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	:	EC00202041133, 16 Oktober 2020
Pencipta		
Nama	:	**Kusrini, Dr., S.Kom, M.Kom, Suputa dkk**
Alamat	:	Dusun Sanggrahan RT 4 RW 36 Desa Wedomartani Kecamatan Ngemplak , Sleman , Di Yogyakarta, 55584
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Kusrini, Dr., S.Kom, M.Kom, Suputa dkk**
Alamat	:	Dusun Sanggrahan RT 4 RW 36 Desa Wedomartani Kecamatan Ngemplak , Sleman, Di Yogyakarta, 55584
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Program Komputer**
Judul Ciptaan	:	**Mango Pest Identifier Versi 2**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	10 September 2020, di Yogyakarta
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	:	000210361

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

[illegible]

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Kusrini, Dr., S.Kom, M.Kom	Dusun Sanggrahan RT 4 RW 36 Desa Wedomartani Kecamatan Ngemplak
2	Suputa	Dusun Jurang Kuali RT/RW 003/005 Desa Sumber Berantas Kecamatan Bumiaji
3	Arief Setyanto	Jl. Gambiran 92 A RT/RW 004/002 Kelurahan Giwangan Kecamatan Umbulharjo
4	I Made Artha Agastya	Terban GK 5/218 RT/RW 008/002 Kelurahan Terban Kecamatan Gondokusuman
5	Herlambang Priantoro	Madusari RT/RW 005/002 Desa Wonosari Kecamatan Wonosari
6	Fadlurahman	Dusun Sumber Bentong RT/RW 006/003 Karang Cempaka Bluto

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Kusrini, Dr., S.Kom, M.Kom	Dusun Sanggrahan RT 4 RW 36 Desa Wedomartani Kecamatan Ngemplak
2	Suputa	Dusun Jurang Kuali RT/RW 003/005 Desa Sumber Berantas Kecamatan Bumiaji
3	Arief Setyanto	Jl. Gambiran 92 A RT/RW 004/002 Kelurahan Giwangan Kecamatan Umbulharjo
4	I Made Artha Agastya	Terban GK 5/218 RT/RW 008/002 Kelurahan Terban Kecamatan Gondokusuman
5	Herlambang Priantoro	Madusari RT/RW 005/002 Desa Wonosari Kecamatan Wonosari
6	Fadlurahman	Dusun Sumber Bentong RT/RW 006/003 Karang Cempaka Bluto

Manual Aplikasi

Aplikasi Mango Pest Identifier Versi 2

Kusrini, Dr., S.Kom, M.Kom
Suputa
Arief Setyanto
I Made Artha Agastya
Herlambang Priantoro
Fadlurahman

A. Deskripsi Program Aplikasi

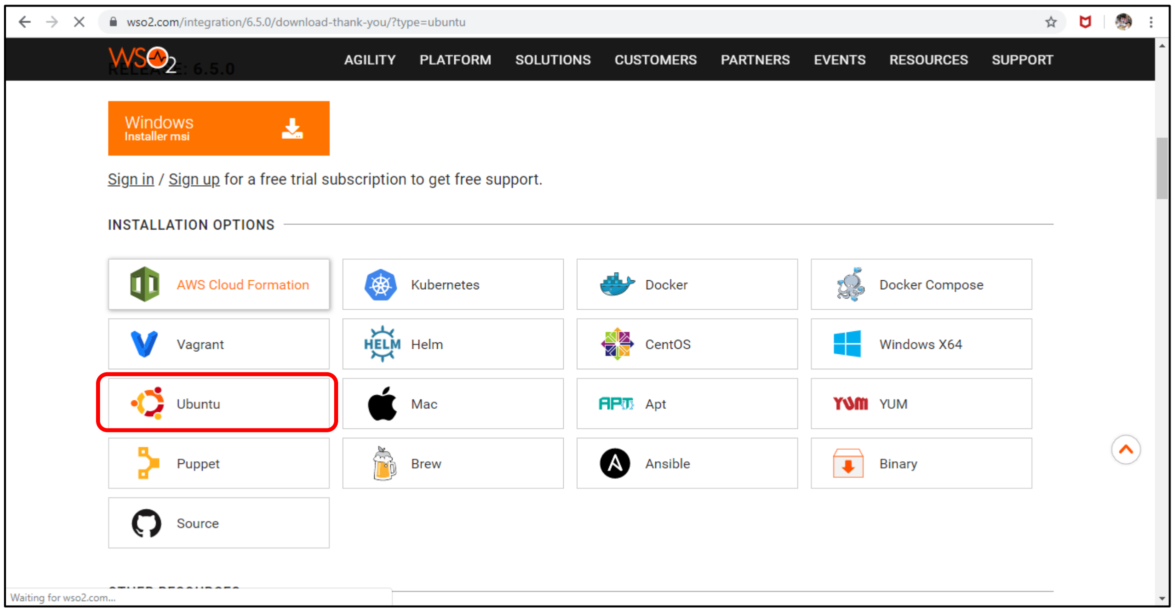
Aplikasi Mango Pest Identifier, merupakan aplikasi mobile berbasis android yang digunakan sebagai antar muka pengenalan hama / penyakit pada buah mangga. Aplikasi ini dapat memproses data dari citra gambar yg di simpan di memory piranti mobile maupun dari kamera langsung. Cara kerja aplikasi ini adalah dengan cara mengubah data image menjadi data encoded kemudian dikirimkan melalui API / webservice ke machine learning engine server. Orkestrasi service akan memproses data encoded tersebut menjadi file, kemudian menjalankan engine augmentasi gambar untuk proses identiikasi, membaca hasil identifikasi dan mengirimkan data hasil identifikasi sebagai response ke mobile apps. Selanjutnya mobile apps akan menampilkan hasil identifikasi, proses ini terjadi dalam hitungan detik. Selain proses identifikasi, pengguna mobile apps juga dapat mengakses knowledge base mengenai hama mangga.

B. Manual Instalasi

1. Middleware / Enterprises Integrator

Enterprise Integrator merupakan middleware yang digunakan untuk membangun webservice berbasis SOA (service oriented architecture). Middleware yang digunakan adalah WSO2 Enterprise Integrator, merupakan ESB (Enterprises Service Bus) platform yang berbasis open source yang sangat memudahkan dalam proses integrasi. Middleware ini support 2 jenis service yaitu SOAP dan REST. Dalam 1 proses development, dapat sekaligus menghasilkan 2 jenis service baik SOAP maupun REST. Adanya Service orchestration dalam platform ESB memungkinkan kita untuk dapat mengintegrasikan berbagai jenis service dengan cukup mudah. Berikut ini adalah langkah – langkah instalasi WSO2 Enterprises Integrator :

- Download Aplikasi WSO2 Integrator melalui halaman website, Pilih jenis system operasi yang digunakan, dalam hal ini adalah Ubuntu Server :

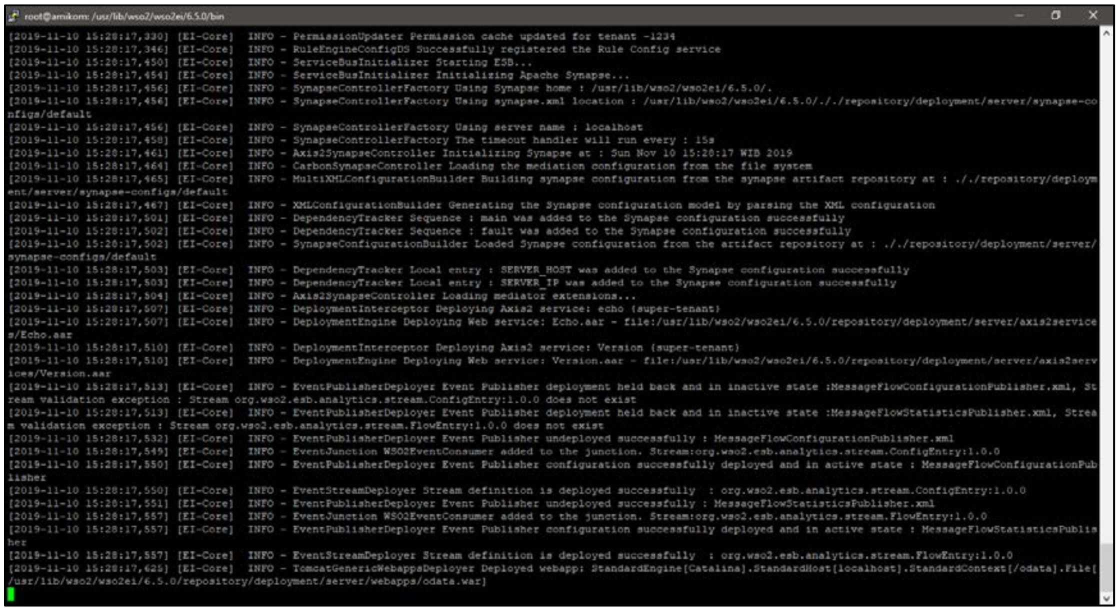


- Install aplikasi ke server yang akan digunakan dengan menjalankan perintah :

```
cvis@amikom$ sudo dpkg -i wso2ei-linux-installer-x64-6.5.0.deb
```

