

**Name: Andi Dwi Saputro**

**Date: April 15<sup>th</sup>, 2020**

---

Saya memiliki repository **nusabank.api** di github dan mencoba untuk mempraktekkan bagaimana cara membuat branch, manage codebase dengan **commit, push, pull request** serta **merge** dari **branch repo** ke **master** juga mengkonfigurasi file **.gitignore**. Pada kesempatan kali ini saya menggunakan tools diantaranya GitHub Desktop dan Git CMD/Bash untuk melakukan operasi management repository.

Untuk selengkapnya bisa dicek history commit di repository saya:

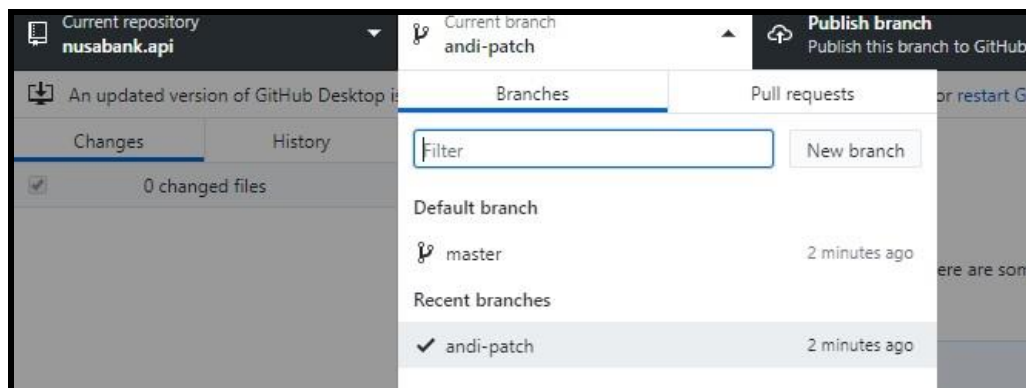
<https://github.com/andiads/nusabank.api>

### 1. Create new Branch from Master repository

Disini saya akan membuat sebuah branch/cabang baru dari master repository **nusabank.api** guna dari branch pada git adalah untuk menghindari adanya konflik atau bentrokan saat melakukan kolaborasi.

Sebagai contoh apabila suatu repository tidak memiliki branch, dan repo tersebut diakses dan dimaintain oleh beberapa orang anggota team yang misal saja pada repository tersebut terdapat sebuah file untuk proses login yang dikerjakan oleh lebih dari satu orang, kemudian kedua orang tersebut melakukan commit dan push ke master, ini akan terjadi konflik/bentrokan karena orang pertama dan lainnya masing-masing akan memiliki algoritma yang berbeda untuk membuat proses login tersebut dari sini leader harus mengapprove salah satu atau memblame salah satu dari commit tersebut.

Tentu hal ini tidak efisien untuk kolaborasi oleh sebab itu dengan adanya Branch maka anggota team yg lain tidak perlu takut untuk merusak master repository karena masing2 memiliki branchnya sendiri dan bebas untuk melakukan perubahan-perubahan tanpa merusak master tersebut.



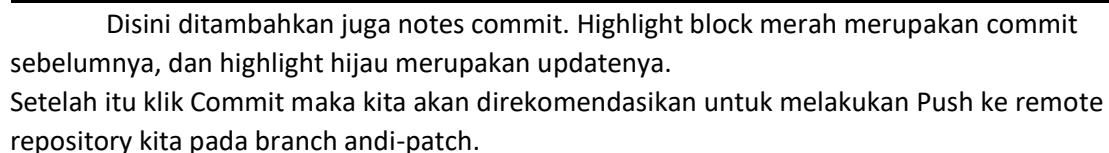
### 2. Commit, Push, Pull Request and Merge from Branch to Master Repo

