Python DTS PRoA 2022

Library Session: Miscellaneous (Telegram BOT)

Muhammad Ogin Hasanuddin

KK Teknik Komputer
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Pendahuluan

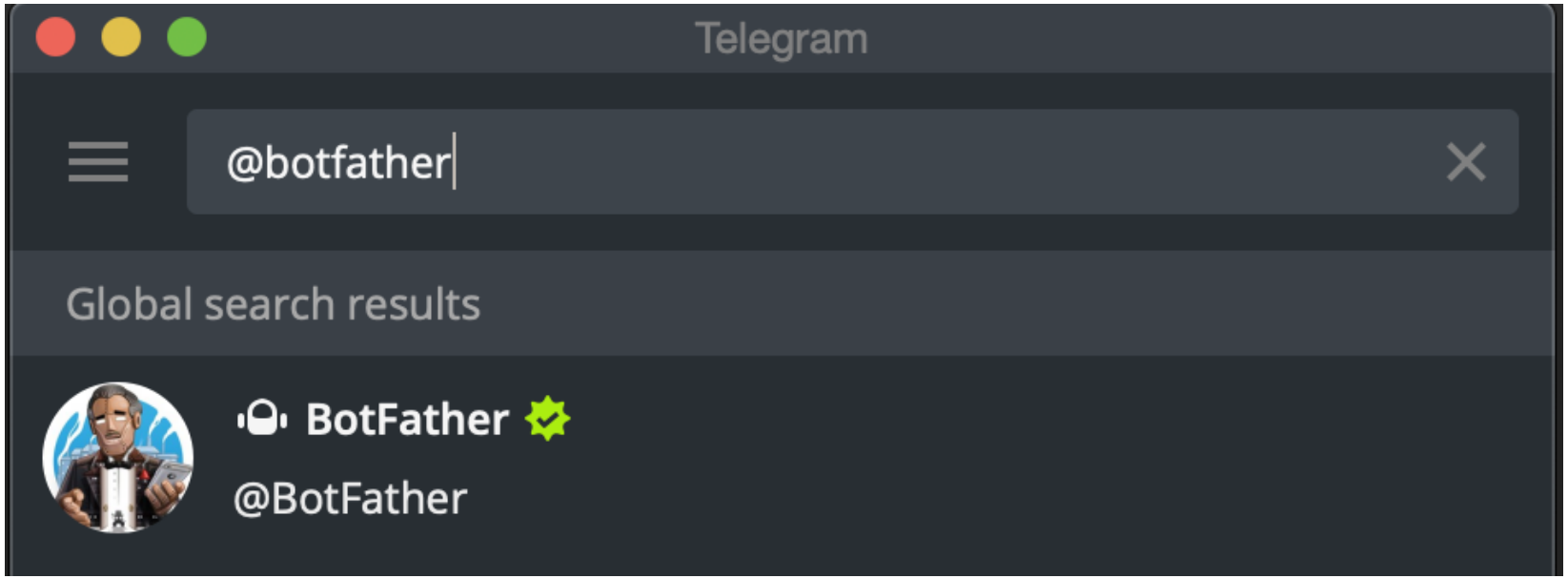
- Bot adalah kependekan dari “robot” yang dapat diartikan sebagai sistem untuk melakukan tugas tertentu secara otomatis dan repetitif. Karena bot beroperasi secara otomatis, mereka dapat bekerja jauh lebih cepat dibandingkan manusia. Selain itu, bot dapat berjalan setiap waktu tanpa harus menunggu manusia mengoperasikannya secara manual.
- Ada beberapa bot yang dapat membantu pekerjaan manusia, contohnya seperti web crawler dan chatbot. Sementara itu, ada juga bot yang buruk dan digunakan untuk tindak kejahatan, contohnya seperti malicious bot.
- Bot telegram adalah aplikasi pihak ketiga yang berjalan di dalam Telegram. Pengguna dapat berinteraksi dengan bot dengan mengirimi mereka pesan, perintah, dan permintaan sebaris. Anda mengontrol bot Anda menggunakan permintaan HTTPS ke Bot API Telegram.

How do I create a bot?

