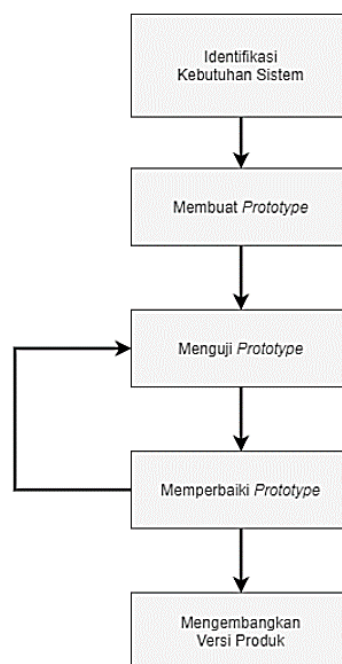


BAB III METODOLOGI PENELITIAN

Metodologi pada penelitian ini berupa tahapan atau langkah-langkah yang akan dilakukan dalam penelitian. Dalam tiap tahap penelitian disusun secara sistematis sehingga dapat memudahkan proses pengembangan sistem.

3.1. Metode Pengembangan Perangkat Lunak

Metode yang digunakan dalam penelitian ini yaitu *prototype*. *Prototype* adalah tahap awal dari sistem perangkat lunak yang digunakan untuk mendemonstrasikan konsep, eksperimen desain, dan menemukan lebih banyak masalah dan kemungkinan solusi [34]. *Prototype* merupakan suatu model atau metode yang dilakukan dalam pengembangan sistem perangkat lunak menggunakan pendekatan untuk membuat sesuatu dengan cepat dan bertahap sehingga dapat dievaluasi oleh pengguna [35].



Gambar 3. 1 Model *Prototype*

Sumber : Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) 2021

3.2. Subjek dan Objek Penelitian

Pada proses perancangan sistem jarak mata terhadap komputer, peneliti membutuhkan sumber referensi sebagai acuan pembuatan sistem. Kemudian pada penelitian dibutuhkan parameter sebagai objek atau sasaran penelitian. Berikut data subjek dan objek penelitian sensor jarak mata terhadap komputer dengan NodeMCU.

3.1.1. Subjek Penelitian

Subjek penelitian ini adalah anak usia 5 – 8 tahun yang berhadapan pada monitor komputer dengan berbagai kajian literatur dengan sumber dan referensi yang sudah jelas dan dapat dipertanggungjawabkan kebenarannya.

3.1.2. Objek Penelitian

Objek yang diamati pada rancang alat ini yaitu durasi / limit yang dibatasi yaitu maksimal 1 jam untuk anak berusia 2-8 tahun, jarak mata ke monitor yaitu sekitar 55-80 cm dan pencahayaan di lingkungan pengguna komputer yaitu 100-200 lux.

3.3. Alat dan Bahan

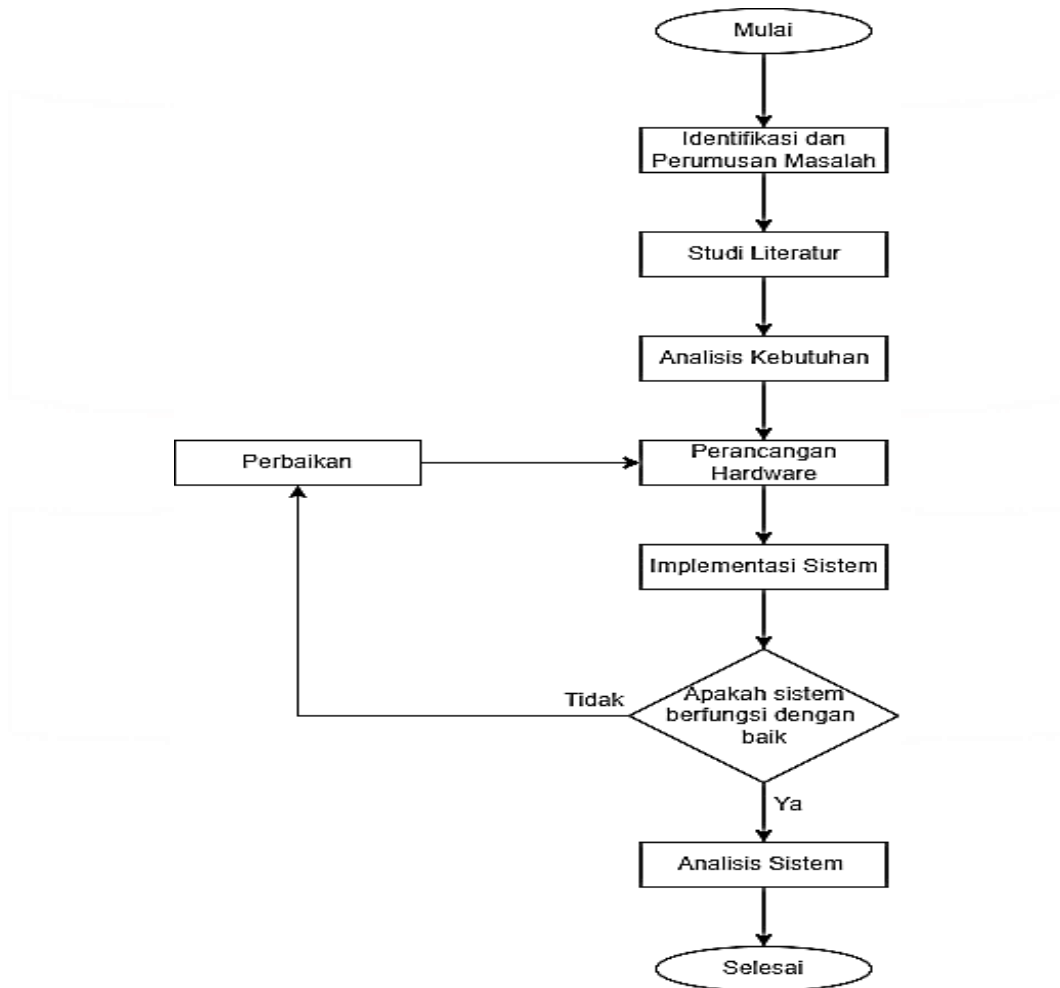
Perancangan alat sensor jarak ini membutuhkan beberapa komponen untuk penunjang pembuatan sistem, alat dan bahan sebagai berikut:

1. NodeMCU ESP8266
2. Sensor *Ultrasonic* HC-SR04
3. Sensor BH1750
4. Sensor kamera ESP32 cam
5. *Downloader* ESP32
6. *Blynk App*
7. Kabel Jumper
8. Laptop

3.4. Diagram Alir Penelitian

Pada penelitian ini dirancang sebuah diagram alir penelitian guna mempermudah proses pelaksanaan penelitian. Diagram ini merupakan diagram alir

penelitian yang bertujuan untuk memfokuskan dan memberikan alur penelitian agar terarah dan berjalan sesuai dengan rencana.

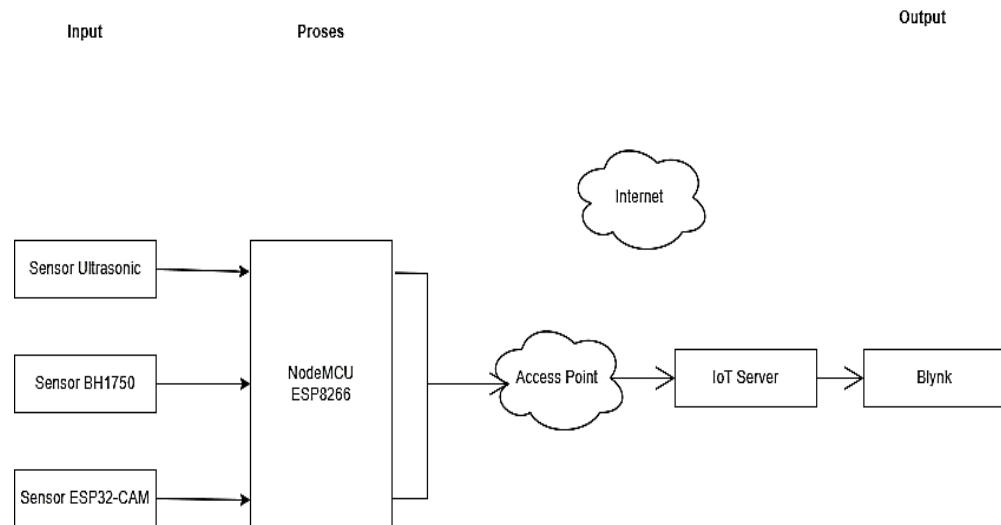


Gambar 3. 2 Diagram Alir Penelitian

3.5. Perancangan Hardware

3.4.1. Diagram Blok

Diagram blok merupakan diagram proses pada sistem yang menggambarkan ringkasan dari hubungan sistem yang dibuat. Bentuk diagram tersusun dari gabungan blok-blok fungsional masing-masing komponen sistem dengan memperhatikan aliran sinyalnya. Perancangan diagram blok dilakukan untuk mengetahui prinsip kerja dari rangkaian elektronika yang akan dibangun.



Gambar 3. 3 Diagram blok

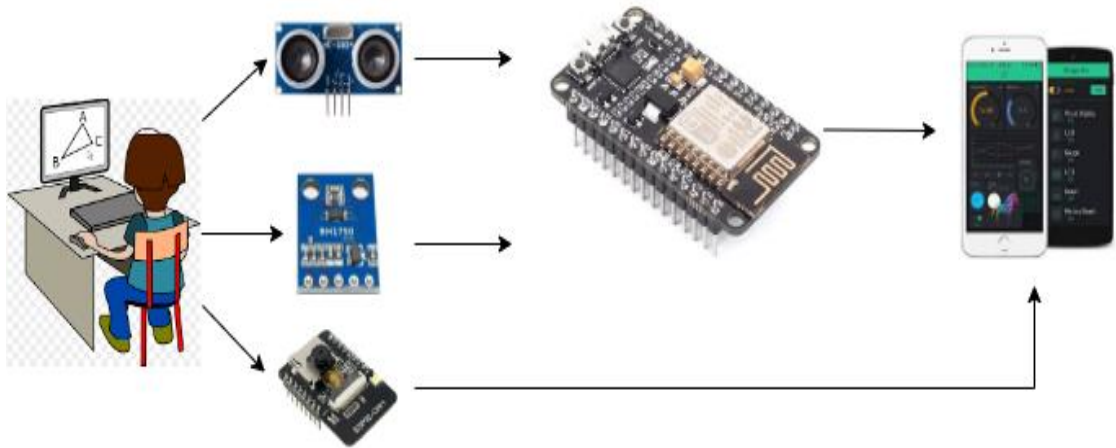
Fungsi tiap blok pada diagram:

