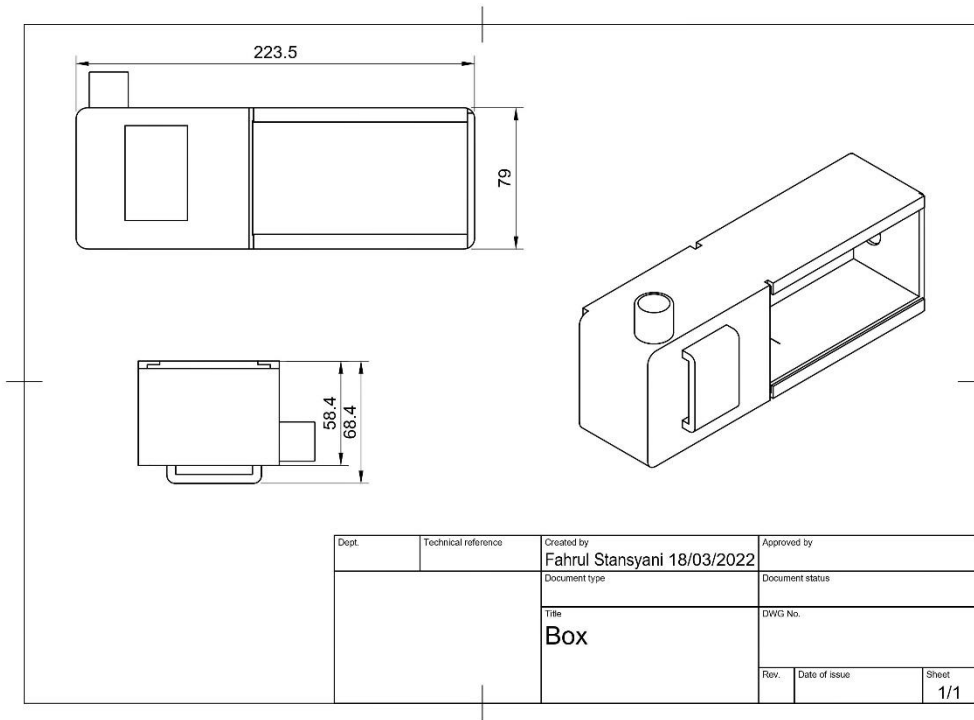


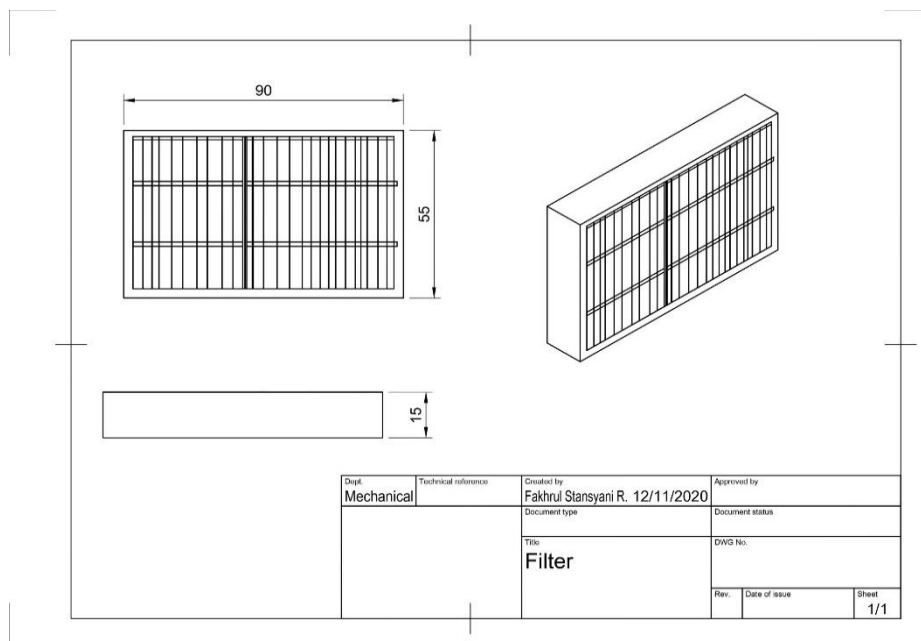
## DAFTAR PUSTAKA

- Church, T. J. (2019). Bacterial Contamination Study of Loose-Fitting Power Air Purifying Respirators (PAPR) Compared to N-95 FFR and Surgical Mask, in a Simulated Sterile Environment.
- Herman, J. A., Urits, I., Kaye, A. D., Urman, R. D., & Viswanath, O. (2020). COVID-19: General anesthesia precautions. *Journal of Clinical Anesthesia*.
- Faisal, H. D., & Susanto, A. D. (2017). Peran Masker/Respirator dalam Pencegahan Dampak Kesehatan Paru Akibat Polusi Udara. *Jurnal Respirasi*, 3(1), 18-25.
- Powell, J. B., Kim, J. H., & Roberge, R. J. (2017). Powered air-purifying respirator use in healthcare: effects on thermal sensations and comfort. *Journal of occupational and environmental hygiene*, 14(12), 947-954.
- Hospital Respiratory Protection Program Toolkit. OSHA, May 2015.
- Occupational Safety and Health Administration. (2009). Assigned protection factors for the revised respiratory protection standard; OSHA 3352-02. 2009. *Occupational Safety and Health Administration* ([www.osha.gov](http://www.osha.gov)), Washington.
- Utam, S. Y. A. (2018). *Buku Ajar Keperawatan Medikal Bedah Sistem Respirasi*. Deepublish.
- Curran, D. T., Murphy, A., Walker, G. J., Sayers, T. M., Henderson, C. P., & Kallman, B. (2013). *U.S. Patent Application No. 13/634,777*.
- Gregory James Fernandez, dr. Tjokorda Istri Anom Saturti, SpPD. (2017). *Sistem Pernafasan` Universitas Udayana*.

