

# Dasar-Dasar Pemrograman 2

## Getting Started with a Java Project



FAKULTAS  
ILMU  
KOMPUTER

### Adopted from:

- Getting Started with a Java Project, Tim Asisten Dosen DDP2 2019/2020 Genap
  - Getting Started with a Java Project, Tim Asisten Dosen DDP2 2020/2021 Genap
- 

Setelah berhasil membuat program Java sederhana pada lab, mari kita coba menjalankan sebuah proyek Java.

## Gradle

Terdapat beberapa masalah yang mungkin terjadi saat kita membangun suatu proyek Java yang besar:

1. Biasanya sebuah proyek membutuhkan **dependency** atau **library-library non-standar**. Mungkin saja sebuah proyek memerlukan ratusan *dependency* untuk diinstal oleh orang yang ingin terlibat dalam proyek tersebut. Tentunya hal ini sangat menyulitkan orang tersebut.
2. Setiap ada **perubahan** pada suatu file, maka seluruh bagian dari file yang berubah **harus dikompilasi**. Tidak mengkompilasi seluruh file memungkinkan program tidak berjalan sesuai ekspektasi. Namun, hal ini kurang optimal. Oleh karena itu, diperlukan *tools* yang bisa mengkompilasi file yang berubah saja.
3. Biasanya suatu proyek akan dilengkapi **script**, misalnya untuk menjalankan **test**. *Test* tersebut berguna untuk memastikan bahwa perubahan yang

dilakukan *programmer* benar atau sesuai ekspektasi. Oleh karena itu, kita memerlukan *tools* untuk menjalankan *script* tersebut.

Masalah-masalah yang mungkin terjadi saat pembangunan proyek besar, mendukung diciptakannya sebuah *build tools* di mana salah satunya adalah **Gradle**.

## Git

Mengapa kita memerlukan Git? Misalnya Anda sedang membangun sebuah program. Kemudian Anda terpikirkan untuk menambah sebuah fitur baru. Namun, ternyata kode baru Anda tidak tepat dan mengakibatkan seluruh program tidak bisa dijalankan. Tentunya Anda ingin meng-*undo* perubahan yang telah kamu lakukan tersebut. Dengan adanya Git, Anda bisa mengembalikan kode tersebut ke versi sebelum adanya perubahan. Git merupakan sebuah *Version Control System*. Dengan Git, Anda dapat mengunggah kode Anda ke situs penyedia layanan *hosting* repositori Git, salah satunya GitLab.

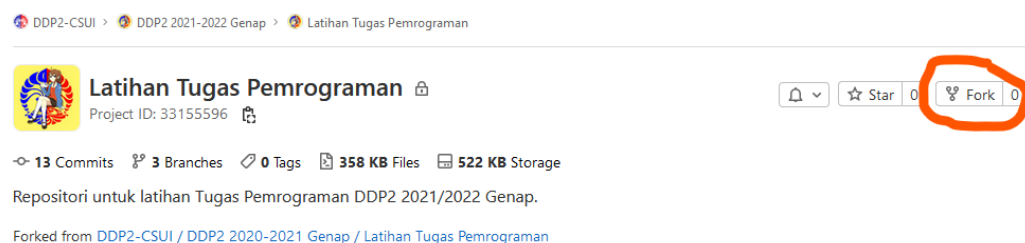
# Proyek Java Pertamaku

## Gitlab sebagai Repository

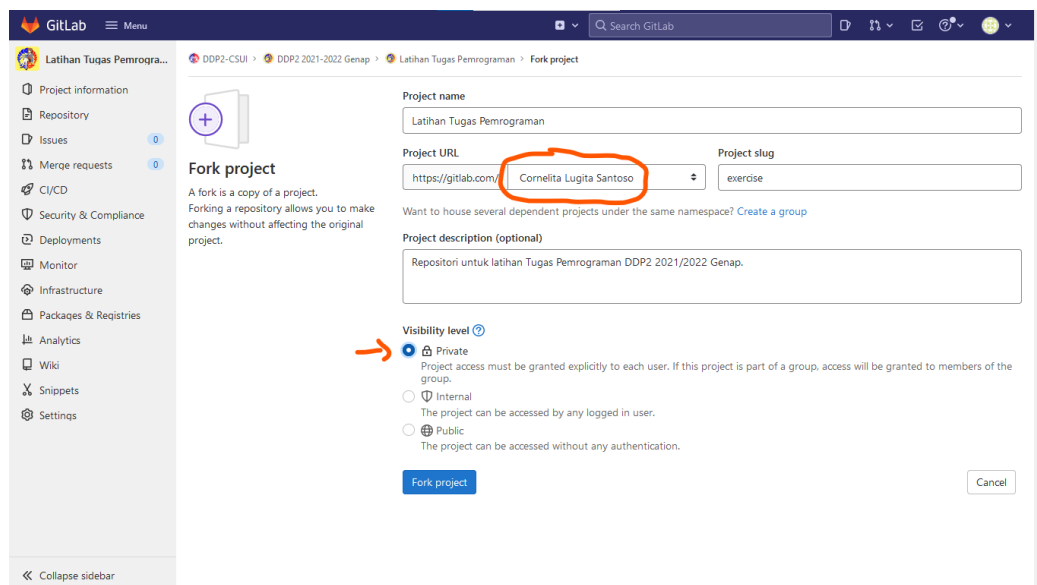
**Notes:** Panduan ini menggunakan Java versi 17 (JDK17)

Pada tutorial ini, Anda akan dipandu untuk menggunakan Git dengan cara mengunduh sebuah proyek Java, memodifikasinya, dan mengunggah perubahan tersebut ke Gitlab. Sebelum memulai, **pastikan Anda telah memiliki akun [GitLab](#)**.

