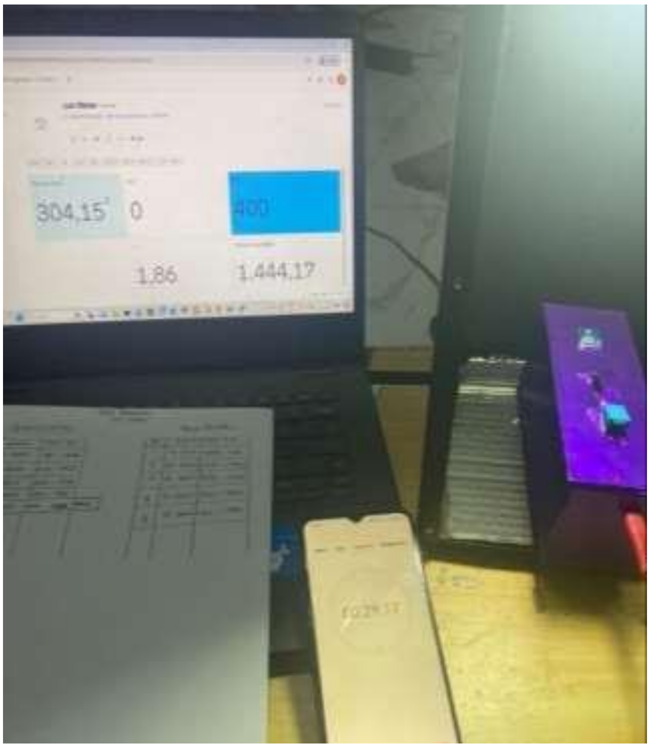
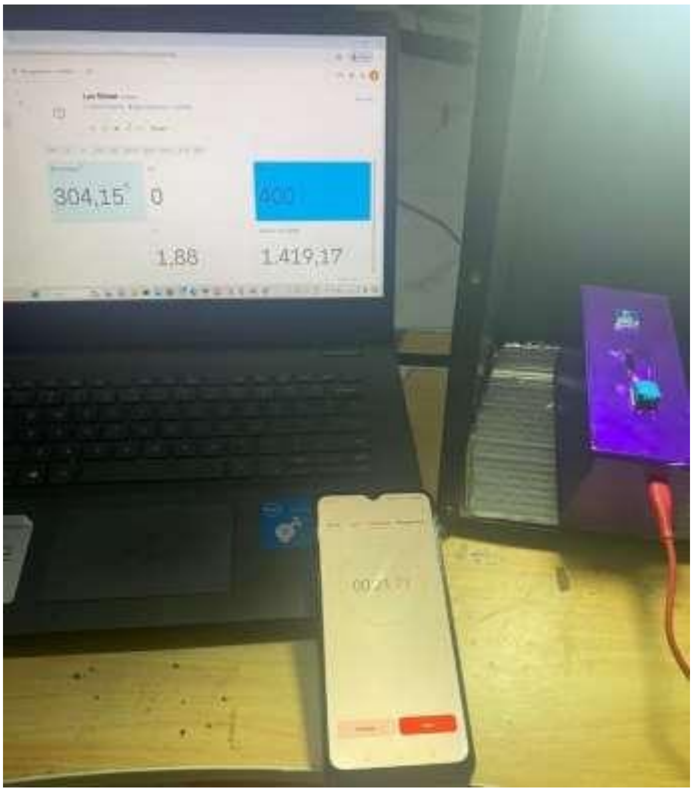
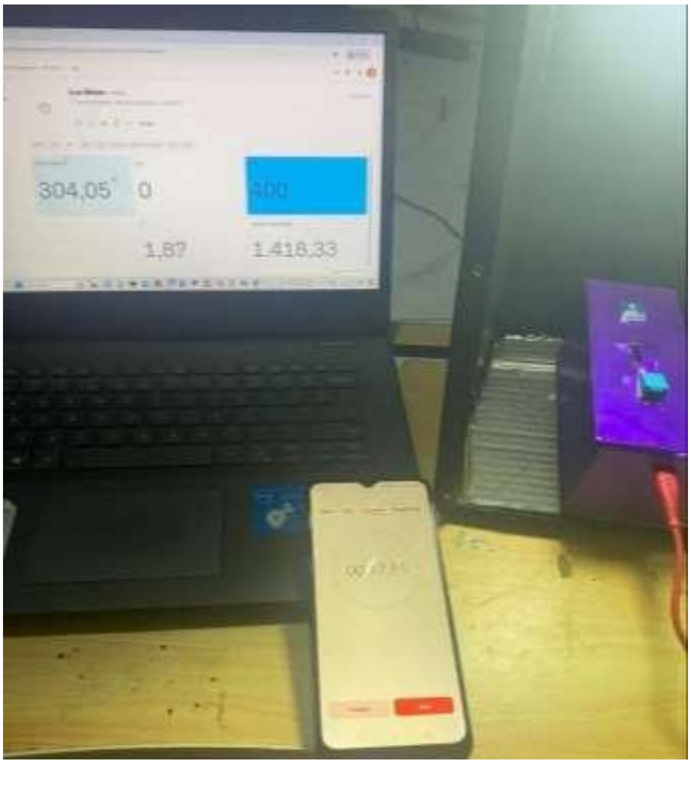
LAMPIRAN




