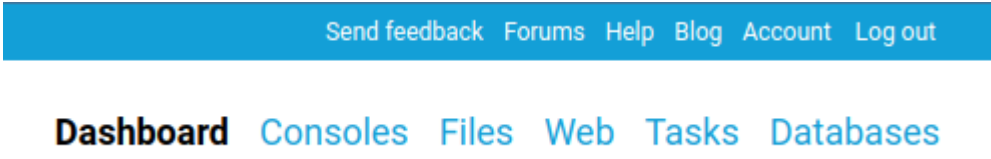

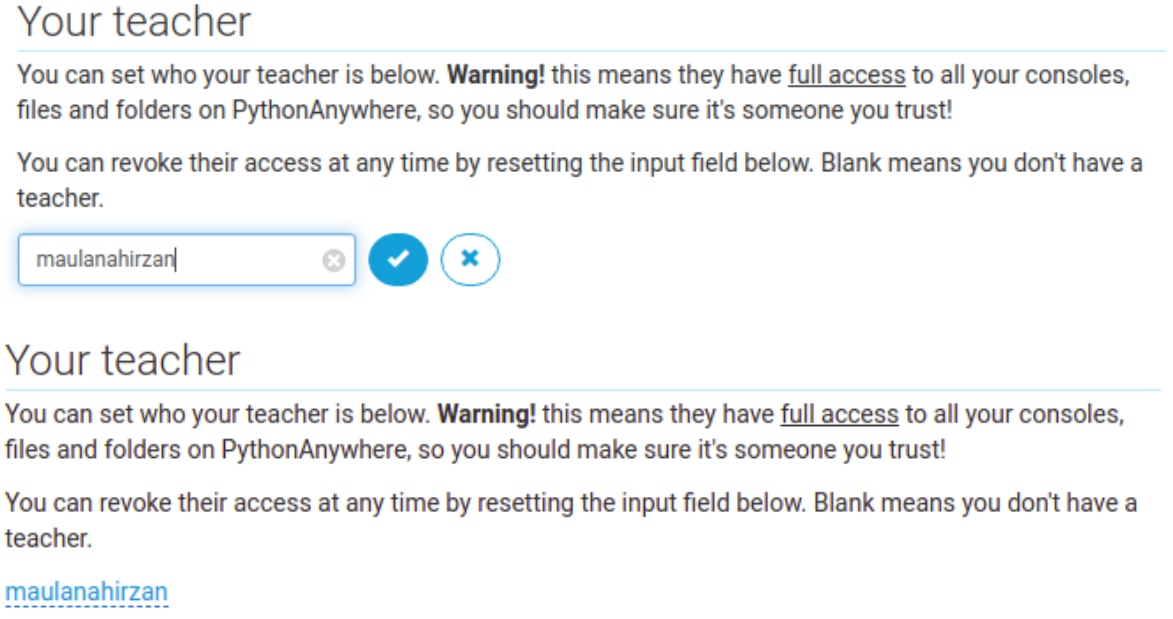


Praktikum 6 – Internet of Things

No	Langkah-langkah
1	Buka Dashboard dari pythonanywhere , kemudian klik Account di bagian atas kanan
	
2	Klik tab Education ,
	
3	Klik Enter your teacher's username lalu masukkan maulanahirzan . Hal ini dilakukan untuk penilaian nantinya. Kemudian klik Centang
	
4	Dipertemuan kali ini kita akan membuat proses scheduling pesan secara otomatis dengan Telegram Bot . Sehingga dapat digunakan untuk mengirim kondisi rumah secara periodik. Kembali ke Dashboard , dan buatlah file baru