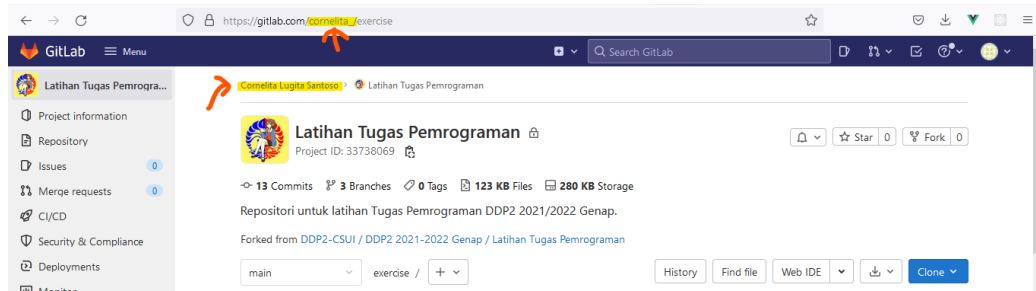
1. **Fork:** Copy proyek milik akun GitLab DDP ke akun GitLab milik Anda dengan cara membuka [proyek exercise milik akun GitLab DDP](#) dan tekan tombol *fork*.



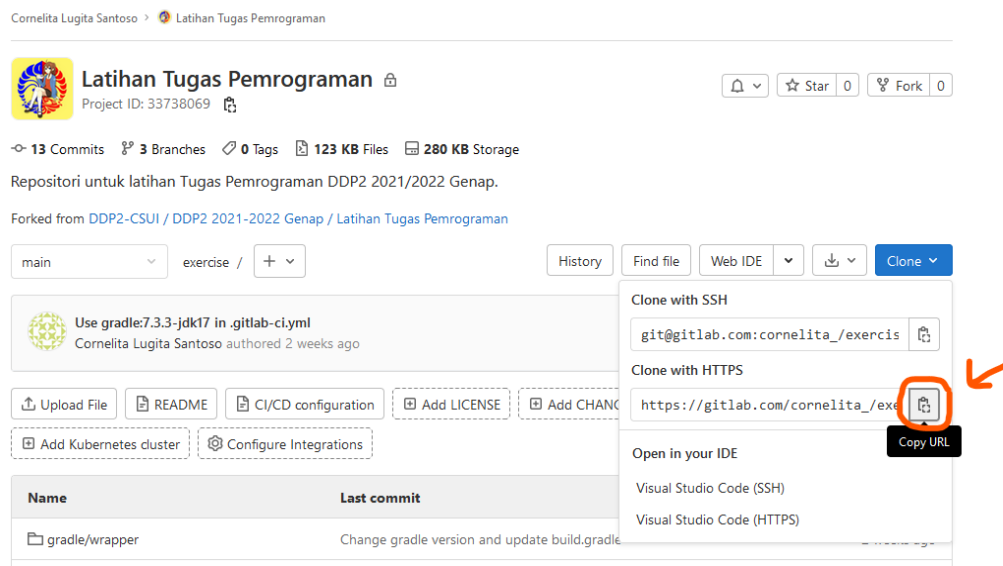
2. Anda akan diarahkan menuju halaman berikut. Pastikan field “Project URL” adalah akun gitlab yang Anda miliki. Pastikan juga anda memilih “Private” pada field “Visibility level” agar tidak semua orang dapat melihat proyek ini.



3. Jika proyek berhasil di-fork, Anda dapat melihat *username* dan nama Anda sebagai *owner*.



4. **Clone:** Unduh proyek tersebut ke komputer Anda dengan menekan tombol **clone** dan salin URL **HTTPS** (SSH memerlukan *setup* khusus).



5. **Clone:** Buka cmd dan masuk ke direktori yang diinginkan untuk menyimpan proyek. Kemudian masukkan perintah: `git clone [HTTPS yang baru saja Anda copy]`

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2>git clone https://gitlab.com/cornelita_/exercise.git
Cloning into 'exercise'...
remote: Enumerating objects: 86, done.
remote: Counting objects: 100% (86/86), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 86 (delta 28), reused 86 (delta 28), pack-reused 0
Receiving objects: 100% (86/86), 67.25 KiB | 1.68 MiB/s, done.
Resolving deltas: 100% (28/28), done.
E:\Lita\1. Kuliah\6. Semester 6\DDP 2>
```

6. Buka proyek tersebut dengan *text editor* atau IDE favoritmu.
7. Jalankan fungsi main yang berada pada file `src/main/java/exercise/Solver.java`. Kalian dapat menggunakan *executable* Gradle Wrapper yaitu `gradlew.bat` yang berada pada proyek tersebut. Cukup jalankan `gradlew.bat run` pada cmd untuk menjalankan fungsi main tersebut.

**Note:** Jika kalian menggunakan macOS atau Linux, ganti seluruh perintah `gradlew.bat` pada dokumen ini menggunakan `./gradlew`, misalnya `./gradlew run`

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>gradlew.bat run
Starting a Gradle Daemon, 1 incompatible Daemon could not be reused, use --status for details

> Task :run
true
false

BUILD SUCCESSFUL in 18s
2 actionable tasks: 2 executed
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>
```

8. Jalankan *test* yang berada pada `src/test/java/exercise/SolverTest.java` dengan cara menjalankan perintah `gradlew.bat test` pada cmd. Test membantu *programmer* untuk memastikan bahwa kode yang dituliskan sesuai ekspektasi walaupun ada perubahan.

