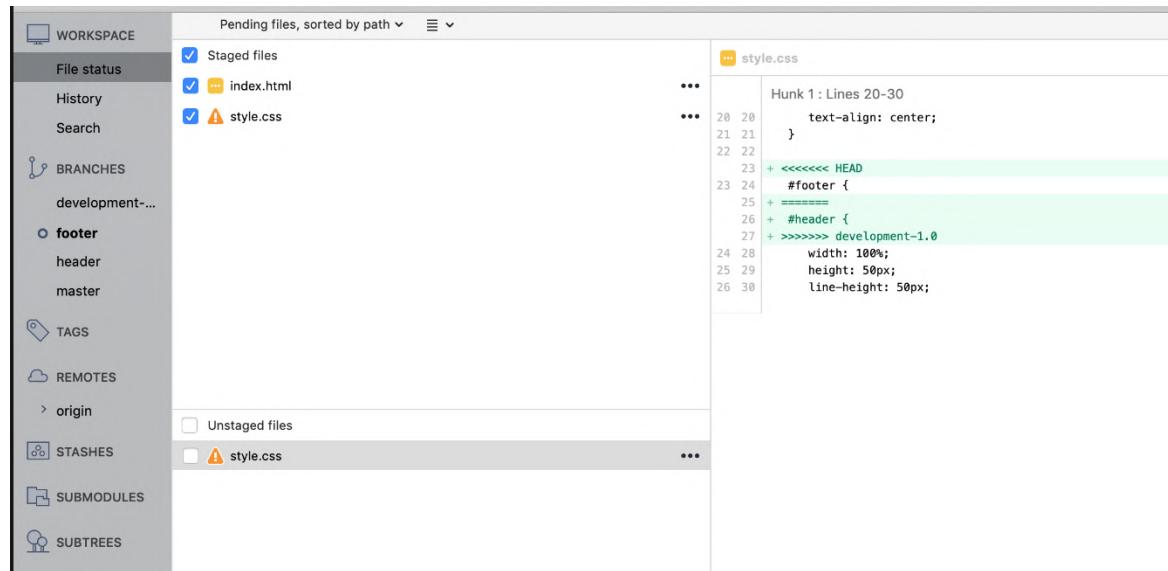


Hal ini wajar saja, karena kita tadi menambahkan style untuk header dan footer sama-sama di line terakhir dari file style.css.

Jangan centang "Do not show this message again" agar selalu ada peringatan jika terjadi konflik. Klik OK.

17. Sekarang kita akan memahami konfliknya dahulu. Pastikan branch yang aktif saat ini adalah masih branch footer. **Klik File Status**
Akan seperti ini tampilan pada Sourcetree:

Modul Praktikum Kakas Pengembangan Perangkat Lunak



dan seperti ini tampilan VS Code:

The screenshot shows a dark-themed VS Code editor. The file being edited is 'style.css'. A conflict is visible at the bottom of the code, indicated by colored highlights and markers:

```
...  
# style.css x  
1 body {  
2     width: 100%;  
3     margin: 30px;  
4     background-color: #antiquewhite;  
5 }  
6  
7 #profile-container{  
8     max-width: 300px;  
9     background-color: #white;  
10    border: 1px solid #gray;  
11    border-radius: 6px;  
12 }  
13  
14 #profile-image{  
15     width: 100%;  
16     height: auto;  
17 }  
18  
19 #profile-name{  
20     text-align: center;  
21 }  
22  
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes  
<<<<< HEAD (Current Change)  
24 #footer {  
25 =====  
26 #header [br/>27 >>>>> development-1.0 (Incoming Change)  
28     width: 100%;  
29     height: 50px;  
30     line-height: 50px;  
31     background-color: burlywood;  
32     text-align: center;  
33     color: white;  
34 }
```

Konflik terjadi karena kita menambahkan style untuk header dan footer sama-sama di line terakhir dengan style value yang juga sama.

Hal tersebut membuat Git menganggap perubahan yang terjadi hanya selectornya saja, yaitu :

- #header yg berasal dari branch development-1.0 (Incoming Change)
- #footer yang berasal dari branch footer (Current Change/HEAD)

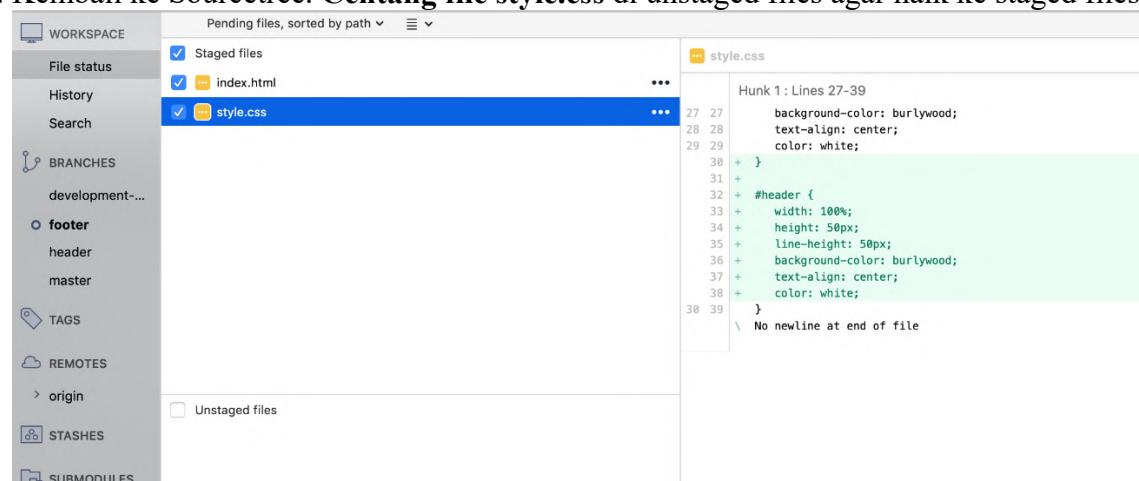
Modul Praktikum Kakas Pengembangan Perangkat Lunak

18. Perrbaiki konflik nya dengan menghapus TAG yang diberikan oleh Git seperti <<< HEAD dan >>> "incoming branch", lalu menulis ulang kodennya menjadi sebagaimana seharusnya

Atur kode pada style.css menjadi seperti ini:

```
# style.css *  
1 body {  
2     width: 100%;  
3     margin: 30px;  
4     background-color: #antiquewhite;  
5 }  
6  
7 #profile-container{  
8     max-width: 300px;  
9     background-color: #white;  
10    border: 1px solid #gray;  
11    border-radius: 6px;  
12 }  
13  
14 #profile-image{  
15     width: 100%;  
16     height: auto;  
17 }  
18  
19 #profile-name{  
20     text-align: center;  
21 }  
22  
23 #footer {  
24     width: 100%;  
25     height: 50px;  
26     line-height: 50px;  
27     background-color: #burlywood;  
28     text-align: center;  
29     color: #white;  
30 }  
31  
32 #header {  
33     width: 100%;  
34     height: 50px;  
35     line-height: 50px;  
36     background-color: #burlywood;  
37     text-align: center;  
38     color: #white;  
39 }
```

19. Kembali ke Sourcetree. Centang file style.css di unstaged files agar naik ke staged files.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

20. Klik icon plus untuk melakukan **Commit**,



Akan muncul default message tanda bahwa kita telah selesai memperbaiki konflik:

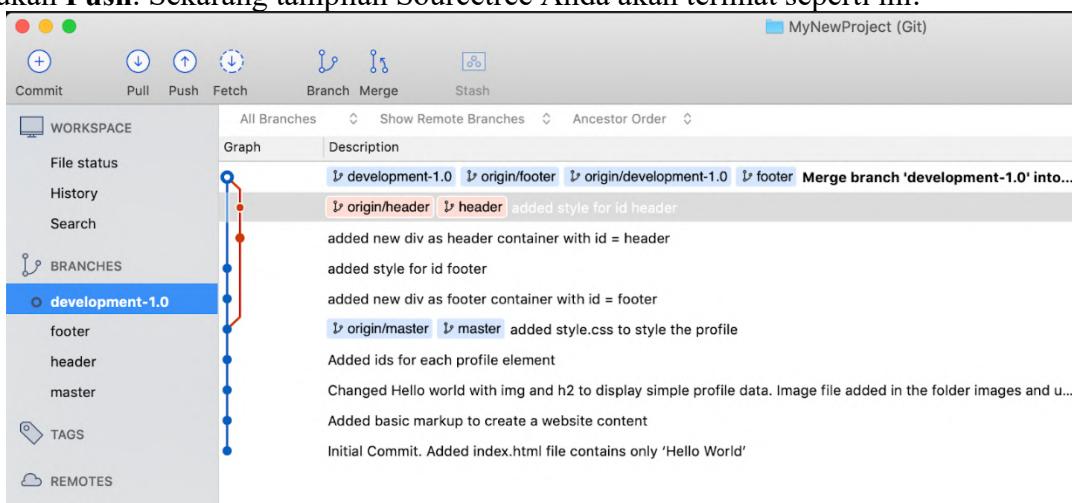
```
Angga Pratama <aa.angga.pratama@gmail.com>
Merge branch 'development-1.0' into footer

# Conflicts:
#   style.css

 Push changes immediately to origin/footer
```

Klik **Commit**.

