



CI/CD (Integration)

Chapter 5 - Topic 5

Selamat datang di Chapter 5 Topic 5
online course React Native dari Binar Academy!





Ketemu lagi nih kita!

Setelah di topik 4 kita bahas permukaannya CI/CD, di topik 5 chapter 5 ini kita akan bahas lebih dalam lagi nih mengenai CI/CD.

Wah, apa tuh?

Cuci sepatu bayarnya pake poin, kalau mau tahu langsung ya kepoin!



Detailnya, kita bakal bahas hal-hal berikut ini:

1. Integrasi CI/CD ke Telegram





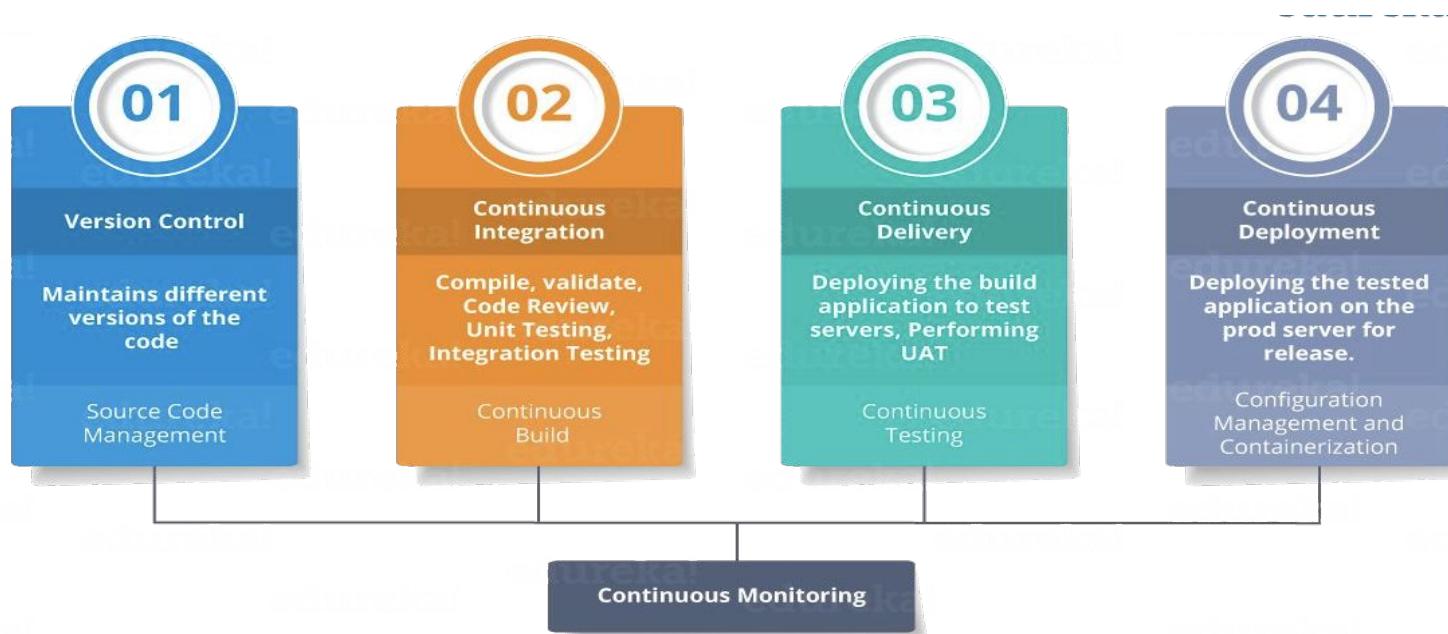
Setelah kita mempelajari dan mengetahui tentang gitlab secara overview, nah disini kita akan melakukan integrasi CI/CD ke Telegram. Biar apa ya?

Yuk Cekidot~





Review sedikit yuk ke cara kerja CI/CD





Bikin CI/CD makin mudah kamu jangkau!

Untuk sedikit mengingat, CI/CD sendiri tujuannya untuk melakukan continuous integration dan continuous deployment.

Lalu bagaimana continuous integration dan continuous deployment bisa lebih mudah dilakukan?

Nah, caranya adalah dengan memungkinkan adanya notifikasi setiap ada perubahan yang dilakukan di build aplikasi



Lalu bagaimana caranya memungkinkan adanya notifikasi setiap ada perubahan yang dilakukan di build aplikasi kita?

