**# Electrical Test**

**Diagram, schematic

Description automatically generated**

**Diagram

Description automatically generated with medium confidence**

**Diagram, schematic

Description automatically generatedDiagram, schematic

Description automatically generated**

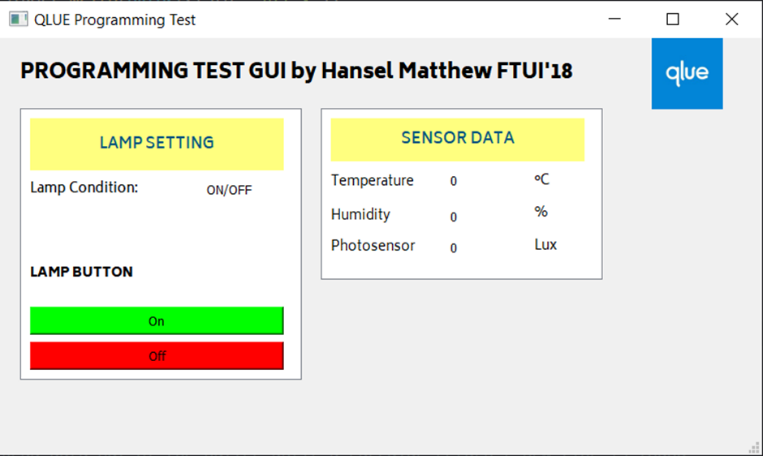
**# Programming Test**

**#MERANCANG GUI**

Tahapan awal adalah membuat rancangan GUI di QT Designer, untuk membuat sebuah GUI dapat menggunakan QT atau Tkinter namun saya menggunakan QT untuk simplifitas dan menghemat waktu. Pada QT Designer dibuat tombol – tombol, label, judul dan aset GUI lainnya. File GUI disimpan dalam file **sensorgui.ui.** Berikut merupakan gambar GUI yang didesain pada QT Designer.

Graphical user interface, application, Word

Description automatically generated



Pada GUI terdapat 3 buah komponen utama yaitu:

* Lamp conditition yang akan memberitahu kondisi lampu
* Tombol ON/OFF untuk menyalakan atau mematikan lampu
* Ketiga buah label untuk menampilkan bacaan dari sensor.

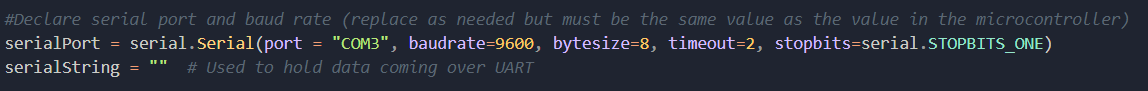
**#KODE PYTHON**

Selanjutnya adalah membuat kode python dari GUI tersebut. Kode tugas ini saya buat di file **sensorcode.py.** Kode dimulai dengan memasukkan library yang akan digunakan pada kode sumber

Text

Description automatically generated

Kemudian saya membuat variabel serialPort yang akan bertanggung jawab atas komunikasi antara kode python dengan mikrokontroller.



Selanjutnya saya membuat class dari GUI dan pada constructor class tersebut mensetting agar class tersebut terkoneksi dengan file **sensorgui.ui** yang sudah dibuat. Pada constructor juga dihubungkan method yang akan dijalankan ketika tombol pada gui ditekan

Text

Description automatically generated

Selanjutnya saya membuat dua buah method yakni satu untuk tombol ON dan satu untuk tombol OFF. Ketika tombol ON ditekan, maka kode python akan mendisplay tulisan “ON” pada GUI dan mengirimkan pesan state on ke mikrokontroller. Ketika tombol off ditekan, GUI akan mendisplay tulisan “OFF” dan mengirimkan pesan state off juga ke mikrokontroller

Text

Description automatically generated

Selanjutnya adalah membuat objek dari QT dan objek dari class yang telah dibuat sebelumnya.