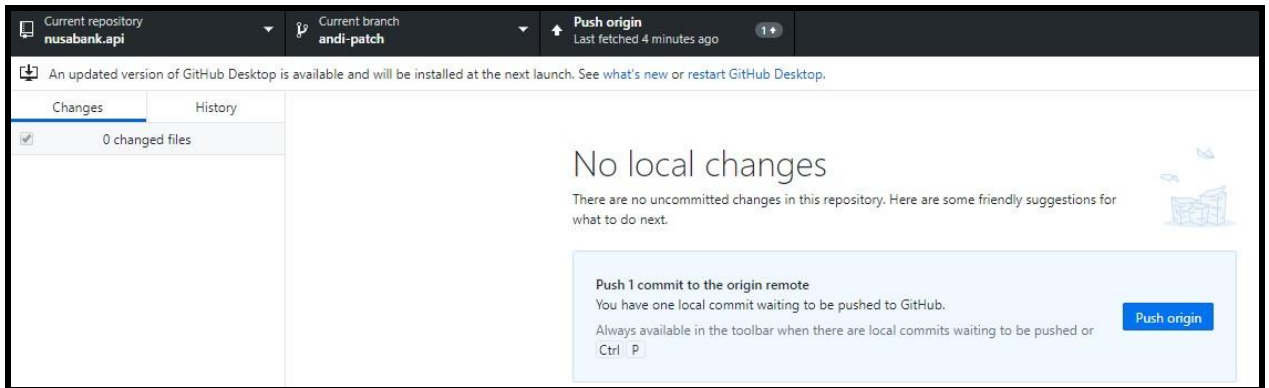
**Commit:** merupakan proses untuk memastikan dimana perubahan-perubahan hanya terjadi pada local repository tidak berdampak pada remote repository yang ada pada GitHub.

**Push:** merupakan proses untuk mengupload segala macam perubahan yang terjadi pada local maupun remote (melalui github website) ke remote repository.

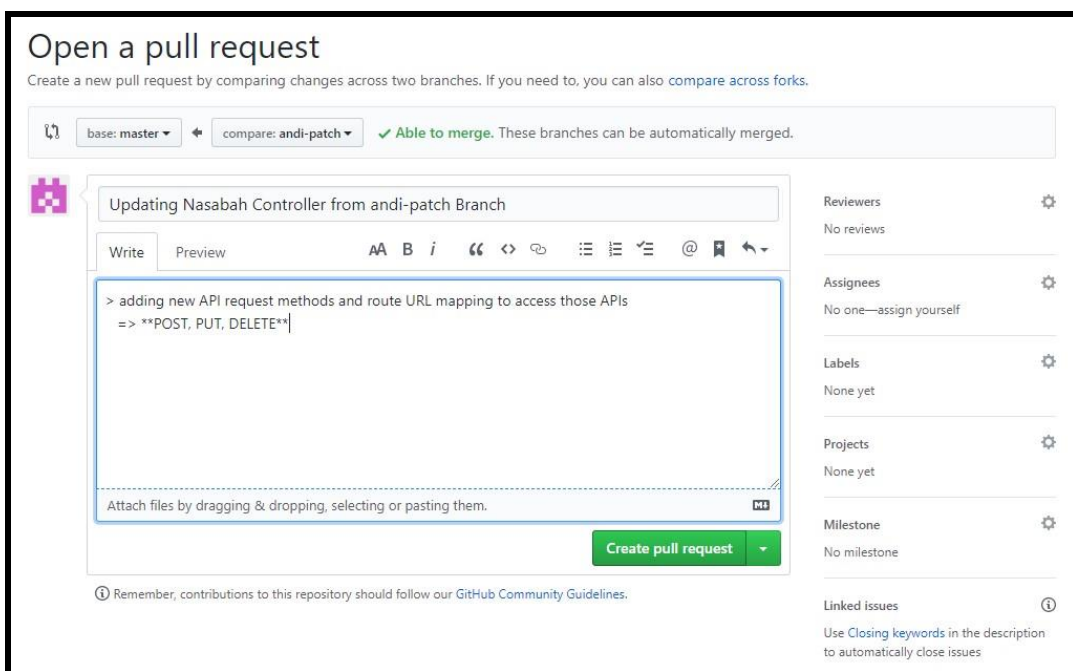
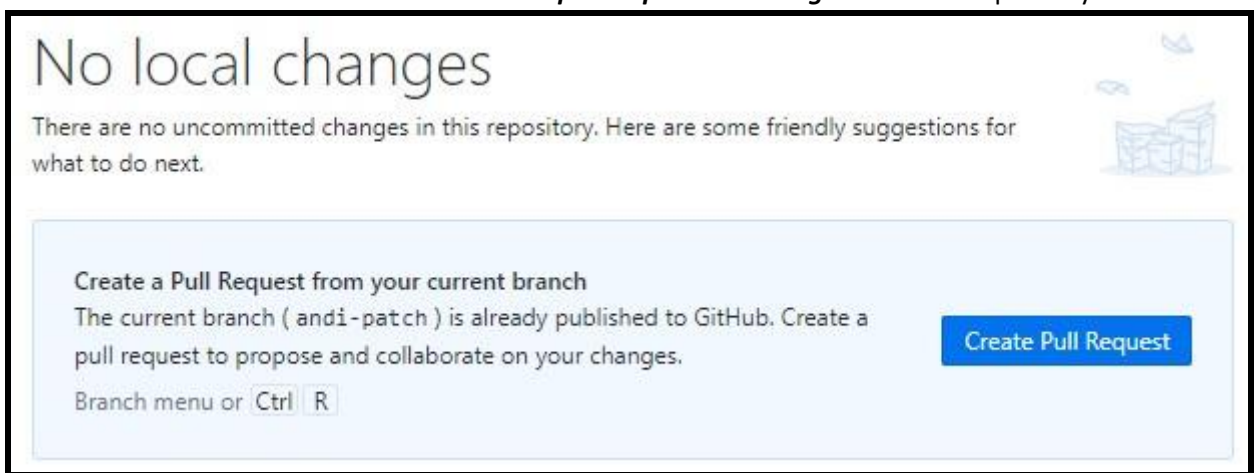
**Merge:** merupakan proses penggabungan perubahan yang terjadi pada suatu source branch ke target branch setelah melakukan pull request.

