

Python Firebase

Menggunakan Database dan
Authentication Firebase
dengan Python 3

Disusun oleh Tim BCS

Dilengkapi dengan link source code
dan studi kasus



**BATAVIA
CODING
SCHOOLS**
semua bisa belajar

Python Firebase

*Menggunakan Database dan Authentication
Firebase dengan Python 3*

*Tim Batavia Coding Schools (BCS)
2020, Jakarta*

Daftar Isi

Pendahuluan.....	6
Apa Itu Firebase.....	6
Push Notifications.....	6
Authentication.....	6
Realtime Database.....	6
Persiapan.....	8
Membuat Project Firebase.....	11
Membuat Project.....	11
Membuat Aplikasi di Firebase.....	15
Membuat Aplikasi.....	15
Membuat File Konfigurasi di Python.....	16
Kesimpulan.....	19
Login ke Firebase dengan Python.....	20
Mengaktifkan Metode Login.....	20
Membuat User.....	22
Mencoba Login.....	23
Kesimpulan.....	24
Membuat Database Firebase.....	25
Membuat Database.....	25
Kesimpulan.....	27
Insert Data pada Database Firebase.....	28
Insert Data	28
Method Set.....	30
Insert Banyak Data Sekaligus.....	32
Kesimpulan.....	33
Mengambil Data pada Database Firebase	34
Mengambil Data.....	34
Mengambil Id.....	36
Kesimpulan.....	37
Update Data pada Database Firebase.....	38
Update Data.....	38
Kesimpulan.....	39

Hapus Data pada Database Firebase	40
Hapus Data.....	40
Kesimpulan.....	42
Studi Kasus Aplikasi cli Sederhana.....	43
Menjalankan Program.....	56
Kesimpulan.....	60
Lampiran 1 – Cara Mengunduh Source Code.....	61
Lampiran 2 – Cara Menjalankan Source Code.....	63
Lampiran 3 – Cara Donasi.....	65
Bagaimana Cara Berdonasi.....	65

Tentang Buku

Pada serial buku ini, kita akan belajar memanfaatkan layanan database realtime Firebase dan terakhir kita akan membuat aplikasi CLI sederhana dengan Python, dimana kita dapat melakukan registrasi, login, mengganti password pengguna dan melakukan serangkaian crud data produk ke database di Firebase.

Buku ini ditulis dengan bahasa yang sederhana dan isinya menekankan pada praktik. Buku ini ditujukan kepada teman-teman yang sudah mempelajari dasar Python.

Saat penulisan buku ini, kami menggunakan Python 3 dan Sistem Operasi Windows untuk menguji kode-nya.

Pendahuluan

Apa Itu Firebase

Firebase adalah Cloud Service Provider dan Backend as a Service yang dimiliki oleh Google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah dalam pengembangan aplikasi mobile maupun web. Kita tidak perlu membangun fitur-fitur yang dibuat pada backend dan infrastruktur dari awal sehingga kita dapat fokus untuk mengembangkan aplikasi yang berkualitas tinggi tanpa perlu mengeluarkan effort yang besar.

Push Notifications

Dengan Push Notifications, kita bisa mengirimkan notifikasi atau pesan kepada pengguna aplikasi kita secara realtime. Push Notifications merupakan bagian dari layanan Grow pada Firebase.

Authentication

Dengan layanan ini kita dapat membuat authentication untuk aplikasi kita yang disediakan dan ditangani oleh Firebase. Jadi kita tidak perlu repot-repot membuat sistem login untuk aplikasi kita. Kita bisa memanfaatkan layanan ini dari Firebase.

Realtime Database

Realtime Database adalah sebuah Cloud-Hosted database yang dapat kita gunakan untuk menyimpan dan melakukan sinkronisasi data secara realtime untuk setiap client yang terhubung. Setiap kali user

memperbarui data, data akan tersimpan pada cloud dan sekaligus memberitahu ke semua client yang terhubung dan secara otomatis menerima pembaruan dengan data terbaru.

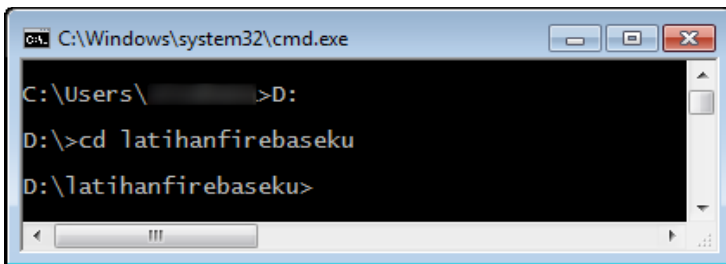
Sebenarnya masih banyak lagi fitur-fitur yang disediakan oleh Firebase. Di sini hanya disebutkan fitur-fitur umum yang sering sekali digunakan.

Persiapan

Kita akan menyiapkan folder project untuk menyimpan file latihan-latihan yang ada di buku ini. Di sini saya menyiapkan folder project di drive D dengan nama **latihanfirebaseku**:

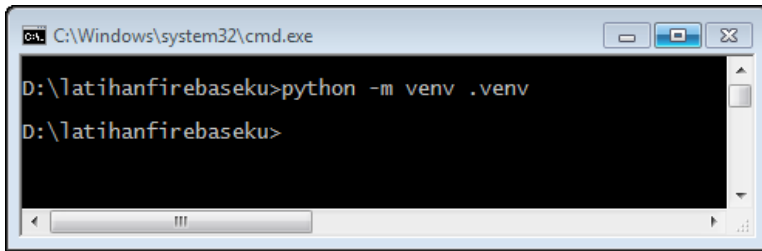
Selanjutnya, buka cmd dan masuk ke folder tersebut dengan perintah:

```
D:  
cd latihanfirebaseku
```



Setelah itu, jalankan perintah berikut untuk membuat virtual environment dengan nama **.venv**:

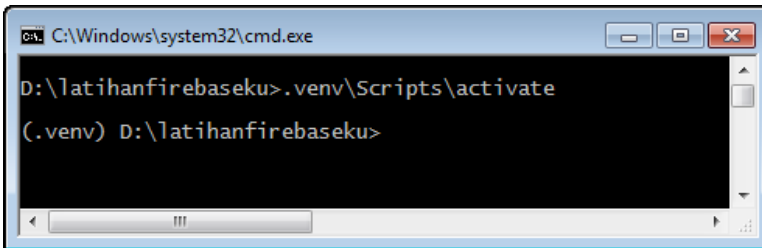
```
python -m venv .venv
```

```
C:\Windows\system32\cmd.exe
D:\latihanfirebaseku>python -m venv .venv
D:\latihanfirebaseku>
```

Jika sudah, aktifkan virtual environment yang sudah kita buat dengan menjalankan perintah berikut di cmd saat ini:

```
.venv\Scripts\activate
```



```
C:\Windows\system32\cmd.exe
D:\latihanfirebaseku>.venv\Scripts\activate
(.venv) D:\latihanfirebaseku>
```

Selanjutnya, kita akan menginstal package Pyrebase dengan perintah berikut di cmd saat ini (pada langkah ini, pastikan Anda terkoneksi dengan internet, karena pip akan mengunduh terlebih dahulu package Pyrebase sebelum proses instalasi):

```
pip install pyrebase
```

Pyrebase merupakan package Python yang akan kita gunakan untuk berkomunikasi dengan Firebase. Package tersebut diibaratkan sebagai penghubung atau jembatan antara Python dan Firebase.

Silahkan buka folder project Anda dengan IDE atau Editor kesayangan Anda.

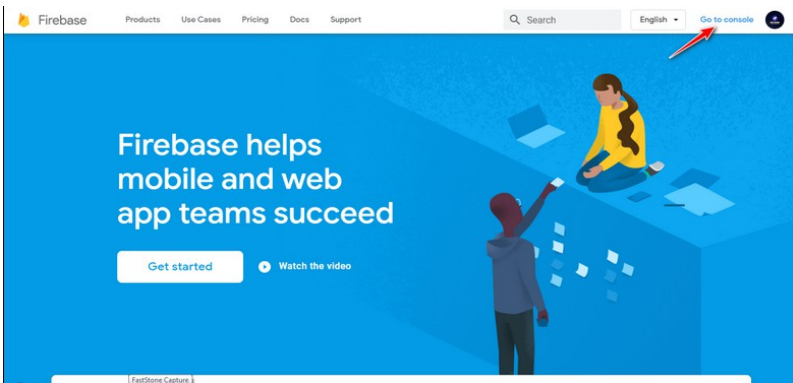
Sebagai catatan, ketika masuk ke bab selanjutnya, jangan lupa membuka cmd, lalu masuk ke folder project-nya dan aktifkan virtual environment-nya.