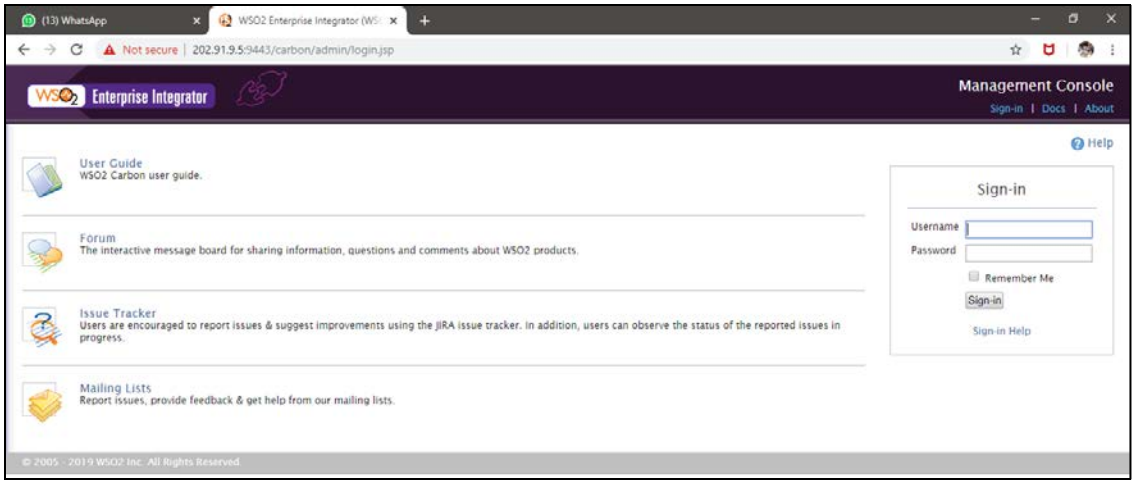
- Pastikan sebelum proses instalasi, sudah terinstall package JDK/JRE pada server
- Setelah proses instalasi selesai, jalankan middleware dengan menjalankan perintah berikut :

```
cvis@amikom$ service wso2ei-6.5.0-integrator start
```



```
root@amikom: /usr/lib/wso2/wso2ei/6.5.0/bin
[2019-11-10 15:28:17,330] [EI-Core]  INFO - PermissionUpdater Permission cache updated for tenant -1234
[2019-11-10 15:28:17,346] [EI-Core]  INFO - RuleEngineConfigDS Successfully registered the Rule Config service
[2019-11-10 15:28:17,450] [EI-Core]  INFO - ServiceBusInitializer Starting ESB...
[2019-11-10 15:28:17,454] [EI-Core]  INFO - ServiceBusInitializer Initializing Apache Synapse...
[2019-11-10 15:28:17,456] [EI-Core]  INFO - SynapseControllerFactory Using Synapse home : /usr/lib/wso2/wso2ei/6.5.0/.
[2019-11-10 15:28:17,456] [EI-Core]  INFO - SynapseControllerFactory Using synapse.xml location : /usr/lib/wso2/wso2ei/6.5.0/././repository/deployment/server/synapse-co
nfigs/default
[2019-11-10 15:28:17,456] [EI-Core]  INFO - SynapseControllerFactory Using server name : localhost
[2019-11-10 15:28:17,458] [EI-Core]  INFO - SynapseControllerFactory The timeout handler will run every : 15s
[2019-11-10 15:28:17,461] [EI-Core]  INFO - Axis2SynapseController Initializing Synapse at : Sun Nov 10 15:28:17 WIB 2019
[2019-11-10 15:28:17,464] [EI-Core]  INFO - CarbonSynapseController Loading the mediation configuration from the file system
[2019-11-10 15:28:17,465] [EI-Core]  INFO - MultiXMLConfigurationBuilder Building synapse configuration from the synapse artifact repository at : ././repository/deploym
ent/server/synapse-configs/default
[2019-11-10 15:28:17,467] [EI-Core]  INFO - XMLConfigurationBuilder Generating the Synapse configuration model by parsing the XML configuration
[2019-11-10 15:28:17,501] [EI-Core]  INFO - DependencyTracker Sequence : main was added to the Synapse configuration successfully
[2019-11-10 15:28:17,502] [EI-Core]  INFO - DependencyTracker Sequence : fault was added to the Synapse configuration successfully
[2019-11-10 15:28:17,502] [EI-Core]  INFO - SynapseConfigurationBuilder Loaded Synapse configuration from the artifact repository at : ././repository/deployment/server/
synapse-configs/default
[2019-11-10 15:28:17,503] [EI-Core]  INFO - DependencyTracker Local entry : SERVER_HOST was added to the Synapse configuration successfully
[2019-11-10 15:28:17,503] [EI-Core]  INFO - DependencyTracker Local entry : SERVER_IP was added to the Synapse configuration successfully
[2019-11-10 15:28:17,504] [EI-Core]  INFO - Axis2SynapseController Loading mediator extensions...
[2019-11-10 15:28:17,507] [EI-Core]  INFO - DeploymentInterceptor Deploying Axis2 service: echo {super-tenant}
[2019-11-10 15:28:17,507] [EI-Core]  INFO - DeploymentEngine Deploying Web service: Echo.aar - file:/usr/lib/wso2/wso2ei/6.5.0/repository/deployment/server/axis2service
s/Echo.aar
[2019-11-10 15:28:17,510] [EI-Core]  INFO - DeploymentInterceptor Deploying Axis2 service: Version {super-tenant}
[2019-11-10 15:28:17,510] [EI-Core]  INFO - DeploymentEngine Deploying Web service: Version.aar - file:/usr/lib/wso2/wso2ei/6.5.0/repository/deployment/server/axis2serv
ices/Version.aar
[2019-11-10 15:28:17,513] [EI-Core]  INFO - EventPublisherDeployer Event Publisher deployment held back and in inactive state :MessageFlowConfigurationPublisher.xml, St
ream validation exception : Stream org.wso2.esb.analytics.stream.ConfigEntry:1.0.0 does not exist
[2019-11-10 15:28:17,513] [EI-Core]  INFO - EventPublisherDeployer Event Publisher deployment held back and in inactive state :MessageFlowStatisticsPublisher.xml, Strea
m validation exception : Stream org.wso2.esb.analytics.stream.FlowEntry:1.0.0 does not exist
[2019-11-10 15:28:17,532] [EI-Core]  INFO - EventPublisherDeployer Event Publisher undeployed successfully : MessageFlowConfigurationPublisher.xml
[2019-11-10 15:28:17,549] [EI-Core]  INFO - EventJunction WSO2EventConsumer added to the junction. Stream:org.wso2.esb.analytics.stream.ConfigEntry:1.0.0
[2019-11-10 15:28:17,550] [EI-Core]  INFO - EventPublisherDeployer Event Publisher configuration successfully deployed and in active state : MessageFlowConfigurationPub
lisher
[2019-11-10 15:28:17,550] [EI-Core]  INFO - EventStreamDeployer Stream definition is deployed successfully  : org.wso2.esb.analytics.stream.ConfigEntry:1.0.0
[2019-11-10 15:28:17,551] [EI-Core]  INFO - EventPublisherDeployer Event Publisher undeployed successfully : MessageFlowStatisticsPublisher.xml
[2019-11-10 15:28:17,557] [EI-Core]  INFO - EventJunction WSO2EventConsumer added to the junction. Stream:org.wso2.esb.analytics.stream.FlowEntry:1.0.0
[2019-11-10 15:28:17,557] [EI-Core]  INFO - EventPublisherDeployer Event Publisher configuration successfully deployed and in active state : MessageFlowStatisticsPublis
her
[2019-11-10 15:28:17,557] [EI-Core]  INFO - EventStreamDeployer Stream definition is deployed successfully  : org.wso2.esb.analytics.stream.FlowEntry:1.0.0
[2019-11-10 15:28:17,625] [EI-Core]  INFO - TomcatGenericWebappsDeployer Deployed webapp: StandardEngine[Catalina].StandardHost[localhost].StandardContext[/odata].File[
/usr/lib/wso2/wso2ei/6.5.0/repository/deployment/server/webapps/odata.war]
```

- Akses middleware melalui alamat IP server, misal : https://202.91.9.5:9443/carbon



(13) WhatsApp | WSO2 Enterprise Integrator (WS
Not secure | 202.91.9.5:9443/carbon/admin/login.jsp

WSO2 Enterprise Integrator — Management Console
Sign-in | Docs | About

Help

User Guide
WSO2 Carbon user guide.

Forum
The interactive message board for sharing information, questions and comments about WSO2 products.

Issue Tracker
Users are encouraged to report issues & suggest improvements using the JIRA issue tracker. In addition, users can observe the status of the reported issues in progress.