- Blok sensor *ultrasonic* untuk mengetahui jarak pengguna dengan objek
- Blok sensor BH1750 untuk mengetahui intensitas cahaya pengguna
- Blok sensor ESP32 cam untuk mengambil gambar sekitar pengguna
- Blok NodeMCU ESP8266 sebagai modul wifi yang berperan untuk mengirim data yang diterima
- Blok Internet yaitu jaringan yang saling terkoneksi
- Blok Blynk sebagai *platform IoT* yang menampilkan jarak dan intensitas cahaya

Pada proses *input* kondisi dari diperoleh dari hasil deteksi dari sensor *ultrasonic* yang sudah terprogram sesuai dengan kebutuhan. Terdapat batasan jarak yang sudah ditentukan sesuai standar kesehatan penggunaan laptop yaitu 50 cm [15]. Setelah melakukan identifikasi jarak, sistem akan membaca intensitas cahaya di sekitar pengguna. Deteksi intensitas akan memberi informasi besaran intensitas yang diterima. Pada penelitian ini hanya menampilkan intensitasnya yang di tangkap. Namun peneliti sarankan untuk dapat memenuhi perhitungan fluks cahaya yang diukur dalam lumen sebagai ukuran intensitas pencahayaan suatu permukaan [13]. Pada penelitian ini pencahayaan minimal yang digunakan berkisar 0 lux, sedangkan pencahayaan maximal yang digunakan berkisar 300

lux [15]. Kemudian terdapat deteksi visual pengguna dengan ESP32 CAM untuk monitoring selama proses penggunaan komputer.

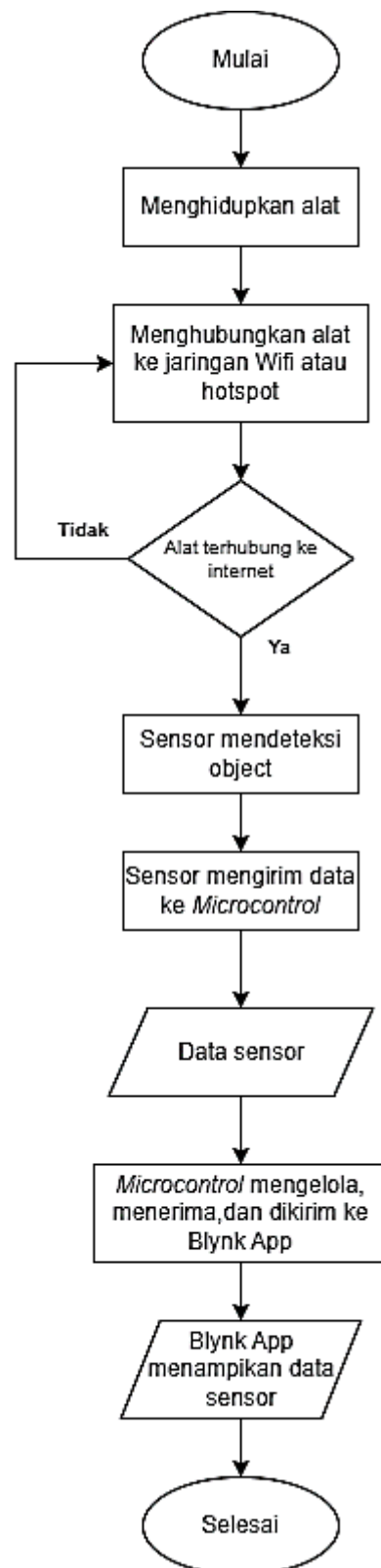
3.4.2. Perancangan Arsitektur Sistem



Gambar 3. 4 Arsitektur Monitoring Jarak pada Komputer

Pada proses perancangan arsitektur sistem menjelaskan alur kerja sistem dari awal proses *input* hingga *output* yang akan diperoleh oleh pengguna. Pertama sensor *ultrasonic*, sensor intensitas cahaya, dan sensor kamera. Sensor yang sudah terpasang dan diprogram akan melakukan deteksi keadaan pengguna. Data yang diperoleh dari pengguna kemudian dikirim ke *microcontroller* yang berperan sebagai pusat kendali dan pusat informasi. Kemudian sistem akan melakukan deteksi jarak, intensitas cahaya dan menangkap keadaan visual pengguna pada waktu yang sudah di pengaturan sebelumnya. Berdasarkan deteksi tersebut kemudian sistem akan melakukan monitoring yang *output*-nya dapat dilihat dengan *Blynk App*.

Adapun skema perancangan dalam menyusun komponen yang sudah tersedia dan dirangkai sesuai dengan desain *prototype* yang sudah dirancang sebelumnya. Skema rangkaian yang dibuat terdiri dari 2 *microcontroller* yaitu ESP8266 dan ESP32. Pada ESP8266 menggunakan NodeMCU.

