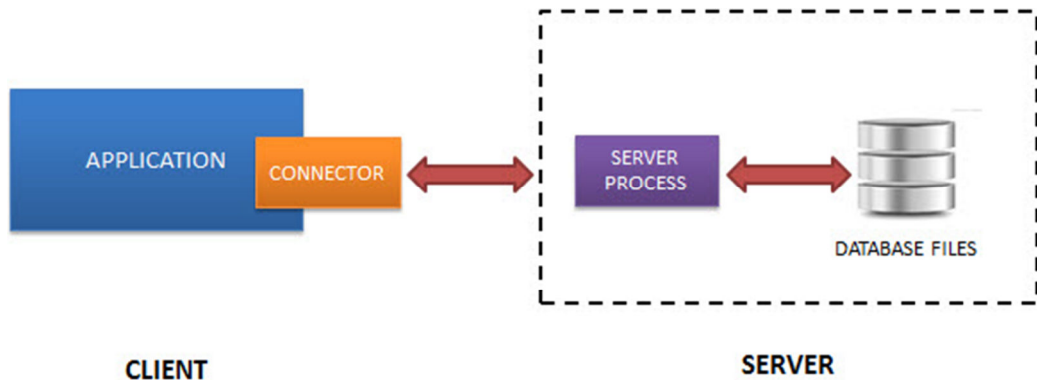
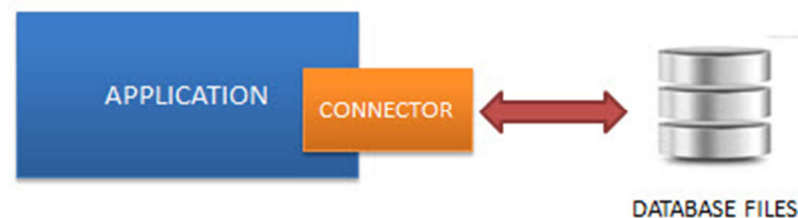


## Python dan SQLite

Umumnya, sebuah RDBMS seperti MySQL, SQLBase, dll memerlukan proses server terpisah untuk beroperasi. Aplikasi yang ingin mengakses database server menggunakan protokol TCP/IP untuk mengirim dan menerima request. Ini disebut arsitektur client/server.



SQLite tidak memerlukan server untuk menjalankannya. Database SQLite di-integrasikan dengan aplikasi yang mengakses databasae. Aplikasi berinteraksi dengan SQLite database read dan write secara langsung dengan file database yang disimpan dalam disk.



Limits are documented on the SQLite's official website. We mention a few of them:

- Maximum database size is 128 TiB or 140 TB.
- Maximum number of tables in a schema is 2147483646.
- Maximum of 64 tables can be used in a JOIN.
- Maximum number of rows in a table is 264.
- Maximum number of columns in a table/index/view is 1000 but can be increased to 32767.
- Maximum string/BLOB length is 1 billion bytes by default but this can be increased to 2<sup>31</sup>-1.
- Maximum length of an SQL statement is 1 million bytes but can be increased to 1073741824.

Untuk download, <https://www.sqlite.org/download.html>

Sebagai contoh, untuk bekerja dengan SQLite di Windows, kita download program command-line shell berikut:

**Precompiled Binaries for Windows**

[sqlite-dll-win32-x86-3290000.zip](#) (474.63 KiB) 32-bit DLL (x86) for SQLite version 3.29.0.  
(sha1: 00435a36f5e6059287cde2cebb2882669cdba3a5)

[sqlite-dll-win64-x64-3290000.zip](#) (788.61 KiB) 64-bit DLL (x64) for SQLite version 3.29.0.  
(sha1: c88204328d6ee3ff49ca0d58cbb2ee05243172c3a)

[sqlite-tools-win32-x86-3290000.zip](#) (1.71 MiB) A bundle of command-line tools for managing SQLite database files, including the [command-line shell](#) program, the [sqldiff.exe](#) program, and the [sqlite3\\_analyzer.exe](#) program.  
(sha1: f009ff42b8c22886675005e3e57c94d62bca12b3)