Misalkan saya ingin mengupdate file NasabahController lalu melakukan commit dan push ke branch andi-patch berikut adalah gambarannya:

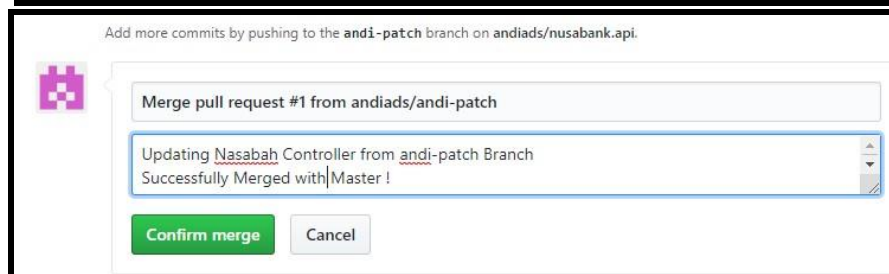
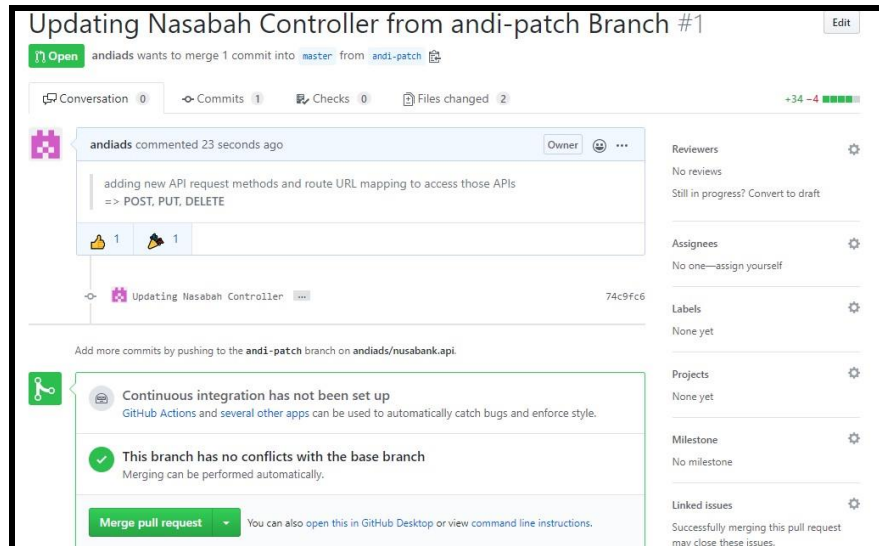