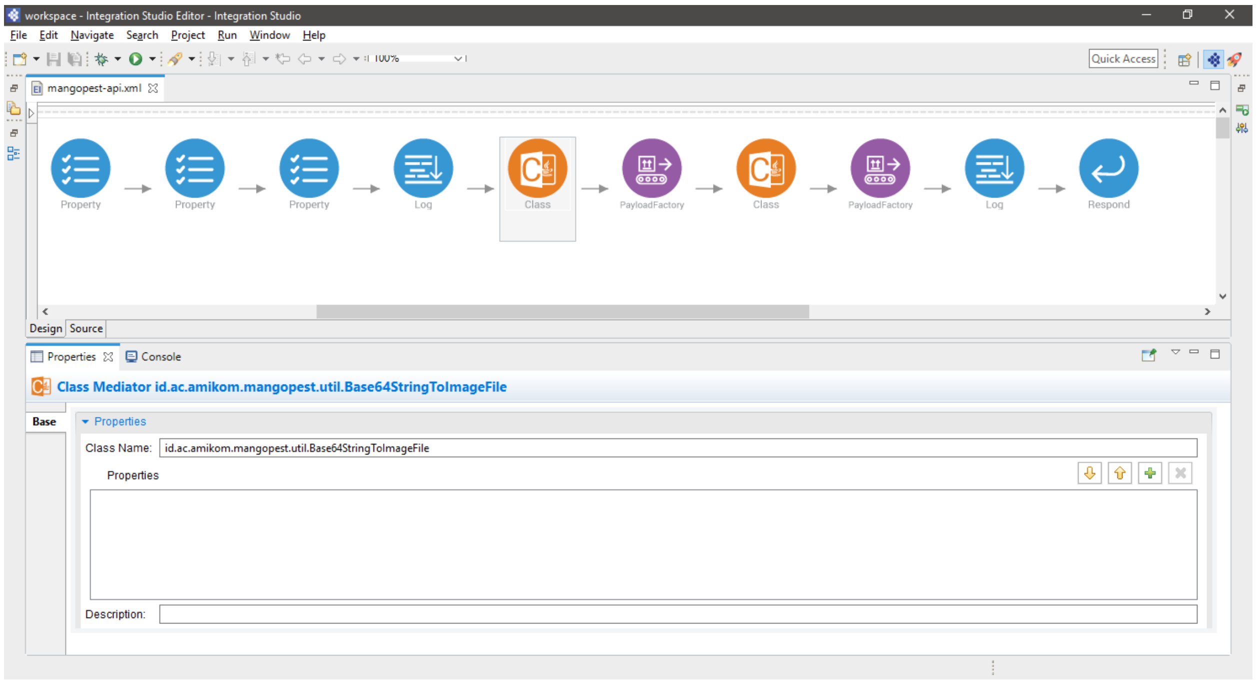
Mailing Lists
Report issues, provide feedback & get help from our mailing lists.

Sign-in
Username
Password
Remember Me
Sign-in
Sign-in Help

© 2005 - 2019 WSO2 Inc. All Rights Reserved.

2. Service Orchestration dengan WSO2 Integration Studio

- Buka Aplikasi Integration Studio
- Buat service orchestration sesuai dengan yang diinginkan
- Export project menjadi artifact untuk di deploy ke WSO2 Integrator



Class Mediator id.ac.amikom.mangopest.util.Base64StringToImageFile

Base

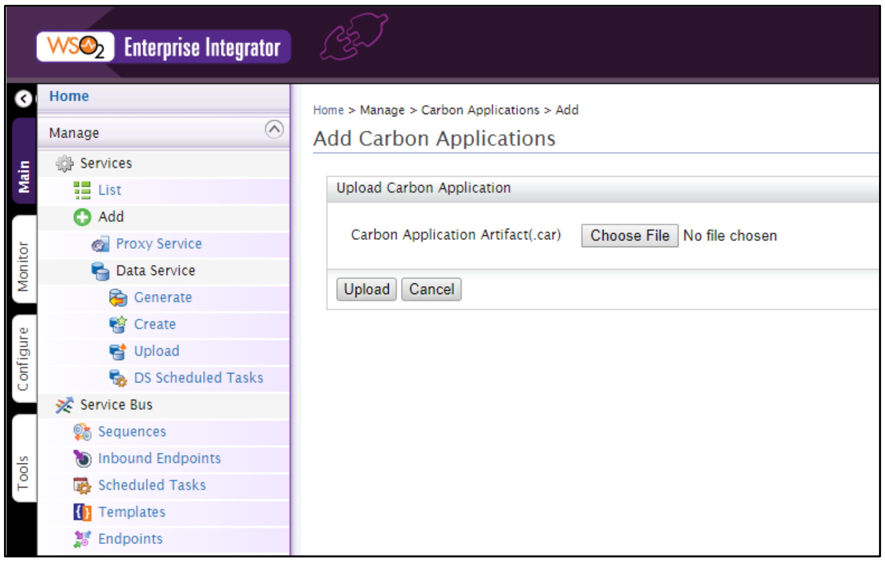
Properties

Class Name: id.ac.amikom.mangopest.util.Base64StringToImageFile

Properties

Description:

3. **Deploy Carbon Apps (Service / API Orchestration)**
   - Masuk ke webconsole, kemudian Pilih menu Main > Carbon Applications > Add
   - Upload file artifact yang telah di generate dari wso2 project
   - Jika berhasil maka akan muncul list API pada menu Main > Service Bus > API
   - API siap untuk di consume



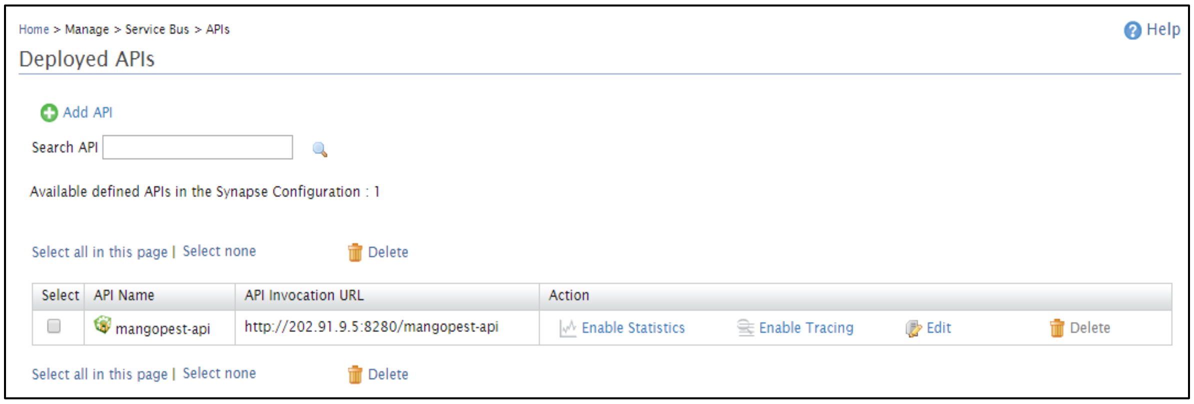
Home > Manage > Carbon Applications > Add

Add Carbon Applications

Upload Carbon Application

Carbon Application Artifact(.car) Choose File No file chosen

Upload Cancel


Home > Manage > Service Bus > APIs

Deployed APIs

Available defined APIs in the Synapse Configuration : 1

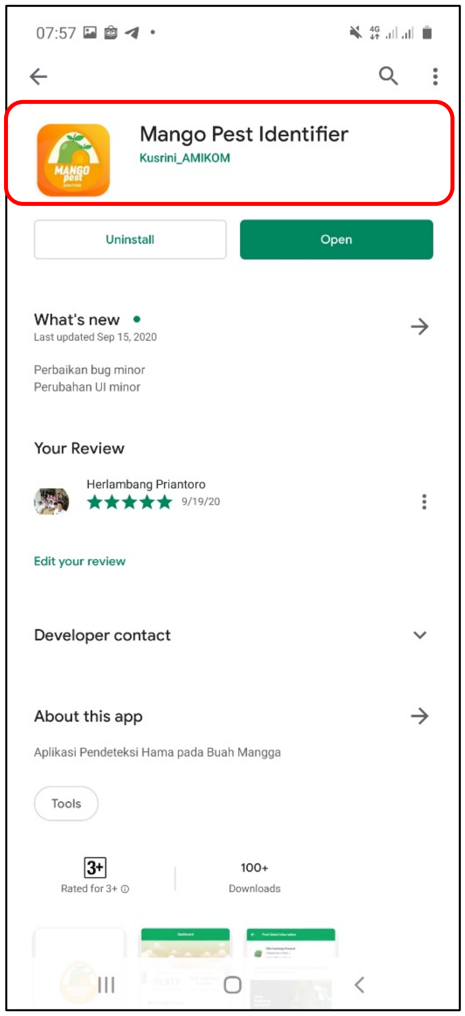
Select all in this page | Select none
Delete