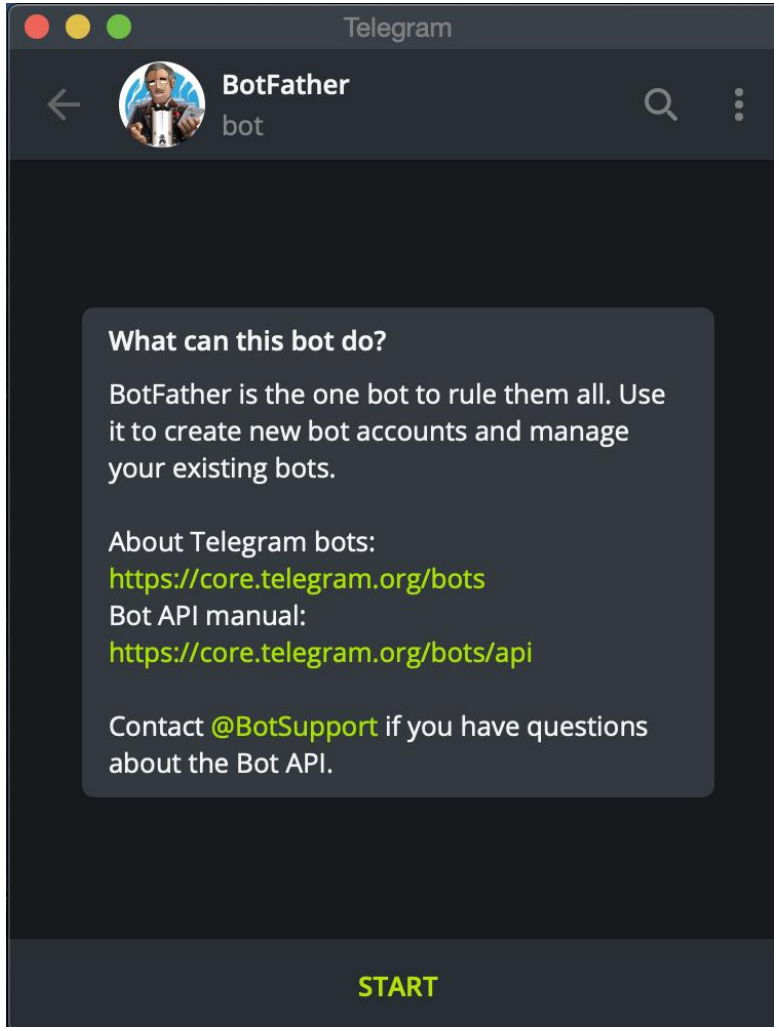
- BotFather is the one bot to rule them all. It will help you create new bots and change settings for existing ones.

Set up your Bot's profile

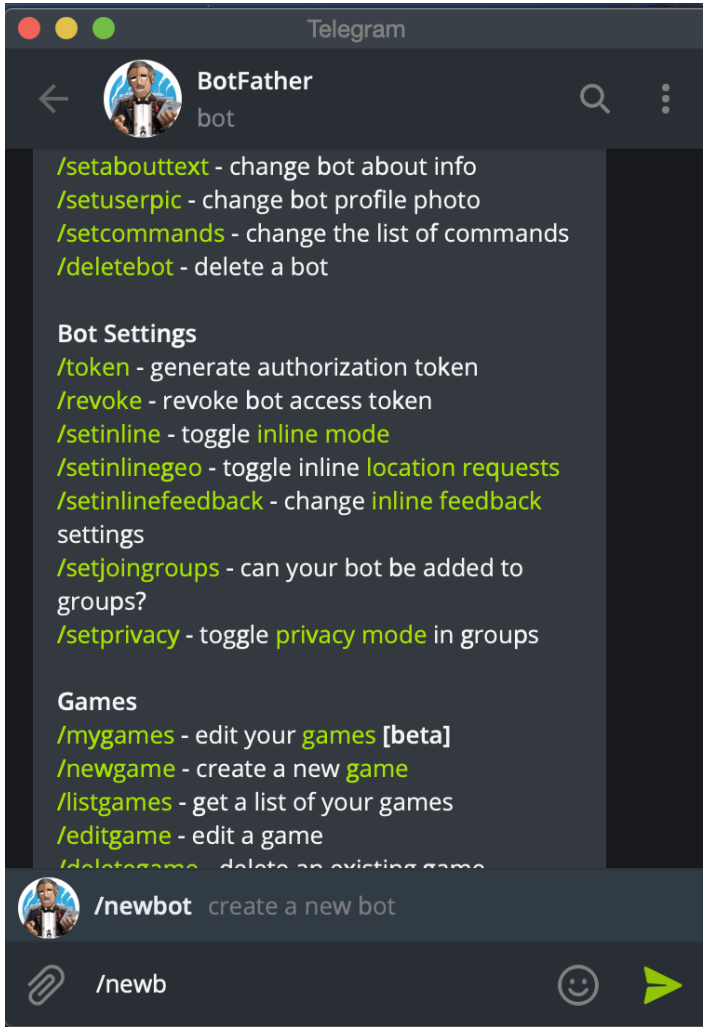
- To set up a new bot, start the conversation with BotFather (@BotFather). BotFather will help us in creating the new bot.
- Search for @botfather in Telegram.