21. Lakukan **Push**. Sekarang tampilan Sourcetree Anda akan terlihat seperti ini:



Branch development-1.0 berada sejajar dengan footer.

22. Silakan Anda buka VS Code, lalu lihat kode di file index.html dan style.css. Kode dari branch header dan footer sekarang ada bersama-sama. Seperti ini kode di index.html:

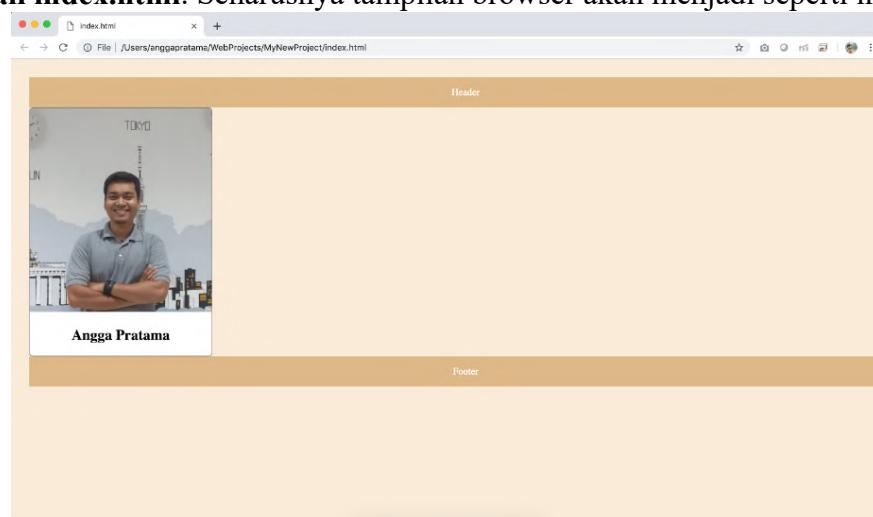
```
index.html — MyNewProject
index.html # style.css
1  <!DOCTYPE html>
2  <html>
3  |  <head>
4  |  |  <link rel="stylesheet" href="style.css">
5  |  </head>
6  |  <body>
7  |  |  <div id="header">Header</div>
8  |  |  <div id="profile-container">
9  |  |  |  
10 |  |  |  <h2 id="profile-name">Angga Pratama</h2>
11 |  |  </div>
12 |  |  <div id="footer">Footer</div>
13 |  </body>
14 </html>
```

Modul Praktikum Kakas Pengembangan Perangkat Lunak

Dan seperti ini kode yang ada di style.css:

```
# style.css x
1 body {
2     width: 100%;
3     margin: 30px;
4     background-color: #antiquewhite;
5 }
6
7 #profile-container{
8     max-width: 300px;
9     background-color: #white;
10    border: 1px solid #gray;
11    border-radius: 6px;
12 }
13
14 #profile-image{
15     width: 100%;
16     height: auto;
17 }
18
19 #profile-name{
20     text-align: center;
21 }
22
23 #footer {
24     width: 100%;
25     height: 50px;
26     line-height: 50px;
27     background-color: #burlywood;
28     text-align: center;
29     color: #white;
30 }
31
32 #header {
33     width: 100%;
34     height: 50px;
35     line-height: 50px;
36     background-color: #burlywood;
37     text-align: center;
38     color: #white;
39 }
```

23. Run kembali index.html. Seharusnya tampilan browser akan menjadi seperti ini:



Header dan Footer kini sudah muncul mengapit layout profile.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Selamat, kini Anda sudah mahir melakukan commit, push, create branch, checkout branch, dan merge branch. Anda juga sudah belajar bagaimana memperbaiki konflik. Mudah sekali bukan? Semua akan menjadi mudah dipelajari jika kita mendapatkan visualisasi dari apa yang kita pelajari dan kita kerjakan.

Lalu selanjutnya apa? Jika diperhatikan baik-baik keseluruhan tampilan website yang telah kita buat, tampilan seperti itu jelek sekali bukan? Betul, tidak nampak keindahan sama sekali. Mari kita beri keindahan pada website tersebut. Pada modul berikutnya kita akan latihan sekali lagi untuk membiasakan diri kita dengan atomic commit dan branching management.

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





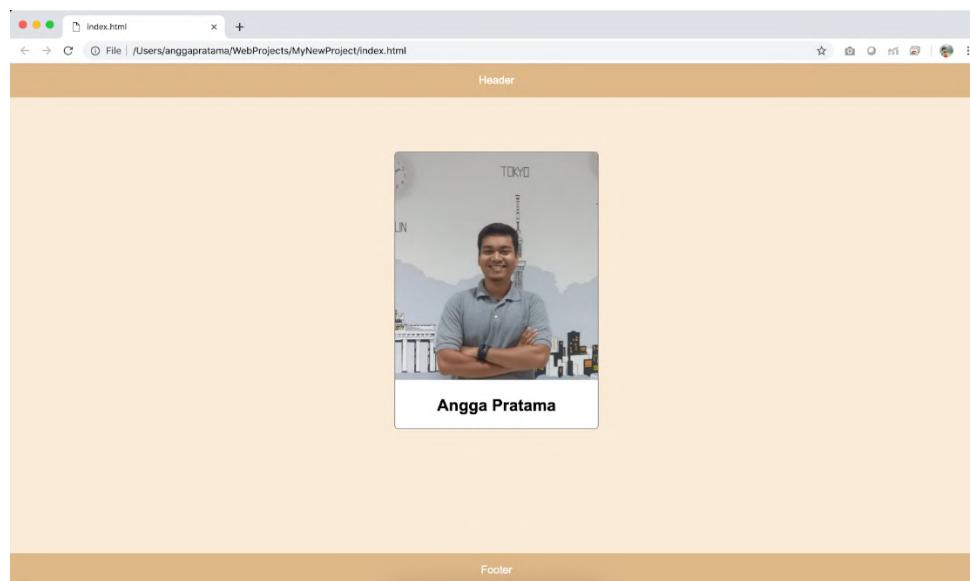
Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 7: Latihan lanjutan Atomic Commit and Branch Management

Deskripsi Pertemuan

Di modul ini kita akan latihan lagi melakukan atomic commit dan branch management dengan mempercantik tampilan website yang sudah kita buat. Hasil akhirnya akan menjadi seperti ini :



Syarat Kompetensi

1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
2. Sed mollis aliquam nibh.
3. Pellentesque pellentesque dictum urna.

Standar Kompetensi

1. [standar kompetensi, capaian minimal setelah praktikum]
2. [standar kompetensi, capaian minimal]

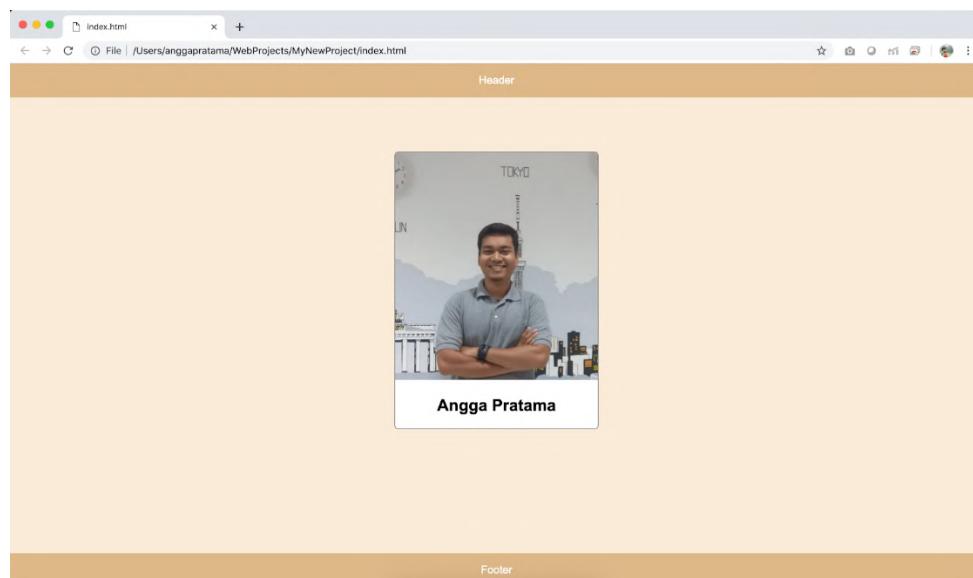




Modul Praktikum Kakas Pengembangan Perangkat Lunak

Dasar Teori

Di modul ini kita akan latihan lagi melakukan atomic commit dan branch management dengan mempercantik tampilan website yang sudah kita buat. Hasil akhirnya akan menjadi seperti ini :



Well, masih belum cantik betul, tetapi tampilannya kini sudah cukup responsif dan mengikuti style website kekinian.