## LAMPIRAN



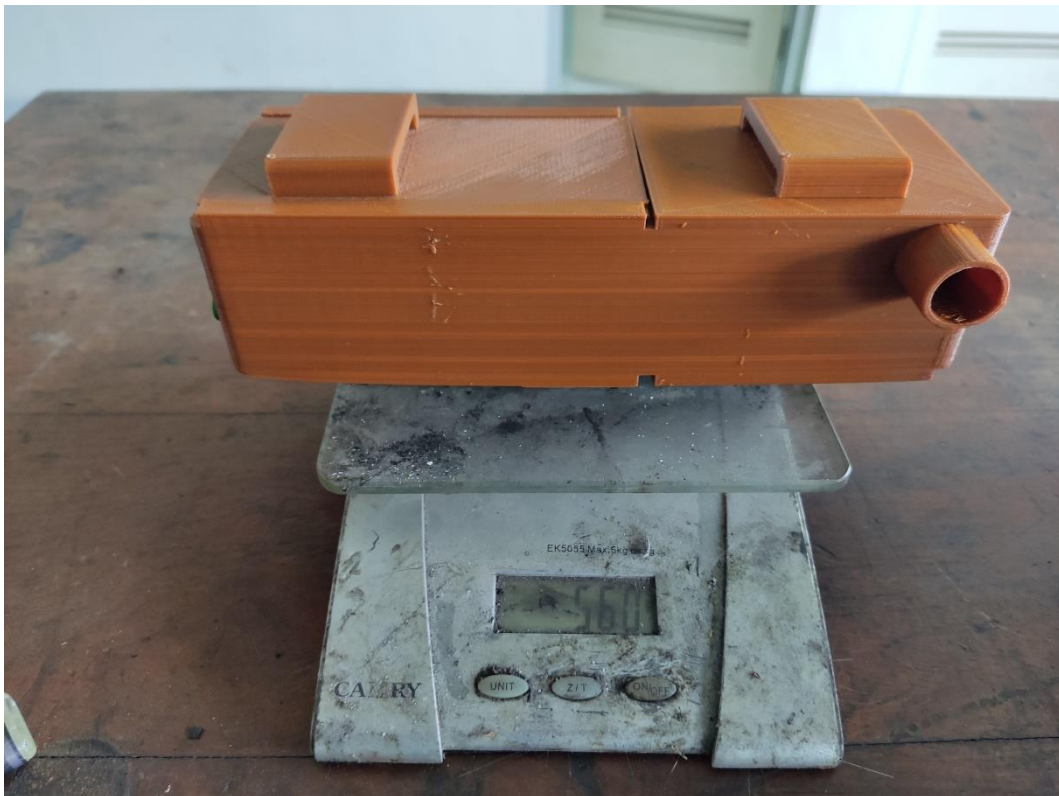
a. Dimensi box PAPR



b. Dimensi HEPA filter



c. Perakitan komponen



d. Mengukur berat PAPR



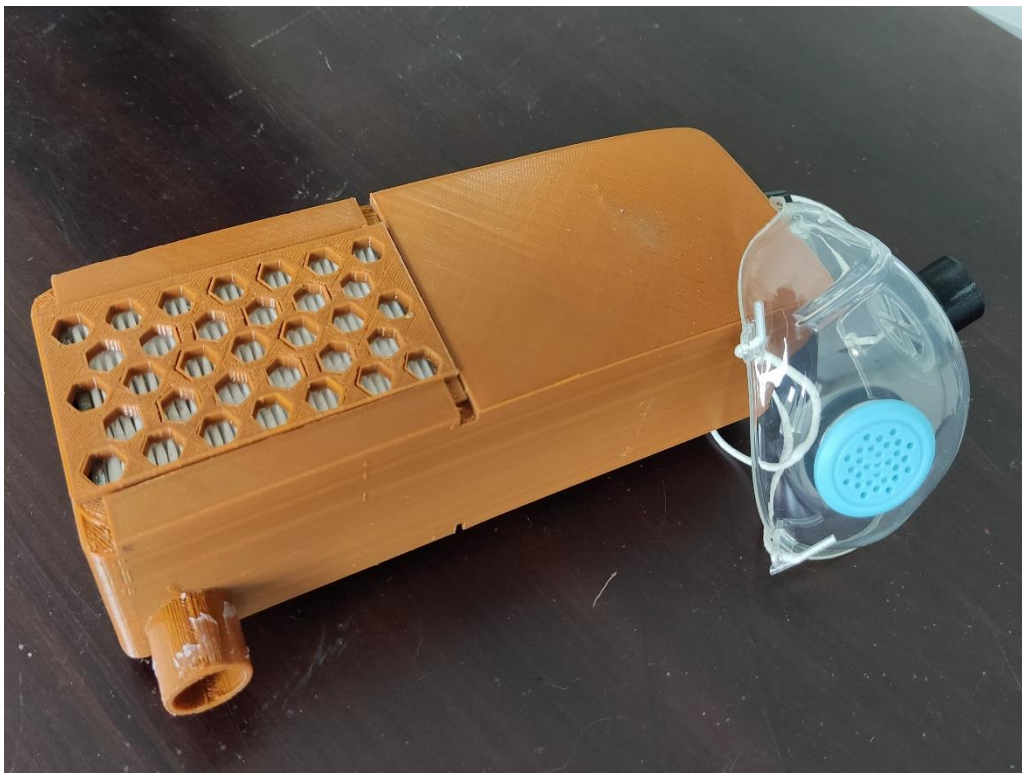
e. Penyambungan komponen elektronik



f. Perakitan PAPR



g.Pemasangan elektrikal



h. Produk PAPR

```

#define motor 5
#define potensio 0
int kecepatan = 0;
void setup() {
  pinMode(motor, OUTPUT);
  pinMode(potensio, INPUT);
}
void loop() {
  kecepatan = analogRead(potensio);
  kecepatan = map(kecepatan, 0, 1023, 0, 255); /
  analogWrite(motor, kecepatan);
  delay(20);
}

```

i. Program Arduino PAPR

```

byte sensorInt = 0;
byte flowSensor_pin = 2;

float konst = 4.5;
float flow_rate;
volatile byte count;

unsigned int flow_mlt;
unsigned long total_volume;
unsigned long oldTime;

void setup() {
  Serial.begin(9600);
  pinMode(flowSensor_pin, INPUT);
  digitalWrite(flowSensor_pin, HIGH);

  count = 0;
  flow_rate = 0.0;
  flow_mlt = 0;

```

```

total_volume = 0;
oldTime = 0;

attachInterrupt(sensorInt, countPulse, FALLING);
Serial.println("Menggunakan Flow Sensor");
}
void loop() {
if ((millis() - oldTime) > 1000) {
detachInterrupt(sensorInt);
flow_rate = ((1000.0 / (millis() - oldTime)) * count) / konst;
oldTime = millis();
flow_mlt = (flow_rate / 60) * 1000;
total_volume += flow_mlt;

unsigned int frac;

Serial.print("Flow Rate: ");
Serial.print(int(flow_rate));
Serial.println(" L/min");

count = 0;

attachInterrupt(sensorInt, countPulse, FALLING);
}
}
void countPulse(){
count++;
}

```

j. Program Arduino flow sensor