Membuat Project Firebase

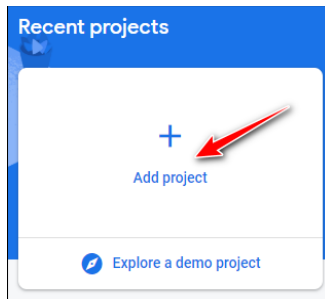
Membuat Project

Sebelum lebih jauh kita mempelajari buku ini, kita wajib membuat project di Firebase. Pastikan Anda memiliki akun email Google. Untuk membuat project, ikuti langkah-langkah berikut ini.

Buka browser dan akses url <https://firebase.google.com/>:



Klik **Go to console** di sebelah kanan atas. Lalu kita akan di bawa ke halaman **Recent projects**. Klik bagian **Add project**:



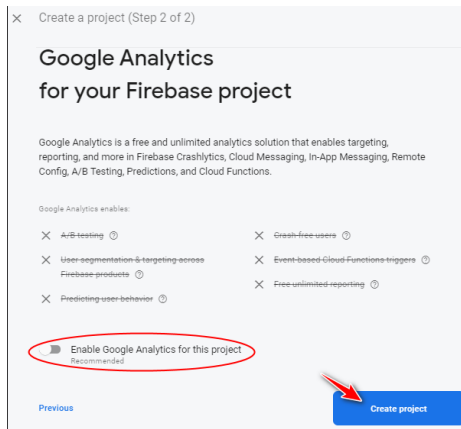
Selanjutnya, isi nama project-nya misalnya **project-todo**:

Setelah itu, klik tombol **Continue**:

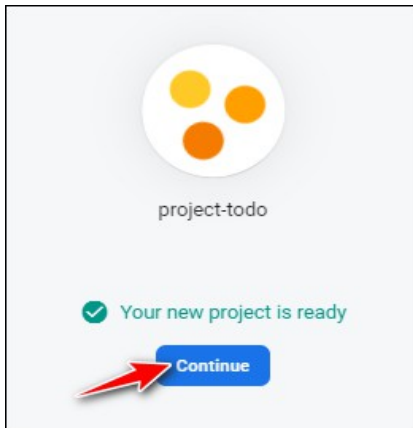


Akan muncul form **Google Analytics for your Firebase project**. Matikan **Enable Google Analytics for this project** lalu klik tombol **Continue**:

Membuat Project Firebase

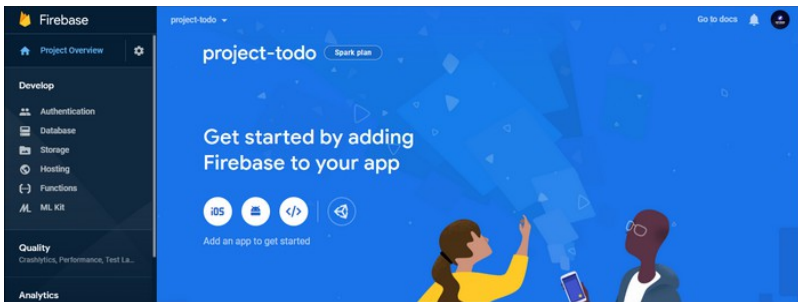


Tunggu hingga pembuatan project selesai. Jika sudah, klik tombol **Continue**:



Kita akan di bawa ke halaman **Dashboard Firebase** untuk project yang baru saja kita buat:

Membuat Project Firebase

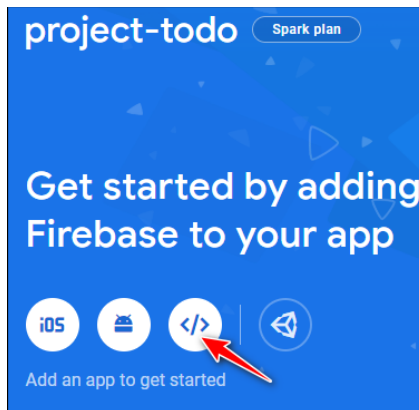


Membuat Aplikasi di Firebase

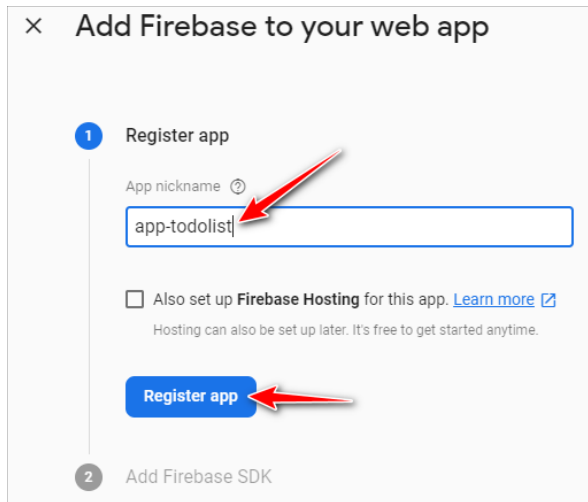
Membuat Aplikasi

Pada bab sebelumnya kita sudah berhasil membuat project di Firebase. Pada bab ini kita perlu membuat aplikasi di dalam project tersebut. Dengan membuat aplikasi di Firebase, kita akan mendapatkan konfigurasi untuk Pyrebase agar Python dan Firebase bisa saling berkomunikasi. Untuk membuat aplikasi, ikuti langkah-langkah berikut ini.

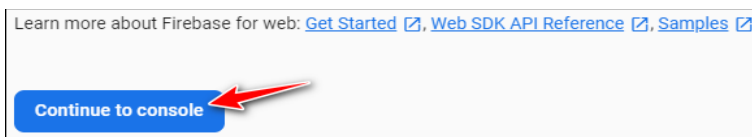
Buka kembali halaman **Dashboard Firebase** pada **project-todo**. Klik ikon web:



Akan muncul form **Add Firebase to your web app**. Isi bagian **App nickname** dengan nama, misalnya **app-todolist** lalu klik **Register app**:



Setelah itu, akan muncul langkah **Add Firebase SDK**. Untuk saat ini, jangan di salin dulu konfigurasinya, kita lanjutkan saja dengan menekan tombol **Continue to console** untuk kembali ke halaman dashboard:



Sampai sini kita berhasil membuat aplikasi di Firebase. Perhatikan, di halaman Dashboard Firebase, terdapat ikon **apps** dengan tulisan **1 app** di bawah nama project kita.

Membuat File Konfigurasi di Python

Buka folder project **D:\latihanfirebaseku** dengan IDE atau Editor. Buat file bernama **config.py** dan isi dengan kode berikut:


```
import pyrebase

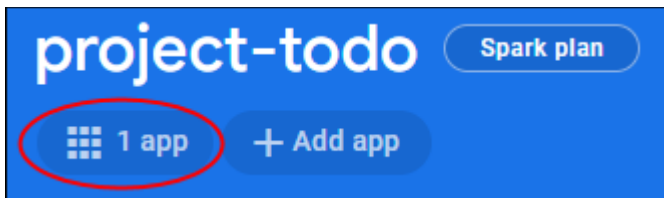
config = {
    "apiKey": "",
    "authDomain": "",
    "databaseURL": "",
    "storageBucket": "",
}

firebase = pyrebase.initialize_app(config)
```

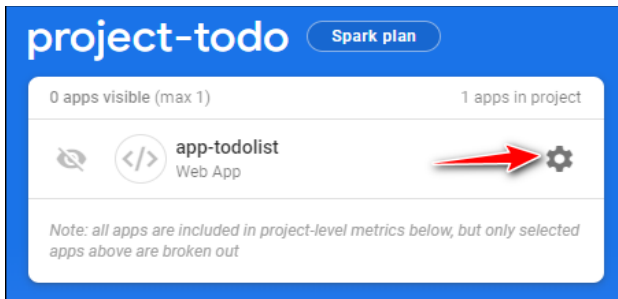
Perhatikan pada variabel **config**. Terdapat beberapa konfigurasi yang harus kita isi nilainya seperti **apiKey**, **authDomain**, **databaseURL** dan **storageBucket**.

Nilai-nilai konfigurasi tersebut bisa kita peroleh di pengaturan aplikasi Firebase. Nilai-nilai konfigurasi tersebut digunakan oleh Pyrebase sebagai kunci untuk berkomunikasi dengan Firebase. Untuk mendapatkannya, ikuti langkah-langkah berikut.

