REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN
CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC00202227677, 26 April 2022
Pencipta		
Nama	:	**Umi Salamah, S.Si., M.Sc., Muhamad Faishal Khairan dkk**
Alamat	:	Sidodari RT 04 RW 08, Mlese, Gantiwarno , Klaten , JAWA TENGAH, 57455
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**UNIVERSITAS AHMAD DAHLAN**
Alamat	:	Jl. Pramuka 5F, Pandeyan, Umbulharjo , Yogyakarta , DI YOGYAKARTA, 55161
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Program Komputer**
Judul Ciptaan	:	**PM-CARDEC**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	20 April 2022, di Yogyakarta
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	:	000343181

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto
NIP.196412081991031002

Disclaimer:
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Umi Salamah, S.Si., M.Sc.	Sidodari RT 04 RW 08, Mlese, Gantiwarno
2	Muhamad Faishal Khairan	Dusun Manis RT 009 RW 002 Kalapagunung Kramatmulya
3	Yuliana Safitri	Jalan Tuk Lanting RT 014 RW 004, Mendawai, Sukamara
4	Dhita Pratama Putra	Dusun Mendut RT 001 RW 003, Ngrapah, Banyubiru
5	Rini Suphia Nuryati	Jalan Karang Jawa RT 004 RW 000, Baroqah, Simpang Empat
6	Amelia	Lingk. Ketib No. 52A RT 003 RW 003, Kotakaler, Sumedang Utara

DISUSUN OLEH :

Umi Salamah, S.Si., M.Sc.
Muhamad Faishal Khairan
Yuliana Safitri
Dhita Pratama Putra
Rini Suphia Nuryati
Amelia