Secara umum, yang kita lakukan hanyalah mempercantik tampilan, sehingga kita cukup hanya bekerja di file style.css. Kita akan membuat Header melekat di layout paling atas, dan footer melekat di layout paling bawah. Juga kita akan menempatkan layout profile di tengah-tengah layout website. Dan terakhir kita akan memastikan layout yang kita buat sudah responsif dengan perbedaan jenis device.

Tugas Pendahuluan

[soal-soal TP yang berhubungan dengan materi pertemuan, berbasis masalah rill]

Langkah-Langkah Praktikum

1. Buatlah branch baru bernama “**web-style**” dengan branch development-1.0 sebagai parent, lalu **Push to Origin**.
2. Checkout ke branch web-style dan pastikan working area yang aktif pada saat ini adalah branch web-style.
3. Buka VSCode lalu buka file style.css.
4. Pada selector body, ubah kodennya menjadi seperti berikut:

```
1. body {  
2.   width: 100%;  
3.   height: 100%;  
4.   margin: 0px;  
5.   padding: 0px;  
6.   background-color: antiquewhite;  
7.   font-family: Arial, Helvetica, sans-serif;  
8. }
```



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Seperti ini perubahan yang terjadi:

Hunk 1 : Lines 1-10

```
1 1     body {  
2 2         width: 100%;  
3 -     margin: 30px;  
3 +     height: 100%;  
4 +     margin: 0px;  
5 +     padding: 0px;  
4 6         background-color: antiquewhite;  
7 +     font-family: Arial, Helvetica, sans-serif;  
5 8     }  
6 9  
7 10 #header {
```

Stage hunk Discard hunk

Commit dengan message “**changed style for selector body to make the layout prettier, and changed font to Arial**”, lalu **Push**.

- Pada selector #profile-container, ubah kodennya menjadi seperti berikut:

```
1. #profile-container{  
2.     max-width: 300px;  
3.     margin: 80px auto;  
4.     background-color: white;  
5.     border: 1px solid gray;  
6.     border-radius: 6px;  
7. }
```

Seperti ini perubahan yang terjadi:

Hunk 2 : Lines 18-24

```
15 18  
16 19     #profile-container{  
17 20         max-width: 300px;  
21 +     margin: 80px auto;  
18 22         background-color: white;  
19 23         border: 1px solid gray;  
20 24         border-radius: 6px;
```

Stage hunk Discard hunk

Commit dengan message “**changed style for selector #profile-container, to make it centered horizontally**”, lalu **Push**.

- Pada selector #footer, ubah kodennya menjadi seperti berikut:

```
1. #footer {  
2.     width: 100%;  
3.     height: 50px;  
4.     position: absolute;  
5.     bottom: 0;  
6.     line-height: 50px;  
7.     background-color: burlywood;  
8.     text-align: center;  
9.     color: white;  
10. }
```

Seperti ini perubahan yang terjadi:

Hunk 3 : Lines 36-43

```
32 36  
33 37     #footer {  
34 38         width: 100%;  
39 +     height: 50px;  
40 +     position: absolute;  
35 41         bottom: 0;  
36 42         line-height: 50px;  
37 43         background-color: burlywood;  
              text-align: center;
```

Stage hunk Discard hunk

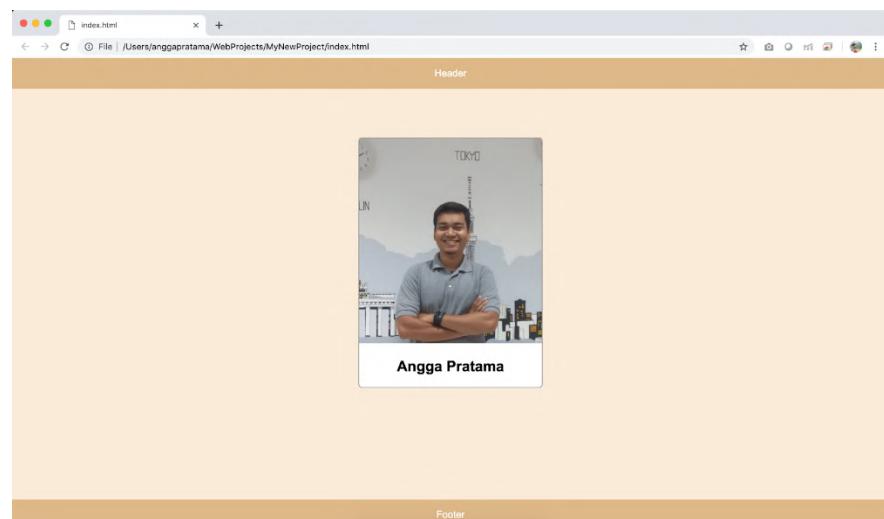
Commit dengan message “**changed style for selector body**”, lalu **Push**.

- Silakan anda run kembali index.html. Seharusnya tampilan website pada layar besar akan seperti ini:

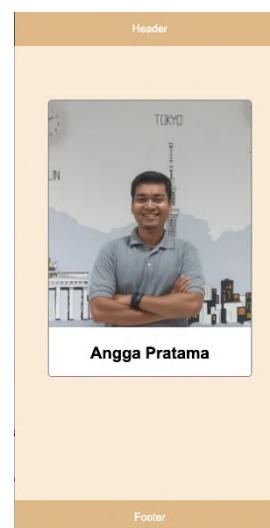




Modul Praktikum Kakas Pengembangan Perangkat Lunak



Dan pada layar device smartphone akan menjadi seperti ini:



8. Jika dirasa penampilan website sudah cukup cantik, kita akan merge branch web-style ke branch development-1.0. Tetapi sebelumnya, untuk memastikan semua perubahan di branch development-1.0 sudah dimiliki oleh branch web-style, kita merge dahulu branch development-1.0 ke branch web-style.
9. Pastikan branch web-style masih dalam kondisi aktif.
10. Klik kanan di branch development-1.0 dan pilih **Merge development-1.0 into web-style**.
11. **Commit immediately** lalu lakukan **Push**.
12. Selanjutnya checkout ke branch development-1.0 dan pastikan working area yang aktif adalah branch development-1.0.
13. Klik kanan pada **branch web-style** dan pilih **Merge web-style into development-1.0**.
14. **Commit immediately** lalu lakukan **Push**.
15. Coba run kembali index.html, jika tampilannya sudah sama, berarti anda sukses melakukan perubahan style di branch yang berbeda.

Selesai. Bagaimana perasaan Anda saat ini? Apakah semakin mahir dengan Atomic Commit dan Branching Management?

Silakan lanjutkan ke modul berikutnya. Collaboration.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 8: Stash

Deskripsi Pertemuan

Satu lagi kemampuan super dari Git, yaitu Stash. Apa itu Stash?

Sederhananya, Stash adalah aksi untuk menyimpan semua perubahan yang masih berada di unstaged area. Untuk apa hal itu dilakukan? Bukankah sudah ada Commit dan Push?

Syarat Kompetensi

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Sed mollis aliquam nibh.
3. Pellentesque pellentesque dictum urna.

Standar Kompetensi

1. [standar kompetensi, capaian minimal setelah praktikum]
2. [standar kompetensi, capaian minimal]

Dasar Teori

Satu lagi kemampuan super dari Git, yaitu Stash. Apa itu Stash?

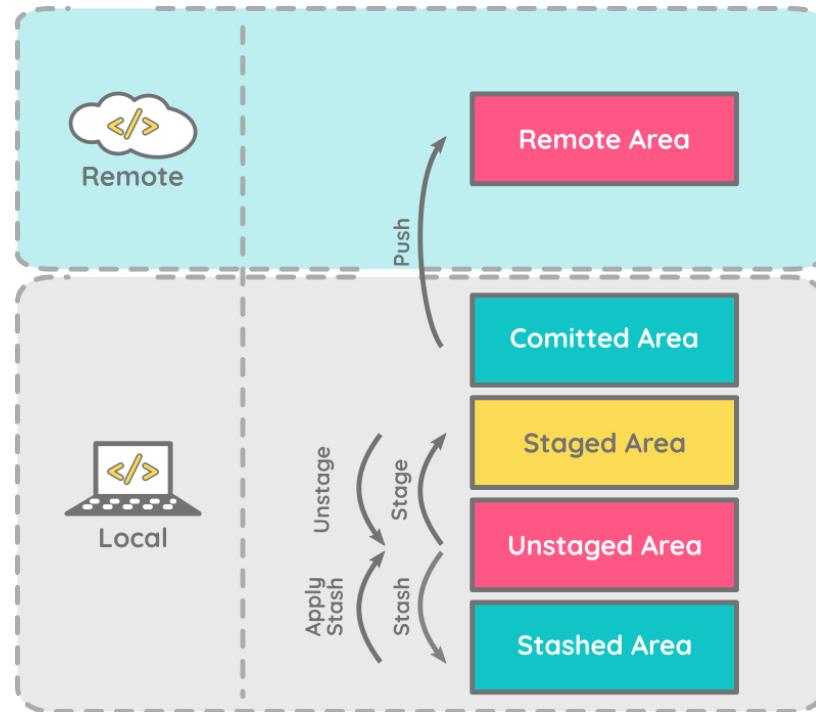
Sederhananya, Stash adalah aksi untuk menyimpan semua perubahan yang masih berada di unstaged area. Untuk apa hal itu dilakukan? Bukankah sudah ada Commit dan Push?