Selanjutnya, buat sebuah folder baru, misal C:\sqlite, kemudian ekstrak isi dari file yang di-download ke folder ini.

```
C:\sqlite>sqlite3
SQLite version 3.29.0 2019-07-10 17:32:03
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

Kita dapat menuliskan perintah **.help** dari prompt untuk melihat seluruh perintah dalam sqlite3.

**Command Line Shell untuk SQLite**

Untuk membuat sebuah database baru, misal database.db:

```
sqlite3 database.db
```

Selanjutnya, membuat sebuah tabel dan mengisinya dengan sejumlah data:

```
create table iot ( sensor varchar(10), nilai float, tanggal date, waktu time,
stamp timestamp );
insert into iot values ( "TEMP", 36.8, current_date, current_time,
current_timestamp );
```

Untuk menampilkan daftar tabel dalam database:

```
sqlite> .tables
```

Untuk keluar dari database:

```
sqlite> .exit
sqlite> .quit
```

**Python dan SQLite**

SQLite3 dapat di-integrasikan dengan Python menggunakan modul sqlite3. Kita tidak perlu meng-instalnya, karena modul ini sudah disertakan dalam instalasi Python versi 2.5.x atau yang lebih baru.

```
import sqlite3

conn = sqlite3.connect("database.db")
cursor = conn.cursor()
rows = cursor.execute("SELECT sensor, nilai, tanggal, waktu FROM iot")
```

```
for row in rows:
    print(row[0], row[1], row[2], row[3])

cursor.close()
conn.close()
```

Digabung dengan aplikasi berbasis MQTT agar dapat menyimpan data untuk topik yang di-subscribe:

```
import paho.mqtt.client as mqtt
import time
import json
import sqlite3

def on_connect(client, data, flags, rc):
    client.subscribe("nsone/smartlamp")

def on_message(client, data, msg):

    conn = sqlite3.connect("database.db")
    cursor = conn.cursor()
    print(msg.topic, msg.payload)
    mdata = msg.payload.decode()
    mdata = json.loads(mdata) # json -> dict
    sensor = mdata["sensor"]
    nilai = mdata["nilai"]
    print(sensor, nilai)
    cursor.execute("INSERT INTO iot (sensor, nilai, tanggal, waktu, stamp) VALUES
( '%s', %s, current_date, current_time, current_timestamp )" % ( sensor, nilai
))
    conn.commit()
    cursor.close()
    conn.close()

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect("broker.hivemq.com",1883)
client.loop_start()

while True:
    conn = sqlite3.connect("database.db")
    cursor = conn.cursor()

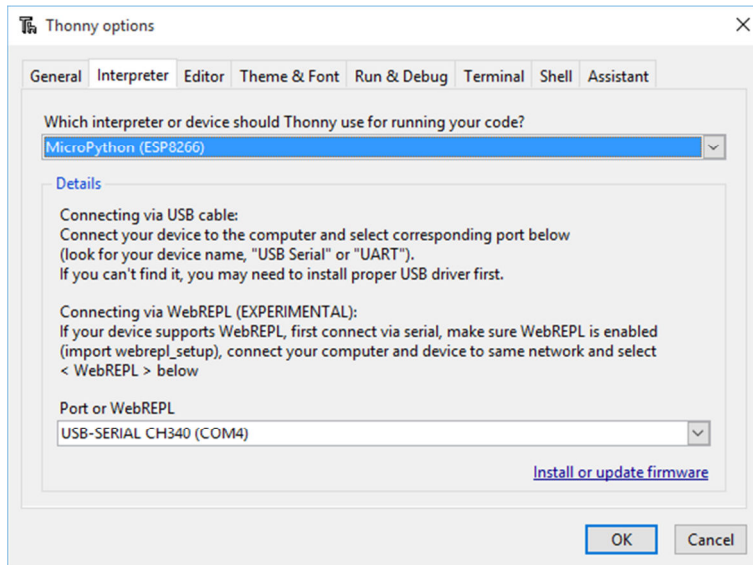
    cursor.execute("SELECT sensor, nilai, tanggal, waktu FROM iot")
    rows = cursor.fetchall()
    for row in rows:
        print(row[0], row[1], row[2], row[3])
    print("---->")
    cursor.close()
    conn.close()