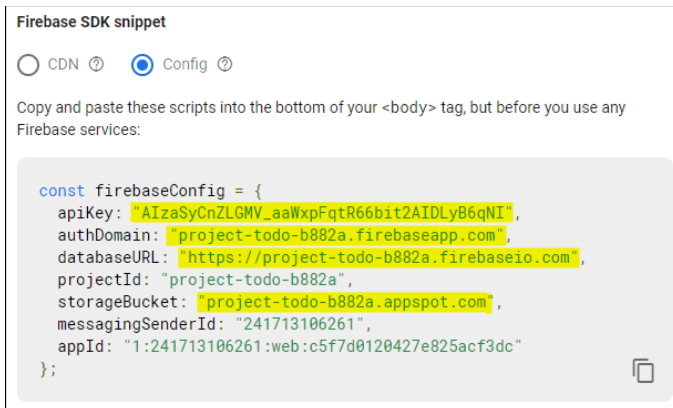
Pada halaman **Dashboard Firebase**, klik ikon **app**:



Muncul pop-up seperti pada gambar di bawah ini. Klik ikon konfigurasi di paling kanan untuk masuk ke halaman **Settings**:



Selanjutnya, kita akan di bawa ke halaman **Settings**. Di halaman tersebut pilih tab **General** lalu scroll ke bawah dan temukan bagian **Firestore SDK snippet** dan pilih opsi **Config**:



Salin nilai-nilai yang kita butuhkan sesuai dengan file **config.py** yang sudah kita buat. Berikut adalah kode keseluruhan **config.py** setelah kita tambahkan nilai-nilai config Firebase:

```
import pyrebase

config = {
    "apiKey": "AIzaSyCnZLGMVaaWxpFqtR66bit2AIDLyB6qNI",
```

```
"authDomain": "project-todo-b882a.firebaseio.com",  
"databaseURL": "https://project-todo-b882a.firebaseio.com",  
"storageBucket": "project-todo-b882a.appspot.com",  
}  
  
firebase = pyrebase.initialize_app(config)
```

Kesimpulan

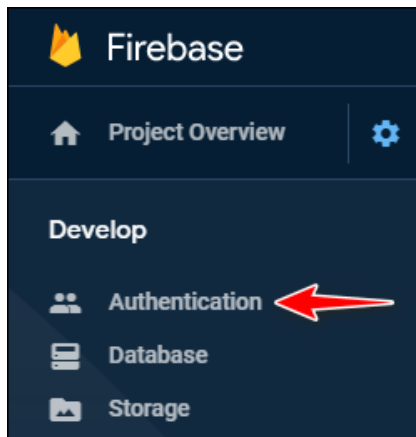
Pada bab ini kita telah membuat aplikasi Firebase dan membuat file konfigurasi di Python untuk aplikasi tersebut. Di bab selanjutnya kita akan coba melakukan login ke Firebase menggunakan Python.

Login ke Firebase dengan Python

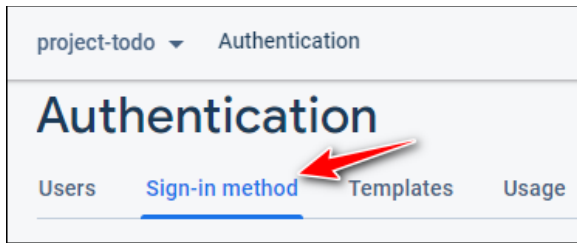
Mengaktifkan Metode Login

Perlu kita ketahui, ada beberapa metode login yang disediakan oleh Firebase. Pada buku ini kita menggunakan metode login dengan **Email/Password**. Kita harus mengaktifkan metode tersebut sebelum kita melakukan login ke Firebase menggunakan Python.

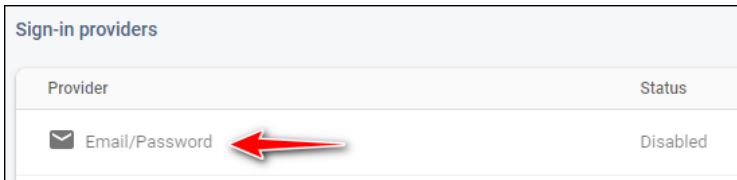
Kembali ke halaman **Dashboard Firebase**. Klik menu **Authentication** di sebelah kiri:



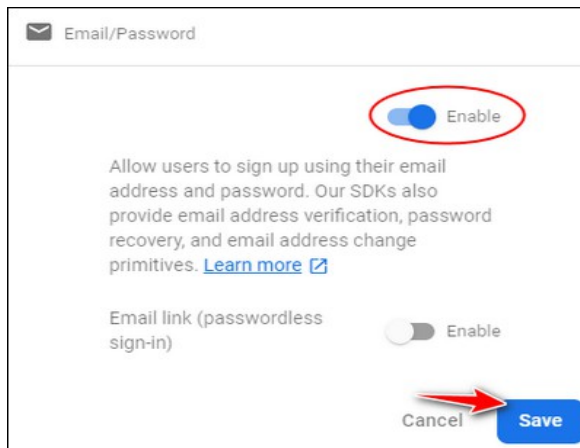
Kita akan di bawa ke halaman **Authentication Firebase**. Pada halaman ini pilih tab **Sign-in method**.



Pada bagian **Sign-in providers**, klik provider **Email/Password**:



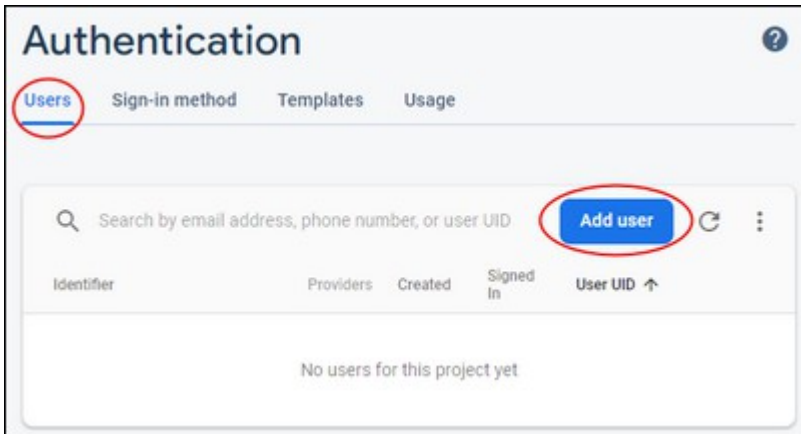
Muncul pop-up seperti gambar di bawah ini. Aktifkan bagian yang sudah ditandai lalu klik tombol **Save**.



Membuat User

Setelah berhasil mengaktifkan metode login untuk Email/Password, kita perlu meregistrasikan user baru yang akan kita gunakan sebagai latihan di buku ini untuk keperluan otentikasi atau login. Berikut adalah langkah-langkah membuat user baru di Firebase.

Masih pada halaman Authentication, pilih tab **Users** dan klik tombol **Add user**:



Ketika muncul pop-up, silahkan isi **Email** dan **Password** user yang ingin Anda masukan. Di sini saya memasukan email dan password user milik saya. Setelah itu klik tombol **Add user**:



Mencoba Login

Setelah kita berhasil mengaktifkan metode login dan berhasil menambahkan user ke Firebase. Sekarang kita akan mencoba untuk melakukan login dengan Python.

Buatlah file di dalam folder project (**D:\latihanfirebaseku**) dengan nama **latihan_01.py** dan ketik kode berikut ini:

```
from config import firebase

EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

print(user)
```

Sebelum menjalankan file tersebut, pastikan **cmd** sudah berada di folder project **D:\latihanfirebaseku** dan jangan lupa untuk mengaktifkan **virtual environment**.

Jalankan file tersebut dengan menjalankan perintah berikut pada cmd:

```
python latihan_01.py
```

Jika login berhasil, akan tercetak:

```
{
    'displayName': '',
    'email': 'yanwarsolah@gmail.com',
    'expiresIn': '3600',
    'idToken': 'eyJhbGciOiJIUzI1NiIsImtpZC.....',
```

```
'kind': 'identitytoolkit#VerifyPasswordResponse',  
'localId': 'iNqTzBcdiHMdFP3VJUVRbbab0nE2',  
'refreshToken': 'AE0u-NdZN4Dgozp0UkvD....',  
'registered': True  
}
```

Keterangan

Dari hasil yang dicetak pada cmd, yang kita perlukan adalah nilai **idToken**. Nilai ini akan kita gunakan untuk operasi-operasi yang membutuhkan token seperti operasi create, update dan delete data pada Database Firebase.