Nah, kita bisa mengintegrasikan CI/CD ke telegram. Biar kalau ada perubahan atau hal lain kita bisa mudah memberikan notifikasi ke tim developer melalui telegram.





**Langkah pertama dari semuanya adalah
build APK nya dulu~**





First of all~

Langkah paling awal dari semuanya adalah tentunya kita harus build APK nya terlebih dahulu.

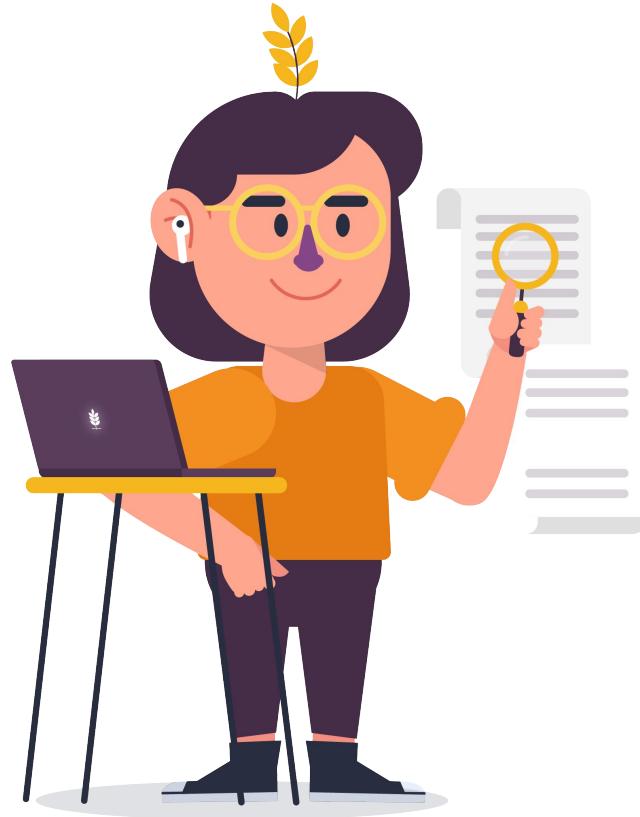
Point utama dari CI/CD adalah melakukan apa yang manual menjadi otomatis. Oleh karena itu kita perlu mengetahui step manualnya terlebih dahulu.





1. Build APK

Untuk android, bukanlah hal yang sulit untuk build APK, Pastikan saja bahwa project kamu sudah siap dan tidak ada error saat melakukan react-native run-android. Tapi, pastikan juga running secara debug terlebih dahulu ya!





Begini langkah-langkah build apk secara manual

Selanjutnya lakukan langkah - langkah berikut:

1. cd [YOUR_RN_PROJECT]
2. npm install
3. cd android
4. ./gradlew clean
5. ./gradlew assembleRelease

```
 admins-MacBook-Pro-2:android 161552.mikhael$ ./gradlew assembleRelease
```

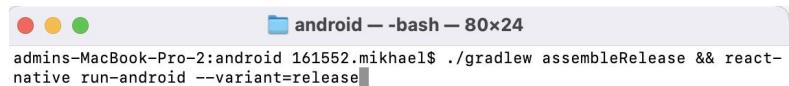
Sampai langkah ke 5, jika proses build success maka apk berhasil di-build dan kamu bisa menemukan apk di folder **/android/app/build/outputs/apk/release/app-release.apk**



Optional

tambahkan command:

react-native run-android --variant=release untuk setelah proses build untuk langsung test ke connected devices



```
 admins-MacBook-Pro-2:android 161552.mikhael$ ./gradlew assembleRelease && react-native run-android --variant=release
```



Nah, kalau udah kita brief tujuan dari kegiatan integrasi ini yuk~





Begini tujuannya

Tujuan yang akan kita lakukan kali ini adalah untuk melakukan deploy pada aplikasi react native kamu yang memiliki pipeline di gitlab CI/CD Server.