Text

Description automatically generated

Setelah itu, bagian terakhir merupakan kode untuk membaca pesan dari mikrokontroller, dimana pesan tersebut dikirimkan dalam sebuah string dan dipisahkan dengan sebuah spasi. Ketika pesan diterima, pesan akan dipisah menjadi tiga buah variabel yang terpisah yaitu photosensor, humidity dan temperature, kemudian ketiga nilai tersebut akan didisplay di GUI menggunakan method .setText()

Text

Description automatically generated

Saya juga membuat file **guionly.py** dimana saya sudah comment kode yang berhubungan dengan komunikasi mikrokontroller jika ingin melihat tampilan gui tanpa terkoneksi atau mengatur mikrokontroller terlebih dahulu.

**#KODE ARDUINO**

Selanjutnya saya membuat kode pada mikrokontroller yaitu **sensor\_arduino.ino yang tersimpan pada folder sensor\_arduino.** Kode dimulai dengan memasukkan library dan mendefine variabel yang akan digunakan. Saya mengambil asumsi bahwa sensor temperatur dan humidity yang digunakan merupakan **sensor DHT11**.

Text

Description automatically generated with medium confidence

Selanjutnya saya mendeclare beberapa variable yang akan digunakan pada kode tersebut.

Graphical user interface, text

Description automatically generated

Pada bagian void setup, dilakukan pemulaian komunikasi serial dan dht serta mengatur jenis pin 11 menjadi output

Text, letter

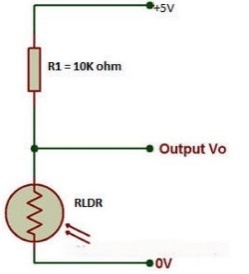
Description automatically generated

Pada bagian awal void loop, terdapat delay untuk reading interval, dan juga kode untuk membaca pesan dari python via serial. Ketika pesan diterima, jika pesan yang diterima bernilai 1, maka mikrokontroller akan menyalakan lampu, dan jika pesan yang diterima bernilai 0, maka mikrokontroller akan mematikan lampu

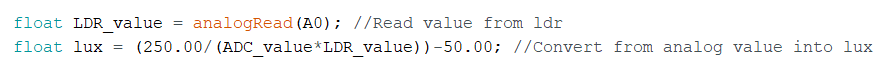
A picture containing text

Description automatically generated

Pada tugas ini, saya mengasumsikan menggunakan photosensor Light Dependent Resistor (LDR) yang mempunyai rangkaian kurang lebih seperti berikut.



Perhitungan konversi ini dapat berubah sesuai dengan tegangan, mikrokontroller yang digunakan, serta besar resistansi yang digunakan. Tegangan dari rangkaian akan dibaca oleh mikrokontroller kemudian dikonversi menjadi satuan lux.

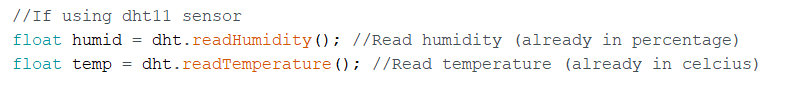


Jika ingin menggunakan sensor temperature dan humidity lainnya dan pembacaan via analogRead, saya membuat kode yang dikomen dimana nilai akan disimpan dan kemudian di scale agar nilai tersebut menjadi dalam satuan celcius dan persentase. Satuan scaler ini dapat diubah nilai perkalian atau penjumlahan maupun pengurangan sesuai dengan spesifikasi sensor pada datasheet.

Text

Description automatically generated

Dan jika menggunakan sensor DHT11, maka dapat menggunakan library dht dimana bacaan sudah dalam satuan celcius dan persentase



Bagian terakhir dari kode merupakan mengubah nilai bacaan ketiga sensor menjadi string, dan ketiga string tersebut di encode menjadi satu buah string **msg.** String **msg** tersebut kemudian dikirimkan ke kode python via serial.

Text

Description automatically generated with medium confidence