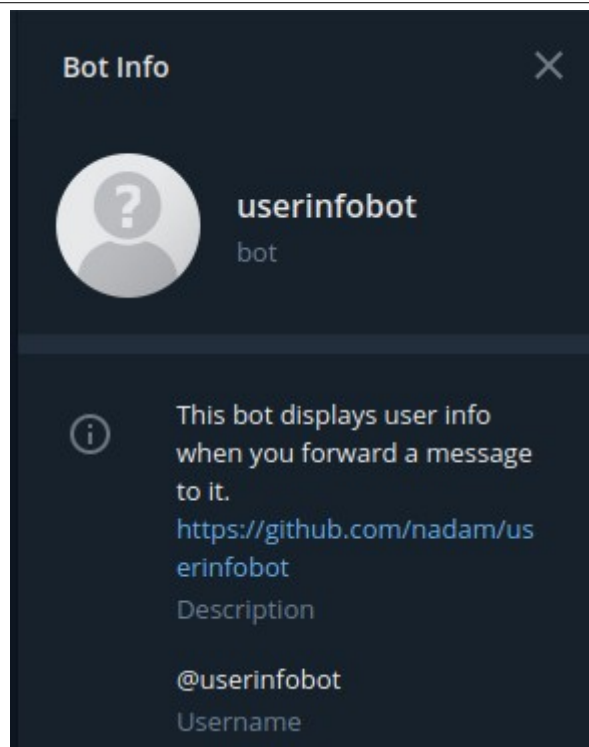
	<div>Files</div> <div>telegram_schedule.py New file</div>
5	<p>Copy isi dari file Praktikum 5.py sebelumnya ke file ini. Abaikan jika ada yang error</p>
	<pre> 1 #!/usr/bin/env python 2 3 import logging 4 import os, psutil 5 6 from telegram import Update, ForceReply 7 from telegram.ext import Updater, CommandHandler, MessageHandler, Filters, CallbackContext 8 9 logging.basicConfig(10 format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO 11) 12 logger = logging.getLogger(__name__) 13 14 15 def start(update: Update, context: CallbackContext) -> None: 16 user = update.effective_user 17 update.message.reply_markdown_v2(18 fr'Hi {user.mention_markdown_v2()}\!', 19 reply_markup=ForceReply(selective=True), 20) 21 22 def memory_usage(update: Update, context: CallbackContext) -> None: 23 process = psutil.Process(os.getpid()) 24 update.message.reply_text(process.memory_info().rss) 25 26 def echo(update: Update, context: CallbackContext) -> None: 27 update.message.reply_text(update.message.text) 28 29 30 def main() -> None: 31 updater = Updater("1901469256:AAHz0864vwPsAS6HWu68GZ4uoQ8k_FS0YU8") 32 33 dispatcher = updater.dispatcher 34 35 # Perintah 36 dispatcher.add_handler(CommandHandler("start", start)) 37 dispatcher.add_handler(CommandHandler("memory", memory_usage)) </pre>
6	<p>Hapus baris kode ini</p>
	<pre> user = update.effective_user update.message.reply_markdown_v2(fr'Hi {user.mention_markdown_v2()}\!', reply_markup=ForceReply(selective=True),) def memory_usage(update: Update, context: CallbackContext) -> None: process = psutil.Process(os.getpid()) update.message.reply_text(process.memory_info().rss) def echo(update: Update, context: CallbackContext) -> None: update.message.reply_text(update.message.text) dispatcher.add_handler(CommandHandler("memory", memory_usage)) dispatcher.add_handler(MessageHandler(Filters.text & ~Filters.command, echo)) </pre>
7	<p>Kode harus terlihat seperti ini</p>

```


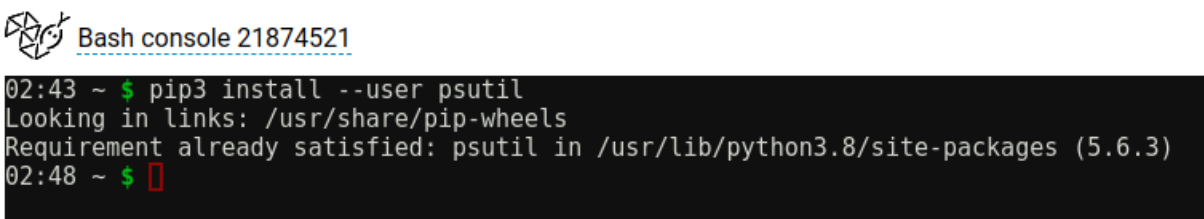
1  #!/usr/bin/env python3
2
3  import logging
4  import os, psutil
5
6  from telegram import Update, ForceReply
7  from telegram.ext import Updater, CommandHandler, MessageHandler, Filters, CallbackContext
8
9  logging.basicConfig(
10     format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO
11 )
12 logger = logging.getLogger(__name__)
13
14 id = 1362682845
15
16 def start(update: Update, context: CallbackContext) -> None:
17
18
19 def main() -> None:
20     updater = Updater("1901469256:AAHz0864vwPsAS6HWu68GZ4uoQ8k_FS0YU8")
21     dispatcher = updater.dispatcher
22
23     # Perintah
24     dispatcher.add_handler(CommandHandler("start", start))
25
26     updater.start_polling()
27     updater.idle()
28
29
30 if __name__ == '__main__':
31     main()


```

- 8 Sebelum mengisi kode, cari **chat id** daripada akun Anda masing-masing dengan cara menghubungi @userinfobot