Untuk variabel **EMAIL** dan **PASSWORD** silahkan disesuaikan dengan email dan password yang sudah Anda buat di **Authentication Firebase**.

Variabel **auth** digunakan untuk menyimpan object Authentication yang kita peroleh dari Pyrebase di file **config.py**.

Variabel **user** digunakan untuk menampung nilai kembali dari method **sign_in_with_email_and_password**. Method tersebut akan melakukan login ke Firebase dengan melewati parameter email dan password.

Kesimpulan

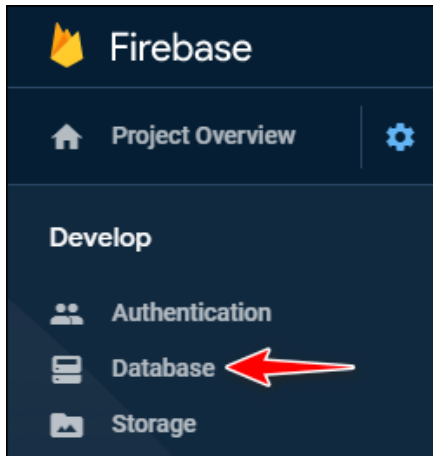
Di bab ini kita sudah mengaktifkan metode login di Firebase. Kita juga sudah belajar bagaimana menambahkan user di Firebase dan terakhir kita coba melakukan login Firebase dengan Python. Pada bab selanjutnya kita akan belajar bagaimana membuat database di Firebase.

Membuat Database Firebase

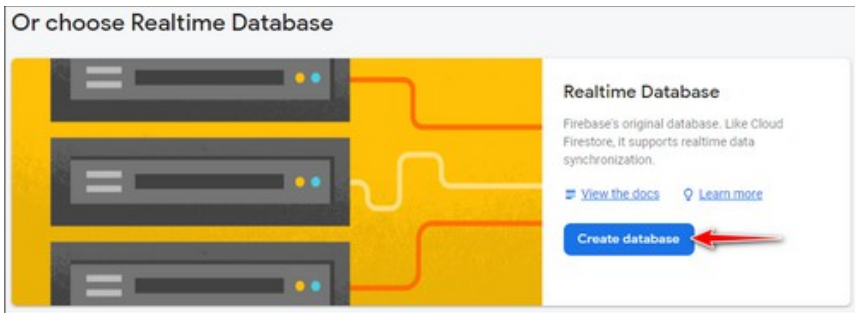
Membuat Database

Di bab ini kita akan membuat database di Firebase. Database yang akan kita buat digunakan untuk melakukan CRUD data, misalnya data produk. Untuk membuat database, ikuti langkah-langkah berikut.

Kembali ke halaman **Dashboard Firebase**. Klik menu **Database** di sebelah kiri:



Kita akan di bawa ke halaman **Database Firebase**. Pada halaman ini, scroll ke bawah, temukan bagian **Realtime Database**. Klik tombol **Create database**:

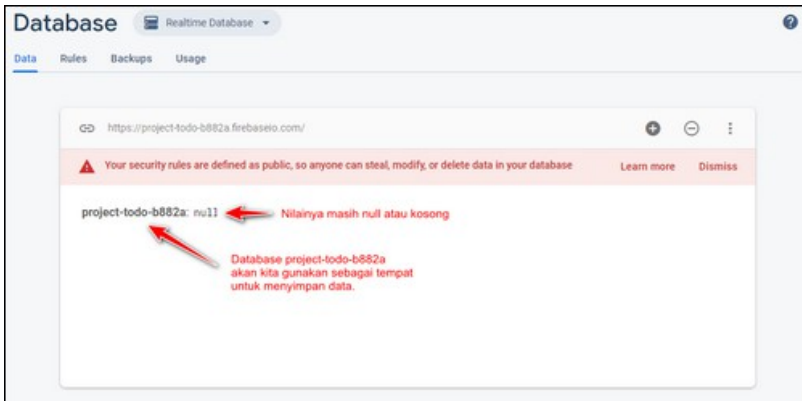


Setelah tombol tersebut di klik, Akan muncul pop-up seperti gambar di bawah ini. Untuk kebutuhan belajar, pilih saja opsi **Start in test mode** dan klik tombol **Enable**:



Ketika kita memilih opsi **Start in test mode**, kita bisa melakukan **read** dan **write** data ke database. Ini karena pengaturan **write** dan **read** pada **rules** memiliki nilai **true**.

Selanjutnya, kita akan di bawa ke halaman **Database Firebase** seperti pada gambar berikut:



Kesimpulan

Pada bab ini kita sudah berhasil membuat database di Firebase. Database di Firebase bersifat realtime. Dalam konteks sederhana, Misalnya saat kita menambahkan data ke database tersebut dengan Python, data yang kita masukan akan muncul sendiri tanpa harus melakukan reload halaman.

Pada bab selanjutnya kita akan belajar bagaimana melakukan insert data.

Insert Data pada Database Firebase

Insert Data

Pada bab ini kita akan melakukan insert data dengan Python (misalnya data produk) ke database Firebase yang sudah kita buat di bab sebelumnya. Untuk melakukan insert data, ikuti langkah-langkah berikut.

Buat file bernama **latihan_02.py** di dalam folder **D:\latihanfirebaseku** dan ketik kode berikut:

```
from config import firebase

EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
database = firebase.database()

user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

table = "produk"
data = {
    "nama": "Air Mineral",
    "harga": 5000
}

result = database.child(table).push(data, user['idToken'])

print(result)
```

Jalankan file tersebut dengan perintah:

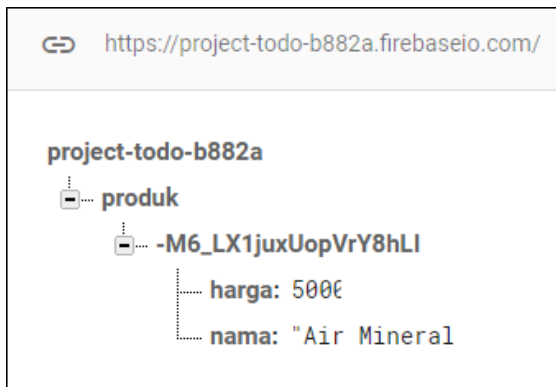
```
python latihan_02.py
```

Jika berhasil melakukan insert data, akan tercetak:

```
{'name': '-M6_LX1juxUopVrY8hLI'}
```

Nilai **-M6_LX1juxUopVrY8hLI** adalah **id** dari data yang kita insert ke database Firebase. Nilai id tersebut digenerate oleh Firebase.

Buka halaman database untuk memeriksa apakah data yang sudah kita insert sudah ada:



Keterangan

File **latihan_02.py** kita gunakan untuk latihan melakukan insert data ke database Firebase.

Variabel **table** adalah nama tabel kita. Sebenarnya ini bukan sebuah tabel melainkan nama **child** atau **node** di dalam struktur database kita di Fiebase.

Variabel **data** berisi nilai berupa **dictionary** yang ingin kita insert ke

database.

Selanjutnya pada bagian **database.child(table).push(data, user['idToken'])**, digunakan untuk melakukan insert **data** dari node **produk** dengan token yang kita peroleh dari variabel **user**. Token diperoleh karena kita berhasil melakukan login ke Firebase.

push adalah method untuk melakukan insert data berdasarkan child atau node yang kita berikan. Dengan menggunakan method ini, id pada data yang kita insert akan di generate oleh Firebase. Lalu, bagaimana jika kita ingin melakukan insert data dengan id yang kita tentukan sendiri? Kita bisa menggunakan method **set**.

Method Set

Untuk menggunakan method **set**, buka kembali file **latihan_02.py** dan ubah kodenya menjadi seperti berikut:

```
from config import firebase

EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
database = firebase.database()

user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

id_produk = "P001"
table = "produk"
data = {
    "nama": "Susu Beruang",
    "harga": 8500
}

result = database\
    .child(table)\
```

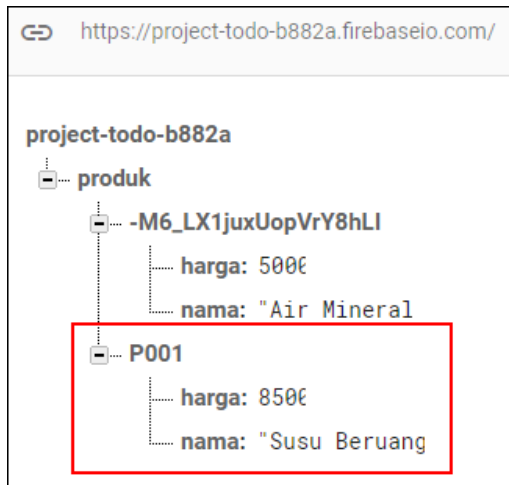
```
.child(id_produk)  
.set(data, user["idToken"])
```

```
print(result)
```

Jalankan kode berikut dan jika insert data berhasil, akan tercetak:

```
{'harga': 8500, 'nama': 'Susu Beruang'}
```

Sekarang, buka kembali halaman Database Firebase, hasilnya:



Dari hasil tersebut, kita bisa membedakan penggunaan method **push** dan method **set**. Dengan method **set**, kita bisa menentukan **id** sendiri dimana **id** tersebut sebenarnya merupakan **child** dari **produk**.