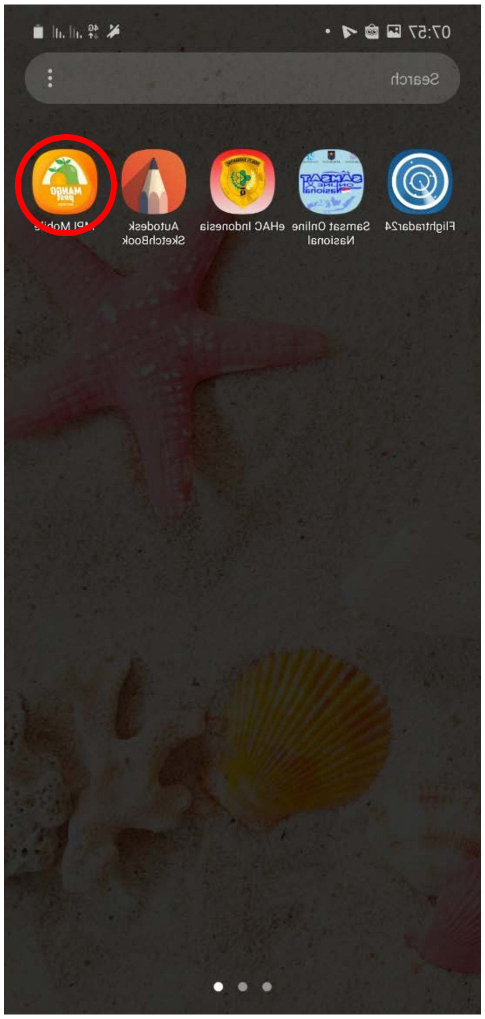
Select	API Name	API Invocation URL	Action			
☐	mangopest-api	http://202.91.9.5:8280/mangopest-api	Enable Statistics	Enable Tracing	Edit	Delete

Select all in this page | Select none
Delete

4. Mobile Apps

- Download Aplikasi Melalui Google Play Store dengan kata kunci Mango Pest Identifier



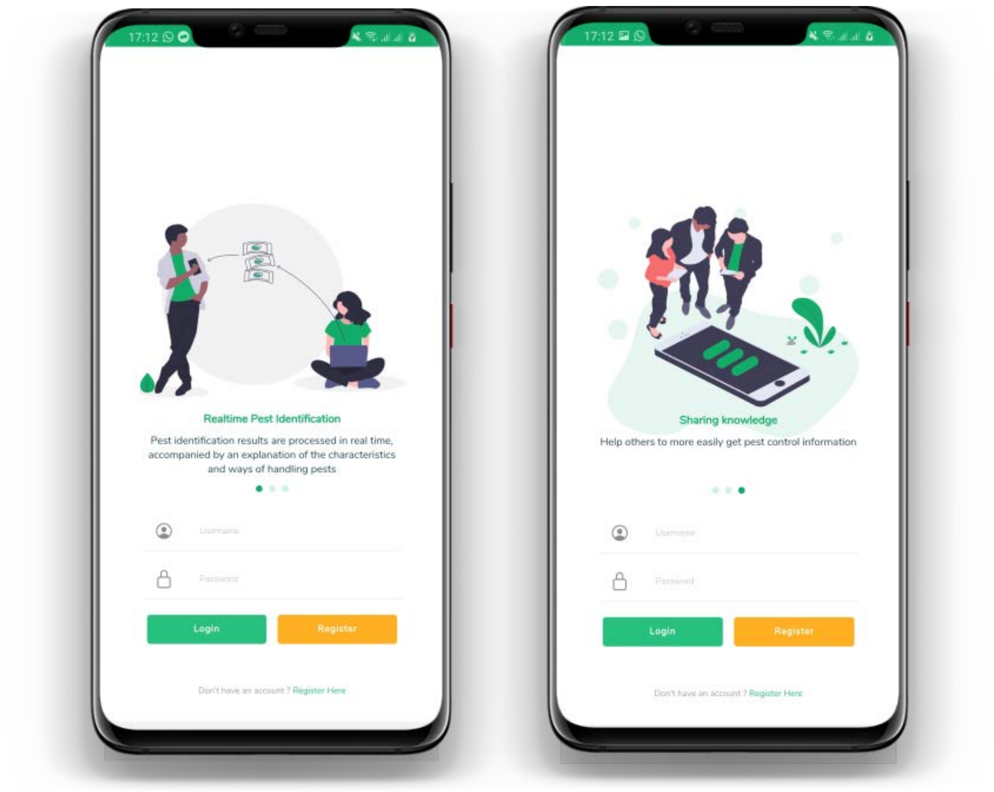


- Klik tombol install untuk memulai proses instalasi aplikasi Mango Pest Identifier
- Jika instalasi berhasil maka aplikasi akan muncul pada layar

C. Manual Penggunaan

1. Login

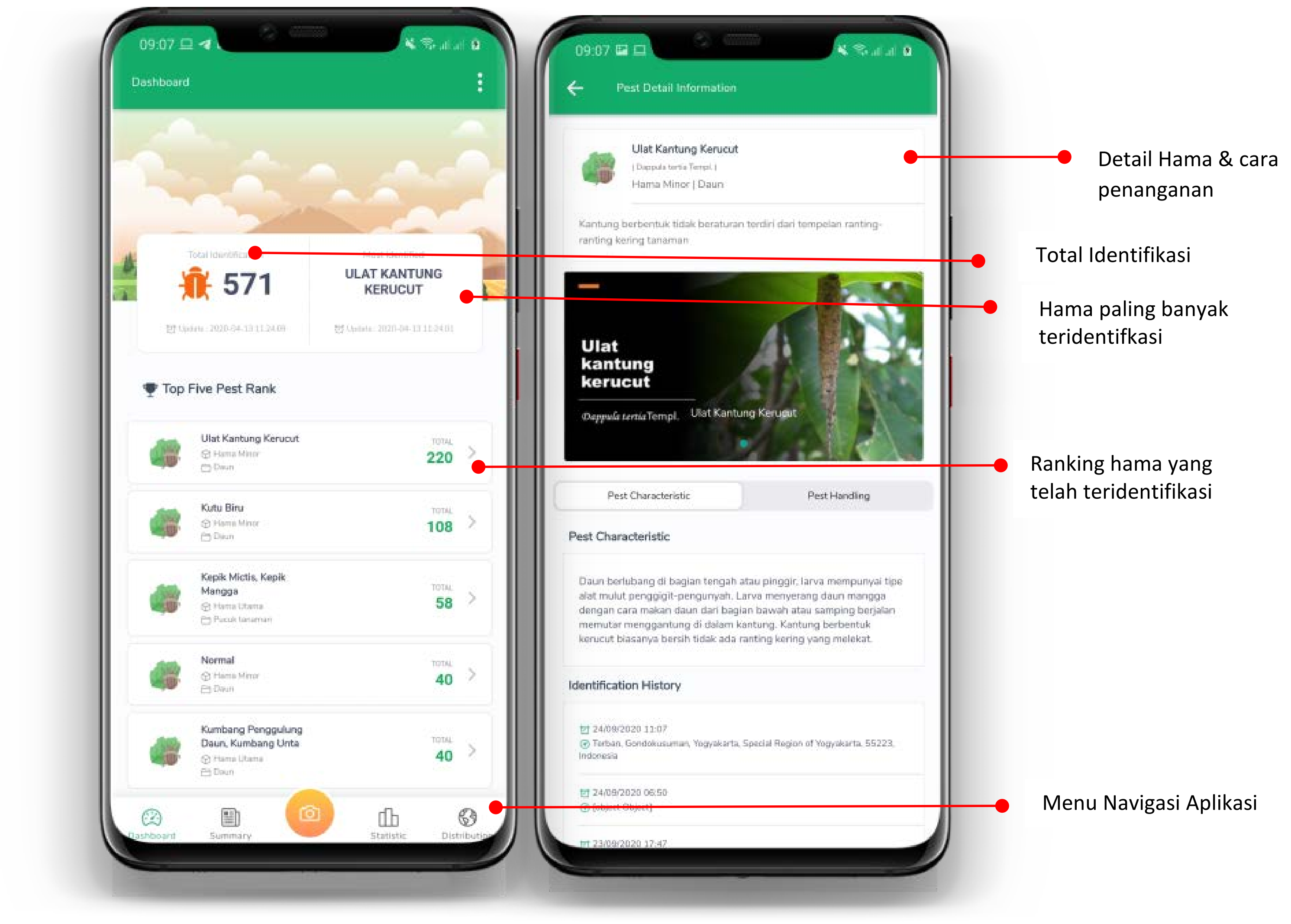
Merupakan halaman yang digunakan untuk meng-autentikasi user yang akan menggunakan aplikasi


17:12
Realtime Pest Identification
Pest identification results are processed in real time, accompanied by an explanation of the characteristics and ways of handling pests
Username
Password
Login
Register
Don't have an account ? Register Here
17:12
Sharing knowledge
Help others to more easily get pest control information
Username
Password
Login
Register
Don't have an account ? Register Here

Untuk masuk ke aplikasi, masukkan user & password yang sesuai kemudian klik tombol Login, jika berhasil maka akan muncul halaman dashboard aplikasi

2. Dashboard

Merupakan menu yang digunakan untuk menampilkan resume hasil identifikasi, ranking hama yang telah teridentifikasi. Untuk mengakses menu ini, tekan menu Dashboard melalui menu navigasi yang terletak di bagian bawah . Untuk melihat detail hama pada ranking dapat dilakukan dengan cara melakukan tap pada list ranking. Kemudian akan muncul halaman detail hama berikut history nya