UNIVERSITAS AHMAD DAHLAN
TAHUN 2022

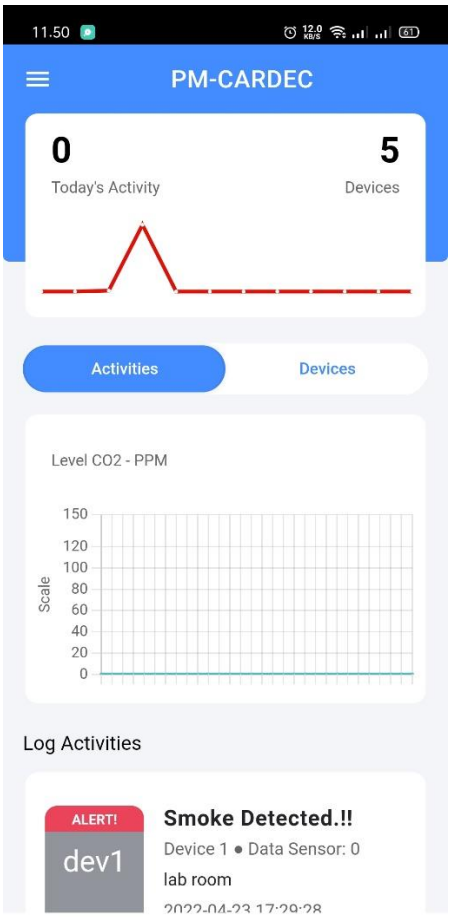
Profil PM-CARDEC

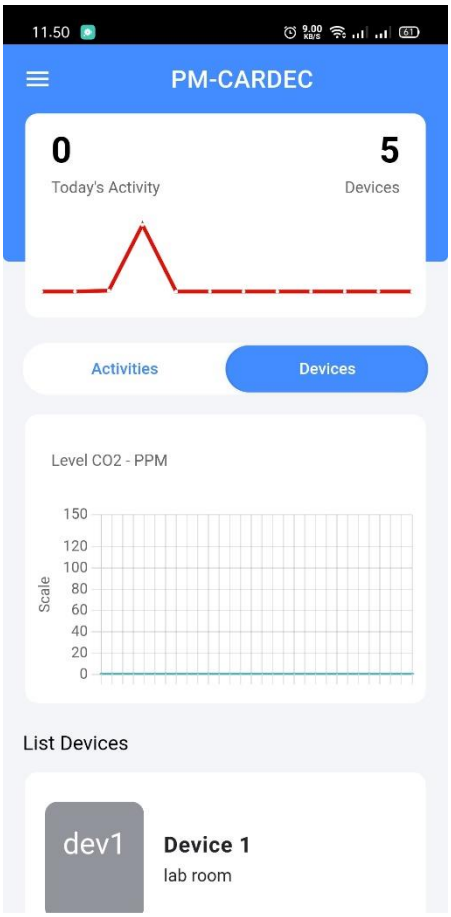
PM-Cardec merupakan aplikasi untuk memonitoring gas CO_2 jarak jauh secara realtime, pada aplikasi ini terdapat beberapa grafik dan informasi yang dikirimkan oleh sistem hardware deteksi gas CO_2 yang menggunakan sensor MQ-135, ESP-32, LCD dan Buzzer. Sistem hardware ini terpasang pada lokasi yang akan dideteksi seperti hutan yang rawan kebakaran, ruang dalam gedung dan lain-lainnya. Aplikasi PM-CARDEC dapat memunculkan peringatan jika terdapat gas CO_2 dalam jumlah ppm tertentu yang berbahaya. Data kadar gas dan aktivitas deteksi ditampilkan dalam bentuk grafik secara *real-time.*

Mekanisme Penggunaan PM-CARDEC

Mekanisme penggunaan PM-CARDEC terdapat beberapa langkah instalasi dan tangkapan layar menu yang ada di aplikasi PM-CRDEC antara lain:

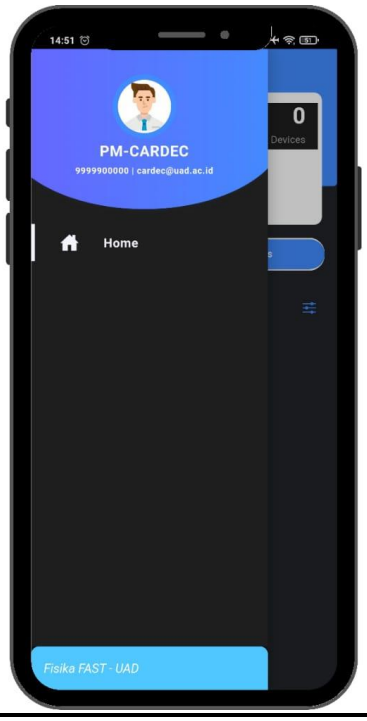
1. Download aplikasi PM-Cardec melalui tautan berikut ini PM-CARDEC.apk
2. Install pada perangkat anda kemudian buka aplikasi PM-Cardec
3. Pada menu utama akan langsung ditampilkan beberapa informasi yang sudah berhasil dikirimkan oleh perangkat hardware sistem deteksi gas CO_2 yang sudah aktif, berikut ini hasil tangkapan layar beserta deskripsinya:

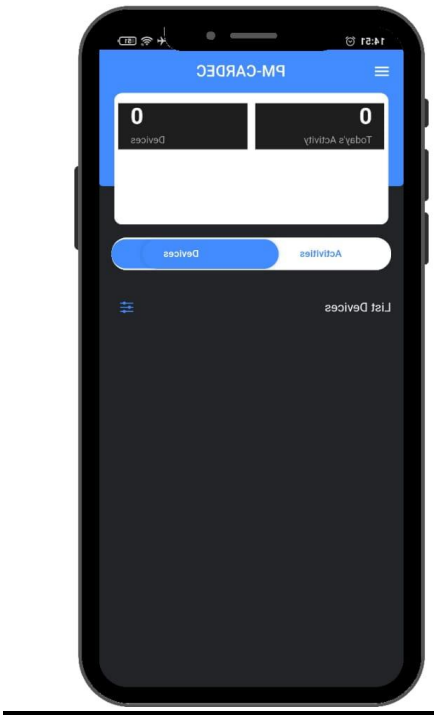




Pada halaman utama pengguna akan langsung ditampilkan informasi seperti jumlah aktivitas alat hari ini, data jumlah perangkat yang ada, grafik bulanan, grafik realtime saat ini dan juga log aktivitas yang sudah di kirimkan oleh perangkat dan pada tab *devices* terdapat daftar informasi perangkat yang ada.