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>gradlew.bat test

> Task :test

SolverTest > testTwoSumNotConsiderSameNumber() FAILED
org.opentest4j.AssertionFailedError at SolverTest.java:25

5 tests completed, 1 failed

> Task :test FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':test'.
> There were failing tests. See the report at: file:///E:/Lita/1.%20Kuliah/6.%20Semester%206/DDP%202/exercise/build/reports/tests/test/index.html

* Try:
> Run with --stacktrace option to get the stack trace.
> Run with --info or --debug option to get more log output.
> Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 5s
3 actionable tasks: 2 executed, 1 up-to-date
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>
```

9. Rupanya *test* kita gagal. Terlihat bahwa

`testTwoSumNotConsiderSameNumber` **FAILED**. Test tersebut rupanya sebagai berikut:

```
@Test
public void testTwoSumNotConsiderSameNumber() {
    assertFalse(new Solver().isExistTwoSum(new int[] {2}, 4), "Two sum should not consider the same number twice");
}
```

Angka 2 yang sama tidak boleh dipilih dua kali untuk menghasilkan 4. Anda langsung sadar bahwa kode pada fungsi `isExistTwoSum` belum tepat sehingga Anda memperbaikinya menjadi:

```

public boolean isExistTwoSum(int[] nums, int K) {
    for (int i = 0; i < nums.length; i++) {
        for (int j = 0; j < nums.length; j++) {
            if (i != j && nums[i] + nums[j] == K) return true;
        }
    }
    return false;
}

```

10. Simpan perubahan pada *file* dan jalankan test lagi.

```

E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>gradlew.bat test
Starting a Gradle Daemon, 1 incompatible and 4 stopped Daemons could not be reused, use --status for details

BUILD SUCCESSFUL in 16s
3 actionable tasks: 3 executed
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>

```

11. Kini kode Anda telah sesuai ekspektasi karena sukses melalui seluruh test.

12. Simpan perubahan tersebut pada Git dengan git. Pertama jalankan perintah

`git status` untuk mengetahui *file* apa saja yang berubah.

```

E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/main/java/exercise/Solver.java

no changes added to commit (use "git add" and/or "git commit -a")
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>

```

13. Untuk melihat perubahannya, jalankan perintah `git diff`. Untuk keluar dari

`git diff`, tekan `q` (jika diperlukan).

```

E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git diff
diff --git a/src/main/java/exercise/Solver.java b/src/main/java/exercise/Solver.java
index 748b55b..ea65a7d 100644
--- a/src/main/java/exercise/Solver.java
+++ b/src/main/java/exercise/Solver.java
@@ -10,7 +10,7 @@ public class Solver {
    public boolean isExistTwoSum(int[] nums, int K) {
        for (int i = 0; i < nums.length; i++) {
            for (int j = 0; j < nums.length; j++) {
-               if (nums[i] + nums[j] == K) return true;
+               if (i != j && nums[i] + nums[j] == K) return true;
            }
        }
        return false;
    }
}
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>

```

14. Tentukan file mana saja yang ingin disimpan perubahannya. Dalam kasus ini, hanya file `src/main/java/exercise/Solver.java` yang berubah. Oleh karena itu, kita akan menambahkan file tersebut ke staging dengan menjalankan perintah:

`git add src/main/java/exercise/Solver.java`. Pastikan perubahan tersebut sudah ada di *staging* dan siap untuk di-*commit* dengan menjalankan perintah: `git status`

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git add src/main/java/exercise/Solver.java
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/main/java/exercise/Solver.java

E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>
```

**Note:**

Ada beberapa karakter khusus yang dapat dimanfaatkan untuk mempersingkat perintah `git add`, misalnya:

- `git add *` : menambahkan semua *file* ke *staging*
- `git add .` : menambahkan semua *file* pada direktori saat ini beserta subdirektori ke *staging*
- `git add -u` : menambahkan hanya *file* yang berubah (bukan file baru) ke *staging*

15. Atur *username* dan *email* git di komputer Anda untuk menandai bahwa *commit* tersebut dibuat oleh Anda dengan menjalankan perintah berikut:

```
git config user.name "Nama Kamu"
```

```
git config user.email "emailkamu@server.tld"
```

**Note:** Tambahkan parameter `--global` jika Anda ingin mengatur konfigurasi secara global, tidak hanya untuk repositori tersebut.

```
git config --global user.name "Nama Kamu"
```

```
git config --global user.email "emailkamu@server.tld"
```

Pastikan *email* yang digunakan sesuai dengan akun GitLab Anda.