Tidak, Stash jauh berbeda dengan apa yang dilakukan Commit atau Push. Commit adalah aksi untuk menyimpan perubahan yang Anda yakini perubahan itu sudah benar, dan anda berkomitmen perubahan itulah yang Anda pilih untuk masuk ke dalam Commit history. Sedangkan Stash, menyimpan perubahan yang Anda rasa belum benar, belum cukup bagus untuk di-Commit, dan perubahan itu hanya akan disimpan di local komputer Anda. Tidak ada orang lain yang bisa melihat perubahan tersebut kecuali anda.





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Tugas Pendahuluan

[soal-soal TP yang berhubungan dengan materi pertemuan, berbasis masalah rill]

Langkah-Langkah Praktikum

1. Buka Sourcetree, checkout ke branch development-1.0
2. Buka VS Code, di index.html masukkan kode berikut di bawah element H2 dengan id = "profile-name". Anda bebas mengisinya dengan apa saja:

```
<p id="profile-job">Software Engineer</p>
```

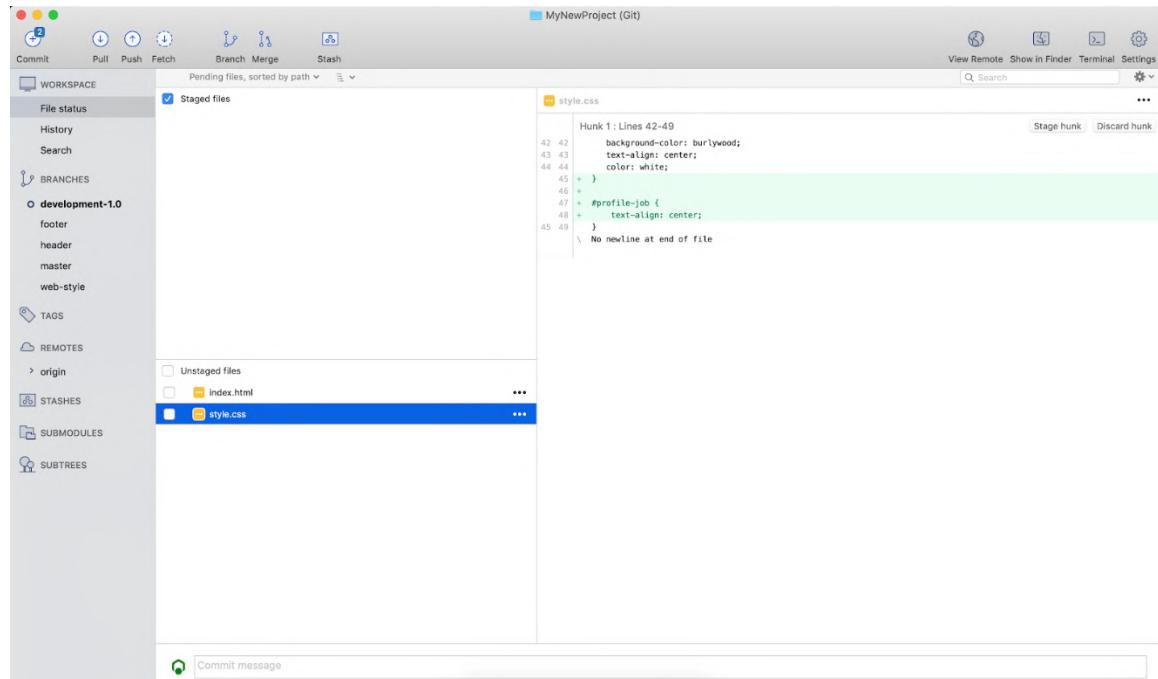
3. Buka style.css dan masukkan kode berikut di line paling akhir:

```
#profile-job {  
    text-align: center;  
}
```

4. Silakan anda run index.html dan lihat hasilnya.
5. Sebelum anda melakukan Commit, misalnya merasa bahwa mencantumkan nama dan job di profile tidaklah bagus. Anda berpikir untuk langsung saja menambahkan deskripsi pada profile. Tapi hal itu juga Anda tidak yakin. Anda masih berpikir menempatkan job pada profile juga cukup bagus. Anda ragu. Ditengah keraguan itulah Stash bisa Anda gunakan.
6. Buka Sourcetree. Klik pada **File Status**. Seharusnya tampilannya seperti ini:

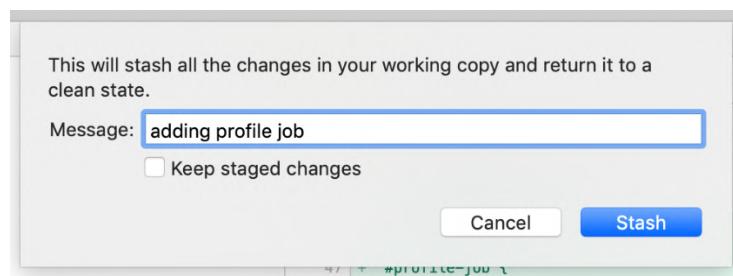


Modul Praktikum Kakas Pengembangan Perangkat Lunak



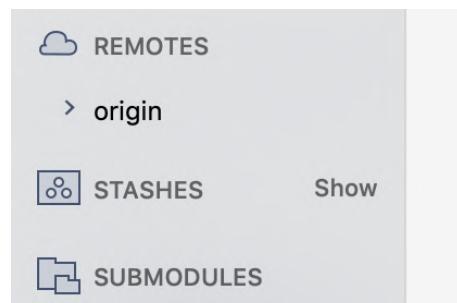
Ada perubahan di 2 file yang Anda tidak yakin perubahan tersebut sudah benar apa belum. Untuk itu perlu untuk melakukan Stash.

7. Klik icon **Stash** di panel atas Sourcetree. Akan muncul dialog seperti berikut:



Isikan message sesuka hati karena hanya akan disimpan di local. Tapi pastikan message dapat dibaca untuk menggambarkan perubahan apa yang dilakukan. Klik tombol Stash.

8. File Status area akan kosong. Untuk memastikannya silakan cek file index.html dan style.css. Perubahan yang terakhir Anda lakukan sudah hilang. Karena jika tombol di Stash diklik, semua perubahan di semua file akan disimpan menjadi suatu archive di Stash area. Untuk melihat Stash area, hover menu **Stash** di panel sebelah, dan klik **Show**.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

- Setelah Stash area terbuka, akan muncul list archive perubahan yang disimpan. Klik perubahan yang barusan Anda Stash. Maka akan muncul panel di sebelah kanan yang menunjukkan detail perubahan dari semua file yang anda Stash.

The screenshot shows the GitKraken application window titled "MyNewProject (Git)". The left sidebar includes buttons for Commit, Pull, Push, Fetch, Branch, Merge, and Stash. The Stash section is selected, showing a list of stashed changes:

- development-1.0**:
 - footer
 - header
 - master
 - web-style
- TAGS**: None
- REMOTES**: origin
- STASHES**:
 - On development-1.0: adding profile job
- SUBMODULES**: None
- SUBTREES**: None

The main pane displays two stashed changes:

- index.html**: Hunk 1: Lines 8-14. Content:

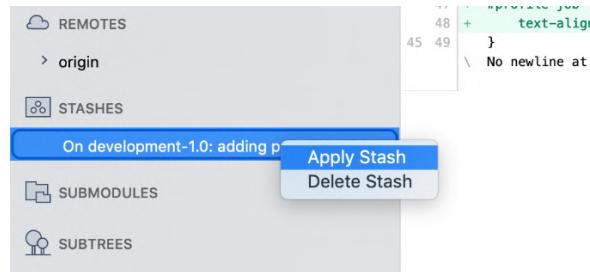
```
<div id="profile-container">
  
  <h2 id="profile-name">Angga Pratama</h2>
  + <p id="profile-job">Software Engineer</p>
</div>
<div id="footer">Footer</div>
```
- style.css**: Hunk 1: Lines 42-49. Content:

```
background-color: burlywood;
text-align: center;
color: white;
+
+
#profile-job {
  text-align: center;
}
\ No newline at end of file
```

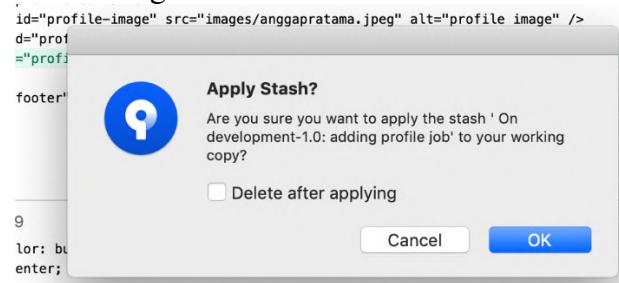
Anda bisa melakukan perubahan lain, dan melakukan Stash untuk perubahan kedua. Maka akan muncul 2 archive di Stash area.