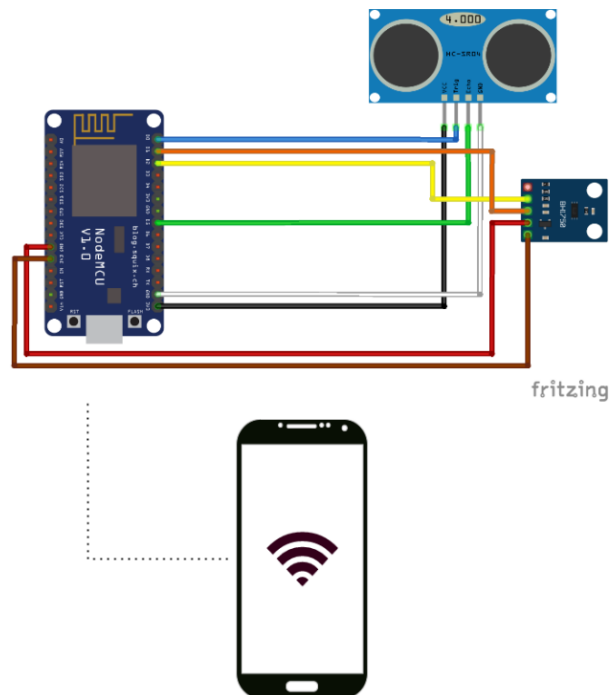


Gambar 3. 5 Flowchart sistem

Flowchart menjelaskan alur dari sistem monitoring monitoring anak dalam penggunaan komputer. Perancangan flowchart sistem dapat membantu untuk mengetahui bagaimana alur dari kerja alat yang dibuat. Proses diawali dengan mulai, kemudian melakukan hubungan dengan jaringan internet via hotspot atau WiFi. Ketika sudah berhasil terhubung maka seluruh sistem akan menjalankan fungsi yang sudah ditentukan pada *microcontroller*. Pada sistem akan menampilkan RTC yang berfungsi untuk *timer* pengguna dalam menggunakan komputer. Pengguna perlu memperhatikan *LED* pada *timer* untuk batas penggunaan komputer. Pengaturan waktu pada RTC dilakukan secara manual dengan *input* hari, jam, menit dan detik. Pada sensor HC-SR04 diatur oleh peneliti dengan indeks jarak maksimal adalah 110cm. ketika pengguna berada pada jarak kurang dari 55 cm maka sistem akan memberikan peringatan dengan indikator *LED*. Kemudian sistem akan melakukan deteksi intensitas cahaya dengan sensor BH1750. Sistem akan memberikan nilai intensitas cahaya dalam besaran lux meter. Nilai intensitas yang diatur pada sistem yaitu 100-300 lux. Setelah proses tersebut akan tampil nilai dari intensitas cahaya dan jarak pengguna pada *Blynk App*. Pada intensitas cahaya ≤ 100 maka *LED* bahaya akan menyala, jika ≥ 101 dan ≤ 200 maka *LED* aman akan nyala, jika ≥ 201 maka *LED* cukup akan menyala. Kemudian terdapat sensor kamera yang dapat digunakan oleh orang tua dalam melakukan monitoring pengguna. Gambar visual tersebut di tangkap dengan sensor dan ditampilkan pada *Blynk App*.

3.4.3. Perancangan Alat

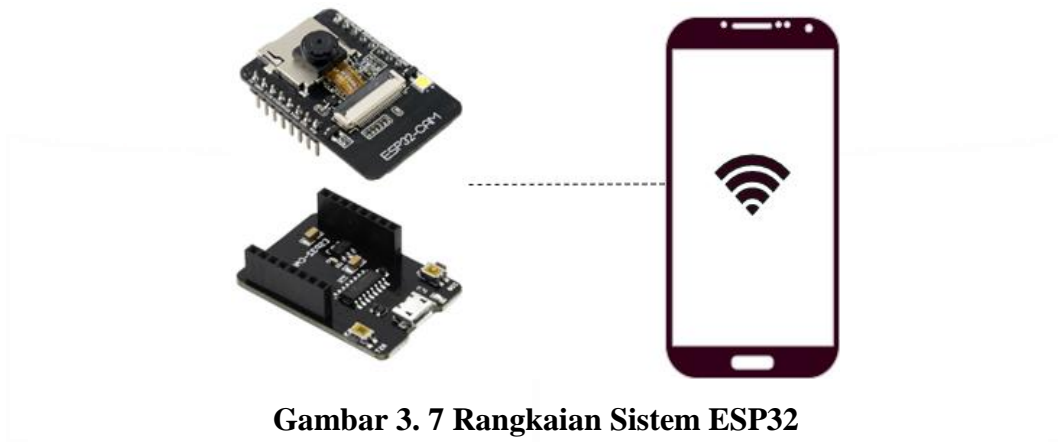
Pada perancangan alat dibangun dengan menjadi 2 rangkaian dan *microcontroller* yang berbeda. Pada rangkaian I yaitu ESP8266 dengan *NodeMCU* dan rangkaian II yaitu *ESP32* yaitu dengan *ESP32 Cam*.



Gambar 3. 6 Rangkaian Sistem ESP8266

Tabel 3. 1 Keterangan rancangan alat Gambar 3.5

No.	Nama Perangkat	Keterangan
1	NodeMCU	Berfungsi sebagai <i>microcontroller</i> pada rangkaian sistem
2	HC-SR04	Suatu sensor yang berfungsi mengukur jarak suatu benda yang terletak di depan sensor
3	BH1750	Suatu sensor pengukur intensitas cahaya yang mempunyai satuan lux