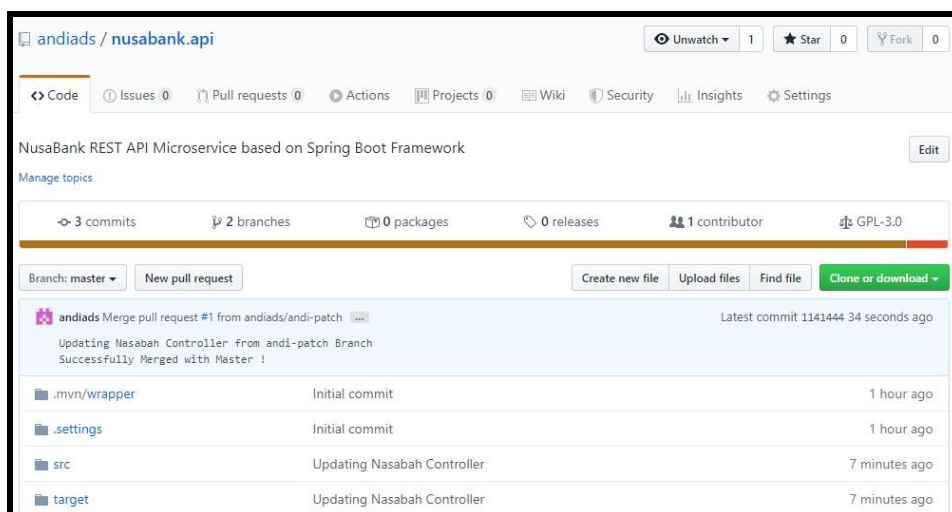
Apabila sudah final dan ingin di publish maka kita lakukan push ke remote. Setelah itu kita akan direkomendasikan untuk melakukan **pull request** dan **merge** ke master repository.



Pada page tersebut apabila tidak ada kendala atau konflik maka statusnya akan hijau dengan tulisan **Able to merge**, apabila terjadi konflik maka harus melakukan compare untuk masing-masing file yang diupdate dari branch dengan master.