Gambaran Aplikasi PM-CARDEC





Output

Untuk *test dapat diakses PM-CARDEC.apk*

***Source Code* Koko**

Berikut adalah beberapa *source code* aplikasi PM-CARDEC:

1. Home UI

```
<ion-header class=”ion-no-border”>
  <ion-toolbar color=”primary”>
    <ion-title class=”ion-text-center” style=”margin-left: -25px;”>
      PM-CARDEC
    </ion-title>
    <ion-buttons slot=”start”>
      <ion-menu-button></ion-menu-button>
    </ion-buttons>
  </ion-toolbar>

  <app-shrink-header [scrollArea]=”myContent” [headerHeight]=”170”>
    <ion-grid [style.padding-bottom.px]=”120 – newHeight”></ion-grid>
    <ion-list class=”popover” lines=”none”
      [style.margin-top.px]=”-120 – newHeight/2”>
      <ion-row>
        <ion-col size=”6”>
          <ion-item>
            <ion-label class=”ion-text-wrap”>
              <h1>
                <b>{{logsToday.length}}</b>
              </h1>
```

```
            <p>Today’s Activity</p>
          </ion-label>
        </ion-item>
      </ion-col>
      <ion-col size=”6”>
        <ion-item>
          <ion-label class=”ion-text-wrap ion-text-right”>
            <h1>
              <b>{{allDevices.length}}</b>
            </h1>
            <p>Devices</p>
          </ion-label>
        </ion-item>
      </ion-col>
      <ion-col size=”12”>
        <canvas #lineCanvas></canvas>
      </ion-col>
    </ion-row>
  </ion-list>
 </app-shrink-header>

 <div class=”ion-padding” (ionChange)=”segmentChanged($event)”>
  <ion-segment mode=”ios” [value]=”segmentValue”>
    <ion-segment-button value=”1”>
      <ion-label>Activities</ion-label>
    </ion-segment-button>
    <ion-segment-button value=”2”>
      <ion-label>Devices</ion-label>
    </ion-segment-button>
  </ion-segment>
 </div>
</ion-header>

<ion-content [scrollEvents]=”true” (ionScroll)=”scroll($event)” #myContent>
 <ion-refresher slot=”fixed” (ionRefresh)=”doRefresh($event)”>
  <ion-refresher-content></ion-refresher-content>
 </ion-refresher>
 <div class=”statistic”>
  <app-shrink-header>
    <ion-list class=”popover” lines=”none”>
      <ion-row>
        <ion-col size=”6”>
          <ion-item>
            <ion-label class=”ion-text-wrap”>
              <p>Level CO<span style=”font-size: x-small;”>2</span> - PPM</p>
            </ion-label>
          </ion-item>
        </ion-col>
        <ion-col size=”12”>
          <canvas #lineChartStatisticCanvas></canvas>
        </ion-col>
      </ion-row>
    </ion-list>
  </app-shrink-header>
 </div>
```

```
<ion-list lines=”none”>
 <ion-item color=”light”>
  <ion-label class=”heading”>{{segmentValue == ‘1’ ? ‘ Log Activities’ : ‘ List Devices’}}</ion-label>
 </ion-item>

 <ion-item-group>
  <ion-item class=”ion-padding-vertical popItem”
   *ngFor=”let item of (segmentValue == ‘1’) ? allLogs : allDevices”>
   <ion-thumbnail slot=”start”>
    <div class=”ion-text-center” *ngIf=”segmentValue == ‘1’”>
     <ion-text color=”white”>ALERT!</ion-text>
    </div>
    <ion-text color=”white” class=”deviceCode” [style.margin]=”segmentValue == ‘1’ ? ‘10px 0’ : ‘20px 0’”>{{item?.deviceCode}}</ion-text>
   </ion-thumbnail>
   <ion-label>
    <ion-note color=”dark”><b>{{segmentValue == ‘1’ ? item?.title:item?.deviceName}}</b></ion-note>
    <p class=”ion-text-wrap” *ngIf=”segmentValue == ‘1’”>{{item?.deviceName}}
    <span>&#9679; Data Sensor: {{item?.dataSensor}}</span> </p>
    <p>
     <ion-text color=”dark”>{{segmentValue == ‘1’ ? item?.position:item?.position}}</ion-text>
    </p>
    <p class=”ion-text-wrap” *ngIf=”segmentValue == ‘1’”>{{item?.created_at}}</p>
   </ion-label>
  </ion-item>
 </ion-item-group>
</ion-list>

</ion-content>
```