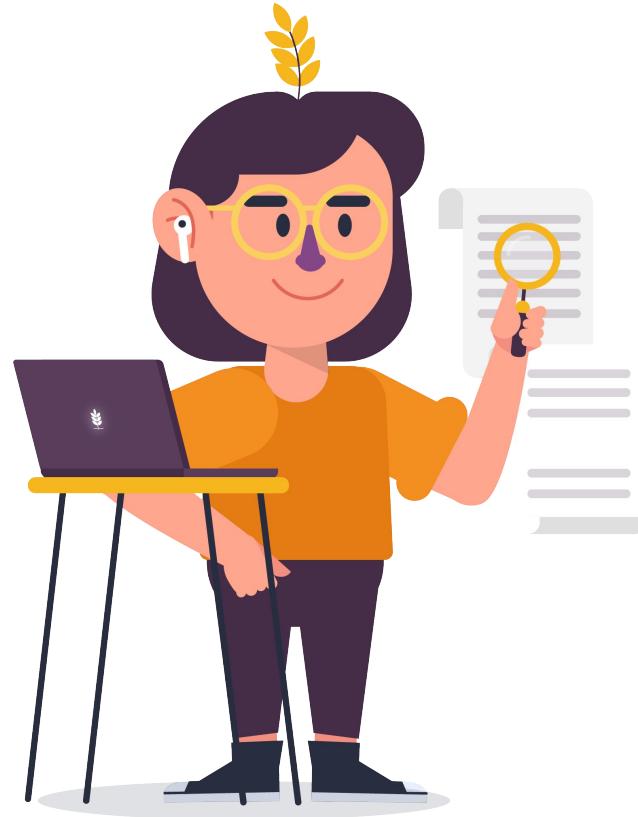
Sebagai hasilnya, code terbaru akan langsung ter-deploy setelah kita melakukan push di Gitlab.





Lalu keuntungannya~

Keuntungan menerapkan hal ini adalah kamu bisa melakukan publish atas perubahan pada code yang kamu buat, **tanpa melalui validasi yang dibutuhkan oleh app store/ playstore.** Jadi, ini bisa membantu develop pada level staging project kamu.





**Udah siap melakukan integrasi CI/CD ke
Telegram? Yuk mulai!**





Pastikan kamu sudah memiliki repository yang tersimpan di gitlab. Kali ini kita akan menggunakan telegram dan telegram bot yaitu bot father.





Langkah pertama, setup environment! Begini caranya

1. Search & Add User: BotFather





Lanjut~

2. Type: /newbot
3. Buatlah nama untuk bot misal RN/Telebot atau apapun.
4. Buatlah username untuk bot, isikan apapun misal rnbuidbot yang mudah diingat aja ya!





5. Selanjutnya kita akan mendapat Token. Jika semua proses selesai melalui chat dengan botFather, copy token kita dapatkan pada use this token to access the HTTP API: <Token>. Sekaligus telegram memberikan dokumentasi api telegram untuk kita baca.

Done! Congratulations on your new bot. You will find it at t.me/rnbuildBot. You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

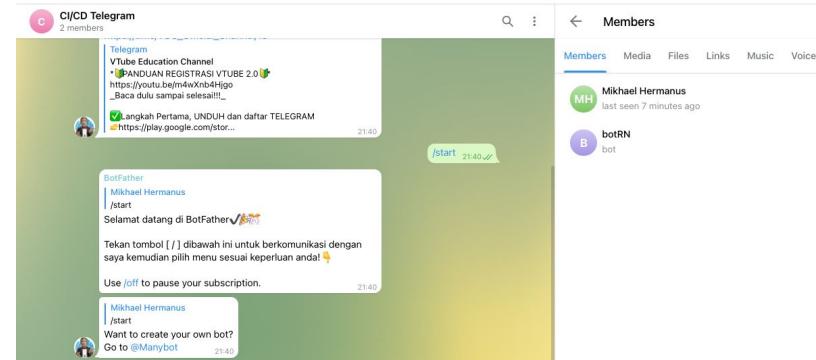
Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:

<https://core.telegram.org/bots/api>



6. Buatlah grup sederhana dan add telegram bot ke dalam grup search @bot_name (name yang kita buat)
7. kirim dummy message contoh /my_id @botRN



