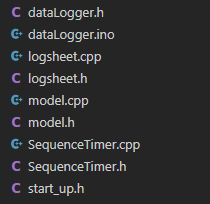
***Software Structure***

*Software* pada sistem *data logger* ini terdapat struktur dalam perancangan dan pembuatannya. Dan di dalam struktur tersebut terdapat fungsi-fungsi tersendiri, yaitu:

1. *Logsheet*:
   * Mendapatkan data dari sensor lengkap dengan hasil setelah di filter (alfa ema)
   * Tampilan lokal pada OLED
   * Rata-rata data sensor (tiap menit, jam, dan hari)
   * Menyimpan ke local storage (LittleFS)
   * Layanan data untuk web
     + Konfigurasi sensor
     + Logsheet tiap jam
     + Grafik trending data (suhu dan kelembapan)
2. *Web services*:
   * *Server side (back end)*
     + *Static files services* (*Images*, CSS dan JS)
     + HTML
     + AJAX
   * *Client side (Front end) pages / User Interface*
     + *Home*
     + *Report*
     + *Configuration*
     + *Login*
     + *Logout*
     + *About*
3. Data Model
   * Parameter sensor (*High, Low, Unit*, Alarm, alfa ema)
   * *Logsheet* (suhu dan kelembapan)
   * *User* (*username, password*, email, *level*)
4. Ketepatan waktu untuk menyinkronkan pengambilan sampel
5. *Start up*
   * Menampilkan logo ketika *start up*
   * Menampilkan *welcome messages*

Adapun fungsi tersebut di atas, dibagi menjadi beberapa bagian *software* / program dengan memisahkan sesuai dengan fungsinya (prinsip OOP). Pada gambar dibawah ini terlihat bahwa terdapat 5 bagian program (1 *main* program dan 4 *class*), datalogger.ino sebagai *main* program dan yang lainnya sebagai *class*. Ke 4 *class* ini sudah tercantum pada *file-file* dibawah ini yaitu logsheet, model, SequenceTimer dan start\_up. File-file ini merupakan *library*, jadi bisa disimpulkan bahwa class ini sudah terwakili oleh *library*. *File* cpp merupakan detail semua fungsi yang didefinisikan pada *file* h.



* File datalogger

Sebagai program utama dataLogger.ino sebagai pintu masuk (*entry point*) yang di dalamnya terdapat struktur dasar dari program ESP8266 (Wemos D1 mini). Sedangkan dataLogger.h berisi seluruh definisi *global variables* dan konstanta dalam fungsi yang digunakan dalam keseluruhan program ESP8266 ini. Struktur program Arduino minimal tampak seperti dalam cuplikan program dari IDE Arduino berikut ini:

void setup() {

// put your setup code here, to run once:

}

void loop() {

// put your main code here, to run repeatedly:

}

Struktur minimal pada program Arduino (yang dipakai oleh ESP8266 ini) setidaknya memuat fungsi void setup() dan void loop(). Pada file dataLogger ini terdapat beberapa urutan dalam pembuatannya, yaitu:

1. *Importing libraries*

Diperlukan beberapa *libraries* untuk mengembangkan *software* IoT *data logger* ini, baik yang langsung di-*import* melalui datalogger.ino ataupun saat pembuatan *class* (*Logsheet*, Model, *StartUp, SequenceTimer*). *Libraries* yang dimaksud adalah untuk mempermudah dalam pengembangan *software* sehingga tidak harus membuat program sendiri mulai dari awal (*scratch*). *Libraries* ini dibagi sesuai dengan kepentinganya,yaitu:

* *Core* arduino – ESP8266
  + #include <Arduino.h>
* *Libraries* untuk WIFI and WEB
  + #include <ESP8266WiFi.h>
  + #include <ESPAsyncTCP.h>
  + #include <ESPAsyncWebServer.h>
  + #include <ESP8266WiFiMulti.h>
* *Library* dibuat sendiri:
  + #include "dataLogger.h"
  + #include "SequenceTimer.h"
  + #include "logsheet.h"
  + #include "start\_up.h"
* *Libraries* untuk pembuatan *Logsheet*:
  + #include <Arduino.h>
  + #include <ArduinoJson.h>
  + #include <DHT.h>
  + #include <Wire.h> // Include Wire if you're using I2C
  + #include <SPI.h>
  + #include <Adafruit\_GFX.h>
  + #include <Adafruit\_SSD1306.h>
  + #include <LittleFS.h>
  + #include "model.h"
* *Library* untuk pembuatan *SequenceTimer*:
  + #include <Arduino.h>
* *Libraries* untuk pembuatan *StartUp*:
  + #include <Arduino.h>
  + #include <Wire.h> // Include Wire if you're using I2C
  + #include <SPI.h>
  + #include <Adafruit\_SSD1306.h>

Bisa dilihat terdapat perintah #include yang digunakan untuk memanggil *library* masuk ke dalam program. Terdapat perintah #include yang dilanjutkan dengan tanda kurung sudut (< >) dan tanda petik dua (""). Untuk tanda kurung sudut (< >) ini merupakan *library* yang telah di-*loading* atau diinstall dari *library manager* Arduino IDE dan untuk tanda petik dua ("") merupakan library yang dibuat sendiri dan berada dalam satu folder dengan dataLogger.ino.