2. Home Functionality

```
Import { DatePipe } from ‘@angular/common’;
Import { HttpClient } from ‘@angular/common/http’;
Import { Component, ElementRef, OnInit, ViewChild } from ‘@angular/core’;
Import Chart from ‘chart.js/auto’;
Import { serverUrl } from “../../config”;

@Component({
 Selector: ‘app-home’,
 templateUrl: ‘home.page.html’,
 styleUrls: [‘home.page.scss’],
})
Export class HomePage implements OnInit {

 @ViewChild(‘lineCanvas’) lineCanvas: ElementRef;
 @ViewChild(‘lineChartStatisticCanvas’) lineChartStatisticCanvas: ElementRef;
 segmentValue = ‘1’;
 lineChart: any;
 lineChartStatistic: any;
 newHeight = 0;
```

```
constructor(
  public datePipe: DatePipe,
  public http:HttpClient
) {}

ngOnInit() {
  this.getLog();
  this.getLineChart();
  this.getDevices();
  this.getStatistics();
  this.allStatistics = setInterval(()=> { this.getStatisticsAgain() }, 2000);
}

ngAfterViewInit() {
}

Async doRefresh(event) {
  This.lineChart.destroy();
  This.lineChartStatistic.destroy();
  This.allLogs = [];
  This.logsToday = [];
  This.allDevices = [];
  This.allStatistics = [];
  This.dataLines = {};
  This.getLog();
  This.getLineChart();
  This.getDevices();
  This.getStatistics();
  setTimeout(() => {
    event.target.complete();
  }, 2000);
}

allLogs:any = [];
getLog() {
  this.http.get(serverUrl).subscribe(res => {
    this.allLogs = res;
    this.parseGetTodayLogs();
  })
}

getLineChart() {
  this.http.get(serverUrl+'lineChart').subscribe(res => {
    this.parseDataChart(res);
  })
}

allDevices:any = [];
getDevices() {
  this.http.get(serverUrl+'devices').subscribe(res => {
    this.allDevices = res;
  })
}

allStatistics:any = [];
```

```
getStatistics() {
  this.http.get(serverUrl+'statistics').subscribe(res => {
    this.allStatistics = res;
    this.parseDataStatistic(this.allStatistics);
  })
}

getStatisticsAgain() {
  this.http.get(serverUrl+'statistics').subscribe(res => {
    this.allStatistics = res;
    this.lineChartStatistic.destroy();
    this.parseDataStatistic(this.allStatistics);
  })
}

dataStatistic:any = {};
showData:any = [];
labelStatistics:any = [];
parseDataStatistic(res) {
  this.showData = [];
  this.labelStatistics = [];
  res.forEach((e, index) => {
    this.dataStatistic[index+1] = e;
    this.showData.push(Number(e.dataSensor))
    this.labelStatistics.push(e.created_at)
  });
  This.lineChartStatistics();
}

logsToday:any = [];
parseGetTodayLogs() {
  let dateNow = this.datePipe.transform(new Date(), 'dd MMMM yyyy');
  this.allLogs.forEach(e => {
    let created_at = this.datePipe.transform(new Date(e.created_at), 'dd MMMM yyyy');
    if(created_at == dateNow) {
      let idx = this.logsToday.indexOf€;
      if(idx == -1) {
        this.logsToday.push€;
      }
    }
  });
}

dataLines:any = {};
parseDataChart(res) {
  res.forEach(e => {
    this.dataLines[e.month] = e;
  });
  This.lineChartMethod();
}

segmentChanged(event) {
  this.segmentValue = event.detail.value;
}
```

```
lineChartMethod() {
  this.lineChart = new Chart(this.lineCanvas.nativeElement, {
    type: 'line',
    data: {
      labels: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sept', 'Oct', 'Nov',
    'Dec'],
      datasets: [
        {
          Label: 'Activity',
          Fill: 'false',
          backgroundColor: '#fff',
          borderColor: '#E31007',
          borderCapStyle: 'butt',
          borderDash: [],
          borderDashOffset: 0.0,
          borderJoinStyle: 'miter',
          pointBorderColor: '#fff',
          pointBackgroundColor: '#fff',
          pointBorderWidth: 1,
          pointHoverRadius: 5,