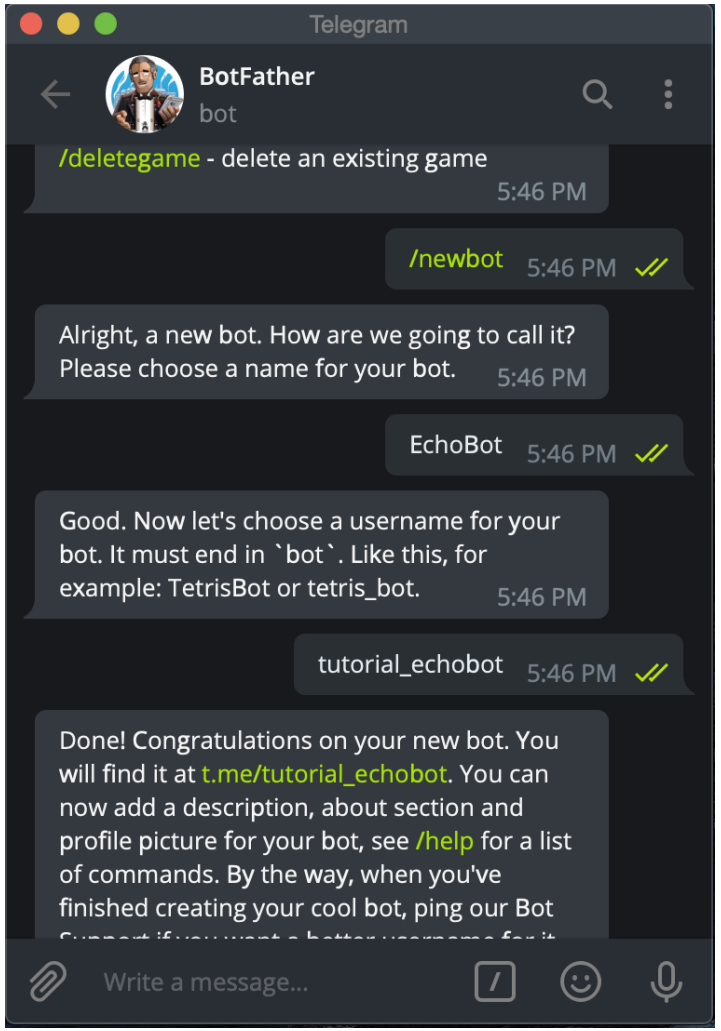
Start your conversation by pressing the Start button.