Suatu saat, Anda sudah memutuskan perubahan mana yang akan di-commit, Anda hanya perlu mengambil kembali perubahan dari Stash area. Aksi itu disebut dengan Apply Stash. Lakukan langkah berikut untuk Apply Stash:

- Klik kanan pada salah satu archive Stash yang Anda inginkan, maka akan muncul menu seperti berikut:



- Pilih **Apply Stash**, maka dialog baru akan muncul:



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Silakan baca baik-baik pesan dari dialog tersebut. Biarkan saja tombol Delete after applying dalam keadaan tidak tercentang. Jika Anda centang, maka archive akan terhapus setelah Stash di Apply. Klik **OK**.

3. Klik **File Status**, maka anda akan melihat hasil Apply Stash. Silakan cek index.html dan style.css untuk membuktikan:

```
MyNewProject (Git)
Commit Pull Push Fetch Branch Merge Stash
Pending files, sorted by path ...
File status
History Search
BRANCHES
development-1.0
footer header master web-style
TAGS
REMOTES origin
STASHES On development-1.0: adding profile job
SUBMODULES
SUBTREES
index.html
Hunk 1 : Lines 8-14
8 8 <div id="profile-container">
9 9 
10 10 <h2 id="profile-name">Angga Pratama</h2>
11 + <p id="profile-job">Software Engineer</p>
12 12 </div>
13 13 <div id="footer">Footer</div>
14 </body>
Unstaged files
index.html
style.css
```

Perlu diingat kembali bahwa Stash hanya menyimpan perubahan di local dalam bentuk archives. Jika sesuatu terjadi pada komputer anda sehingga kehilangan data, maka archive akan ikut hilang.

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 9: Collaboration

Deskripsi Pertemuan

Sebelumnya kami ucapkan, Selamat! Pada tahap ini Anda sudah semakin mahir menggunakan Sourcetree, khususnya Atomic Commit dan Branching Management. Anda juga sudah mendapatkan gambaran umum tentang code versioning atau Source Code Management.

Tetapi perlu Anda sadari, sampai saat ini Anda bekerja sendirian. Bagaimana jika Anda berada dalam sebuah team, bekerja sama mengembangkan sebuah software? Apakah akan semakin rumit?

Bertambah rumit, ya. Tetapi tidak serumit yang Anda bayangkan. Jika Anda sudah pandai melakukan commit, push, create branch, checkout, dan merge, maka Anda sudah mampu untuk bekerja dalam sebuah team, baik itu hanya segelintir orang, ataupun ratusan. Wow!

Modul ini akan membahas mengenai Collaboration, yaitu berkolaborasi dengan developer lain, mengembangkan sebuah software dalam repository yang sama. Tetapi mulai dari modul ini sampai modul-modul berikutnya, kami tidak akan membahas hal-hal detail. Kami hanya akan menunjukkan cara-caranya saja secara umum.

Syarat Kompetensi

1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
2. Sed mollis aliquam nibh.
3. Pellentesque pellentesque dictum urna.

Standar Kompetensi

1. [standar kompetensi, capaian minimal setelah praktikum]
2. [standar kompetensi, capaian minimal]

Dasar Teori





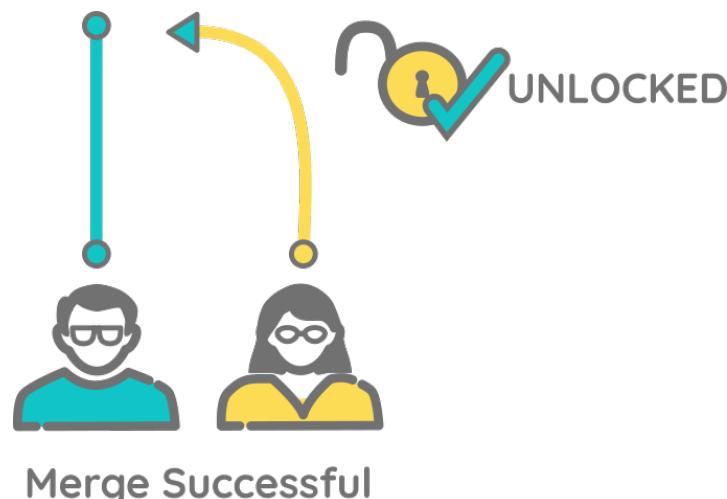
Modul Praktikum Kakas Pengembangan Perangkat Lunak

Collaborators

Collaborators adalah sebuah feature pada penyedia cloud repository sehingga pemilik repository dapat memberikan hak akses kepada temannya untuk berkolaborasi pada repository tersebut.

Menambahkan collaborator memiliki benefit seperti berikut:

1. Dengan menambahkan collaborator, teman Anda bisa mengakses repository private Anda. Perlu diketahui bahwa GitHub membatasi 3 collaborator untuk repository private. Sedangkan BitBucket membatasi sampai 5 collaborator untuk repository private.
2. Collaborator bisa bebas melakukan merge branch ke branch milik creator repository tanpa perlu izin dari creator repository.



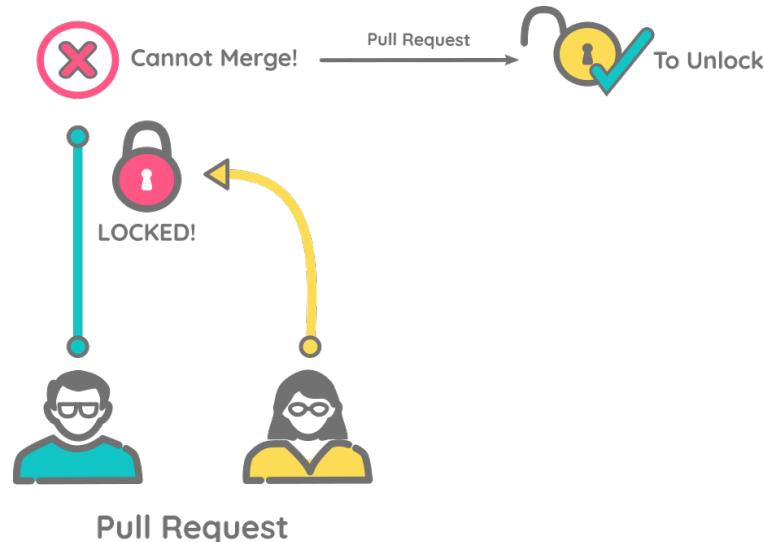
Pull Request

Jika Anda membuat sebuah public repository, siapa saja dapat mengakses repository Anda. Batasannya adalah mereka tidak bisa bebas melakukan push ke branch Anda atau pun merge branch yang mereka buat ke branch milik Anda. Anda bisa saja menjadikan orang tersebut *collaborator*, akan tetapi hal itu terlalu risikan. Anda wajib menjaga branch milik Anda agar kode anda tidak rusak oleh sembarang orang.

Tetapi walaupun begitu, orang lain masih bisa berkolaborasi walau statusnya bukan *collaborator*. Ketika mereka membuat branch baru, dan sudah selesai melakukan perubahan pada branch tersebut, mereka merasa perlu untuk melakukan merge ke branch utama. Karena aksi tersebut dibatasi, maka mereka perlu melakukan sebuah request agar Anda melakukan Checkout pada branch mereka, melakukan Pull, lalu melakukan Merge ke branch utama. Request itu disebut dengan Pull Request, yang jika diterjemahkan ke bahasa Indonesia akan berbunyi : "Hi, saya ada *request*, tolong Pull (tarik) perubahan di branch saya, ke branch utama milikmu!"



Modul Praktikum Kakas Pengembangan Perangkat Lunak



Log dan Annotation Tracing

Jika Anda bekerja dalam team yang cukup besar, lalu pada branch yang sedang anda kerjakan terdapat kode yang menyebabkan masalah. Bisa itu bug, atau penulisannya kurang baik sehingga memperburuk performance. Tetapi Anda yakin bahwa kode tersebut bukanlah Anda yang membuatnya, dan Anda ingin tahu siapa yang menulis kode seperti itu. Sourcetree bisa memberikan jawabannya.