Contoh:

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git config user.name "Cornelita Lugita Santoso"
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git config user.email "lugitasantoso127@gmail.com"
```

16. Simpan perubahan dengan menjalankan `git commit -m "<perubahan yang kamu lakukan>"`. Contoh:

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git commit -m "Fix bug on selecting number twice"
[main c4966e8] Fix bug on selecting number twice
 1 file changed, 1 insertion(+), 1 deletion(-)

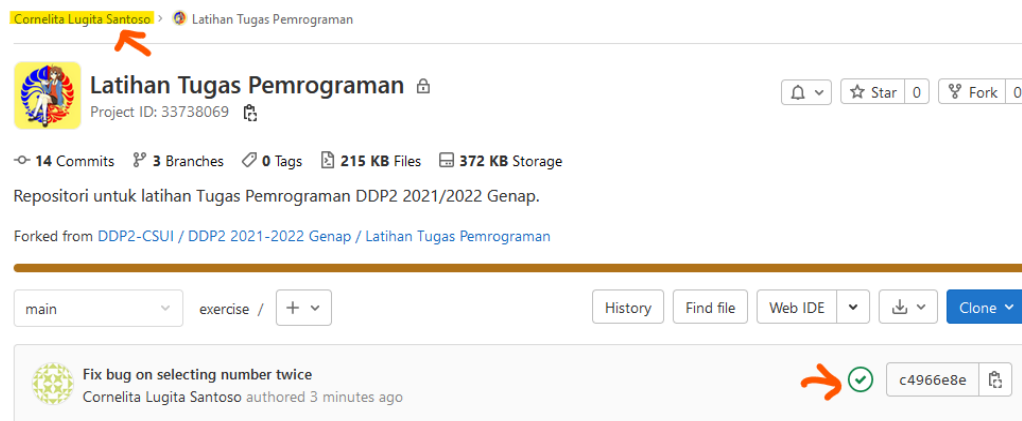
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>
```

17. Unggah perubahan ke repositori *online* Anda yang berada di GitLab dengan menjalankan perintah: `git push origin main`. Jika diminta *username/email* dan *password*, gunakan detail akun GitLab Anda.

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 535 bytes | 178.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/cornelita_/exercise.git
7343720..c4966e8 main -> main

E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>
```

18. Setelah pembaruan diunggah, [GitLab CI pipeline](#) akan menjalankan *build* termasuk *test*. Jika berhasil seperti di komputer Anda, maka akan ada centang hijau pada repositori GitLab milik Anda (**bukan DDP2-CSUI**).



19. Jika ada soal baru atau revisi soal, Anda bisa mendapat pembaruan dengan menambahkan repositori GitLab milik DDP2-CSUI sebagai *upstream*. Jalankan perintah `git remote add upstream [HTTPS yang dicopy dari Repositori DDP2-CSUI]` dan pastikan *upstream* berhasil ditambahkan dengan menjalankan perintah: `git remote -v`.



DDP2-CSUI > DDP2 2021-2022 Genap > Latihan Tugas Pemrograman

**Latihan Tugas Pemrograman**

Project ID: 33155596

🔔 14 Commits 3 Branches 0 Tags 451 KB Files 621 KB Storage

Repositori untuk latihan Tugas Pemrograman DDP2 2021/2022 Genap.

Forked from DDP2-CSUI / DDP2 2020-2021 Genap / Latihan Tugas Pemrograman

main exercise / +

History Find file Web IDE Clone

Fix indentation  
Cornelita Lugita Santoso authored 24 minutes ago

Upload File README CI/CD configuration Add LICENSE Add CHANGELOG

Add Kubernetes cluster Configure Integrations

Name	Last commit
gradle/wrapper	Change gradle version and update build.gradle

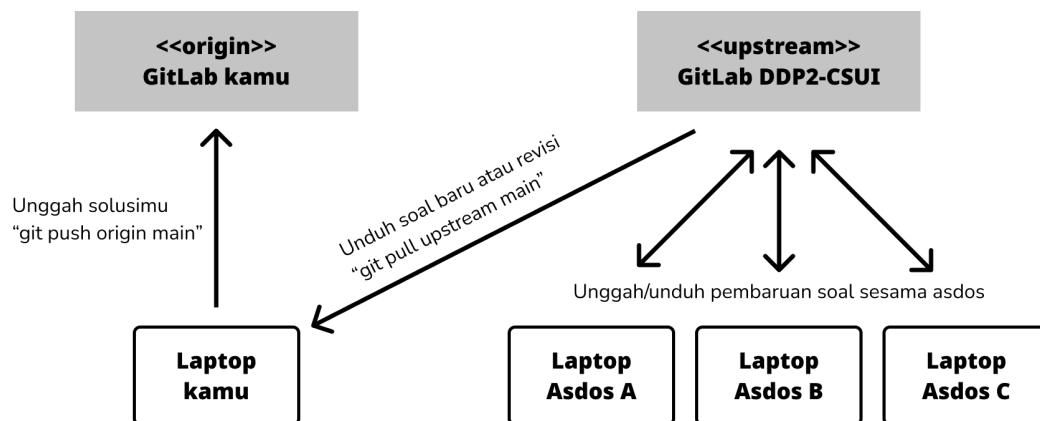
Clone with SSH  
git@gitlab.com:DDP2-CSUI/2021-2022

Clone with HTTPS  
https://gitlab.com/DDP2-CSUI/2021-2022

Open in your IDE  
Visual Studio Code (SSH)  
Visual Studio Code (HTTPS)

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git remote add upstream https://gitlab.com/DDP2-CSUI/2021-2022-genap/exercise.git
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git remote -v
origin https://gitlab.com/cornelita_/exercise.git (fetch)
origin https://gitlab.com/cornelita_/exercise.git (push)
upstream https://gitlab.com/DDP2-CSUI/2021-2022-genap/exercise.git (fetch)
upstream https://gitlab.com/DDP2-CSUI/2021-2022-genap/exercise.git (push)
```

20. Nantinya, Anda bisa mendapat pembaruan soal atau revisi dengan menjalankan perintah `git pull upstream main`. Berikut ini alur yang akan terjadi pada Tugas Pemrograman Anda:



## Git sebagai Version Control System

1. Jalankan perintah `git log`.

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git log
commit c4966e8ecd8101b083f842621c48b54f3b2df84f (HEAD -> main, origin/main, origin/HEAD)
Author: Cornelita Lugita Santoso <lugitasantoso127@gmail.com>
Date: Tue Feb 15 17:26:46 2022 +0700

    Fix bug on selecting number twice

commit 7343720e3fc71b1b2267bd6d6850d2e5dd0c427c (upstream/main)
Author: cornelita_ <lugitasantoso127@gmail.com>
Date: Tue Feb 15 17:08:17 2022 +0700

    Fix indentation
```

2. Pilih versi yang Anda inginkan dengan menyalin *commit hash* (cukup beberapa karakter pertama)

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git log
commit c4966e8ecd8101b083f842621c48b54f3b2df84f (HEAD -> main, origin/main, origin/HEAD)
Author: Cornelita Lugita Santoso <lugitasantoso127@gmail.com>
Date: Tue Feb 15 17:26:46 2022 +0700

    Fix bug on selecting number twice

commit 7343720e3fc71b1b2267bd6d6850d2e5dd0c427c (upstream/main)
Author: cornelita_ <lugitasantoso127@gmail.com>
Date: Tue Feb 15 17:08:17 2022 +0700

    Fix indentation
```

3. Keluar dari `git log` dengan menekan karakter `q` jika diperlukan
4. Jalankan `git checkout <commit_hash>` ke versi tersebut.

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git checkout 7343720e3fc7
Note: switching to '7343720e3fc7'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 7343720 Fix indentation
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>
```

5. Lihat kembali file `src/main/java/exercise/Solver.java`. Perubahan yang telah Anda lakukan sebelumnya kini tidak ada.

```
public boolean isExistTwoSum(int[] nums, int K) {
    for (int i = 0; i < nums.length; i++) {
        for (int j = 0; j < nums.length; j++) {
            if (nums[i] + nums[j] == K) return true;
        }
    }
    return false;
}
```

6. Untuk kembali ke versi terbaru, jalankan perintah `git checkout main`.

```
E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>git checkout main
Previous HEAD position was 7343720 Fix indentation
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

E:\Lita\1. Kuliah\6. Semester 6\DDP 2\exercise>
```

7. Kini *bug fix* yang telah Anda kerjakan kembali ada.