Untuk catatan tambahan, method **set** akan menimpa nilai data tersebut jika kita melakukan insert data dengan id yang sama. Anda bisa melihat referensi method ini di

<https://github.com/thisbejim/Pyrebase#save-data>.

Insert Banyak Data Sekaligus

Bagaimana jika kita memiliki data produk lebih dari satu? Bagaimana cara melakukan insert data list tersebut? Sebenarnya, untuk melakukan insert banyak data sekaligus, kita perlu mengumpulkan data tersebut ke dalam **list** dan melakukan iterasi atau loop untuk melakukan insert data ke Database Firebase. Ini hanya sebuah trik, Anda bebas menggunakan trik ini atau memiliki cara yang lebih baik.

Untuk lebih jelasnya buka kembali file **latihan_02.py** dan ubah kodenya menjadi:

```
from config import firebase

EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
database = firebase.database()

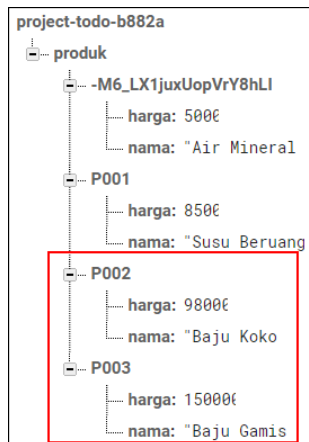
user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

table = "produk"
data_list = [
    {
        "id": "P002",
        "isi": {
            "nama": "Baju Koko",
            "harga": 98000
        }
    },
    {
        "id": "P003",
        "isi": {
```



```
        "nama": "Baju Gamis",  
        "harga": 150000  
    }  
}  
]  
  
for data in data_list:  
    database.child(table)\  
        .child(data['id'])\  
        .set(data['isi'], user['idToken'])
```

Jalankan file tersebut, hasilnya akan terlihat seperti gambar berikut:



Kesimpulan

Pada bab ini kita sudah mempelajari bagaimana melakukan insert data ke Database Firebase menggunakan method push dan set. Kita juga belajar perbedaan kedua method tersebut. Pada bab selanjutnya kita akan belajar mengambil data dari database.

Mengambil Data pada Database Firebase

Mengambil Data

Untuk mengambil data dari Database Firebase, kita bisa menggunakan method **get** dan **val**. Anda bisa membaca dokumentasinya di

<https://github.com/thisbejim/Pyrebase#retrieve-data>. Berikut langkah-langkah untuk mengambil data dari Database Firebase.

Buatlah file bernama **latihan_03.py** di dalam folder project dan ketik kode berikut:

```
from config import firebase

EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
database = firebase.database()

user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

table = "produk"

result = database.child(table).get().val()

print(result)
```

Jalankan file tersebut. Hasilnya kurang lebih akan terlihat seperti ini:

```
OrderedDict([('-M6_LX1juxUopVrY8hLI', {'harga': 5000, 'nama': 'Air Mineral'}), ('P001', {'harga': 8500, 'nama': 'Susu Beruang'})])
```

Hasilnya merupakan object **OrderedDict**. Jika kita ingin mengubah object tersebut menjadi dictionary, kita jadikan result sebagai parameter untuk method **dict()**:

```
print(dict(result))
```

Bagaimana jika kita ingin mengambil data berdasarkan id tertentu, misalnya id **P001**. Buka kembali file **latihan_03.py** dan ubah kodenya menjadi seperti berikut:

```
from config import firebase

EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
database = firebase.database()

user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

table = "produk"
id_produk = "P001"

result = database\
    .child(table)\
    .child(id_produk)\
    .get()\
    .val()

print(result)
```

Jika dijalankan, hasil yang akan tercetak adalah:

```
OrderedDict([('harga', 8500), ('nama', 'Susu Beruang')])
```

Jika dilihat dari perintah SQL, query untuk mendapatkan satu data berdasarkan id tersebut ialah:

```
SELECT * FROM produk WHERE id="P001"
```

Mengambil Id

Selain data, id data tersebut bisa kita dapatkan dengan menggunakan method **shallow** yang kita kombinasikan dengan method **get** dan **val**.

Untuk lebih jelasnya, buat file baru bernama **latihan_04.py** di dalam folder project dan ketik kode berikut:

```
from config import firebase

EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
database = firebase.database()

user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

table = "produk"

result = database\
    .child(table)\
    .shallow()\
    .get()\
    .val()

print(result)
```

Jika dijalankan, akan tercetak object **dict_keys** seperti berikut:

```
dict_keys(['-M6_LX1juxUopVrY8hLI', 'P001'])
```

Jika ingin mengubahnya ke list, kita bisa menjadikan result sebagai parameter untuk method **list()**:

```
print(list(results))
```

Kesimpulan

Pada bab ini kita sudah mempelajari bagaimana mengambil semua data, mengambil satu data berdasarkan id dan terakhir mengambil semua id dari Database Firebase. Pada bab berikutnya kita akan belajar bagaimana melakukan update data di Database Firebase.

Update Data pada Database Firebase

Update Data

Pada bab ini kita akan belajar cara melakukan update data ke Database Firebase. Untuk melakukan update, kita bisa menggunakan method **update**. Anda bisa melihat dokumentasi resminya di <https://github.com/thisbejim/Pyrebase#update>. Berikut langkah-langkah untuk melakukan update data ke Database Firebase.

Buatlah file bernama **latihan_05.py** di dalam folder project dan ketik kode berikut:

```
from config import firebase

EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
database = firebase.database()

user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

id_produk = "P001"
table = "produk"

data = {
    "harga": 3500
}

result = database\
    .child(table)\
    .child(id_produk)\
    .update(data, user['idToken'])
```

```
print(result)
```

Jalankan file tersebut. Hasilnya:

```
{'harga': 3500}
```

Jika kita buka halaman Database Firebase, harga untuk produk dengan id **P001** sudah berubah menjadi **3500**.

Kesimpulan

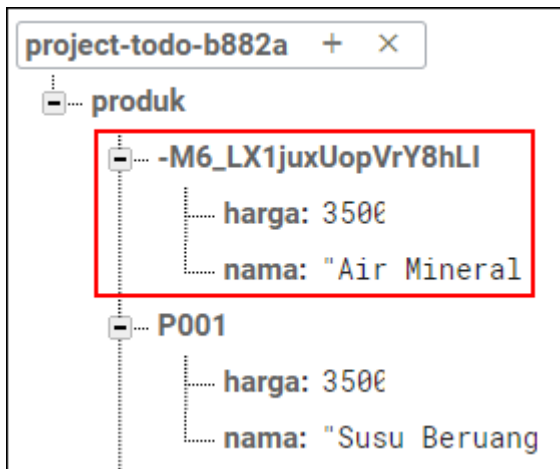
Pada bab ini kita sudah belajar bagaimana melakukan update data ke Database Firebase menggunakan Python. Pada bab selanjutnya kita akan belajar bagaimana menghapus data dari Database Firebase menggunakan Python.

Hapus Data pada Database Firebase

Hapus Data

Pada bab ini kita akan menghapus data yang ada di Database Firebase menggunakan method remove. Berikut ini adalah langkah-langkah untuk menghapus data.