Tugas Pendahuluan

[soal-soal TP yang berhubungan dengan materi pertemuan, berbasis masalah rill]

Langkah-Langkah Praktikum

Collaborators

Silakan Anda buka repository GitHub di browser. Klik **Settings**, lalu klik **Collaborators**. Anda berhak menambahkan teman dengan hanya menginput username GitHub mereka.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

The screenshot shows the GitHub repository settings page for 'arjuna135 / my-repo-from-remote'. The 'Collaborators' tab is selected. On the left, there's a sidebar with options like 'Options', 'Collaborators', 'Branches', 'Webhooks', 'Notifications', 'Integrations & services', 'Deploy keys', 'Moderation', and 'Interaction limits'. In the main area, it says 'This repository doesn't have any collaborators yet. Use the form below to add a collaborator.' Below this is a search bar with 'dicoding' typed in, and two user profiles listed: 'dicodingcontentwriter' and 'dicodingreviewer'. A blue button labeled 'Add collaborator' is visible.

Klik Add collaborator untuk menambahkan teman Anda sebagai collaborator. Email invitation akan dikirimkan oleh GitHub ke email teman Anda tersebut.

Pull Request

Silakan Anda buka repository anda di GitHub, lalu klik **Pull Request**.

The screenshot shows the GitHub repository page for 'arjuna135 / my-repo-from-remote' with the 'Pull requests' tab selected. At the top, there's a message about labeling issues and pull requests for new contributors. Below this, it shows 'Your recently pushed branches:' with 'web-style' highlighted. There are filters for 'is:pr is:open', 'Labels', 'Milestones', and a green 'New pull request' button. A large empty box is labeled 'Welcome to Pull Requests!' with a note: 'Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#)'.

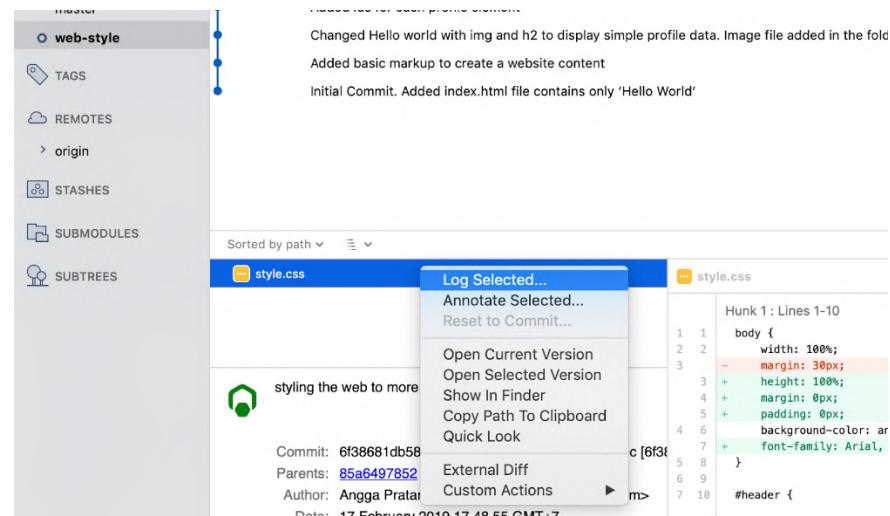
Di halaman ini, bisa dibuat Pull Request baru. Di halaman ini juga Anda bisa melihat List Request dari orang-orang yang melakukan Pull Request di repository Anda. Anda bebas menerima request tersebut atau menolaknya. Jika Anda menolak, pastikan Anda menanggapi mereka dan memberikan alasan kenapa Anda membiarkan atau menolak Request tersebut.

Log dan Annotation Tracing

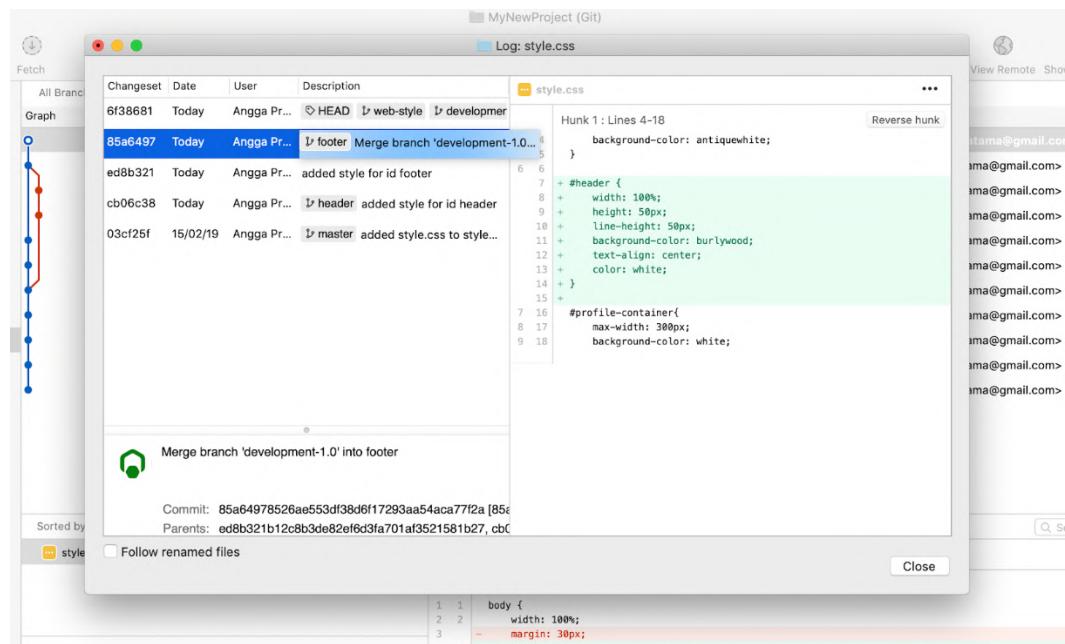
Kita akan gunakan repository yang selama ini kita gunakan. Buka Sourcetree, pilih **branch web-style**, klik pada **style.css**, lalu klik **kanan**. Akan muncul dialog seperti berikut:



Modul Praktikum Kakas Pengembangan Perangkat Lunak



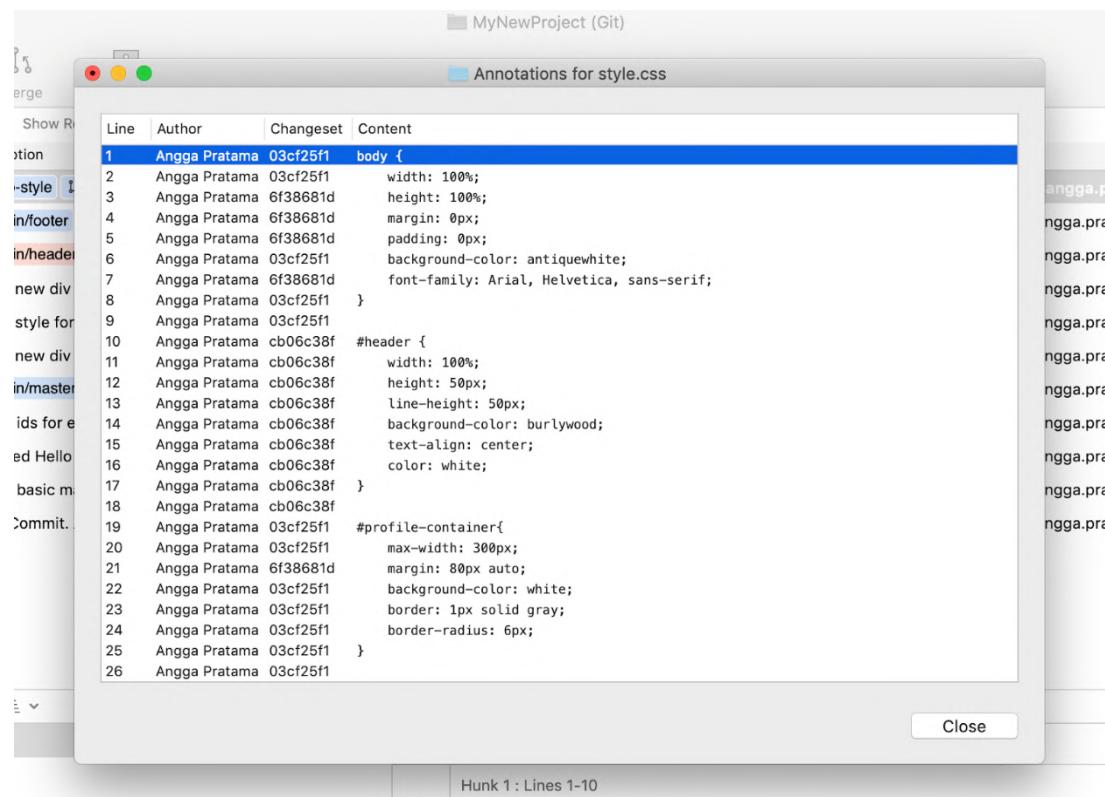
Anda bisa memilih Log Selected, atau Annotate Selected. Untuk saat ini pilih **Log Selected**. Akan muncul jendela seperti ini:



Log Files adalah sebuah fitur untuk melihat rekam jejak suatu file menurut Commit yang dilakukan. Panel sebelah kiri menunjukkan list commit-an yang terjadi pada file tersebut, sedangkan panel sebelah kanan adalah detail perubahan pada commit-an yang Anda pilih. Anda juga bisa melihat siapa yang melakukan commit, tanggal berapa, dan deskripsinya.