- 9 Kirim pesan /start untuk mendapatkan **ID**. Ini digunakan untuk melakukan **Direct Message / PM** dari **Bot** ke **Akun**

	
10	Copy ID tersebut dan Paste ke kode Telegram Sebelumnya
	<pre> logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO) logger = logging.getLogger(__name__) id = [redacted] def start(update: Update, context: CallbackContext) -> None: </pre>
11	Sekarang kita dapat memulai mengisikan fungsi di dalam kode. Dimulai dari Start
	<pre> def start(update: Update, context: CallbackContext) -> None: message = "Selamat Datang di BOT Telegram Asisten IoT" context.bot.send_message(chat_id=id, text=message) </pre>
12	Berikutnya untuk membuat fungsi untuk mengambil data suhu maupun proses IoT. Namun sebelumnya wajib menginstall tool psutil dari bash/console . Gunakan perintah berikut
	
13	Jika sudah, kembali ke file kode, dan Import tool tersebut.
	<pre> import logging import os, psutil </pre>
14	Buatlah fungsi untuk memonitor penggunaan CPU dalam bentuk percent
	<pre> def cpu_percent(context: CallbackContext) -> None: cpu = psutil.cpu_percent() message = "Penggunaan CPU saat ini : " + str(cpu) + "%" context.bot.send_message(chat_id=id, text=message) </pre>
15	Namun fungsi ini tidak akan berjalan jika belum di set scheduler nya. Definisikan JobQueue setelah baris kode dispatcher
	<pre> updater = Updater('1561707230:AAH...') dispatcher = updater.dispatcher jq = updater.job_queue </pre>

16	Lalu tambahkan scheduler setelah perintah dispatcher.addhandler milik start . Baca langkah 17 untuk keterangan masing-masing scheduler
	<pre># Perintah dispatcher.add_handler(CommandHandler("start", start)) #jq.run_once(cpu_percent,10) #time = datetime.time(10, 00, 00, 000000) #jq.run_daily(cpu_percent,time, days=(0, 1, 2, 3, 4, 5, 6)) #jq.run_repeating(cpu_percent,interval=10)</pre>
17	Sintaks run_once() berarti fungsi akan dijalankan sekali setelah terlewat beberapa detik dari menjalankan script. Contoh di bawah ini menjalankan fungsi setelah 10 detik
	<pre>#jq.run_once(cpu_percent,10)</pre>
18	Sintaks run_daily() menjalankan fungsi setiap hari yang ditentukan dengan list tuple : 0 : Senin, 1 : Selasa, 2 : Rabu, 3 : Kamis, 4 : Jumat, 5 : Sabtu, 6 : Minggu . Pilih salah satu nomor, atau lebih. Selain itu sintaks ini dilengkapi dengan time untuk menentukan jam berapa fungsi dijalankan. Harus diformat dalam bentuk date.time . Contoh: Fungsi dijalankan Setiap Hari tepat jam 10:00:00 pagi sesuai zona waktu IoT
	<pre>#time = datetime.time(10, 00, 00, 000000) #jq.run_daily(cpu_percent,time, days=(0, 1, 2, 3, 4, 5, 6))</pre>
19	Sintaks run_repeating() menjalankan waktu berulang kali setiap berapa detik. Contoh: Setiap 10 Detik , Fungsi akan dijalankan
	<pre>#jq.run_repeating(cpu_percent,interval=10)</pre>
20	Hilangkan komentari dari sintaks run_repeating untuk mengetes fungsi
	<pre>jq.run_repeating(cpu_percent,interval=10)</pre>
21	Jalankan dari Console masing-masing
	
	<pre>2021-10-30 03:15:54,672 - apscheduler.scheduler - INFO - Adding job tentatively -- it will be properly scheduled when the scheduler starts 2021-10-30 03:15:54,672 - apscheduler.scheduler - INFO - Added job "cpu_percent" to job store "default" 2021-10-30 03:15:54,672 - apscheduler.scheduler - INFO - Scheduler started 2021-10-30 03:16:04,672 - apscheduler.executors.default - INFO - Running job "cpu_percent (trigger: interval[0:00:10], next run at: 2021-10-30 03:16:14 UTC)" (scheduled at 2021-10-30 03:16:04,672288+00:00) 2021-10-30 03:16:05,197 - apscheduler.executors.default - INFO - Job "cpu_percent (trigger: interval[0:00:10], next run at: 2021-10-30 03:16:14 UTC)" executed successfully</pre>
	SELESAI