- copy token yang kalian dapat ke dalam url : `api.telegram.org/bot<Token>/getUpdates`, untuk mendapat chat_id yang akan kalian gunakan, pastikan anda mendapatkan response seperti berikut: ulangi langkah untuk mengirim send dummy message ke bot jika mendapatkan result = []

```
← → C https://api.telegram.org/bot5/getUpdates
API documentation SPT 1770S | Direkt... Login | Direktorat... Building Chatty — react-instantsearch... mic

{"ok":true,"result":[{"update_id":870085016,"my_chat_member":{"chat":{"id":-645714642,"title":"CI/CD Telegram","type":"group","all_members_are_administrators":true,"id":1821171439,"is_bot":false,"first_name":"Mikhael","last_name":"Hermanus","username":"mikhaelhrm"}, {"id":5121326688,"is_bot":true,"first_name":"botRN","username":"rn_CI_CD_BOT"}, {"status":"left"}, {"new_chat_member":{"id":5121326688,"is_bot":true,"first_name":"botRN","username":"rn_CI_CD_BOT"}, {"status":"member"}]}, {"message":{"message_id":1,"from":{"id":1821171439,"is_bot":false,"first_name":"Mikhael","last_name":"Hermanus","username":"mikhaelhrm"}, "chat":{"id":-645714642,"title":"CI/CD Telegram","type":"group","all_members_are_administrators":true}, "date":1649084077,"new_chat_participant":{"id":5121326688,"is_bot":true,"first_name":"botRN","username":"rn_CI_CD_BOT"}, "new_chat_members":[{"id":5121326688,"is_bot":true,"first_name":"botRN","username":"rn_CI_CD_BOT"}]}]}]
```



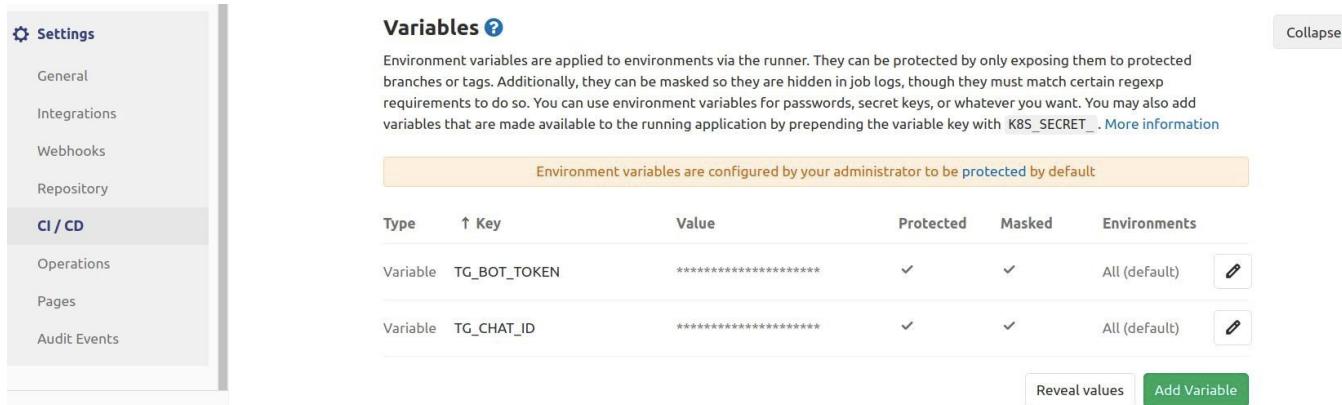
9. Test mengirimkan pesan melalui url : https://api.telegram.org/bot<token>/sendMessage?chat_id=<chat_id>&text=yes



```
{"ok":true,"result":{"message_id":9,"from":{"id":5121326688,"is_bot":true,"first_name":"botRN","username":"rn_CI_CD_BOT"},"chat":{"id":1821171439,"first_name":"Mikhael","last_name":"Hermanus","username":"mikhaelhrm","type":"private"},"date":1649085154,"text":"hello %trythis"}}
```



10. Pada Settings pilih CI/CD ke menu variables kemudian klik collapse , kita akan menambahkan variables dan token & chat ID



The screenshot shows the 'Variables' section within the 'CI / CD' settings of a GitLab project. The left sidebar lists various settings like General, Integrations, Webhooks, Repository, and Audit Events, with 'CI / CD' currently selected. The main area has a heading 'Variables' with a help icon and a 'Collapse' button. A note states: 'Environment variables are applied to environments via the runner. They can be protected by only exposing them to protected branches or tags. Additionally, they can be masked so they are hidden in job logs, though they must match certain regexp requirements to do so. You can use environment variables for passwords, secret keys, or whatever you want. You may also add variables that are made available to the running application by prepending the variable key with K8S_SECRET_. [More information](#)'.