Sekarang coba pilih **Annotate Selected**, akan muncul jendela seperti ini:

Modul Praktikum Kakas Pengembangan Perangkat Lunak



The screenshot shows a SourceTree interface with a window titled "Annotations for style.css". The main pane displays a list of code changes with the following details:

Line	Author	Changeset	Content
1	Angga Pratama	03cf25f1	body { width: 100%; height: 100%; margin: 0px; padding: 0px; background-color: antiquewhite; font-family: Arial, Helvetica, sans-serif; }
2	Angga Pratama	03cf25f1	#header { width: 100%; height: 50px; line-height: 50px; background-color: burlywood; text-align: center; color: white; }
3	Angga Pratama	6f38681d	#profile-container{ max-width: 300px; margin: 80px auto; background-color: white; border: 1px solid gray; border-radius: 6px; }
4	Angga Pratama	6f38681d	
5	Angga Pratama	6f38681d	
6	Angga Pratama	03cf25f1	
7	Angga Pratama	6f38681d	
8	Angga Pratama	03cf25f1	
9	Angga Pratama	03cf25f1	
10	Angga Pratama	cb06c38f	
11	Angga Pratama	cb06c38f	
12	Angga Pratama	cb06c38f	
13	Angga Pratama	cb06c38f	
14	Angga Pratama	cb06c38f	
15	Angga Pratama	cb06c38f	
16	Angga Pratama	cb06c38f	
17	Angga Pratama	cb06c38f	
18	Angga Pratama	cb06c38f	
19	Angga Pratama	03cf25f1	
20	Angga Pratama	03cf25f1	
21	Angga Pratama	6f38681d	
22	Angga Pratama	03cf25f1	
23	Angga Pratama	03cf25f1	
24	Angga Pratama	03cf25f1	
25	Angga Pratama	03cf25f1	
26	Angga Pratama	03cf25f1	

A vertical sidebar on the left lists commit messages, and a right sidebar shows a list of names starting with "angga.pr". A "Close" button is at the bottom right of the main pane.

Jendela tersebut menampilkan keseluruhan kode pada file yang Anda pilih. Hebatnya, setiap line dari kode tersebut menginformasikan siapa yang menulisnya. Fitur ini sangatlah berguna pada project yang berumur tahunan dan sering mengalami pergantian developer. Kita bisa mencari tahu siapa yang menuliskan kode tersebut, kemudian bertanya langsung kepada si penulis kode untuk mendapatkan informasi lebih banyak mengenai kode yang dituliskan. Pada gambar, semua nama bertuliskan Angga Pratama, hal ini karena belum ada developer lain yang melakukan perubahan pada file tersebut.

Sedikit fun fact. Pada versi Sourcetree tahun lalu, Fitur Annotation dinamai dengan “Blame”. Mungkin karena kata tersebut berkonotasi negatif sehingga di versi terbaru Sourcetree diganti dengan nama “Annotations.”

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]





Modul Praktikum Kakas Pengembangan Perangkat Lunak



Modul Praktikum 10: SCM Component and Operations

Deskripsi Pertemuan

Telah sampailah kita ke modul terakhir. Memang bukan benar-benar yang terakhir, tapi modul ini adalah modul terakhir bagi pemula yang belajar Source Code Management. Modul-modul yang ada setelah modul ini adalah modul bagi Anda yang sudah merasa terbiasa dengan SCM, bisa dibilang untuk tingkatan intermediate atau advance.

Pada modul ini tidak akan ada latihan, hanya beberapa penjelasan mengenai apa yang sudah kita lakukan pada modul-modul sebelumnya. Pada modul ini akan dibahas mengenai komponen-komponen dan operasi-operasi yang ada dalam Source Code Management. Ya, seharusnya modul ini diletakkan sebagai modul paling awal, akan tetapi penulis memiliki tujuan untuk membuat Anda melakukan praktik dahulu sebelum membaca teori.

Syarat Kompetensi

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Sed mollis aliquam nibh.
3. Pellentesque pellentesque dictum urna.

Standar Kompetensi

1. [standar kompetensi, capaian minimal setelah praktikum]
2. [standar kompetensi, capaian minimal]

Dasar Teori

Sebelum membahas komponen dan operasi SCM, mari kita bahas kembali apa itu SCM. Source Code Management (SCM) - beberapa orang menyebutnya sebagai *Code Versioning* - adalah suatu bentuk manajemen *source code* yang didesain untuk melacak jejak perubahan-perubahan kode selama pengembangan sebuah software berlangsung. Keuntungan dari penerapan SCM adalah :

1. **Cloud storage.** Kode anda akan aman berada di awan.
2. **Private workspace.** Anda bebas melakukan perubahan kode di mesin pribadi Anda sampai Anda melakukan Push untuk Commit yang Anda lakukan.





Modul Praktikum Kakas Pengembangan Perangkat Lunak

3. **Distributed source code.** Anda bisa mendistribusikan kode Anda kepada siapa saja yang Anda kehendaki dan berkolaborasi dengan mereka.
4. **Traceable and trackable changes.** Dengan menerapkan SCM, Anda bisa dengan mudah melacak perubahan kode yang Anda dan Team lakukan.
5. **Version control.** Jika ternyata perubahan baru yang dilakukan tidak lebih baik dari versi sebelumnya, Anda bisa kembali ke versi kode yang lawas.

Tools utama yang kita gunakan untuk belajar SCM diantaranya :

1. **Git**, sebagai open source code versioning control system.
2. **GitHub**, sebagai web-based cloud hosting platform untuk menyimpan dan mendistribusikan kode anda yang di kelola menggunakan Git.
3. **Sourcetree**, sebagai Git Graphic User Interface (GUI) yang memvisualisasikan komponen beserta operasi SCM pada Git.

Baiklah sekarang kita akan membahas tentang komponen SCM dan operasinya yang sebagian besar telah Anda lakukan di modul modul sebelum ini.

SCM Components

Secara umum, komponen SCM terbagi menjadi 3 komponen utama, yaitu remote environment, local environment, dan developer. Remote environment berada di cloud storage, sedangkan local environment berada di komputer yang anda gunakan. Developer dikategorikan juga menjadi SCM component sebagai SCM user yang dalam hal ini adalah Anda dan Team.

Remote environment hanya memiliki satu komponen, yaitu remote repository. Local environment memiliki komponen seperti berikut:

1. Local repository, Berisi folder dan file yang berisi kode-kode Anda.
2. Workspace atau Working Directory, yang bisa juga disebut sebagai unstaged area. Berisikan perubahan yang sedang Anda lakukan.
3. Stage area, berisi perubahan yang anda simpan sementara untuk kemudian Anda commit. Perubahan di stage area masih bisa Anda ganti.
4. Commit area, berisikan perubahan-perubahan yang telah Anda commit. Perubahan di commit area tidak bisa Anda ganti. Untuk mengganti perubahan yang telah di-commit Anda perlu melakukan commit yg lain.
5. Stash area. Untuk menyimpan semua perubahan yang ada di working area sehingga working menjadi kosong.
6. Branch. Sebuah pointer di dalam repository untuk menentukan arah commit, sehingga Anda bisa membuat cabang dari serangkaian commit yang Anda lakukan.
7. Tag. Sebuah label yang diberikan pada sebuah Commit. Label apapun bisa diberikan, contohnya label Release point.

Dari komponen komponen yang sudah disebutkan, dapat dibuat diagram seperti berikut untuk menggambarkan keseluruhan komponen SCM:





Modul Praktikum Kakas Pengembangan Perangkat Lunak

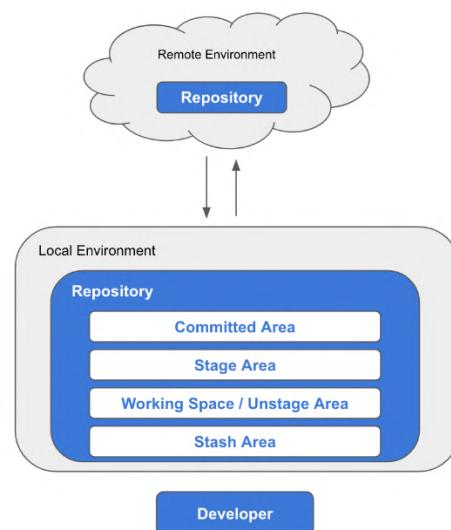


Diagram komponen SCM

SCM Operations

Komponen komponen SCM dihubungkan dengan operasi SCM. Operasi SCM adalah aktivitas yang dilakukan oleh Developer dalam mengelola *source code*. Mirip seperti komponen SCM, operasi SCM juga terbagi menjadi *remote activities* dan *local activities*.