Kembali ke halaman Database Firebase. Di sini saya akan menghapus data produk yang memiliki id **-M6_LX1juxUopVrY8hLI**:



Buatlah file bernama **latihan_06.py** di dalam folder project dan ketik kode berikut:

```
from config import firebase
```



```
EMAIL = "yanwarsolah@gmail.com"
PASSWORD = "AntiBucin001"

auth = firebase.auth()
database = firebase.database()

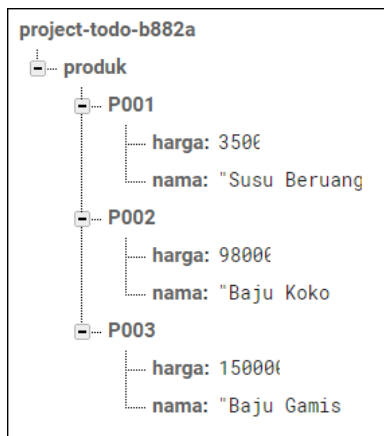
user = auth.sign_in_with_email_and_password(EMAIL, PASSWORD)

id_produk = "-M6_LX1juxUopVrY8hLI"
table = "produk"

result = database\
    .child(table)\
    .child(id_produk)\
    .remove(user['idToken'])

print(result)
```

Lihat kembali ke halaman Database Firebase, jika data produk yang memiliki id **-M6_LX1juxUopVrY8hLI** sudah tidak ada lagi, itu artinya kita berhasil menghapusnya:



Kesimpulan

Di bab ini kita sudah belajar bagaimana menghapus data dari Database Firebase. Sampai bab ini kita sudah belajar CRUD lengkap dengan Python dan Firebase. Pada bab selanjutnya kita akan mencoba studi kasus membuat aplikasi cli sederhana yang.

Studi Kasus Aplikasi cli Sederhana

Pada bab ini kita akan membuat program atau aplikasi berbasis CLI dengan Python. Pada aplikasi yang akan kita buat ini, kita bisa melakukan registrasi, login, logout, verifikasi akun, forgot password dan crud untuk data produk dengan memanfaatkan Firebase.

Aplikasinya sangat sederhana, ini hanya sebagai pelengkap agar apa yang sebelumnya kita pelajari dapat kita langsung terapkan untuk membuat aplikasi. Tujuannya adalah agar kita mengerti konsep dasar menggunakan Firebase dengan Python. Baik, langsung saja kita mulai!

Buatlah file bernama **mycli.py** di folder project. Lalu ketik kode berikut:

```
import sys
from config import firebase

def set_token(token):
    with open("dbsource.txt", "w") as f:
        f.write(token)

def get_token():
    with open("dbsource.txt", "r") as f:
        token = f.read()
    return token

def remove_token():
    with open("dbsource.txt", "w") as f:
        f.write("")
```

```
def is_login():
    if get_token():
        return True

def register(auth):
    """Tidak butuh login"""
    email = input("Masukan email: ")
    password = input("Masukan password: ")
    results = auth.create_user_with_email_and_password(email,
password)
    auth.send_email_verification(results['idToken'])
    return results['idToken']

def login(auth):
    """Tidak butuh login"""
    email = input("Masukan email: ")
    password = input("Masukan password: ")
    results = auth.sign_in_with_email_and_password(email, password)
    print(results)
    set_token(results['idToken'])

def change_password(auth):
    """Tidak butuh login"""
    email = input("Masukan email: ")
    auth.send_password_reset_email(email)
    print("Please check email to reset password")

def logout(auth):
    """Butuh login"""
    try:
        if not is_login():
            raise Exception('You need login')
        remove_token()
```

```
except Exception as e:  
    print(e)
```

```
def get_all_product(db):  
    """Butuh login"""  
    try:  
        if not is_login():  
            raise Exception('You need login')  
  
        products = db.child("product").get().each()  
        if products:  
            for data in products:  
                print(f"ID: {data.key()}")  
                print(f>Nama Produk: {data.val()['nama']}")  
                print(f>Harga Produk: {data.val()['harga']}")  
                print("=====")  
        else:  
            print("No product available")  
  
    except Exception as e:  
        print(e)
```

```
def get_next_id(db):  
    """Butuh login"""  
    try:  
        if not is_login():  
            raise Exception('You need login')  
  
        list_id = db.child("product").shallow().get().val()  
        if list_id:  
            data_length = len(list_id) + 1  
            return '{:04d}'.format(data_length)  
        return "0001"  
  
    except Exception as e:
```

```
print(e)

def create_product(db):
    """Butuh login"""
    try:
        if not is_login():
            raise Exception('You need login')

        token = get_token()
        nama = input('Nama produk: ')
        harga = input('Harga produk: ')

        data = {
            'nama': nama,
            'harga': int(harga)
        }

        id = get_next_id(db)
        print(id)
        db.child('product').child(id).set(data, token)
    except Exception as e:
        print(e)

def update_product(db):
    """Butuh login"""
    try:
        if not is_login():
            raise Exception('You need login')

        token = get_token()
        id = input("Masukan id: ")
        nama = input("Masukan nama: ")
        harga = input("Masukan harga: ")
        data = {}

        if nama:
```

```
        data['nama'] = nama

    if harga:
        data['harga'] = harga

    if data:
        db.child("product").child(id).update(data, token)

except Exception as e:
    print(e)

def remove_product(db):
    """Butuh login"""
    try:
        if not is_login():
            raise Exception("You need login")

        token = get_token()
        id = input('Masukan id: ')
        db.child("product").child(id).remove(token)
    except Exception as e:
        print(e)

def menu():
    if not is_login():
        print("No Logged In")
        print("(1) - Register")
        print("(2) - Login")
        print("(3) - Change Password")
    else:
        print("Logged In")
        print("(4) - Get All Product")
        print("(5) - Create Product")
        print("(6) - Update Product")
        print("(7) - Remove Product")
        print("(8) - Logout")
```

```
print("(X) - Exit")

auth = firebase.auth()
db = firebase.database()

while True:
    menu()
    command = input("Masukan perintah: ")
    if command == "1":
        register(auth)
    elif command == "2":
        login(auth)
    elif command == "3":
        change_password(auth)
    elif command == "4":
        get_all_product(db)
    elif command == "5":
        create_product(db)
    elif command == "6":
        update_product(db)
    elif command == "7":
        remove_product(db)
    elif command == "8":
        logout(auth)
    elif command == "X":
        sys.exit()
    else:
        print("Command not found...")
```

Keterangan

Perhatikan potongan kode **import** di paling atas. Pada kode tersebut, kita akan mengimpor package built-in **sys** dan **firebase**. **firebase** adalah object yang kita impor dari module **config**:


```
import sys
from myfirerebase import firebase
```

Perhatikan fungsi **set_token**. Fungsi ini digunakan untuk menyimpan data token ke dalam file **dbsource.txt**:

```
def set_token(token):
    with open("dbsource.txt", "w") as f:
        f.write(token)
```

Perhatikan fungsi **get_token**. Fungsi ini digunakan untuk mendapatkan data token dari file **dbsource.txt**:

```
def get_token():
    with open("dbsource.txt", "r") as f:
        token = f.read()
    return token
```

Perhatikan fungsi **remove_token**. Fungsi tersebut digunakan untuk menghapus token dari file **dbsource.txt**:

```
def remove_token():
    with open("dbsource.txt", "w") as f:
        f.write("")
```

Perhatikan fungsi **is_login()**. Fungsi tersebut digunakan untuk memeriksa status user login atau tidak.

```
def is_login():
    if get_token():
        return True
```

Perhatikan fungsi **register** berikut. Fungsi tersebut digunakan untuk mendaftarkan user ke User Firebase. Di dalam fungsi tersebut kita sediakan dua inputan berupa email dan password. Setelah user mengisi email dan password, selanjutnya kita akan mendaftarkan user tersebut ke User Firebase menggunakan method **create_user_with_email_and_password** dari object **auth**. Setelah pendaftaran berhasil kita akan mengirimkan email verifikasi akun ke email user yang mendaftar.

```
def register(auth):
    email = input("Masukan email: ")
    password = input("Masukan password: ")
    results = auth.create_user_with_email_and_password(
        email,
        password
    )
    auth.send_email_verification(results['idToken'])
    return results['idToken']
```

Perhatikan fungsi **login** berikut. Fungsi tersebut digunakan untuk mendapatkan token dari Firebase. Di fungsi tersebut terdapat dua inputan email dan password. Email dan password yang dimasukan oleh user akan digunakan untuk login dengan method **sign_in_with_email_and_password**. Setelah login berhasil, nilai tokennya akan disimpan di dalam file dbsource.txt menggunakan method **set_token**.

```
def login(auth):
    email = input("Masukan email: ")
    password = input("Masukan password: ")
    results = auth.sign_in_with_email_and_password(email, password)
    print(results)
    set_token(results['idToken'])
```

Perhatikan fungsi **change_password** berikut. Fungsi ini kita gunakan untuk mengubah password user. Di dalam fungsi tersebut sudah kita berikan inputan email. Kita akan mengirimkan link ubah password ke email tersebut menggunakan method **send_password_reset_email**.

```
def change_password(auth):  
    email = input("Masukan email: ")  
    auth.send_password_reset_email(email)  
    print("Please check email to reset password")
```