Type	↑ Key	Value	Protected	Masked	Environments
Variable	TG_BOT_TOKEN	*****	✓	✓	All (default) Edit
Variable	TG_CHAT_ID	*****	✓	✓	All (default) Edit

At the bottom are buttons for 'Reveal values' and 'Add Variable'.



Setup environment Variables | Gitlab

Update variable

Key

TG_BOT_TOKEN

Value

https://api.telegram.org/bot1293026214:AAHsKlFyWB8mK_D4zHWFXu_PV_G4sB3zyDg

Type

Variable

Environment scope

All (default)

Flags Protect variable ?

Export variable to pipelines running on protected branches and tags only.

 Mask variable ?Variable will be masked in job logs. Requires values to meet regular expression requirements. [More information](#)[Cancel](#)[Delete variable](#)[Update variable](#)

Update variable

Key

TG_CHAT_ID

Value

115120260

Type

Variable

Environment scope

All (default)

Flags Protect variable ?

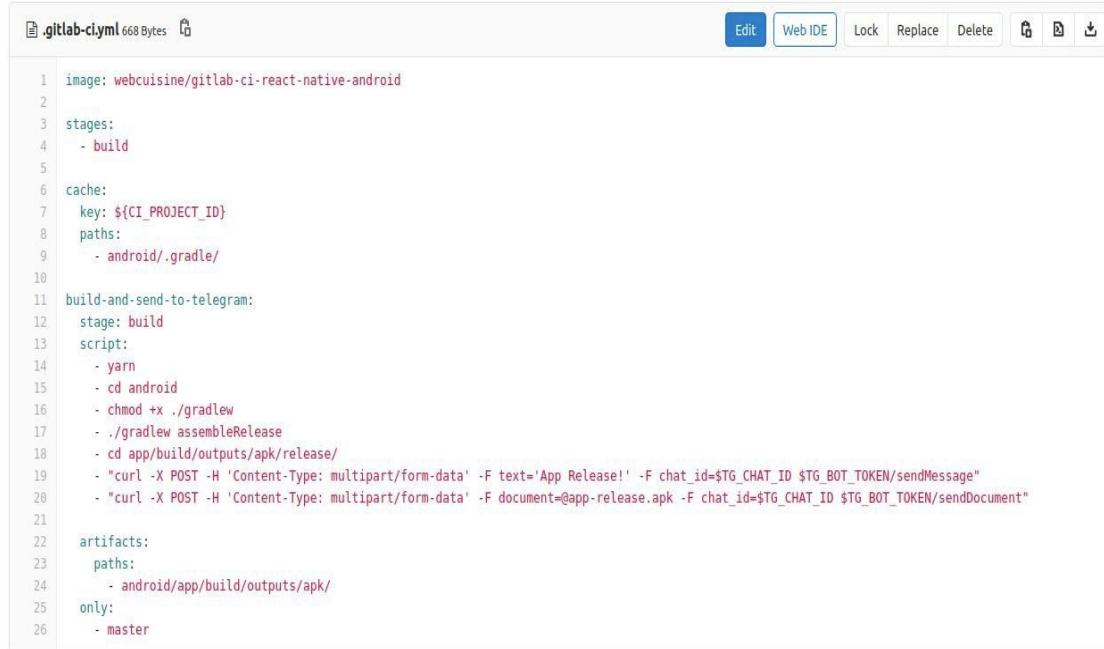
Export variable to pipelines running on protected branches and tags only.

 Mask variable ?Variable will be masked in job logs. Requires values to meet regular expression requirements. [More information](#)[Cancel](#)[Delete variable](#)[Update variable](#)



Langkah kedua, setting script! Begini caranya

1. Buka Project React native kalian , kemudian tambahkan file gitlab-ci.yml seperti pada code di samping.



```
gitlab-ci.yml 668 Bytes 
```

Edit Web IDE Lock Replace Delete   

```
1 image: webcuisine/gitlab-ci-react-native-android
2
3 stages:
4   - build
5
6 cache:
7   key: ${CI_PROJECT_ID}
8   paths:
9     - android/.gradle/
10
11 build-and-send-to-telegram:
12   stage: build
13   script:
14     - yarn
15     - cd android
16     - chmod +x ./gradlew
17     - ./gradlew assembleRelease
18     - cd app/build/outputs/apk/release/
19     - "curl -X POST -H 'Content-Type: multipart/form-data' -F text='App Release!' -F chat_id=$T6_CHAT_ID $T6_BOT_TOKEN/sendMessage"
20     - "curl -X POST -H 'Content-Type: multipart/form-data' -F document=@app-release.apk -F chat_id=$T6_CHAT_ID $T6_BOT_TOKEN/sendDocument"
21
22 artifacts:
23   paths:
24     - android/app/build/outputs/apk/
25 only:
26   - master
```