Detail Hama & cara penanganan
Total Identifikasi
Hama paling banyak teridentifkasi
Ranking hama yang telah teridentifikasi
Menu Navigasi Aplikasi
Dashboard
Summary
Statistic
Distribution

09:07

Dashboard

Total Identification 571 · Update : 2020-04-13 11:24:09

Most Identified ULAT KANTUNG KERUCUT · Update : 2020-04-13 11:24:01

Top Five Pest Rank

Ulat Kantung Kerucut	Hama Minor, Daun	TOTAL 220
Kutu Biru	Hama Minor, Daun	TOTAL 108
Kepik Mictis, Kepik Mangga	Hama Utama, Pucuk tanaman	TOTAL 58
Normal	Hama Minor, Daun	TOTAL 40
Kumbang Penggulung Daun, Kumbang Unta	Hama Utama, Daun	TOTAL 40

09:07

Pest Detail Information

Ulat Kantung Kerucut
| Dappula tertia Templ. |
Hama Minor | Daun

Kantung berbentuk tidak beraturan terdiri dari tempelan ranting-ranting kering tanaman

Ulat kantung kerucut
Dappula tertia Templ. Ulat Kantung Kerucut

Pest Characteristic | Pest Handling

Pest Characteristic

Daun berlubang di bagian tengah atau pinggir, larva mempunyai tipe alat mulut penggigit-pengunyah. Larva menyerang daun mangga dengan cara makan daun dari bagian bawah atau samping berjalan memutar menggantung di dalam kantung. Kantung berbentuk kerucut biasanya bersih tidak ada ranting kering yang melekat.

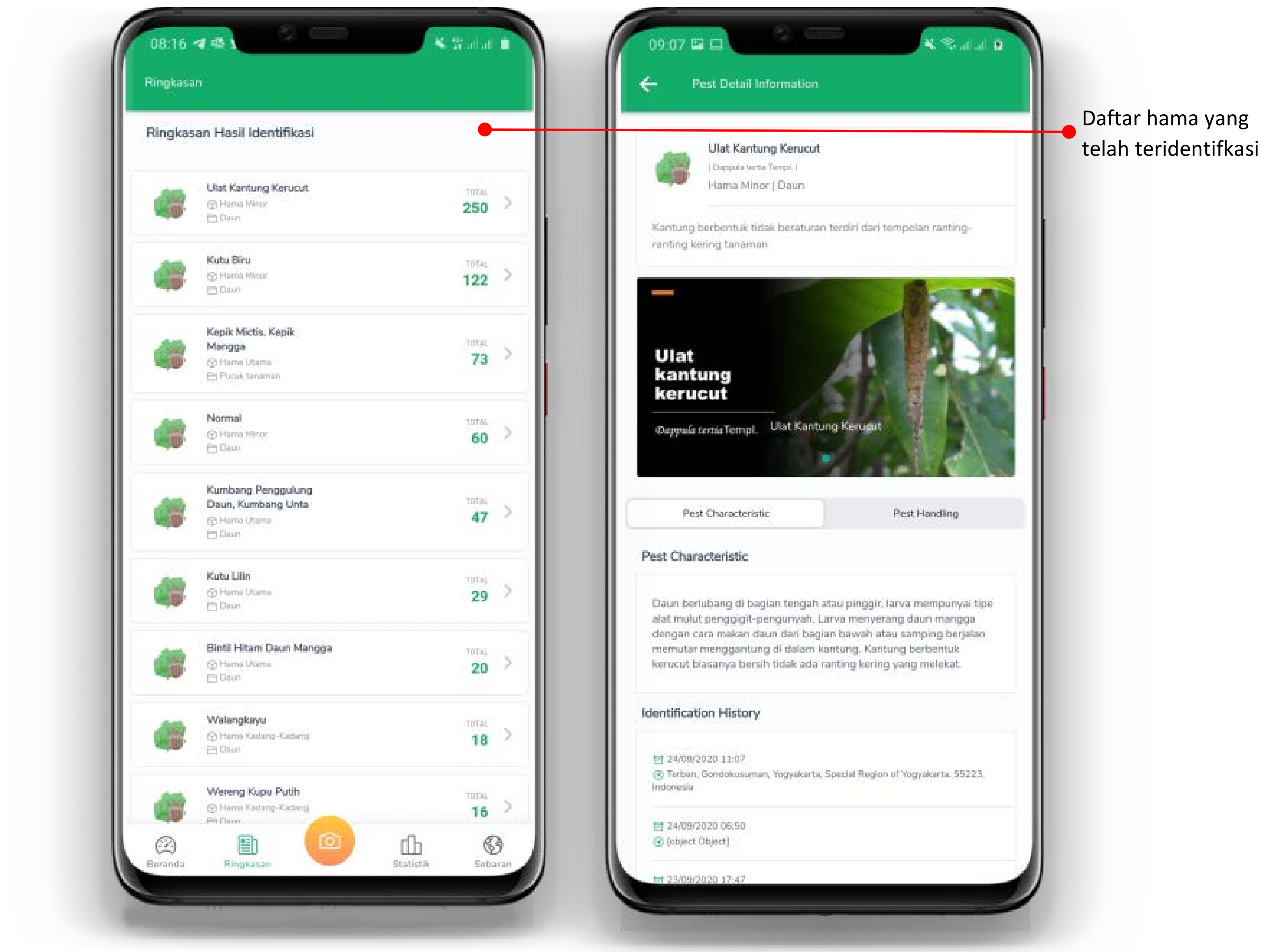
Identification History

24/09/2020 11:07
Terban, Gondokusuman, Yogyakarta, Special Region of Yogyakarta, 55223, Indonesia

24/09/2020 06:50
[object Object]

23/09/2020 17:47

3. **Summary**
Merupakan menu yang digunakan untuk menampilkan summary daftar hama yang telah berhasil teridentifikasi melalui aplikasi. Tekan menu Summary melalui menu navigasi yang terletak pada bagian bawah. Maka akan muncul halaman summary identifikasi yg berisi urutan hama secara keseluruhan berdasarkan jumlah identifikasi. Dari daftar hama dapat dilihat detail masing-masing hama berikut penanganan dan Riwayat identifikasi yang telah dilakukan


Ringkasan
Ringkasan Hasil Identifikasi
Ulat Kantung Kerucut · Hama Minor · Daun · TOTAL 250
Kutu Biru · Hama Minor · Daun · TOTAL 122
Kepik Mictis, Kepik Mangga · Hama Utama · Pucuk tanaman · TOTAL 73
Normal · Hama Minor · Daun · TOTAL 60
Kumbang Penggulung Daun, Kumbang Unta · Hama Utama · Daun · TOTAL 47
Kutu Lilin · Hama Utama · Daun · TOTAL 29
Bintil Hitam Daun Mangga · Hama Utama · Daun · TOTAL 20
Walangkayu · Hama Kadang-Kadang · Daun · TOTAL 18
Wereng Kupu Putih · Hama Kadang-Kadang · Daun · TOTAL 16
Beranda
Ringkasan
Statistik
Sebaran
Pest Detail Information
Ulat Kantung Kerucut
| Dappula tertia Templ. |
Hama Minor | Daun
Ulat kantung kerucut
Dappula tertia Templ.
Ulat Kantung Kerucut
Pest Characteristic
Pest Handling
Pest Characteristic
Identification History
24/09/2020 11:07
Terban, Gondokusuman, Yogyakarta, Special Region of Yogyakarta, 55223, Indonesia
24/09/2020 06:50
[object Object]
23/09/2020 17:47
Daftar hama yang telah teridentifkasi

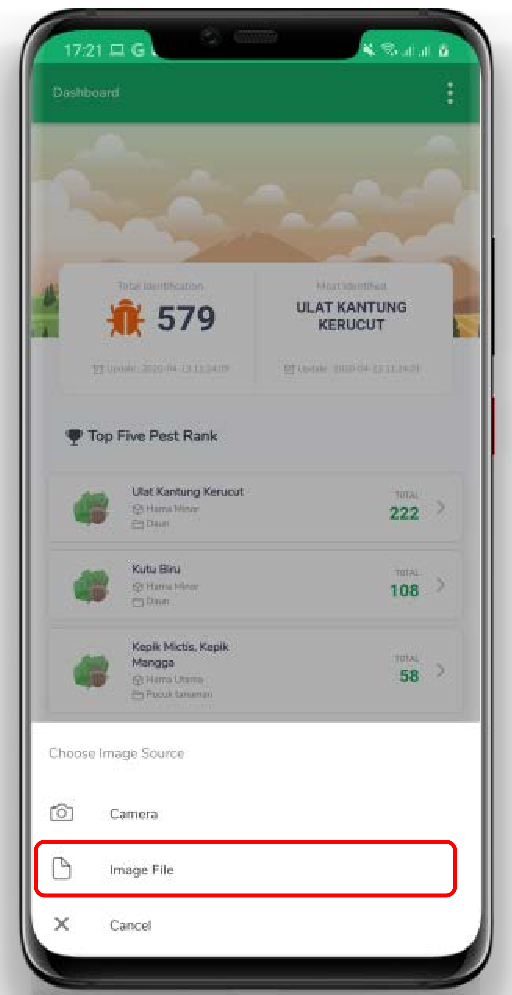
Kantung berbentuk tidak beraturan terdiri dari tempelan ranting-ranting kering tanaman