```
public boolean isExistTwoSum(int[] nums, int K) {
    for (int i = 0; i < nums.length; i++) {
        for (int j = 0; j < nums.length; j++) {
            if (i != j && nums[i] + nums[j] == K) return true;
        }
    }
    return false;
}
```

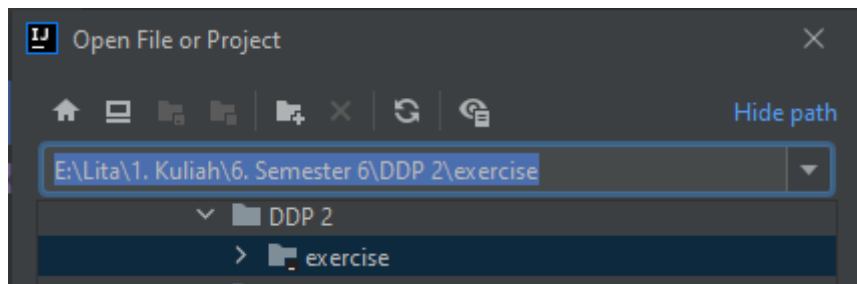
Sekarang, Anda telah dapat menggunakan *tools* yang dimanfaatkan pada tugas pemrograman! Sebenarnya, masih ada banyak kegunaan *tools* tersebut namun tidak dibahas karena tidak diperlukan dalam pengerjaan tugas pemrograman. Misalnya, `git branch` yang dimanfaatkan untuk kolaborasi sesama *developer* namun tidak diperlukan karena tugas pemrograman bersifat individu. Meski demikian, Anda disarankan mengeksplorasi sendiri kegunaan lain *tools* tersebut yang akan berguna dalam mata kuliah ataupun kegiatan lain seperti magang, proyek, dan pekerjaan di masa depanmu. Selamat mengeksplorasi!

## Additional Guide for IntelliJ

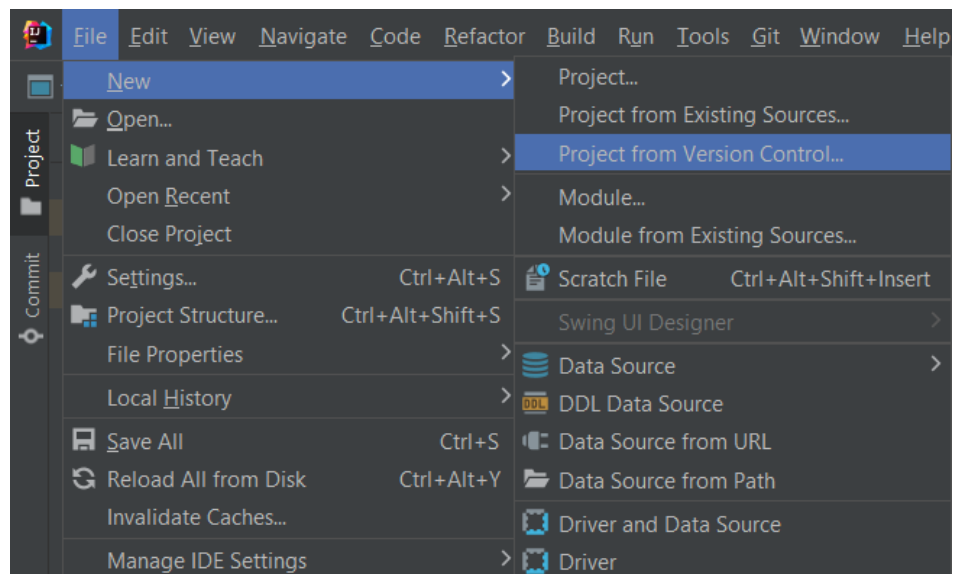
Untuk menginstall IntelliJ IDEA Ultimate, kalian dapat mengikuti panduan berikut

### Getting Started with IntelliJ IDEA Ultimate

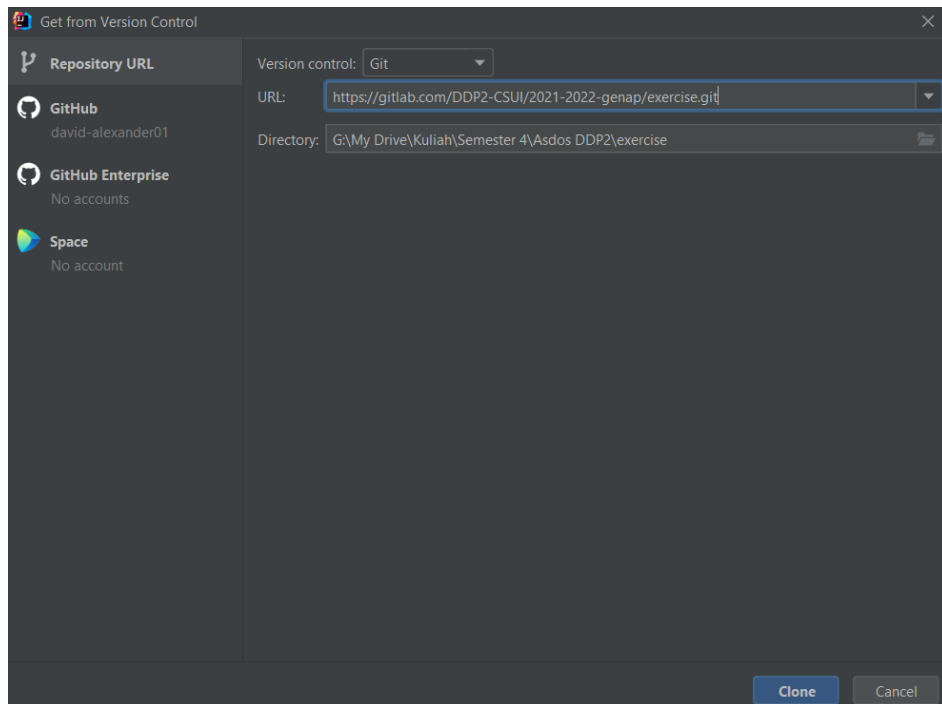
1. Setelah melakukan *fork* pada repository Latihan Tugas Pemrograman, kamu bisa meng-*import* project tersebut ke IntelliJ. Untuk melakukannya, kamu dapat membuka IntelliJ lalu pilih “Open”. Kemudian pilih project yang ingin di-*import*.