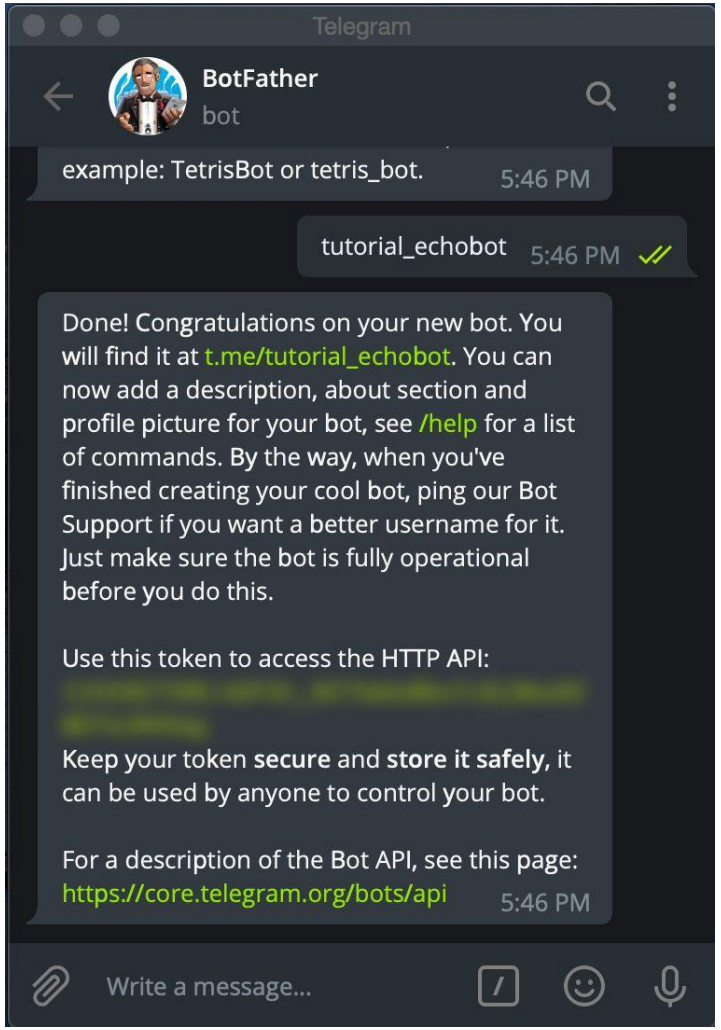
Create the bot by running /newbot command



Enter the Display Name and User Name for the bot.



BotFather will send you a message with the token



Keep access token of the bot securely.
Anyone with your token can manipulate this bot.

Coding the bot

- Open up the terminal and start by creating a new directory first.

```
$ mkdir note-bot
$ cd note-bot/
```

Setup virtual environment

- To create a virtual environment, go to your project's directory and run venv. If you are using Python 2, replace venv with virtualenv in the below commands.

```
$ py -m venv note-bot-venv
```

- The second argument is the location to create the virtual environment. Generally, you can just create this in your project and call it env.
- venv will create a virtual Python installation in the `note-bot-venv` folder.

Note: You should exclude your virtual environment directory from your version control system using .gitignore or similar.

Activating a virtual environment

- Before you can start installing or using packages in your virtual environment you'll need to activate it. Activating a virtual environment will put the virtual environment-specific python and pip executables into your shell's PATH.

```
$ .\note-bot-venv\Scripts\activate
```

Install the package using the following command.

```
$ python -m pip install python-telegram-bot
```

Freezing dependencies

- Pip can export a list of all installed packages and their versions using the freeze command:

```
$ python -m pip freeze
```

- Which will output a list of package specifiers such as:

```
APScheduler==3.6.3
cachetools==4.2.2
certifi==2021.10.8
python-telegram-bot==13.11
pytz==2022.1
pytz-deprecation-shim==0.1.0.post0
six==1.16.0
tornado==6.1
tzdata==2022.1
tzlocal==4.2
```

Create a new file bot.py and paste the following code in it.

- https://github.com/muhammadoginh/library-session-2022.git, cek library session 5

Run the bot

```
$ python bot.py
```

Deploying Telegram Bot on Heroku

Setting up Heroku

- First things first, you will need to create an account on Heroku.
- Install heroku-cli for your specific operating system.
- Login in to your account by running the following command in terminal

```
$ heroku login -i
```

Code Changes

- Sebelumnya kita menggunakan polling untuk menjalankan bot. Untuk mendeploy bot secara online, kita akan menggunakan webhook, sehingga perlu beberapa perubahan pada kode sebelumnya.