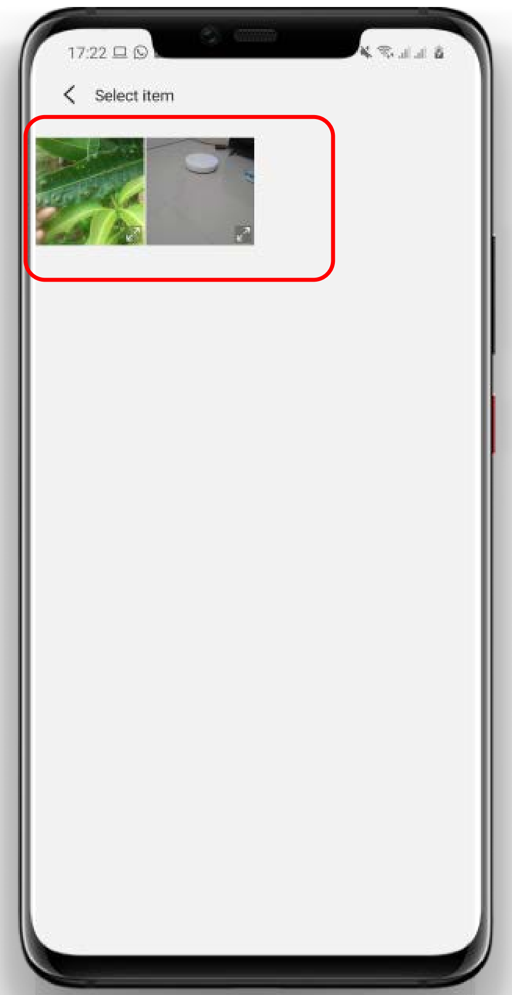
Daun berlubang di bagian tengah atau pinggir, larva mempunyai tipe alat mulut penggigit-pengunyah. Larva menyerang daun mangga dengan cara makan daun dari bagian bawah atau samping berjalan memutar menggantung di dalam kantung. Kantung berbentuk kerucut biasanya bersih tidak ada ranting kering yang melekat.

4. Scanner

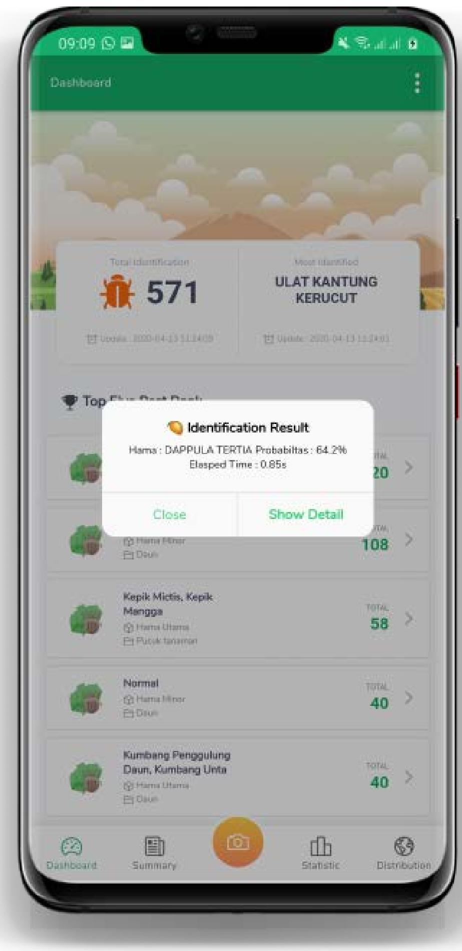
Merupakan menu yang berfungsi untuk melakukan identifkasi baik melalui upload image maupun dengan meng-capture gambar langsung melalui kamera. Untuk dapat memulai proses identifkasi ikuti langkah – langkah berikut :

- Tap pada button berwarna orange, Pilih sumber image yang akan diidentifikasi





- Tap OK, untuk memproses identifikasi. Image akan di decode dan dikirimkan melalui webservice, API akan menjalankan aplikasi augmentasi gambar dan mengembalikan hasil / response identifikasi ke mobile apps
- Hasil identifkasi akan muncul pada layar aplikasi, untuk melihat detail hasil identifikasi dapat dilakukan dengan memilih tombol show detail pada dialog hasil identifikasi
- Tekan Tab Pest Characteristic, untuk mengetahui karakteristik dari hama yang teridentifikasi
- Tekan Tab Pest Handling untuk mengetahui cara penanggulanan hama yang teridentifikasi
- Scroll ke menu history untuk melihat Riwayat identifikasi, termasuk lokasi dilakukan nya identifikasi tersebut

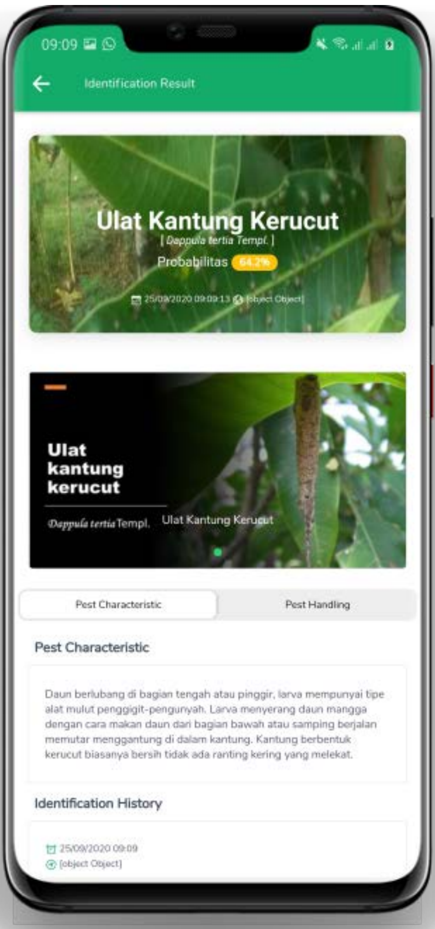


Identification Result

Hama : DAPPULA TERTIA Probabiltas : 64.2%
Elasped Time : 0.85s

Close | Show Detail

Hama Minor, Daun	TOTAL 108
Kepik Mictis, Kepik Mangga, Hama Utama, Pucuk tanaman	TOTAL 58
Normal, Hama Minor, Daun	TOTAL 40
Kumbang Penggulung Daun, Kumbang Unta, Hama Utama, Daun	TOTAL 40

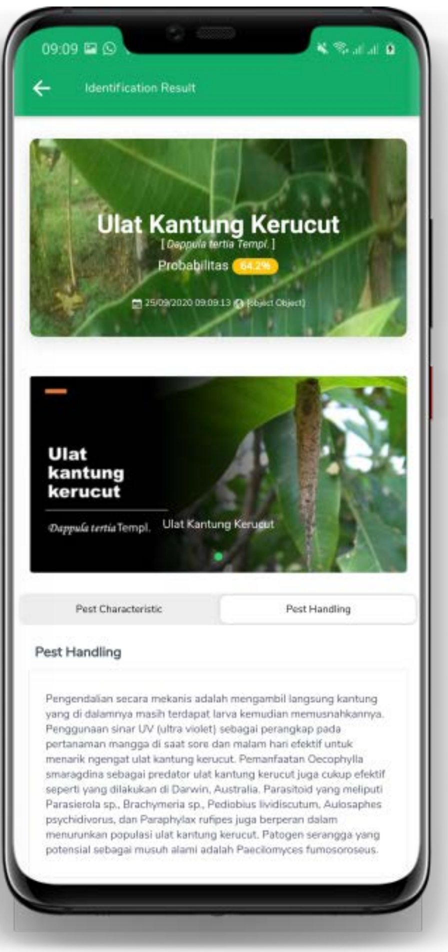


Pest Characteristic

Daun berlubang di bagian tengah atau pinggir, larva mempunyai tipe alat mulut penggigit-pengunyah. Larva menyerang daun mangga dengan cara makan daun dari bagian bawah atau samping berjalan memutar menggantung di dalam kantung. Kantung berbentuk kerucut biasanya bersih tidak ada ranting kering yang melekat.