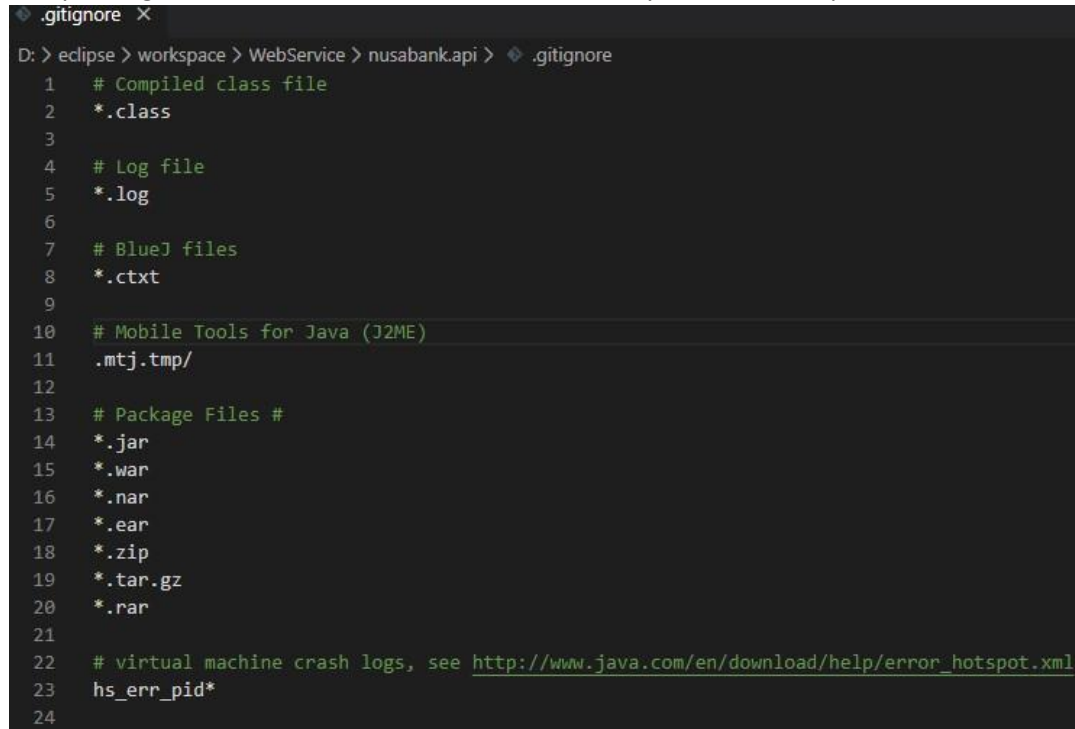


Setelah itu lakukan confirm merge dan perubahan pada branch andi-patch sudah berhasil digabungkan ke codebase yang ada di master.



### 3. Getting Started with File .gitignore

Berikut adalah file .gitignore yang berfungsi untuk meng-exclude/ignore file-file atau folder-folder apa saja yang tidak di track oleh git saat commit/push ke remote repository. File .gitignore dibawah ini merupakan generated saat kita membuat repository baru dengan GitHub Desktop nanti akan ada pilihan template untuk masing-masing jenis project dengan bahasa pemrograman atau framework berbeda, disini saya memilih template Java:



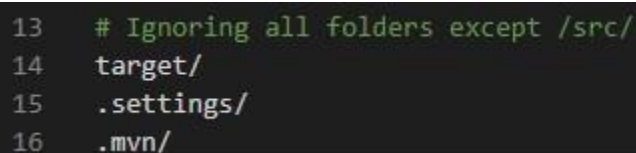
```
.gitignore X
D: > eclipse > workspace > WebService > nusabank.api > .gitignore
1  # Compiled class file
2  *.class
3
4  # Log file
5  *.log
6
7  # BlueJ files
8  *.ctxt
9
10 # Mobile Tools for Java (J2ME)
11 .mtj.tmp/
12
13 # Package Files #
14 *.jar
15 *.war
16 *.nar
17 *.ear
18 *.zip
19 *.tar.gz
20 *.rar
21
22 # virtual machine crash logs, see http://www.java.com/en/download/help/error\_hotspot.xml
23 hs_err_pid*
24
```

Gambar Gitignore File Default Java Template

Seperti yang bisa kita lihat diatas, bahwasanya git akan mengignore segala macam file dengan ekstensi **.log**, **.ctxt**, **.jar**, **.war**, **.zip** dan sebagainya, serta mengexclude directory **.mtj.tmp/** dan keseluruhan sub directory dan files yang ada didalamnya jika ada.