Perhatikan fungsi **logout** berikut. Fungsi logout digunakan untuk keluar dari aplikasi dengan menghapus token user dari file dbsource.txt menggunakan method **remove_token**.

```
def logout(auth):  
    try:  
        if not is_login():  
            raise Exception("You need login")  
        remove_token()  
    except Exception as e:  
        print(e)
```

Perhatikan fungsi **get_all_product**. Fungsi ini untuk mengambil data produk dari Database Firebase. Fungsi ini digunakan ketika user sudah melakukan login dengan memeriksa apakah user login atau tidak menggunakan **is_login**. Parameter yang dilewatkan adalah object database.

```
def get_all_product(db):  
    try:  
        if not is_login():  
            raise Exception("You need login")  
  
        products = db.child("product").get().each()
```

```
if products:
    for data in products:
        print(f"ID: {data.key()}")
        print(f>Nama Produk: {data.val()['nama']}")
        print(f>Harga Produk: {data.val()['harga']}")
        print("=====")
    else:
        print("No product available")

except Exception as e:
    print(e)
```

Perhatikan fungsi **get_next_id**. Fungsi ini digunakan untuk generate id. Id tersebut dapat digunakan sebagai id dari produk yang akan di-insert data ke Database Firebase. Misalnya jika produk terakhir dari Database Firebase adalah **P005** maka fungsi ini secara otomatis mengembalikan nilai **P006**.

```
def get_next_id(db):
    try:
        if not is_login():
            raise Exception("You need login")

        list_id = db.child("product").shallow().get().val()
        if list_id:
            data_length = len(list_id) + 1
            return '{:04d}'.format(data_length)
        return "0001"

    except Exception as e:
        print(e)
```

Perhatikan fungsi **create_product**. Fungsi ini kita gunakan untuk melakukan insert data product ke Database Firebase. Fungsi ini memerlukan token. Jadi fungsi ini akan dipanggil jika user sudah

melakukan login.

```
def create_product(db):
    try:
        if not is_login():
            raise Exception("You need login")

        token = get_token()
        nama = input('Nama produk: ')
        harga = input('Harga produk: ')

        data = {
            'nama': nama,
            'harga': int(harga)
        }

        id = get_next_id(db)
        print(id)
        db.child('product').child(id).set(data, token)
    except Exception as e:
        print(e)
```

Perhatikan fungsi **update_product**. Fungsi ini kita gunakan untuk mengedit data produk. Di sini kita telah menyediakan tiga inputan, yakni id, nama dan harga. Khusus untuk inputan id, inputan ini digunakan untuk mengambil id produk yang diinputkan oleh user dimana id tersebut digunakan untuk mengidentifikasi produk mana yang ingin diubah:

```
def update_product(db):
    try:
        if not is_login():
            raise Exception("You need login")

        token = get_token()
        id = input("Masukan id: ")
```

```
nama = input("Masukan nama: ")
harga = input("Masukan harga: ")
data = {}

if nama:
    data['nama'] = nama

if harga:
    data['harga'] = harga

if data:
    db.child("product").child(id).update(data, token)

except Exception as e:
    print(e)
```

Perhatikan fungsi **remove_product**. Fungsi ini kita gunakan untuk menghapus produk berdasarkan id produk yang dimasukan oleh user:

```
def remove_product(db):
    try:
        if not is_login():
            raise Exception("You need login")

        token = get_token()
        id = input('Masukan id: ')
        db.child("product").child(id).remove(token)
    except Exception as e:
        print(e)
```

Perhatikan fungsi **menu**. Fungsi ini akan menampilkan beberapa menu. Di fungsi ini kita juga sudah memisahkan menu yang mana akan tampil dan menu mana yang tidak akan tampil saat user login:

```
def menu():
```

```
if not is_login():
    print("No Logged In")
    print("(1) - Register")
    print("(2) - Login")
    print("(3) - Change Password")
else:
    print("Logged In")
    print("(4) - Get All Product")
    print("(5) - Create Product")
    print("(6) - Update Product")
    print("(7) - Remove Product")
    print("(8) - Logout")

print("(X) – Exit")
```

Perhatikan kode berikut. Kode tersebut kita gunakan untuk mendapatkan object Auth dan Database. Kita menempatkannya diluar agar jika aplikasi ini dijalankan, object tersebut hanya sekali di inialisasi dan bisa digunakan berkali-kali dalam blok fungsi yang sudah kita buat.

```
auth = firebase.auth()
db = firebase.database()
```

Yang terakhir, perhatikan blok **while** berikut. Blok ini sebagai main app yang kita buat. Di blok ini user akan menerima inputan perintah. Jika user memasukan angka 1, maka akan terpanggil fungsi untuk register, begitu juga seterusnya:

```
while True:
    # Melakukan authentication
    menu()
    command = input("Masukan perintah: ")
    if command == "1":
```

```
    register(auth)
elif command == "2":
    login(auth)
elif command == "3":
    change_password(auth)
elif command == "4":
    get_all_product(db)
elif command == "5":
    create_product(db)
elif command == "6":
    update_product(db)
elif command == "7":
    remove_product(db)
elif command == "8":
    logout(auth)
elif command == "X":
    sys.exit()
else:
    print("Command not found...")
```

Menjalankan Program

Sebelum program dijalankan, buatlah terlebih dahulu file **dbsource.txt** di dalam folder project. Jika tidak, program akan mengalami error seperti berikut karena **dbsource.txt** tidak ditemukan:

```
Traceback (most recent call last):
  File "mycli.py", line 182, in <module>
    menu()
  File "mycli.py", line 161, in menu
    if not is_login():
  File "mycli.py", line 22, in is_login
    if get_token():
  File "mycli.py", line 11, in get_token
```



```
with open("dbsource.txt", "r") as f:  
FileNotFoundError: [Errno 2] No such file or directory: 'dbsource.txt'
```

Sebagai catatan, error terjadi karena kita menggunakan mode **r** pada fungsi open file di dalam fungsi **get_token**. Mode **r** akan membaca file jika file tersebut ada. Jika tidak ada, maka akan terjadi error.

Sekarang jalankan **mycli.py** dengan perintah:

```
python mycli.py
```

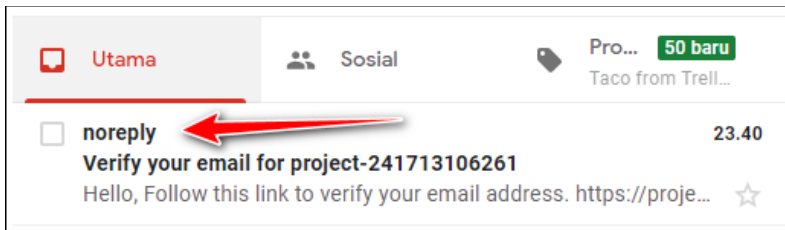
Menu yang tampil pertama kali adalah menu **Register, Login & Change Password**:

```
D:\latihanfirebaseku  
(.venv) λ python mycli.py  
No Logged In  
(1) - Register  
(2) - Login  
(3) - Change Password  
(X) - Exit  
Masukan perintah: |
```

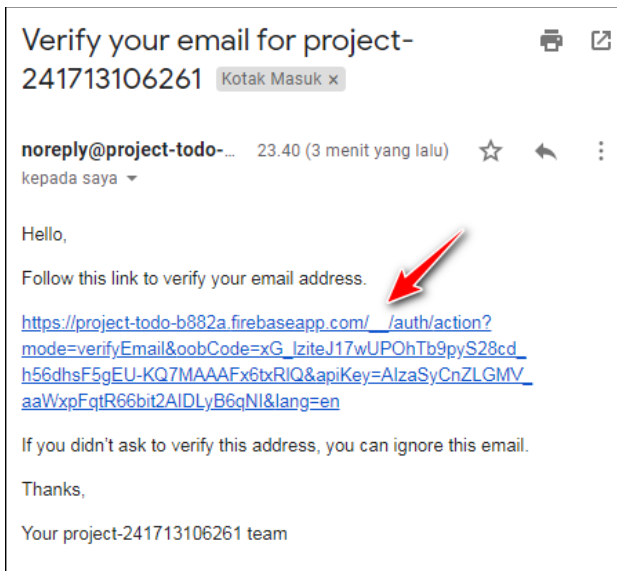
Coba ketikan **1**. Kita akan diminta untuk memasukan email dan password. Isi email dan password tersebut lalu tekan **Enter**:

```
Masukan email: berackin@gmail.com  
Masukan password: tester123
```