Identification History

25/09/2020 09:09
[object Object]



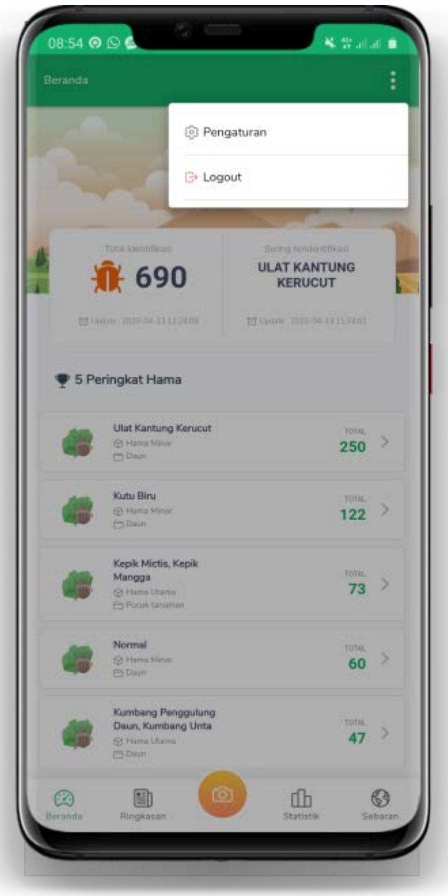
Pest Handling

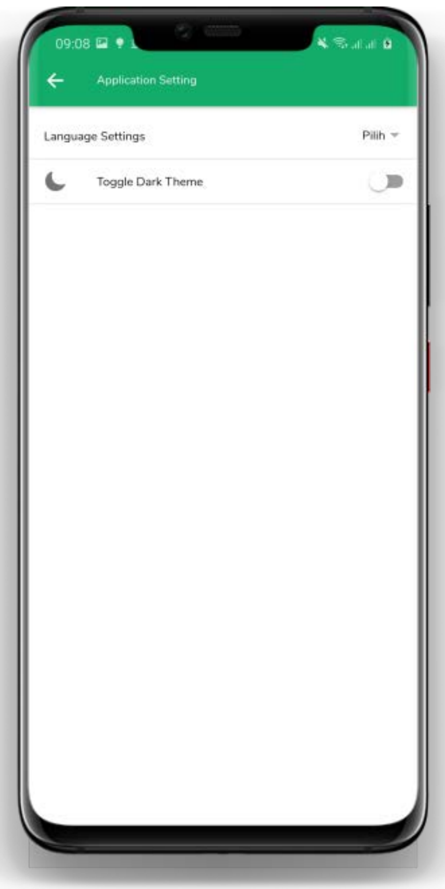
Pengendalian secara mekanis adalah mengambil langsung kantung yang di dalamnya masih terdapat larva kemudian memusnahkannya. Penggunaan sinar UV (ultra violet) sebagai perangkap pada pertanaman mangga di saat sore dan malam hari efektif untuk menarik ngengat ulat kantung kerucut. Pemanfaatan Oecophylla smaragdina sebagai predator ulat kantung kerucut juga cukup efektif seperti yang dilakukan di Darwin, Australia. Parasitoid yang meliputi Parasierola sp., Brachymeria sp., Pediobius lividiscutum, Aulosaphes psychidivorus, dan Paraphylax rufipes juga berperan dalam menurunkan populasi ulat kantung kerucut. Patogen serangga yang potensial sebagai musuh alami adalah Paecilomyces fumosoroseus.

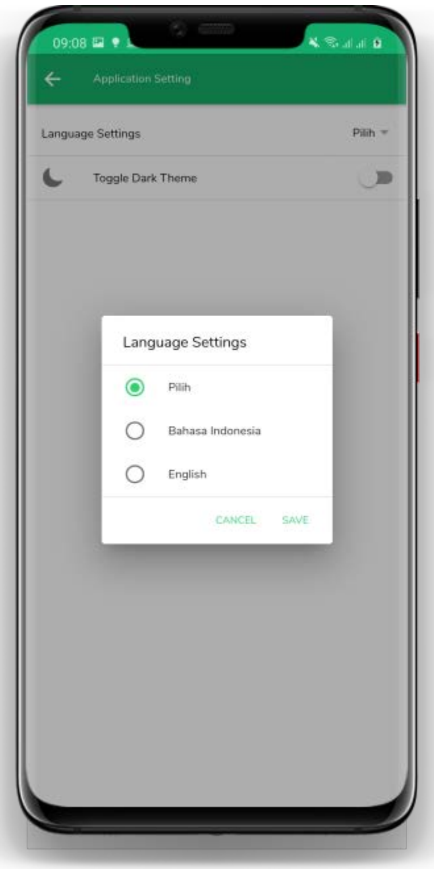
5. **Statistic**
   Merupakan menu yang digunakan untuk menampilkan statistik hasil identifikasi dalam bentuk grafik

6. **Distribution**
   Merupakan menu yang digunakan untuk menampilkan data sebaran hama berdasarkan hasil identifikasi yang telah dilakukan

7. **Settings**
   Merupakan menu yang digunakan untuk melakukan pengaturan aplikasi, pengaturan ini saat ini meliputi pengaturan Bahasa yang digunakan pada aplikasi. Saat ini aplikasi mensupport 2 bahasa yaitu Bahasa Indonesia dan Bahasa inggris. Untuk dapat melakukan pengaturan tap pada menu pojok kanan atas kemudian pilih pengaturan. Maka akan muncul halaman pengaturan. Pilih languages settings , kemudian pilih daftar Bahasa yang akan digunakan. Kemudian Pilih SAVE untuk menyimpan perubahan yang dimaksud.