### 4. Updating .gitignore file and refreshing remote repo statuses

Karena gitignore diatas merupakan default dan sudah di push ke repository remote maka bagaimana cara mengupdatenya? Untuk melakukan update gitignore misalnya pada branch **andi-patch** ingin mengignore supaya hanya direktori pada **/src/** dan keseluruhan isinya saja yang akan ditrack oleh git pada remote repository. Untuk melakukan hal tersebut kita hanya perlu menambahkan pattern seperti digambar dibawah ini.



```
13  # Ignoring all folders except /src/
14  target/
15  .settings/
16  .mvn/
```



Tanda “/” diakhir nama folder merupakan pattern untuk ignoring keseluruhan isi baik itu subfolder ataupun files yang ada pada folder/directory tersebut. Jika sudah maka tinggal kita commit dan push ke branch **andi-patch**. Namun ternyata setelah saya push tidak ada perubahan pada remote repository **nusabank.api** di branch **andi-patch**. Karena seharusnya folder-folder selain **/src/** akan tidak muncul. Untuk mengatasi hal ini saya harus melakukan pembersihan atau **rm cache** dengan perintah pada git versi command line:

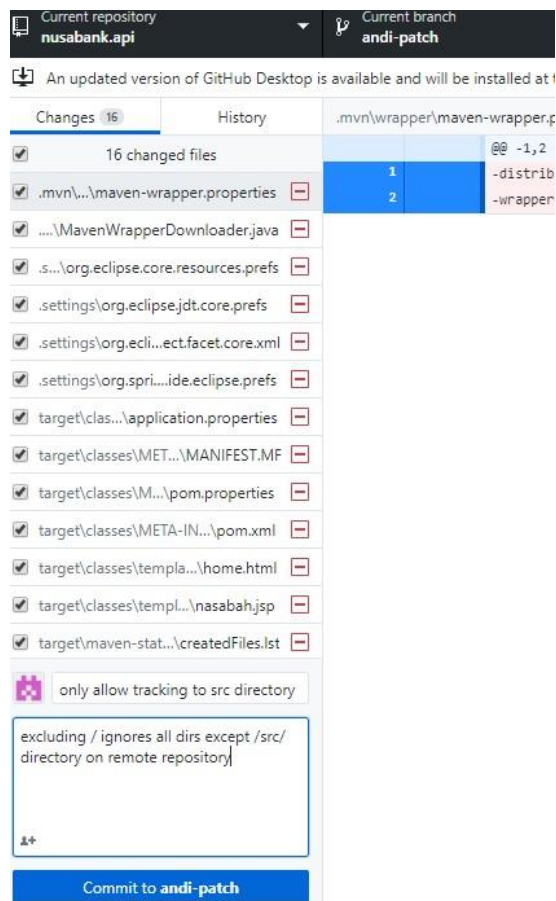
```
$ git rm -r --cache *nama folder tujuan*
```

```
D:\eclipse\workspace\WebService\nusabank.api>git rm -r --cache .mvn/
rm '.mvn\wrapper\MavenWrapperDownloader.java'
rm '.mvn\wrapper\maven-wrapper.properties'

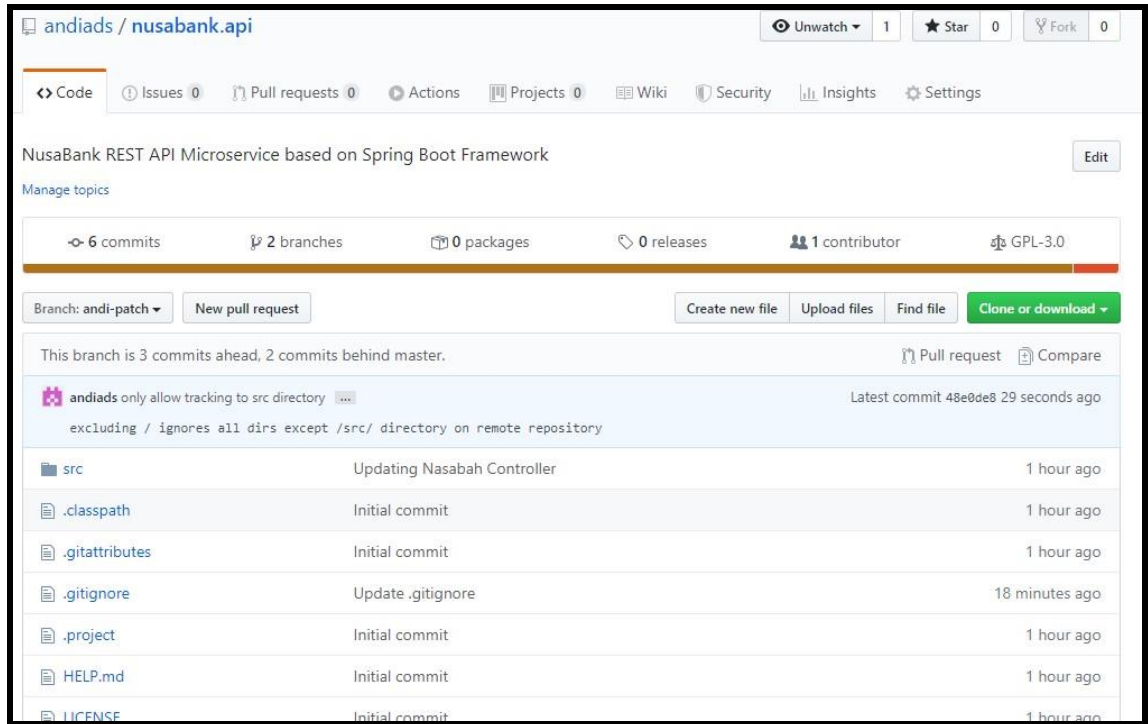
D:\eclipse\workspace\WebService\nusabank.api>git rm -r --cache .settings/
rm '.settings\org.eclipse.core.resources.prefs'
rm '.settings\org.eclipse.jdt.core.prefs'
rm '.settings\org.eclipse.wst.common.project.facet.core.xml'
rm '.settings\org.springframework.ide.eclipse.prefs'

D:\eclipse\workspace\WebService\nusabank.api>git rm -r --cache target/
rm 'target\classes\META-INF\MANIFEST.MF'
rm 'target\classes\META-INF\maven\com.andiads\nusabank.api\pom.properties'
rm 'target\classes\META-INF\maven\com.andiads\nusabank.api\pom.xml'
rm 'target\classes\application.properties'
rm 'target\classes\templates\home.html'
rm 'target\classes\templates\nasabah.jsp'
rm 'target\maven-status\maven-compiler-plugin\compile\default-compile\createdFiles.lst'
rm 'target\maven-status\maven-compiler-plugin\compile\default-compile\inputFiles.lst'
rm 'target\maven-status\maven-compiler-plugin\testCompile\default-testCompile\createdFiles.lst'
rm 'target\maven-status\maven-compiler-plugin\testCompile\default-testCompile\inputFiles.lst'

D:\eclipse\workspace\WebService\nusabank.api>
```