304,15
0
400
1,88
1.419,17


304,05
0
400
1,87
1.418,33

303.65
0
400
1.87
1.431.67

```
  Blynk.begin(auth, ssid, pass, "sgp1.blynk.cloud", 80);
}
  void loop() { Blynk.run();
  if (WiFi.status() != WL_CONNECTED)
  Serial.println("WiFi terputus, menghubungkan
    kembali..."); WiFi.begin(ssid, pass);
    while (WiFi.status() !=
      WL_CONNECTED) { delay(500);
      Serial.print(".");
    }
    Serial.println("WiFi terhubung kembali");
  }

  if (!Blynk.connected()) {
    Serial.println("Blynk terputus, menghubungkan kembali...");
    Blynk.connect();
  }

  sensordht();
  sensorcahaya();
  hitungISO(lux);
  hitungEV(lux, shutterSpeed);
  hitungAperture(iso,
  shutterSpeed, ev);
  kirimdataBlynk();
  delay(1000);
}

void sensordht() {
  suhu =
  dht.readTemperature() +
  273.15; Serial.print("Suhu:
  "); Serial.println(suhu);
}

void sensorcahaya() {
  lux =
  lightMeter.readLightLe
  vel();
  Serial.print("Cahaya:
  "); Serial.println(lux);
}

void
  hitungISO(floa
  t lux) { if (lux
  < 1) {
    iso = 102400;
  } else if
    (lux < 5)
    { iso =
    51200;
  } else if
    (lux <
    10) { iso
    = 25600;
  } else if
    (lux <
    50) { iso
    = 12800;
  } else if (lux
    < 100) { iso
    = 6400;
  } else if (lux
    < 200) { iso
    = 3200;
  } else if (lux
    < 500) { iso
    = 1600;
  } else if (lux
    < 1000) {
    iso = 800;
  } else if (lux
    < 2000) {
    iso = 400;
  } else if (lux
    < 5000) {
    iso = 200;
  } else {
    iso = 100;
  }
}
```

```
void hitungEV(float lux, float
  shutterSpeed) { float K = 12.5; //
  Konstanta kalibrasi daylight ev =
  log2((lux * shutterSpeed) / K);


  if (ev <
    minEV
    ) { ev =
    minEV
    ;
  } else if (ev >
    maxEV) { ev
    = maxEV;
  }
}

void hitungAperture(float iso, float shutterSpeed,
  float ev) { aperture = sqrt(iso * shutterSpeed /
  pow(2, ev));


  if (aperture <
    minAperture) {
    aperture =
    minAperture;
  } else if (aperture >
    maxAperture) { aperture =
    maxAperture;
  }


}

void kirimdataBlynk() {
  Blynk.virtualWrite(V0
  , lux);
  Blynk.virtualWrite(V1
  , suhu);
  Blynk.virtualWrite(V2
  , iso);
  Blynk.virtualWrite(V3
  , ev);
Blynk.virtualWrite(V4,
  aperture);
}
```