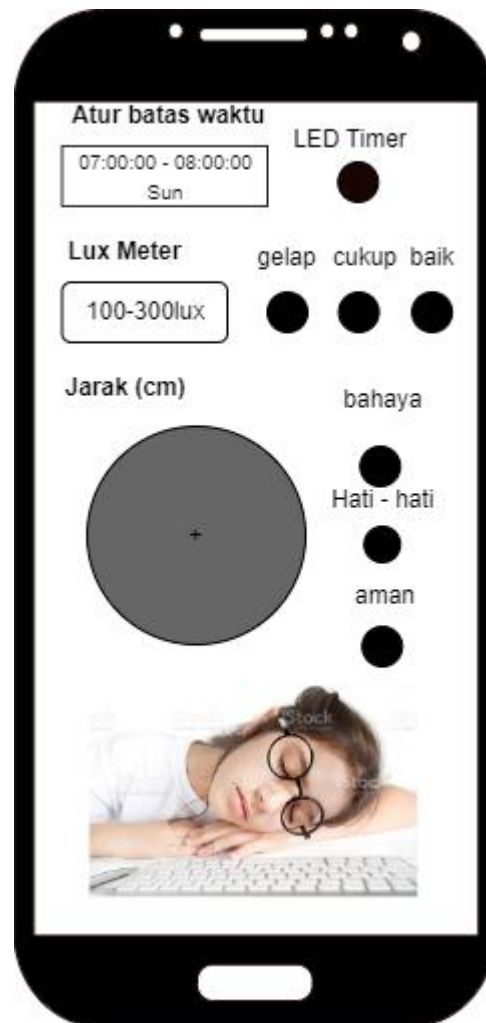
Gambar 3. 7 Rangkaian Sistem ESP32

Tabel 3. 2 Keterangan rangkaian sistem ESP32

No.	Nama Perangkat	Keterangan
1	ESP32 CAM	Berfungsi sebagai <i>microcontroller</i> pada sensor kamera
2	DOWNLOADER ESP32	Merupakan port yang berfungsi untuk menghubungkan modul kamera ke perangkat personal komputer atau laptop

3.6. Perancangan Sistem Monitoring

Perancangan sistem monitoring yang dibangun terhubung pada ESP8266 dan untuk *interface* monitoringnya melalui *Blynk App*. Berikut merupakan rancangan monitoring melalui *Blynk App*.

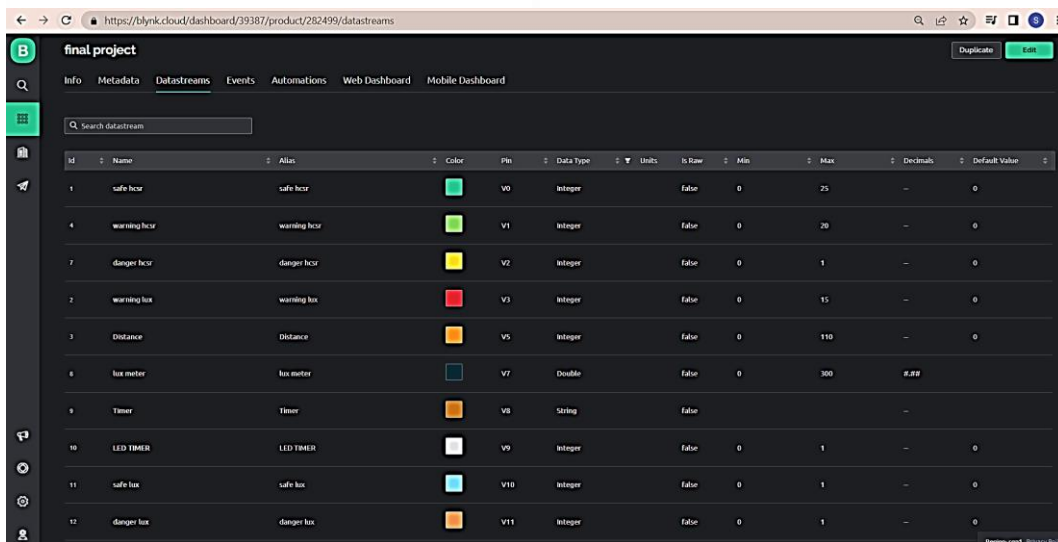


Gambar 3. 8 Mockup Blynk App NodeMCU

Blynk App menjadi *platform IoT* peneliti untuk menampilkan *output* dari sistem yang sudah dirancang. *Output* yang pertama yaitu *software* dari *RTC* yang menjadi tanda batas waktu yang ditandai dengan *LED*. *Output* selanjutnya mengenai sensor intensitas cahaya melalui *BH1750* yang menghitung besaran lux yang diterima. Pada aplikasi akan menampilkan besarannya saja yaitu rentang 100-300 lux. Kemudian *output* *HC-SR04* yaitu data jarak yang ditangkap oleh sensor. Sistem terdapat sistem jarak aman dan bahaya pada pengguna. Kemudian pada tampilan terakhir terdapat *output* dari sensor *ESP32 CAM*. *Output* tersebut bertujuan untuk membantu dalam memantau anak secara langsung. Dengan sensor ini orang tua dapat memantau kegiatan anak tanpa mengganggu konsentrasi anak dengan melakukan kontak langsung.

3.4.4. Perancangan *output* sistem monitoring pada *Blynk App*

Untuk melakukan proses monitoring diperlukan pengaturan *template* pada *Blynk App*. Komunikasi perangkat dengan membuat *device info* dan *coding*. *Device info* diperlukan untuk mendapatkan *auth token*, *template id* dan membuat *datastream* dan *link*. Sedangkan *coding* diperlukan untuk melakukan *input device info*. Penelitian ini menggunakan *microcontroller* ESP32 dan ESP8266.