          pointHoverBackgroundColor: '#E31007',
          pointHoverBorderColor: '#E31007',
          pointHoverBorderWidth: 2,
          pointRadius: 1,
          pointHitRadius: 10,
          data: [
            this.dataLines[1] == undefined ? 0:Number(this.dataLines[1].total_activity),
            this.dataLines[2] == undefined ? 0:Number(this.dataLines[2].total_activity),
            this.dataLines[3] == undefined ? 0:Number(this.dataLines[3].total_activity),
            this.dataLines[4] == undefined ? 0:Number(this.dataLines[4].total_activity),
            this.dataLines[5] == undefined ? 0:Number(this.dataLines[5].total_activity),
            this.dataLines[6] == undefined ? 0:Number(this.dataLines[6].total_activity),
            this.dataLines[7] == undefined ? 0:Number(this.dataLines[7].total_activity),
            this.dataLines[8] == undefined ? 0:Number(this.dataLines[8].total_activity),
            this.dataLines[9] == undefined ? 0:Number(this.dataLines[9].total_activity),
            this.dataLines[10] == undefined ? 0:Number(this.dataLines[10].total_activity),
            this.dataLines[11] == undefined ? 0:Number(this.dataLines[11].total_activity),
            this.dataLines[12] == undefined ? 0:Number(this.dataLines[12].total_activity),
          ],
          spanGaps: false
        }
      ]
    },
    Options: {
      Responsive: true,
      maintainAspectRatio: false,
      plugins: {
        legend: {
          display: false
        }
      },
      Scales: {
        X: {
          Grid: {
            Display: false,
```

```
          drawBorder: false,
        },
        Ticks: {
          Display: false
        }
      },
      Y: {
        Grid: {
          Display: false,
          drawBorder: false,
        },
        Ticks: {
          Display: false
        }
      }
    }
  }
 });
}


lineChartStatistics() {
  this.lineChartStatistic = new Chart(this.lineChartStatisticCanvas.nativeElement, {
    type: 'line',
    data: {
      labels: this.labelStatistics,
      datasets: [
        {
          Label: 'Data Sensor',
          Fill: 'false',
          lineTension: 0.1,
          backgroundColor: 'rgba(75,192,192,0.4)',
          borderColor: 'rgba(75,192,192,1)',
          borderCapStyle: 'butt',
          borderDash: [],
          borderDashOffset: 0.0,
          borderJoinStyle: 'miter',
          pointBorderColor: 'rgba(75,192,192,1)',
          pointBackgroundColor: '#fff',
          pointBorderWidth: 1,
          pointHoverRadius: 5,
          pointHoverBackgroundColor: 'rgba(75,192,192,1)',
          pointHoverBorderColor: 'rgba(220,220,220,1)',
          pointHoverBorderWidth: 2,
          pointRadius: 1,
          pointHitRadius: 10,
          data: this.showData,
          spanGaps: false,
        }
      ]
    },
    Options: {
      Responsive: true,
      maintainAspectRatio: false,
      plugins: {
```

```
        legend: {
          display: false
        }
      },
      Scales: {
        X: {
          Ticks: {
            Display: false
          }
        },
        Y: {
          Min: 0,
          Max: 150,
          stepSize: 1,
          callback: function (label, index, labels) {
            return [150, 100, 80, 50, 25, 10, 0];
          },
          Title: {
            Display: true,
            Text: 'Scale'
          }
        }
      }
    }
  });
 }


 Scroll(event) {
  Const value = event.detail.scrollTop;
  If(value > 40) {
    This.newHeight += 5; // this.newHeight = this.newHeight + 5
  } else {
    This.newHeight = 0;
  }
  If(value > 180 && this.newHeight <= 65) {
    This.newHeight += 50;
  }
 }

}
```