Deploying the application

- we will first need to make this directory as a git repository before pushing the code to Heroku. Use the following command to instantiate your current directory as a git repository.

```
$ git init
```

- Create the Heroku application using the following command, you can use whatever "app_name" you like.

```
$ heroku create "app_name"
```

```
D:\Digital Talent Scholarship 2022\Python DTS PRoA 2022\Library Session 5\note-bot>heroku create note-bot-demo
 »   Warning: heroku update available from 7.53.0 to 7.60.1.
Creating ⬚ note-bot-demo... done
https://note-bot-demo.herokuapp.com/ | https://git.heroku.com/note-bot-demo.git
```

- Now you can use this Heroku application link in the APP_NAME in the code above. (In my case APP_NAME is "https://note-bot-demo.herokuapp.com/")

Remote URL of the application

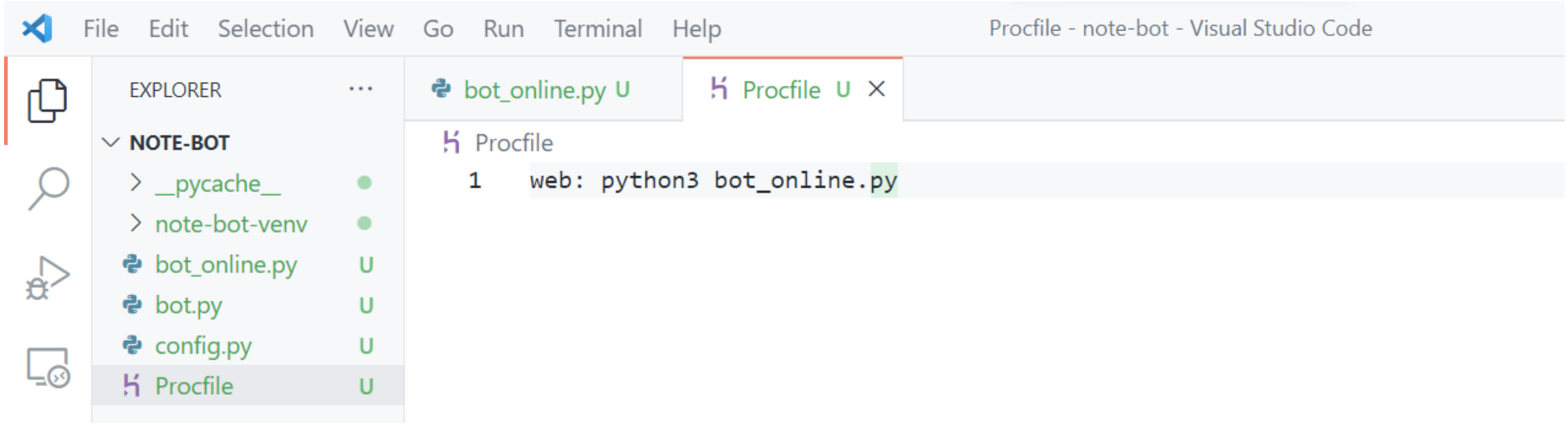
- Once the app is created, you can check the Remote URL of the application. Run the following command to check the emote information.

```
$ git remote -v
```

```
D:\Digital Talent Scholarship 2022\Python DTS PRoA 2022\Library Session 5\note-bot>git remote -v
heroku  https://git.heroku.com/note-bot-demo.git (fetch)
heroku  https://git.heroku.com/note-bot-demo.git (push)
```

- Now we need to tell our application what command to run on startup, for that Heroku uses a file called Procfile. You can read more about Procfile here.

Create a file named “Procfile”.



Tell the dependencies to install on the Heroku server

- Pip can export a list of all installed packages and their versions using the freeze command:

```
$ python -m pip freeze > requirements.txt
```

Last part

- you just need to add the files, commit it and push those changes to Heroku. For our changes, we can add all the files in the git repo. So run the following command.

```
$ git add .
```

- Give commit

```
$ git commit –m “pesan”
```

- Push

```
$ git push heroku master
```

- Now your application will start its build, you can check the logs by running the following command.

```
$ heroku logs -t
```

Referensi

- https://github.com/python-telegram-bot/python-telegram-bot/wiki/Webhooks#heroku
- https://github.com/python-telegram-bot/python-telegram-bot/wiki/Extensions-%E2%80%93-Your-first-Bot
- Link github code library session 1-5: https://github.com/muhammadoginh/library-session-2022.git

Terimakasih!