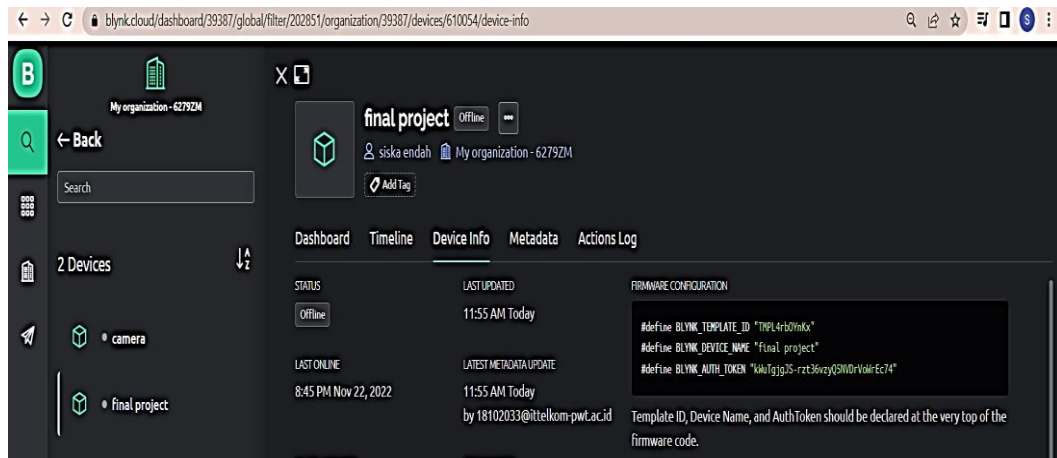


The screenshot shows the Blynk Cloud dashboard for a project named 'final project'. The 'Datastreams' tab is active, displaying a table with 12 datastreams. Each row includes an ID, Name, Alias, Color, Pin, Data Type, Units, Is Raw, Min, Max, Decimals, and Default Value.

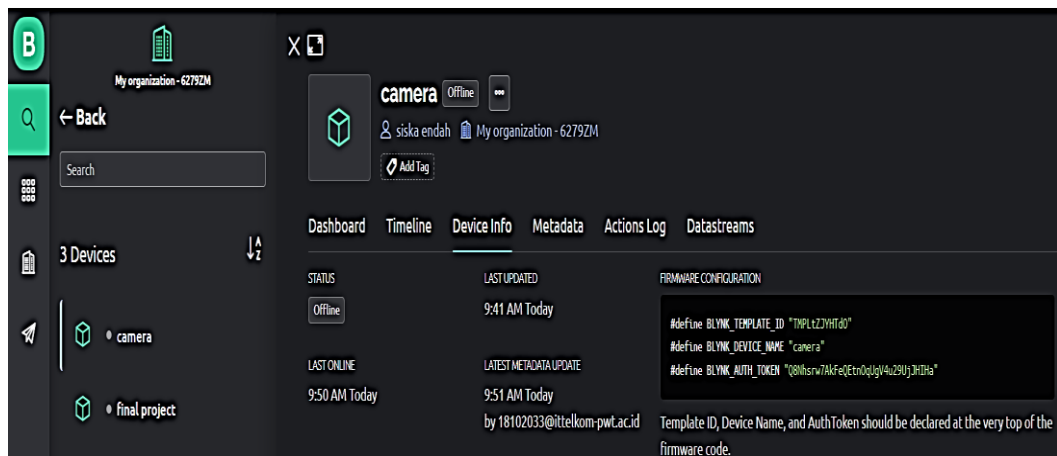
ID	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	Decimals	Default Value
1	safe hcr	safe hcr	Green	V0	Integer		False	0	25	-	0
4	warning hcr	warning hcr	Light Green	V1	Integer		False	0	20	-	0
7	danger hcr	danger hcr	Yellow	V2	Integer		False	0	1	-	0
2	warning lux	warning lux	Red	V3	Integer		False	0	15	-	0
3	Distance	Distance	Orange	V5	Integer		False	0	110	-	0
8	lux meter	lux meter	Blue	V7	Double		False	0	300	###	
9	Timer	Timer	Orange	V8	String		False			-	
10	LED TIMER	LED TIMER	White	V9	Integer		False	0	1	-	0
11	safe lux	safe lux	Cyan	V10	Integer		False	0	1	-	0
12	danger lux	danger lux	Orange	V11	Integer		False	0	1	-	0

Gambar 3. 9 Penambahan *Datastream esp8266*

Penambahan *datastream* dilakukan hanya pada esp8266 pada setiap koneksi pin yang terhubung pada NodeMCU. Sedangkan pada ESP32 semua pin otomatis terhubung pada *downloader* sehingga tidak perlu penambahan *datastream*. Setelah selesai melakukan penambahan *datastream* dapat melakukan *add device* untuk menghubungkan perangkat *mobile*. Setelah berhasil melakukan *add device* akan muncul *auth token template*.