3. Home CSS/Style

```
Ion-header {
  Background: var(--ion-color-light);
  App-shrink-header {
    Display: block;
    Overflow: hidden;
    Ion-grid {
      Background: var(--ion-color-primary);
      Margin-top: -5px;
      Border-radius: 0 0 5px 5px;
      // padding-bottom: 120px;
    }
    .popover {
```

```
      Background: var(--ion-color-white);
      Border-radius: 10px;
      Width: 90%;
      Margin: auto;
      // margin-top: -120px;
      Ion-item {
        Ion-label {
          Margin: 0;
          H1 {
            Font-size: 1.8rem;
          }
          P {
            Margin-top: 3px;
            Font-size: 0.8rem;
          }
          Ion-text {
           Font-size: 0.7rem;
          }
        }
      }
      Canvas {
        Height: 70px !important;
      }
    }
  }
  Div {
    Background: var(--ion-color-light);
    Ion-segment {
      --background: var(--ion-color-white);
      Border-radius: 20px;
      --margin-top: 5px;
      --margin-bottom: 5px;
      Ion-segment-button {
        --background-checked: var(--ion-color-primary);
        --indicator-color: var(--ion-color-primary);
        --border-radius: 20px;
        --color: var(--ion-color-primary);
        --color-checked: var(--ion-color-white);
      }
    }
  }
}

.statistic {
  Background: var(--ion-color-light);
  App-shrink-header {
    Display: block;
    Overflow: hidden;
    Ion-grid {
      Background: var(--ion-color-primary);
      Margin-top: -5px;
      Border-radius: 0 0 5px 5px;
      // padding-bottom: 120px;
    }
    .popover {
```

```
      Background: var(--ion-color-white);
      Border-radius: 10px;
      Width: 90%;
      Margin: auto;
      // margin-top: -120px;
      Ion-item {
        Ion-label {
          Margin: 0;
          H1 {
            Font-size: 1.8rem;
          }
          P {
            Margin-top: 3px;
            Font-size: 0.8rem;
          }
          Ion-text {
           Font-size: 0.7rem;
          }
        }
      }
      Canvas {
        Height: 150px !important;
      }
    }
  }
  Div {
    Background: var(--ion-color-light);
    Ion-segment {
      --background: var(--ion-color-white);
      Border-radius: 20px;
      --margin-top: 5px;
      --margin-bottom: 5px;
      Ion-segment-button {
        --background-checked: var(--ion-color-primary);
        --indicator-color: var(--ion-color-primary);
        --border-radius: 20px;
        --color: var(--ion-color-primary);
        --color-checked: var(--ion-color-white);
      }
    }
  }
}

Ion-content {
  --background: var(--ion-color-light);
  Ion-list {
    Background: var(--ion-color-light);
    .heading {
      Font-family: lato;
    }
    .popItem {
      Width: 90%;
      Margin: auto;
      Margin-bottom: 15px;
      Border-radius: 10px;
```

```
      Padding: 0;
      Ion-thumbnail {
        Min-width: 5rem;
        Min-height: 6rem;
        Border-radius: 10px;
        Background: var(--ion-color-medium);
        Div {
          Background-color: var(--ion-color-danger);
          Border-radius: 10px 10px 0 0;
          Padding-bottom: 2px;
          Ion-text {
            Font-size: 0.7rem;
            Font-weight: bold;
          }
        }
        .deviceCode {
          Font-size: 1.5rem !important;
          Display: flex;
          Justify-content: center;
          Align-content: center;
          Align-items: center;
        }
      }
      Ion-label {
        Ion-note {
          Font-size: 1.1rem;
          Letter-spacing: 0.5px;
        }
        P {
          Margin-top: 5px;
          Ion-text {
            Span {
              Color: var(--ion-color-dark);
              Padding: 0 5px;
            }
          }
        }
      }
      Ion-fab {
        Margin-right: -15px !important;
        Margin-top: -10px;
      }
    }
  }
}
```