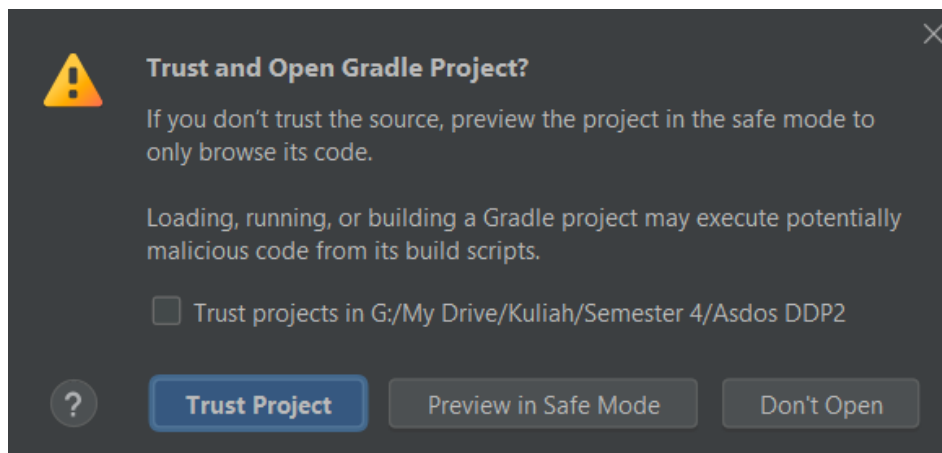
- a. *Alternatively*, IntelliJ memiliki opsi untuk melakukan *clone*, membuat, dan melakukan *build* project langsung dari suatu remote Git repository. Caranya adalah pilih “File -> New -> Project from Version Control...”



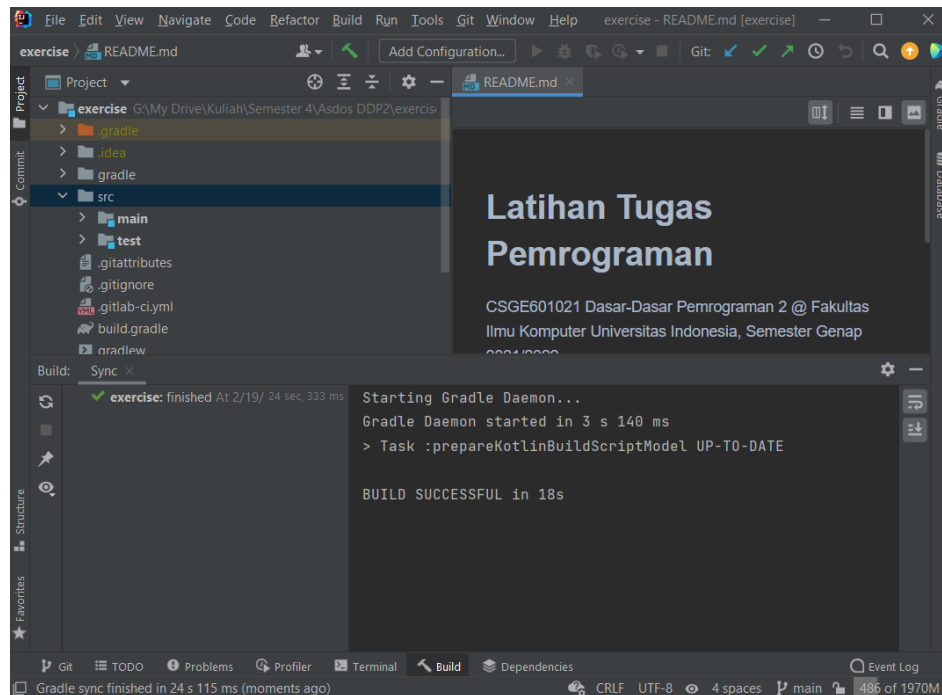
- b. Masukkan URL yang didapatkan dari “Clone from HTTPS”, lalu tekan “Clone”.



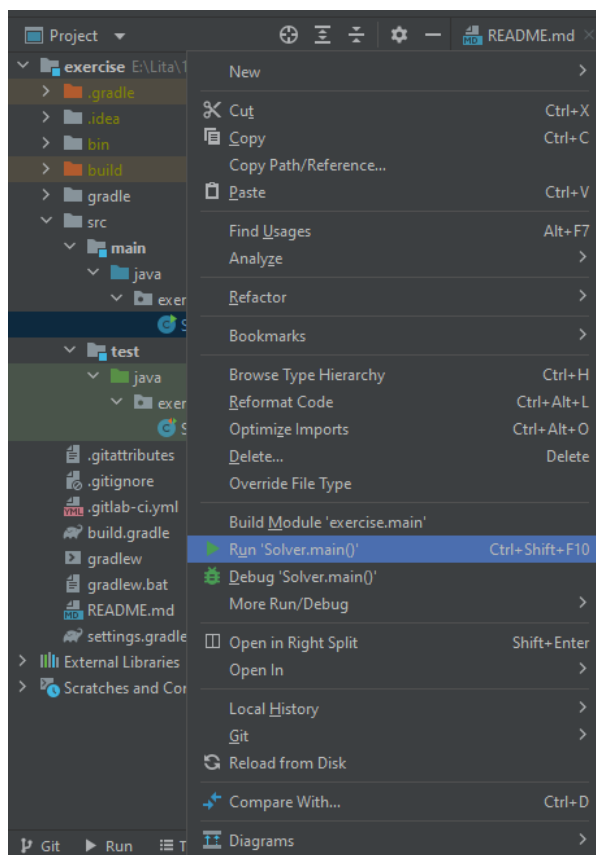
- c. Pilih “Trust Project” (aman koq)



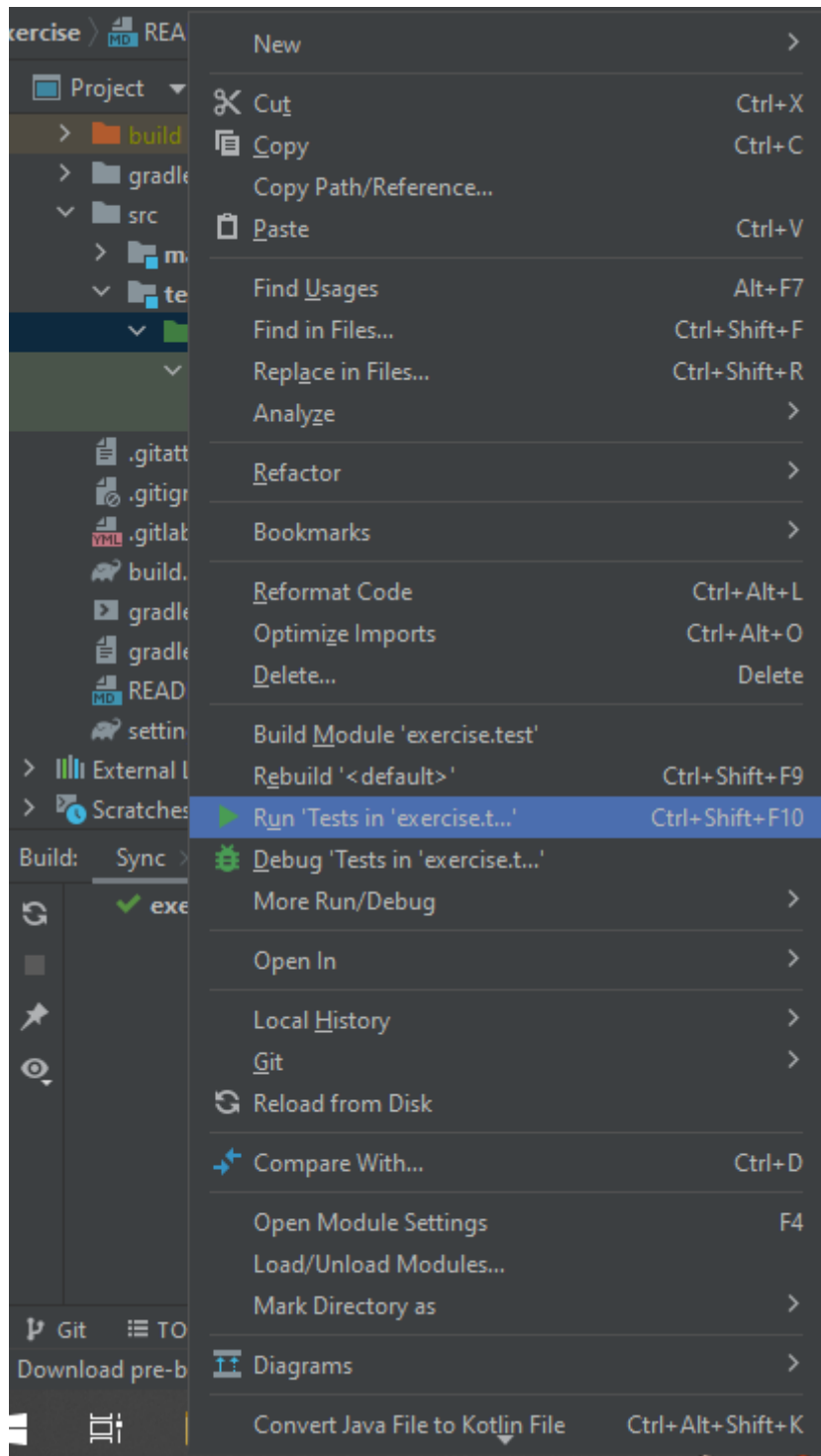
d. IntelliJ akan melakukan build Gradle project secara otomatis



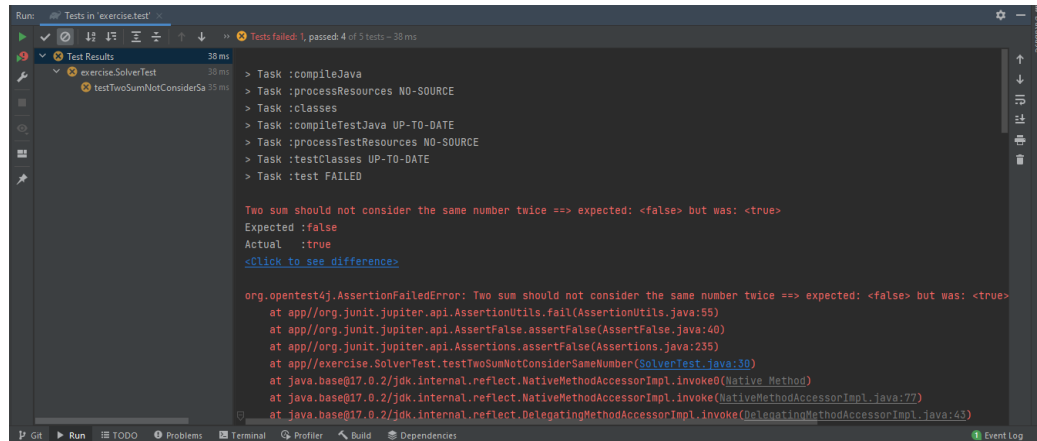
2. Untuk menjalankan programnya, klik kanan pada **Solver.java** dan pilih **Run 'Solver.main()'**



3. Untuk menjalankan test, klik kanan pada folder **test/java** dan pilih **Run 'Tests in 'exercise.test''**



Akan tampil seperti gambar dibawah ini (untuk kode yang belum direvisi)



The screenshot shows an IDE's 'Run' window with a test failure. The test is 'testTwoSumNotConsiderSa' in the 'exercise.SolverTest' class. The failure message is 'Two sum should not consider the same number twice ==> expected: <false> but was: <true>'. The stack trace shows the error originates from 'org.opentest4j.AssertionFailedError' and 'app/org.junit.jupiter.api.Assertions.fail'.

```
Run: Tests in 'exercise.test'
Tests failed: 1, passed: 4 of 5 tests - 38 ms

Test Results: 38 ms
  exercise.SolverTest: 38 ms
    testTwoSumNotConsiderSa: 35 ms

> Task :compileJava
> Task :processResources NO-SOURCE
> Task :classes
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test FAILED

Two sum should not consider the same number twice ==> expected: <false> but was: <true>
Expected :false
Actual   :true
<Click to see difference>

org.opentest4j.AssertionFailedError: Two sum should not consider the same number twice ==> expected: <false> but was: <true>
    at app/org.junit.jupiter.api.AssertionsUtils.fail(AssertionUtils.java:55)
    at app/org.junit.jupiter.api.AssertionsUtils.fail(AssertionUtils.java:40)
    at app/org.junit.jupiter.api.Assertions.fail(Assertions.java:235)
    at app/exercise.SolverTest.testTwoSumNotConsiderSameNumber(SolverTest.java:30)
    at java.base@17.0.2/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at java.base@17.0.2/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
    at java.base@17.0.2/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
```

Feel free to use, reuse, and share this work: the more we share, the more we have!