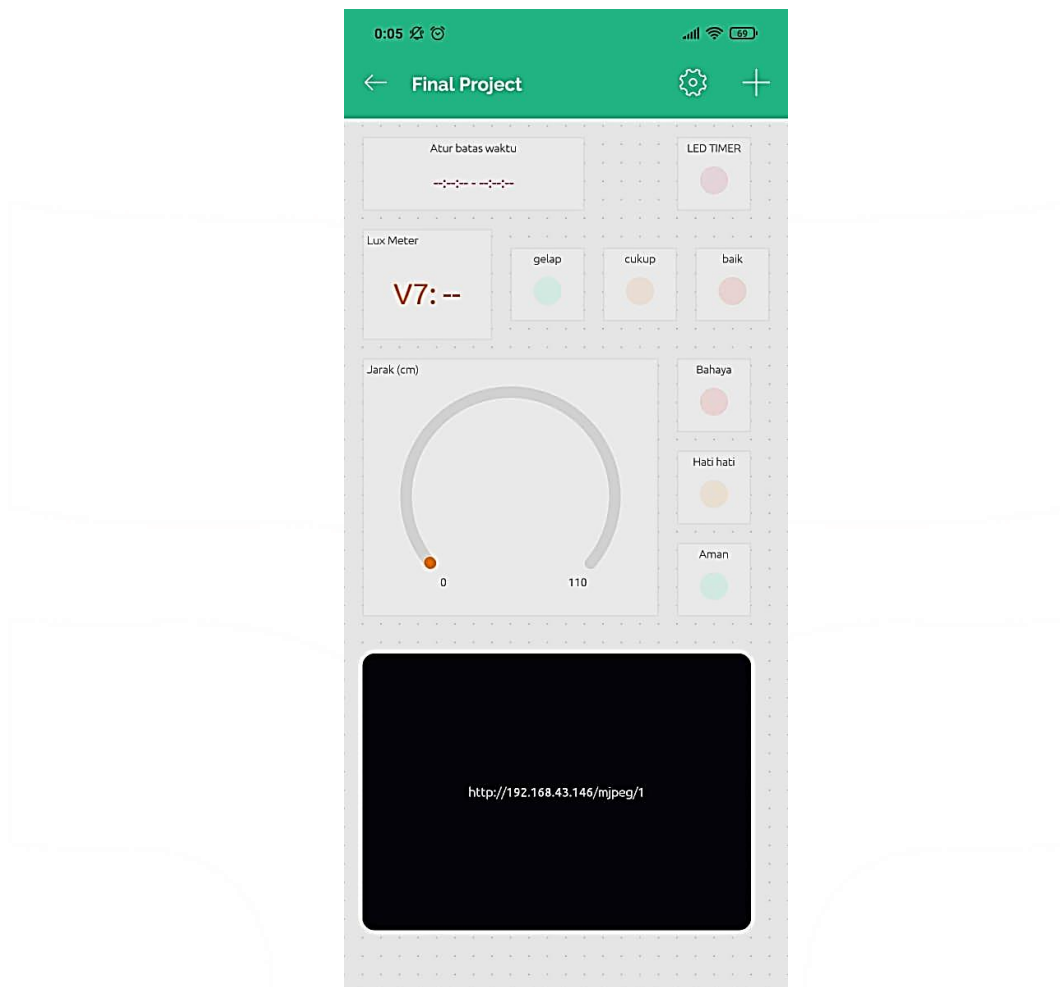


Gambar 3. 10 Informasi auth token esp8266



Gambar 3. 11 Informasi auth token esp32

Kemudian diperlukan kembali pengaturan pada aplikasi *Blynk App*. Pada pengaturan ini melakukan penambahan fungsi melalui menu *Widget Box*. Pengaturan tampilan monitoring terletak pada tahap ini. Setelah menambahkan fungsi yang diperlukan disesuaikan dengan datastream yang sudah dibuat sebelumnya.



Gambar 3. 12 Pengaturan pada *Blynk App*

3.5.1. Koneksi Antar Perangkat

Script yang dibuat terdapat `char ssid[] = "";` dan `char pass[] = "";` yang berfungsi untuk melakukan *input* nama *hotspot* dan *password* sesuai dengan *access point* (*hotspot*). Pada `ssid` dan `pass` merupakan hasil inisiasi dari fungsi *char*.

```
char ssid[] = "xixixi"; //nama WiFi/personal hotspot
char pass[] = "faeja567"; //password
```

Gambar 3. 13 Script untuk koneksi ke *access point*

Sedangkan pada `BLYNK_AUTH_TOKEN;` merupakan inisiasi dari `#define BLYNK_AUTH_TOKEN.`

```

#define BLYNK_AUTH_TOKEN "kWuTgJgJS-rzt36vzyQ5NVDrVoWrEc74" //input auth token milih esp8266
#define BLYNK_PRINT Serial
#define TRIGGER D0
#define ECHO D5
#define APP_DEBUG

//membuat inisiasi untuk tiap fungsi
BH1750 lightMeter; //inisiasi nilai BH1750 menjadi lightMeter
float lux; //inisiasi nilai float menjadi lux
BlynkTimer timer; //inisiasi BlynkTimer menjadi timer

char auth[] = BLYNK_AUTH_TOKEN; //inisiasi nilai BLYNK_AUTH_TOKEN menjadi char auth[]

```

Gambar 3. 14 Auth token untuk NodeMCU ESP8266

```

#define BLYNK_AUTH_TOKEN "Q8Nhsrw7AkFeQEtnOqUgV4u29UjJHIHa" //input auth token milik ESP32
#define BLYNK_PRINT Serial
#define APP_DEBUG
#include "camera_pins.h"
char auth[] = BLYNK_AUTH_TOKEN;