Coba perhatikan job scriptnya deh, setiap perintah menjalankan perintah yang kita lakukan dalam melakukan manual build apk. Kamu juga harus memastikan path untuk mengambil apk sudah benar ya! Lalu Pada only : - akan tereksekusi jika hanya push ke dalam branch master.

.gitlab-ci.yml 668 Bytes 

Edit Web IDE Lock Replace Delete        

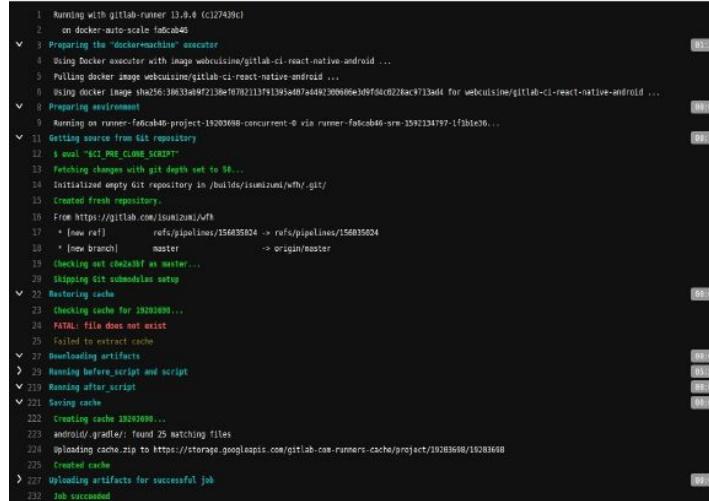
```
1 image: webcuisine/gitlab-ci-react-native-android
2
3 stages:
4   - build
5
6 cache:
7   key: ${CI_PROJECT_ID}
8   paths:
9     - android/.gradle/
10
11 build-and-send-to-telegram:
12   stage: build
13   script:
14     - yarn
15     - cd android
16     - chmod +x ./gradlew
17     - ./gradlew assembleRelease
18     - cd app/build/outputs/apk/release/
19     - "curl -X POST -H 'Content-Type: multipart/form-data' -F text='App Release!' -F chat_id=$TG_CHAT_ID $TG_BOT_TOKEN/sendMessage"
20     - "curl -X POST -H 'Content-Type: multipart/form-data' -F document=@app-release.apk -F chat_id=$TG_CHAT_ID $TG_BOT_TOKEN/sendDocument"
21
22 artifacts:
23   paths:
24     - android/app/build/outputs/apk/
25 only:
26   - master
```



Lanjut~

2. Jalankan git add + git commit + push

maka pipeline akan jalan otomatis, pastikan push ke dalam branch master.