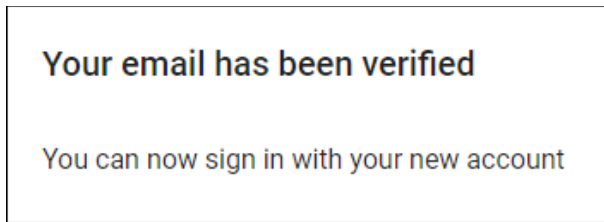
Kita coba periksa kotak masuk email. Di kotak masuk email yang kita daftarkan sudah ada email dari Firebase untuk verifikasi akun:



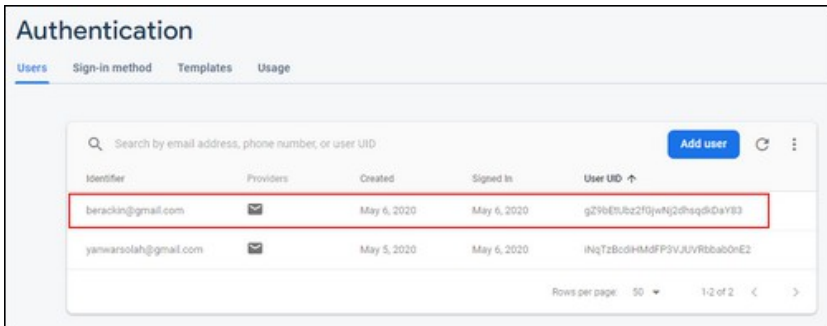
Klik email tersebut. Email tersebut berisi tautan untuk melakukan verifikasi akun:



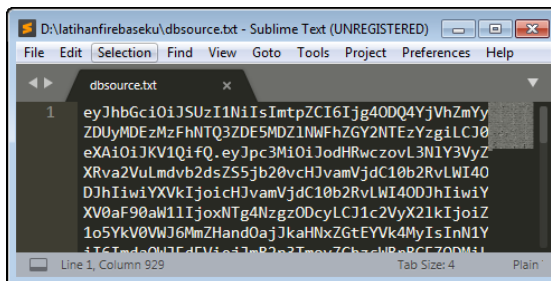
Klik tautan tersebut. Kita akan di bawa ke halaman untuk verifikasi akun. Jika verifikasi berhasil, tampilannya seperti gambar berikut:



Sekarang kembali ke halaman **Authentication Users** di Firebase. Data user yang sudah kita registrasikan sudah masuk ke Users Firebase. Dengan begitu kita bisa melakukan login dengan aplikasi mycli.py:



Sekarang coba kita login. Ketik **2** untuk login. Masukkan email dan password. Jika login berhasil, token akan tersimpan di dbsource.txt:



Untuk menu lainnya, silahkan Anda mencobanya dan bisa Anda kustom sesuai kebutuhan Anda.

Kesimpulan

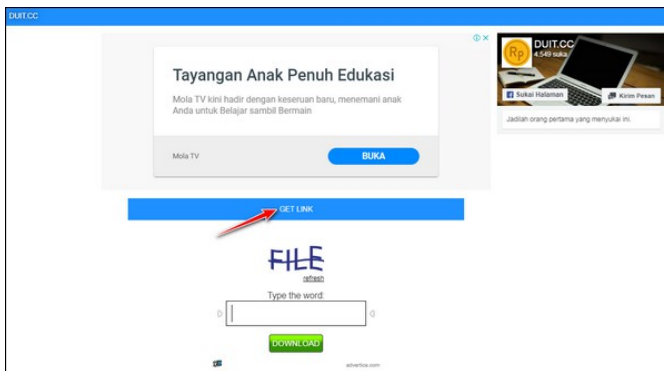
Pada bab ini kita telah belajar bagaimana membuat aplikasi dengan memanfaatkan Firebase. Kita juga belajar cara menggunakan Auth Firebase. Aplikasi tersebut masih sangat sederhana. Kepedannya, Anda bisa melakukan kustomisasi atau diimplementasikan dalam platform misalnya web. Akhir kata dari saya yang penuh kekurangan, mohon maaf jika terdapat kesalahan dalam penulisan. Terimakasih dan salam koding!

Lampiran 1 – Cara Mengunduh Source Code

Buka browser dan akses url unduhan source code berikut:

<https://duit.cc/qsKomDCS>

Pada halaman tersebut, klik tombol **GET LINK** dan tunggu hingga 15 detik:

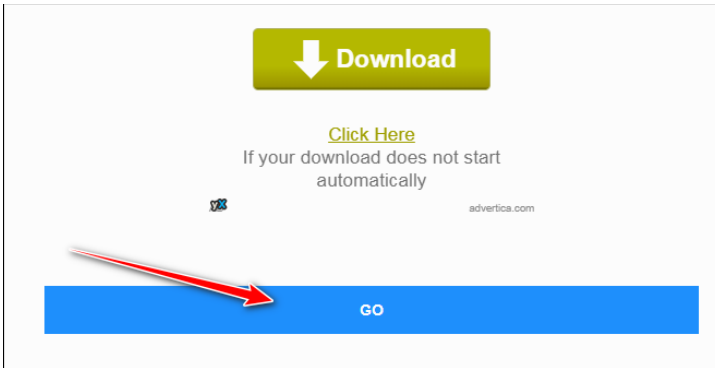


Setelah 15 detik berakhir, klik tombol **DOWNLOAD**:



Pada halaman selanjutnya, klik tombol **GO** untuk mengunduh Source Code:

Lampiran 1 – Cara Mengunduh Source Code



Jika tombol **GO** tersebut membuka tab baru setelah di klik (tab iklan), klik tombol **GO** sekali lagi (biasanya cukup sekali) untuk mendapatkan Source Code.

Catatan: jika link gagal, silahkan kirim informasi ke yanwarsolah@gmail.com.

Lampiran 2 – Cara Menjalankan Source Code

Setelah unduhan selesai, silahkan extract source code lalu masuk ke folder tersebut dan jalankan perintah berikut (direkomendasikan untuk menggunakan virtual environment):

```
pip install -r requirements.txt
```

Jika sudah, pastikan Anda memiliki Project dan Aplikasi Firebase. Isikan file **config.py** dengan konfigurasi Firebase Anda.

```
import pyrebase

config = {
    "apiKey": "", # isi dengan API Key Firebase Anda
    "authDomain": "", # isi dengan Auth Domain Firebase Anda
    "databaseURL": "", # isi dengan Database URL Firebase Anda
    "storageBucket": "", # isi dengan Storage Bucket Firebase Anda
}

firebase = pyrebase.initialize_app(config)
```

Untuk menjalankan file **latihan_01.py** sampai **latihan_06.py**, pastikan **EMAIL** dan **PASSWORD** yang telah didaftarkan di **Authentication User Firebase**, sudah di isi pada tiap file:

```
# Isi Email dan Password
# yang terdaftar di user firebase Anda
EMAIL = ""
PASSWORD = ""
```

Untuk menjalankannya, jalankan perintah berikut:

```
python namafile.py
```


Lampiran 3 – Cara Donasi

Bagi Anda yang ingin berdonasi, kami akan selalu terbuka. Kami menerima dalam bentuk apapun dan dalam jumlah berapapun. Donasi dari Anda sangat berarti untuk kami. Donasi tersebut akan kami gunakan untuk membuat ebook dan aplikasi lainnya. Sebagai ucapan terimakasih, Kami akan mengirimkan link unduhan source code tanpa iklan ke email Anda.

Bagaimana Cara Berdonasi

Jika Anda berminat, Anda bisa menghubungi kami di:

WhatsApp	+62 895 3380 94488
Email	yanwarsolah@gmail.com

Daftar Pustaka

Firebase Realtime Database dengan Android - Javan Cipta Solusi. (n.d.). Retrieved April 25, 2020, from <https://blog.javan.co.id/firebase-realtime-database-dengan-android-e8ac94dc18c8>

Firebase. (n.d.). Retrieved April 25, 2020, from <https://firebase.google.com/>

thisbejim/Pyrebase: A simple python wrapper for the Firebase API. (n.d.). Retrieved April 26, 2020, from <https://github.com/thisbejim/Pyrebase>

Structure Your Database | Firebase Realtime Database. (n.d.). Retrieved April 26, 2020, from <https://firebase.google.com/docs/database/android/structure-data>

Firebase Realtime Database. (n.d.). Retrieved April 27, 2020, from <https://firebase.google.com/docs/database/>