```

Gambar 3. 15 Auth token untuk ESP32

Kemudian dilakukan fungsi char untuk menginisiasi fungsi tersebut menjadi ssid dan pass. Kemudian terdapat script `Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);` berfungsi untuk melakukan koneksi ke *Blynk App*. *Auth* merupakan token template yang sudah diinisiasi sebelumnya.

```

Blynk.begin(auth, ssid, pass, "blynk.cloud", 80); //Blynk akan berjalan ketika auth, ssid, password inputan sync
timer.setInterval(10000L, checkTime);
Serial.begin (9600);

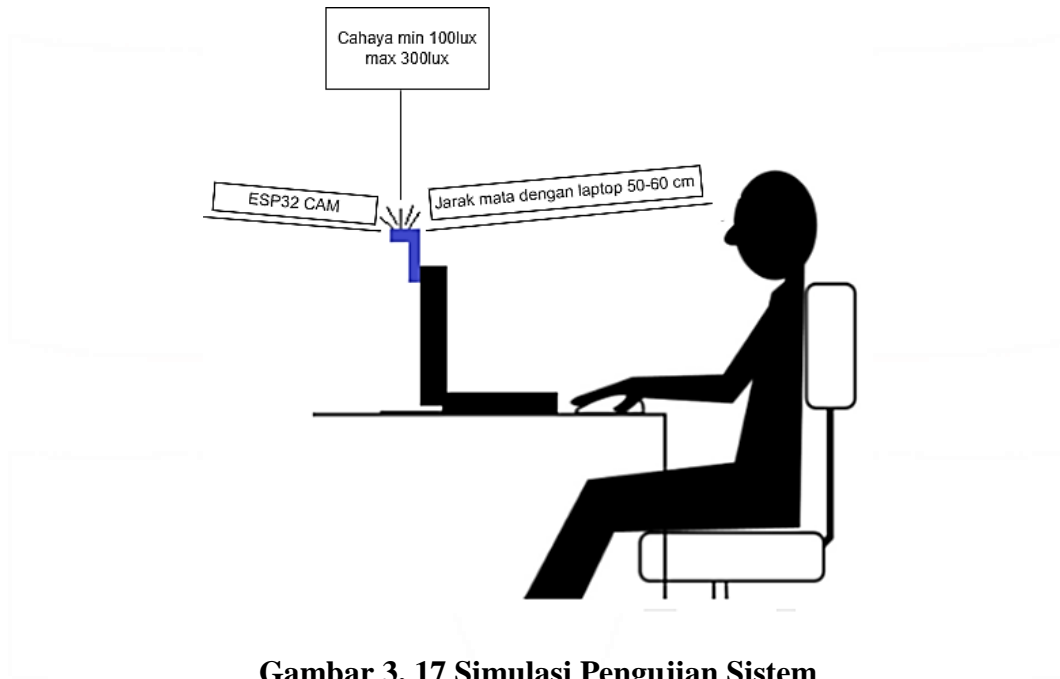
```

Gambar 3. 16 Script untuk koneksi ke Blynk App

3.5.2. Pengujian sistem monitoring

Sistem monitoring yang dibangun terdapat 3 sensor yaitu sensor jarak HC-SR04, sensor intensitas cahaya BH1750 dan sensor kamera. Perancangan alat yang sudah dibangun yaitu sistem monitoring dapat dilihat melalui *Blynk App*. Pada suatu sistem yang dijalankan atau baru dikembangkan perlu dilakukan pengujian atau tahap uji coba. Pengujian dilakukan sesuai dengan fungsionalitas yang diharapkan serta kesalahan maupun kekurangan dapat dideteksi dan diperbaharui secepat mungkin. Apabila pengujian sistem tidak dilakukan, maka akan menyebabkan kegagalan dan menimbulkan kerugian, sehingga mengakibatkan berbagai masalah ketika perangkat lunak tersebut digunakan. Pengujian yang dilakukan hanya mengambil hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak dan perangkat keras [36]. Dari sisi hardware, ada empat fungsional yang diuji yaitu fungsi koneksi

ESP8266, fungsi sensor *ultrasonic*, fungsi sensor intensitas pada NodeMCU dan ESP32 Cam. Sedangkan fungsional dari sisi *Blynk App* dengan mengidentifikasi keadaan realtime.



Gambar 3. 17 Simulasi Pengujian Sistem

Sumber : REPOSITOR, Vol. 2, No. 8, Agustus 2020, Pp. 1055-1066

a. Pengujian kalibrasi

Kalibrasi terbentuk dari kegiatan yang membentuk hubungan antara nilai yang ditunjukkan suatu alat ukur atau dinilai yang didapat alat ukur, terhadap nilai yang telah diketahui pada besaran terukur dalam kondisi tertentu. Kebenaran nilai konvensional sebuah alat ukur dapat dilakukan dengan cara membandingkannya dengan standar ukur. Uji kalibrasi dilakukan untuk menjaga keakuratan nilai alat ukur, menjamin mutu sistem, dan nilai penyimpangan sebuah alat ukur. [37] Pengujian kalibrasi dilakukan peneliti sebanyak 30 kali. Pengujian setiap percobaan kemudian dihitung selisih hasil alat yang dirancang dengan alat pabrikan. Perhitungan kalibrasi pengujian yang telah dilakukan yaitu nilai Error didapat dari meteran – sensor.

b. Pengujian efektivitas delay

Dilakukan dengan cara stopwatch berjalan selama *load* (muat gambar) objek dan ketika objek mulai terdeteksi di ESP32 CAM akan berhenti. Uji efektivitas dilakukan sebanyak 30 kali dengan berbagai objek.

c. Pengujian ketahanan alat

Pengujian dilakukan ketahanan alat diuji sebanyak 30 kali. Kemudian data yang sudah didapat dimasukkan dalam tabel. Kemudian dilakukan perhitungan presentasi untuk dilihat tingkat keberhasilan sistem monitoring.

3.7. Evaluasi Hasil Pengujian

Dalam tahap ini akan melakukan pengumpulan data yang sudah didapat dan kemudian dianalisis. Peneliti akan melakukan proses pengambilan data untuk proses monitoring anak. Jika alat yang sudah dirancang dapat melakukan mengirimkan data ke *Blynk App* dan kemudian melakukan notifikasi yang sudah diatur pada *software* dan *coding*, maka alat sudah siap untuk diimplementasikan. Namun jika terdapat error pada suatu fungsi maka peneliti harus melakukan perbaikan alat dan melakukan pengujian kembali.