Remote activities:

1. **Clone.** mendownload *remote repository* ke dalam komputer sebagai *local repository*.
2. **Fetch.** Sinkronisasi versi teranyar pada remote repository ke local repository.
3. **Pull.** Mengambil semua perubahan dari remote repository ke local repository pada suatu branch.
4. **Push.** Meng-upload commit dari local repository ke remote repository pada suatu branch.

Local activities:

1. **Checkout.** Mengganti working area ke suatu branch.
2. **Stash.** Menyimpan semua perubahan di working directory ke stash area dan membersihkan semua perubahan di working directory.
3. **Apply stash.** Mengambil kembali perubahan yang disimpan di stash area ke working directory.
4. **Stage.** Menyimpan sementara perubahan dari working area untuk kemudian dilakukan commit.
5. **Commit.** Membuat sebuah snapshot atau versi dari perubahan yang telah di-stage.
6. **Branching.** Membuat sebuah pointer untuk menentukan arah penyimpanan snapshot/commit.
7. **Merge.** Mengambil semua perubahan dari sebuah branch (pointer) ke branch lain sehingga pointernya menjadi satu arah.
8. **Tagging.** Memberikan label dari sebuah commit. Biasanya dilakukan untuk menandakan Release point.

Berikut diagram yang menggambarkan komponen SCM beserta operasinya:



Modul Praktikum Kakas Pengembangan Perangkat Lunak

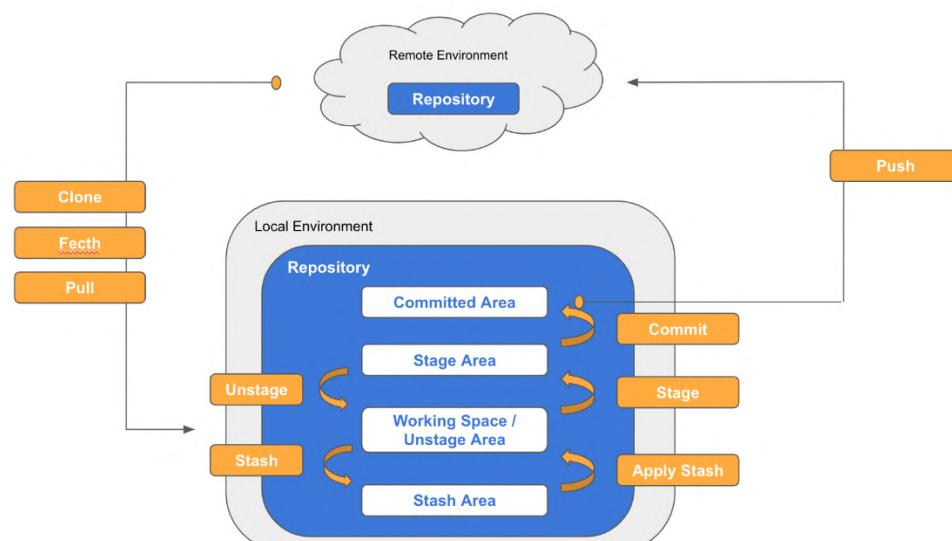


Diagram SCM Operations 1.

Dan berikut diagram yang menggambarkan komponen SCM beserta operasinya yang lebih detail:

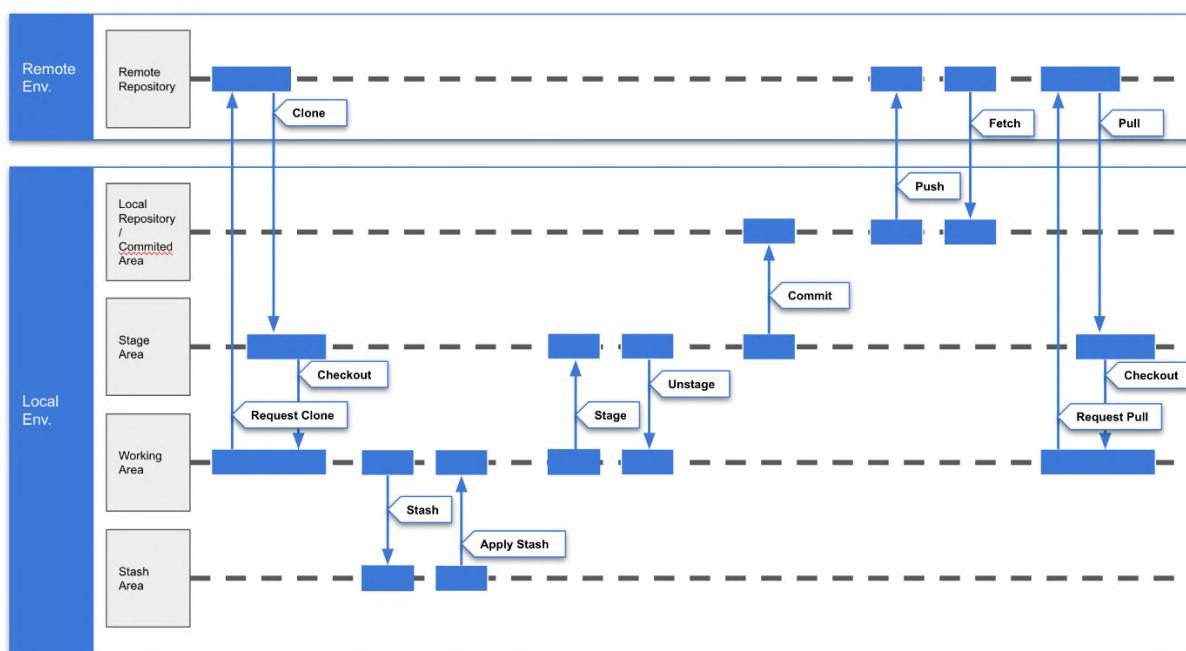


Diagram SCM Operations 2.

Dari semua operasi SCM yang disebutkan, Fetch, Pull, dan Tagging belum pernah kita lakukan. Untuk bisa melakukan Fetch dan Pull, carilah seorang teman untuk berkolaborasi di repository yang Anda miliki. Minta teman Anda untuk melakukan beberapa commit di branch milik Anda. Anda bisa klik tombol Fetch di Sourcetree pada komputer anda, dan melihat label yang muncul pada icon Pull. Selanjutnya klik icon **Pull**, dan Anda bisa melihat sendiri apa yang terjadi. Anda tidak akan pernah membutuhkan tombol Fetch dan Pull jika Anda tidak berkolaborasi dengan developer lain di branch yang sama. Sedangkan Tagging akan kita lakukan di modul berikutnya.



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Baiklah, Anda sudah selesai di materi SCM untuk pemula. Sampai titik ini, kami tidak lagi bisa menyebut Anda sebagai pemula karena Anda kini sudah mahir menerapkan SCM. Anda bebas untuk lanjut ke modul berikutnya untuk belajar materi SCM yang lebih advance atau mencoba untuk lebih membiasakan diri dengan SCM di projek yang sedang Anda garap. Satu hal yang pasti, skill SCM dibangun dan berkembang dari pengalamannya.

Tugas Pendahuluan

[soal-soal TP yang berhubungan dengan materi pertemuan, berbasis masalah rill]

Langkah-Langkah Praktikum

[Uraikan langkah-langkah praktikum/percobaan di sini].

Tugas Laporan

[Format dan deskripsi Tugas laporan praktikum/percobaan]



Lampiran-Lampiran



Modul Praktikum Kakas Pengembangan Perangkat Lunak

Format Tugas Pendahuluan

TUGAS PENDAHULUAN PRAKTIKUM 1

Nama :

Nim :

Soal :

Penyelesaian:

- a. Algoritma Deskripsi (jika ada)

.....
.....
.....

- b. Flowchart (Jika ada)

.....
.....
.....

- c. Program

.....
.....
.....

- d. Penjelasan kode program

.....
.....
.....

Tanda Tangan Asistensi

Asisten





Modul Praktikum Kakas Pengembangan Perangkat Lunak

Format Laporan Praktikum

LAPORAN PRAKTIKUM 1

Nama :

Nim :

Soal :

Penyelesaian:

- Algoritma Deskripsi (jika ada)

- Flowchart (Jika ada)

- Program

- Penjelasan kode program

Tanda Tangan Asistensi

Asisten