The screenshot shows a terminal window displaying the output of a GitLab CI pipeline. The logs are color-coded by step: green for success, red for errors, and blue for informational messages. The pipeline starts with setting up a runner and preparing the Docker executor, then cloning the repository from GitHub. It proceeds through various stages including fetching changes, checking out branches, and running before and after scripts. Finally, it creates a cache, uploads artifacts to Google Cloud Storage, and concludes with a success message. The log ends with a timestamp of 09:54.

```
1 Running with gitlab-runner 13.9.4 (c127439c)
2 on docker-auto-scale fadca46
3 Preparing the 'docker-machine' executor
4 Using Docker executor with image webcuisine/gitlab-ci-react-native-android ...
5 Pulling docker image webcuisine/gitlab-ci-react-native-android ...
6 Using docker image sha256:30633a9f7213be10f82113f91394a07a4d930660e3d9f04cd228ac9713add for webcuisine/gitlab-ci-react-native-android ...
7 Preparing environment
8 Running on runner-fadca46-project-19203698-concurrent-0 via runner-fadca46-srm-159234797-1fb1c06...
9 eval "$CI PIPELINE_SCRIPT"
10 Getting source from Git repository
11 Fetching changes with git depth set to 50...
12 Initialized empty Git repository in /builds/ismunizumi/wfh/.git/
13 Created fresh repository.
14 From https://github.com/ismunizumi/wfh
 * [new ref]      refs/pipelines/156835024 => refs/pipelines/156835024
 * [new branch]   master          -> origin/master
15 Checking out cb223ff as master...
16 Skipping Git submodules setup
17 Restoring cache
18 Checking cache for 19203699...
19 FATAL: file does not exist
20 Failed to extract cache
21 Downloading artifacts
22 Running before_script and script
23 Saving cache
24 Creating cache 19203699...
25 android/gradle/ found 25 matching files
26 uploading cache.zip to https://storage.googleapis.com/gitlab-ci-runners-cache/project/10283698/19203698
27 Created cache
28 Uploading artifacts for successful job
29 Job succeeded
```



```
1  Running with gitlab-runner 13.8.0 (c12748c)
2    on docker-auto-scale fa6cab46
3  Preparing the "docker-machine" executor
4  Using Docker executor with image webcuisine/gitlab-ci-react-native-android ...
5  Pulling docker image webcuisine/gitlab-ci-react-native-android ...
6  Using docker image sha256:36e33a8f9f2138e10782113f9139540f44492360606e3d9f00c6228ac9713ad4 for webcuisine/gitlab-ci-react-native-android ...
7
8  Preparing environment
9  Running on runner-fa6cab46-project-19203698-concurrent-0 via runner-fa6cab46-srm-1592134797-1fbfe36...
10 Getting source from Git repository
11 $ eval "$CI_PRE_CLONE_SCRIPT"
12 Fetching changes with git depth set to $CI_COMMIT_DEPTH...
13 Initialized empty Git repository in /builds/isumizumi/wfh/.git/
14 Created fresh repository.
15 From https://gitlab.com/isumizumi/wfh
16   * [new ref]      refs/pipelines/156035024 -> refs/pipelines/156035024
17   * [new branch]   master           -> origin/master
18 Checking out c6e2a3bf as master...
19 Skipping Git submodules setup
20 Restoring cache
21 Checking cache for 19203698...
22 FATAL: file does not exist
23 Failed to extract cache
24 Downloading artifacts
25 Running before_script and script
26 Running after_script
27 Saving cache
28 Creating cache 19203698...
29 android/gradle: found 25 matching files
30 Uploading cache.zip to https://storage.googleapis.com/gitlab-com-runners-cache/project/19203698/19203698
31 Created cache
32 Uploading artifacts for successful job
33 Job succeeded
```



Nah, begini proses pipeline kamu yang seharusnya jika sudah berjalan.

All 5	Finished	Branches	Tags				
				Filter pipelines			
Status	Pipeline	Triggerer	Stages				
<div>passed</div>	Update README.md #137255411 ⏺ master -o 51da4ffd latest		✓ ✓				
<div>passed</div>	Add bottom sheet for Subjects. #135889850 ⏺ master -o 4e01fa5d		✓ ✓				



Selamaaatt!! Sampai di sini kamu telah mempelajari bagaimana proses CI/CD bekerja dengan memanfaatkan telegram.

Tapi, sebenarnya selain menggunakan gitlab masih banyak cara lain lho! Kamu boleh mencari tahu sendiri ya!





Saatnya kita Quiz!





1. Apa kegunaan telegram menyediakan bot token botFather?

- A. Untuk kita bisa melakukan chat dengan botFather
- B. Memanfaatkan bot dalam mengirimkan otomatis chat.
- C. Menjadikan Bot notifikasi pipeline



2. Pernyataan dibawah ini yang benar adalah

- A. Kita perlu menyimpan token Bot ID dan Chat ID ke gitlab
- B. Kita perlu melakukan send dummy message url setiap kali pipeline selesai
- C. Kita tidak perlu menggunakan gitlab



3. Apa dasar command untuk membuild apk ?

- A. cd android ./gradlew clean
- B. cd android ./gradlew assembleDebug
- C. cd android ./gradlew assembleRelease



4. Secara singkat dalam CI/CD apa yang dimaksud dengan Continuous Development

- A. memanfaatkan service gitlab agar development bergantung terhadap deployment
- B. memanfaatkan service dari gitlab yang tujuannya untuk bisa melakukan development secara paralel dengan deployment
- C. memanfaatkan service gitlab untuk melakukan pengecekan setiap commit yang diberikan



Thank you