4. Rest API - backend database

```
<?php
Defined('BASEPATH') OR exit('No direct script access allowed');
Use \Firebase\JWT\JWT;

Class Cardec extends BD_Controller {

  Function __construct()
  {
```

```
    // Construct the parent class
    Parent::__construct();
    // Configure limits on our controller methods
    // Ensure you have created the 'limits' table and enabled 'limits' within application/config/rest.php
    Header("Access-Control-Allow-Origin: *");
    $this->methods['users_get']['limit'] = 10000; // 500 requests per hour per user/key
    $this->methods['users_post']['limit'] = 10000; // 100 requests per hour per user/key
    $this->methods['users_delete']['limit'] = 500; // 50 requests per hour per user/key
    $this->kunci='34242342343244';
    $this->appUrl='https://apis.ruang-ekspresi.id';
    Date_default_timezone_set('Asia/Jakarta');
}

Function index_get()
{
    $query = $this->db->query('SELECT *
            FROM `logs`
            ORDER BY created_at DESC
            LIMIT 30');

    $this->response($query->result());

}

Function config_get()
{
    $this->db->select('maxudarabersih');
    $this->db->from('config');
    $data = $this->db->get();
    $value = intval($data->row()->maxudarabersih);
    $this->response($value);

}

Function lineChart_get()
{
    $tahunNow = date('Y');
    $query = $this->db->query('select year(created_at) as year, month(created_at) as month, count(id) as total_activity from logs where year(created_at) = '.$tahunNow.' Group by year(created_at), month(created_at)');

    $this->response($query->result());
}

Function devices_get()
{
    $query = $this->db->query('select position, deviceCode, deviceName, count(id) as deviceID from logs group by deviceCode');

    $this->response($query->result());
}

Function index_post()
{
```

```
    $data=$this->post();
    $cek = $this->db->get_where('token',array('token'=>$data['token']))->num_rows();
    If($cek == 0)
    {
        $this->db->insert($db,$data);
    }else{
        $this->db->where('token',$data['token']);
        $this->db->update('token');
    }
     $this->response($data);
}

Function saveStatistic_get()
{
    $data=$this->post();
    $params=array(
        'deviceCode' => $this->get('deviceCode'),
        'deviceName' => $this->get('deviceName'),
        'position' => $this->get('position'),
        'dataSensor' => $this->get('dataSensor')
    );

    $id=uniqid();
    $data['id'] = 's_'.$id;
    $data['deviceCode'] = $params['deviceCode'];
    $data['deviceName'] = $params['deviceName'];
    $data['position'] = $params['position'];
    $data['dataSensor'] = $params['dataSensor'];
    $this->db->insert('statistics',$data);
    $this->deleteOldStatistic();
     $this->response($data);
}

Function deleteOldStatistic() {
    $this->db->query("DELETE FROM `statistics`
        WHERE id NOT IN (
          SELECT id
          FROM (
            SELECT id
            FROM `statistics`
            ORDER BY created_at DESC
            LIMIT 30 – keep this many records
          ) foo
        )");
}

Function statistics_get()
{
    $query = $this->db->query('SELECT id, created_at, dataSensor