    time.sleep(5)
```

## MicroPython dan Thonny

Thonny dapat digunakan untuk memrogram dan meng-install firmware micropython ke dalam chip.

Download firmware micropython dari [micropython.org](http://micropython.org), sesuai dengan board yang digunakan, kemudian install melalui menu Tools | Options... berikut:



## MicroPython Basic

Ada dua file penting yang dicermati MicroPython dalam root dari file system. File-file ini berisi kode MicroPython yang akan di-eksekusi ketika board diberikan catu daya atau di-reset.

- `/boot.py` – file ini yang pertama dijalankan saat power up / reset, dan biasanya berisi low level code yang men-setup board untuk menyelesaikan booting. Secara tipikal, kita tidak perlu merubah file ini, kecuali kita memang memerlukan kustomisasi.
- `/main.py` – jika file ini ada, dia akan dijalankan setelah `boot.py` dan biasanya berisi skrip utama yang kita ingin jalankan ketika board di power up / reset.

Bila kita ingin skrip yang kita buat, langsung berjalan ketika board diberikan catu daya atau di-reset, maka kita perlu menempatkannya dalam `main.py`.

## Hello World

Di-asumsikan, pada pin GPIO2 terhubung LED secara active LOW.

```
import time
import machine

led = machine.Pin(2, machine.Pin.OUT)

while True:
    led.value(0)
```

```
time.sleep(0.5)
led.values(1)
time.sleep(0.5)
```

## Networking

Berikut adalah sejumlah langkah untuk menjadikan ESP sebagai station:

```
import time
import network

wlan = network.WLAN(network.STA_IF)
if not wlan.isconnected():
    print('connecting to network...')
    wlan.active(True)
    wlan.connect('NEXTSYS8', 'Brg68F2Siang123')
    while not wlan.isconnected():
        print(".", end='')
        time.sleep(0.5)
print('\nnetwork config:', wlan.ifconfig())
wlan.disconnect()
wlan.active(False)
```

## MicroPython HTTP Client

```
import urequests

response = urequests.get('http://jsonplaceholder.typicode.com/albums/1')

print(response.text)

parsed = response.json()
print(parsed["userId"])
print(parsed["id"])
print(parsed["title"])
```

## MicroPython MQTT

Untuk aplikasi ini, kita memerlukan bantuan satu pustaka terkait dengan implementasi protokol mqtt, yang dapat di-download dari <https://raw.githubusercontent.com/RuiSantosdotme/ESP-MicroPython/master/code/MQTT/umqttsimple.py>

```
import network
import time
import umqttsimple as mqtt
import machine
import ubinascii
import json
from os import urandom as rnd

def rand( floor, mod=0, negative = False):

    sign = 1 if ord(rnd(1))%10 > 5 else -1
    sign = sign if negative else 1
    if mod:
        value = float('{{{}}}.format(ord(rnd(1))%floor, ord(rnd(1))%mod))
```

```
    else:
        value = int('{{')).format(ord(rnd(1))%floor))
    return sign*value

led = machine.Pin(2, machine.Pin.OUT)
led.value(1)

clientID = ubinascii.hexlify(machine.unique_id())

wlan = network.WLAN(network.STA_IF)
if not wlan.isconnected():
    print("Connecting ...")
    wlan.active(True)
    wlan.connect("NEXTSYSTEM", "Brg68SiangF2")
    while not wlan.isconnected():
        print(".", end='')
        time.sleep(0.5)
    print("\nNetwork config: ", wlan.ifconfig())

def subcallback(topic, msg):
    print(topic, msg)
    mdata = json.loads(msg.decode())
    print(mdata["sensor"], mdata["nilai"])
    if mdata["nilai"] > 37.0:
        led.value(0)
        print("led on")
    else:
        led.value(1)
        print("led off")

print("ID: ", clientID)
client = mqtt.MQTTClient(clientID, "broker.hivemq.com")
client.set_callback(subcallback)
client.connect()
client.subscribe(b"nsone/smartlamp")
print("Connected")

lastime = time.time()
while True:
    client.check_msg()
    if time.time() - lastime > 5:
        nilai = rand(50,10,False)
        mdata = {"sensor":"SUHU", "nilai":nilai}
        message = json.dumps(mdata)
        client.publish(b"nsone/smartlamp", message.encode())
        lastime = time.time()
```