Setelah clear cache selanjutnya kita bisa commit dan push update gitignore kembali ke remote repository pada branch andi-patch. Dan berhasil, pada repository nusabank.api di branch andi-patch sudah tidak muncul lagi folder **.mvn**, **.settings**, dan **target** melainkan hanya menampilkan folder **src** saja.



The screenshot shows the GitHub interface for the repository `andiads / nusabank.api`. The repository description is "NusaBank REST API Microservice based on Spring Boot Framework". The repository statistics show 6 commits, 2 branches, 0 packages, 0 releases, 1 contributor, and GPL-3.0 license. The current branch is `andi-patch`. A message indicates that the branch is 3 commits ahead and 2 commits behind master. A warning message states: "andiads only allow tracking to src directory ... excluding / ignores all dirs except /src/ directory on remote repository". The file list shows the following files and their commit history:

File	Commit Message	Time
<code>src</code>	Updating Nasabah Controller	1 hour ago
<code>.classpath</code>	Initial commit	1 hour ago
<code>.gitattributes</code>	Initial commit	1 hour ago
<code>.gitignore</code>	Update .gitignore	18 minutes ago
<code>.project</code>	Initial commit	1 hour ago
<code>HELP.md</code>	Initial commit	1 hour ago
<code>LICENSE</code>	Initial commit	1 hour ago