            FROM `statistics`
            ORDER BY created_at ASC');

    $this->response($query->result());
}
```

```
Function saveLog_get()
{
   $data=$this->post();
   $params=array(
      'deviceCode' => $this->get('deviceCode'),
      'deviceName' => $this->get('deviceName'),
      'position' => $this->get('position'),
      'status' => $this->get('status'),
      'dataSensor' => $this->get('dataSensor')
   );

   $id=uniqid();
   $data['id'] = 'l_'.$id;
   $data['title'] = 'Smoke Detected.!!';
   $data['body'] = $params['status'];
   $data['deviceCode'] = $params['deviceCode'];
   $data['deviceName'] = $params['deviceName'];
   $data['position'] = $params['position'];
   $data['dataSensor'] = $params['dataSensor'];
   $this->db->insert('logs',$data);

   $this->sendNotif($data);
    $this->response($data);
}

Function sendNotif($data) {
   $tokens = $this->db->get('token')->result();

   Foreach($tokens as $token)
   {
    // FCM API Url
    $url = 'https://fcm.googleapis.com/fcm/send';

    // Put your Server Response Key here
    $apiKey = "AAAAi-
dpDLU:APA91bFqk2J_iLCznTbpZj4JjeEw2g_uZu70kARPKezBzfAHF9D5H-
OgjFSyxqxIkaadc1TzcpyKPBCHnM3jKNt3du5UaddWR_3jDMWT9_u0o5ShULT3
H5voIGoIUHNBt1SqbSGtWMex";

    // Compile headers in one variable
    $headers = array (
     'Authorization:key=' . $apiKey,
     'Content-Type:application/json'
    );

    // Add notification content to a variable for easy reference
    $notifData = [
     "title" => $data['title'],
     "body" => $data['body'],
     "sound" => "android.resource://com.pmcardec.uad/raw/buzzer",
     "image" => https://www.pngmart.com/files/21/Red-Alert-PNG-HD.png
    ];

    $dataToSend = [
     "deviceCode" => $data['deviceCode'],
```

```
      "deviceName" => $data['deviceName'],
      "position" => $data['position']
    ];

    // Create the api body
    $apiBody = [
      'notification' => $notifData,
      'data' => $dataToSend,
      'to' => $token->token
    ];

    // Initialize curl with the prepared headers and body
    $ch = curl_init();
    Curl_setopt ($ch, CURLOPT_URL, $url );
    Curl_setopt ($ch, CURLOPT_POST, true );
    Curl_setopt ($ch, CURLOPT_HTTPHEADER, $headers);
    Curl_setopt ($ch, CURLOPT_RETURNTRANSFER, true );
    Curl_setopt ($ch, CURLOPT_POSTFIELDS, json_encode($apiBody));

    // Execute call and save result
    $result = curl_exec ( $ch );

    // Close curl after call
    Curl_close ( $ch );
  }
}

Function saveToken_post()
{
  $data=$this->post();
  $cek = $this->db->get_where('token',array('token'=>$data['token']))->num_rows();
  If($cek == 0)
  {
    $id=uniqid();
    $data['id'] = 't_'.$id;
    $this->db->insert('token',$data);
  }else{
    $this->db->where('token',$data['token']);
    $this->db->update('token', $data);
  }
   $this->response($data);
}

Function index_put()
{
  $this->auth();
  $db=$this->uri->segment('2');
  $id=$this->uri->segment('3');
  $data=$this->put();
  If($id!=null)
  {
    //$this->db->set($data);
    $this->db->where('id',$id);
    $this->db->update($db,$data);
   $this->response('OK');
```

```
        }else{
            $this->response('FAILURE');
        }
    }
}
```