D. Kode Program

Berikut adalah kode program untuk proses pengiriman data image ke API melalui mobile apps :

```
import { PestIdentificationModel } from "./../model/pest-identification-model";
import { Component } from "@angular/core";
import { ActionSheetController } from "@ionic/angular";
import { Camera, CameraOptions } from "@ionic-native/camera/ngx";
import { Geolocation } from "@ionic-native/geolocation/ngx";
import { Device } from "@ionic-native/device/ngx";
import { Router } from "@angular/router";
import {
  NavController,
  AlertController,
  LoadingController,
} from "@ionic/angular";
import { AndroidPermissions } from "@ionic-native/android-permissions/ngx";
import { LocationAccuracy } from "@ionic-native/location-accuracy/ngx";
import { PestModel } from '../model/pest-model';

@Component({
  selector: "app-tabs",
  templateUrl: "tabs.page.html",
  styleUrls: ["tabs.page.scss"],
})
```

```
export class TabsPage {
  public currentImage: any;
  public latLon: any;
  public locationDetail: any;
  public locationID: any;
  public imageResponse: any;

  constructor(
    public locationAccuracy: LocationAccuracy,
    public permission: AndroidPermissions,
    public loadingController: LoadingController,
    public alertCtrl: AlertController,
    public navController: NavController,
    public actionSheetController: ActionSheetController,
    public camera: Camera,
    public geolocation: Geolocation,
    public identification: PestIdentificationModel,
    public pestDetailModel : PestModel,
    public device: Device,
    public router: Router
  ) {}

  async presentLoading() {
    const loading = await this.loadingController.create({
      cssClass: "loading",
      message: "Identification Process ..",
      mode: "ios",
      backdropDismiss: false,
      spinner: "dots",
      duration: 4000,
    });
    await loading.present();
    const { role, data } = await loading.onDidDismiss();
  }

  async presentActionSheet() {
    const actionSheet = await this.actionSheetController.create({
      header: "Choose Image Source",
      cssClass: "my-custom-class",
      buttons: [
        {
          text: "Camera",
          role: "destructive",
          icon: "camera-outline",
          handler: () => {
            //open camera
            console.log("Camera clicked");
            this.takePicture();
          },
        },
        {
          text: "Image File",
          icon: "document-outline",
          handler: () => {
            this.selectPicture();
          },
        },
        {
          text: "Cancel",
          icon: "close",
          role: "cancel",
          handler: () => {
            console.log("Cancel clicked");
          },
        },
      ],
    });
    await actionSheet.present();
  }

  async presentalert(message: string, id: string, file_name: string) {
    const alert = await this.alertCtrl.create({
      header: "□ Identification Result",
      message: message,
      cssClass: "alertCancel",
      mode: "ios",
      buttons: [
        {
          text: "Close",
          role: "cancel",
          cssClass: "alertButton",
          handler: () => {
            console.log("Confirm Cancel");
          },
        },
        {
          text: "Show Detail",
          cssClass: "alertButton",
          handler: () => {
            //navigate to result page
            this.router.navigateByUrl(
              "/identification-result/" + id + "/" + file_name
            );
          },
        },
      ],
    });
    await alert.present();
  }

  selectPicture() {
    this.permission.requestPermissions([
      this.permission.PERMISSION.WRITE_EXTERNAL_STORAGE,
      this.permission.PERMISSION.ACCESS_COARSE_LOCATION,
      this.permission.PERMISSION.ACCESS_FINE_LOCATION,
    ]);
    const optionsGallery: CameraOptions = {
      quality: 50,
      targetWidth: 500,
      targetHeight: 500,
      sourceType: this.camera.PictureSourceType.PHOTOLIBRARY,
      destinationType: this.camera.DestinationType.DATA_URL,
      encodingType: this.camera.EncodingType.JPEG,
      mediaType: this.camera.MediaType.PICTURE,
      correctOrientation: true,
    };

    //request location
    this.locationAccuracy.canRequest().then((canRequest: boolean) => {
      this.locationAccuracy.request(this.locationAccuracy.REQUEST_PRIORITY_HIGH_ACCURACY);
      if (canRequest) {
        //get location detail
        this.geolocation
          .getCurrentPosition()
          .then((resp) => {
            this.latLon = resp.coords.latitude + "," + resp.coords.longitude;
            this.identification
              .getOpenStreetMapLocationDetail(
                resp.coords.latitude,
                resp.coords.longitude
              )
              .toPromise()
              .then((data: any) => {
                this.locationDetail = data;
                this.locationID = data.place_id;

                //insert master location
                this.identification
                  .saveLocationInfo(
                    data.place_id,
                    data.display_name,
                    data.address.county,
                    data.address.municipality,
                    data.address.state_district,
                    data.address.city,
                    data.address.postcode,
                    data.address.country,
                    data.address.country_code
                  )
                  .toPromise()
                  .then((data): any => {
                    //do nothing
                  });
              });
          })
          .catch((error) => {
            console.log("Error getting location", error);
            alert("Gagal menyimpan informasi lokasi " + JSON.stringify(error));
          });

        //get picture
        this.camera.getPicture(optionsGallery).then(
          (imageData) => {
            this.currentImage = "data:image/jpeg;base64," + imageData;
            var deviceID = this.device.uuid + "-" + this.device.model;
            this.presentLoading();
            //invoke identification request
            this.identification
              .doIdentificationProcess(
                imageData,
                deviceID,
                this.locationID,
                this.latLon
              )
              .toPromise()
              .then(
                (data: any) => {
                  var identificationResult = data.result;
                  this.pestDetailModel.getDetailPestByName(identificationResult.pest_type).toPromise
().then((detail:any)=>{
                    var pestDetail = detail.PestDetailResponse.PestDetail;
                    this.presentalert(
                      "Hama : " +
                      pestDetail.pestNameLatin.toUpperCase() +
                        "\nProbabiltas : " +
                        identificationResult.probability +
                        "\n Elapsed Time : " +
                        identificationResult.proc_time,
                      identificationResult.identification_id,
                      identificationResult.file_name
                    );
                  });
                },
                (error) => {
                  alert("Gagal melakukan identifikasi");
                }
              );
          },
          (err) => {
            // Handle error
            console.log("Camera issue:" + err);
          }
        );
      }
    });
  }

  takePicture() {
    //request permisisons
    this.permission.requestPermissions([
      this.permission.PERMISSION.CAMERA,
      this.permission.PERMISSION.WRITE_EXTERNAL_STORAGE,
      this.permission.PERMISSION.ACCESS_COARSE_LOCATION,
      this.permission.PERMISSION.ACCESS_FINE_LOCATION,
    ]);
    //camera setting
    const optionsCamera: CameraOptions = {
      quality: 75,
      destinationType: this.camera.DestinationType.DATA_URL,
      encodingType: this.camera.EncodingType.JPEG,
      mediaType: this.camera.MediaType.PICTURE,
      targetWidth: 500,
      targetHeight: 500,
      saveToPhotoAlbum: false,
      correctOrientation: true,
    };

    //get location detail
```

```
this.locationAccuracy.canRequest().then((canRequest: boolean) => {
  this.locationAccuracy.request(this.locationAccuracy.REQUEST_PRIORITY_HIGH_ACCURACY);
  if (canRequest) {
    this.geolocation
      .getCurrentPosition()
      .then((resp) => {
        this.latLon = resp.coords.latitude + "," + resp.coords.longitude;
        this.identification
          .getOpenStreetMapLocationDetail(
            resp.coords.latitude,
            resp.coords.longitude
          )
          .toPromise()
          .then((data: any) => {
            this.locationDetail = data;
            this.locationID = data.place_id;

            //insert master location
            this.identification
              .saveLocationInfo(
                data.place_id,
                data.display_name,
                data.address.county,
                data.address.municipality,
                data.address.state_district,
                data.address.city,
                data.address.postcode,
                data.address.country,
                data.address.country_code
              )
              .toPromise()
              .then((data): any => {
                //do nothing
              });
          });
      })
      .catch((error) => {
        console.log("Error getting location", error);
        alert("Gagal melakukan identifikasi " + error);
      });

    //get picture
    this.camera.getPicture(optionsCamera).then(
      (imageData) => {
        this.currentImage = "data:image/jpeg;base64," + imageData;
        var deviceID = this.device.uuid + "-" + this.device.model;

        this.presentLoading();

        //invoke identification request
        this.identification
          .doIdentificationProcess(
            imageData,
            deviceID,
            this.locationID,
            this.latLon
          )
```

```
              .toPromise()
              .then(
                (data: any) => {
                  var identificationResult = data.result;
                  this.pestDetailModel.getDetailPestByName(identificationResult.pest_type).toPromise
().then((detail:any)=>{
                    var pestDetail = detail.PestDetailResponse.PestDetail;
                    this.presentalert(
                      "Hama : " +
                      pestDetail.pestNameLatin.toUpperCase() +
                        "\nProbabiltas : " +
                        identificationResult.probability +
                        "\n Elapsed Time : " +
                        identificationResult.proc_time,
                      identificationResult.identification_id,
                      identificationResult.file_name
                    );
                  });
                },
                (error) => {
                  alert("Gagal melakukan identifikasi");
                }
              );
          },
          (err) => {
            // Handle error
            console.log("Camera issue:" + err);
            alert("Camera issue " + err);
          }
        );
      }
    });
  }
}
```

```
import { HttpClient, HttpHeaders,HttpParams } from "@angular/common/http";
import { Injectable } from '@angular/core';

@Injectable({
    providedIn: 'root'
})

export class PestIdentificationModel {

    constructor(public httpClient: HttpClient) {}

    //insert master location
    public saveLocationInfo(locationID:string, displayName:string, villageName:string, municipality:
string, stateDistrict:string, state:string, postCode:string, country:string, countryCode:string){
        //endpoint
        var endpoint = "https://202.91.9.5:8246/mango-pest-identification-
api/v1/insertmasterlocation";

        //set form data
        let form = {
            locationID:locationID,
            displayName:displayName,
            villageName:villageName,
            municipality:municipality,
            stateDistrict:stateDistrict,
            state:state,
            postCode:postCode,
            country:country,
            countryCode:countryCode
            }

            //set header
            const header = new HttpHeaders({
                "Content-Type":"application/x-www-form-urlencoded",
                "Accept":"*/*"
            });

        //set options
        const httpOptions = { headers: header }
        return this.httpClient.post(endpoint, this.getFormUrlEncoded(form), httpOptions);
    }

    //submit identification data
    public doIdentificationProcess(base64data: any, deviceid : string, locationID:string, latLon:str
ing){
        //endpoint
        var endpoint = "https://202.91.9.5:8246/mango-pest-identification-api/v1/uploadimage";

        //generate filename
        var filename = this.generateFilename()+".jpg";

        //set form data
        let form = {
            deviceID:deviceid,
            fileName:filename,
            base64Data:base64data,
            command:'python3',
            coreFilename:'/home/cvis/mango-apps/core-apps/coba_mangga.py',
            imageFilepath:'/home/cvis/mango-apps/data-upload-mangga',
            imageFilename:filename,
            model:'/home/cvis/mango-apps/core-apps/C12.model',
            locationID:locationID,
            latLon:latLon
            }

            //set header
            const header = new HttpHeaders({
                "Content-Type":"application/x-www-form-urlencoded",
                "Accept":"*/*"
            });

        //set options
        const httpOptions = { headers: header }
        return this.httpClient.post(endpoint, this.getFormUrlEncoded(form), httpOptions);
    }

    //get location detail
    public getOpenStreetMapLocationDetail(lat:any, lon:any){
        var endpoint = "https://nominatim.openstreetmap.org/reverse?format=json&lat="+lat+"&lon="+lo
n;
        console.log(endpoint);
        return this.httpClient.get(endpoint);
    }

    //generate format file
    public generateFilename(): string {
        var currentDate = new Date();
        var year = currentDate.getFullYear();
        var month = currentDate.getMonth();
        var date = currentDate.getDate();
        var hours = currentDate.getHours();
        var minutes = currentDate.getMinutes();
        var second = currentDate.getSeconds();
        var mili = currentDate.getMilliseconds();
        var fileName = year.toString() + month.toString() + date.toString() + hours.toString() + min
utes.toString() + second.toString() + mili.toString();
        return fileName;
    }

    getFormUrlEncoded(toConvert) {
        const formBody = [];
        for (const property in toConvert) {
            const encodedKey = encodeURIComponent(property);
            const encodedValue = encodeURIComponent(toConvert[property]);
            formBody.push(encodedKey + '=' + encodedValue);
        }
        return formBody.join('&');
    }

}
```

Kode Program untuk API :

```
<api xmlns="http://ws.apache.org/ns/synapse" name="mangopest-api" context="/mangopest-api">
   <resource methods="POST" uri-template="/uploadimage">
      <inSequence>
         <property name="uri.var.deviceID" expression="//xformValues//deviceID//text()"
scope="default" type="STRING"/>
         <property name="uri.var.fileName" expression="//xformValues//fileName//text()"
scope="default" type="STRING"/>
         <property name="uri.var.base64Data" expression="//xformValues//base64Data//text()"
scope="default" type="STRING"/>
         <property name="uri.var.command" expression="//xformValues//command//text()"
scope="default" type="STRING"/>
         <property name="uri.var.coreFilename"
expression="//xformValues//coreFilename//text()" scope="default" type="STRING"/>
         <property name="uri.var.imageFilepath"
expression="//xformValues//imageFilepath//text()" scope="default" type="STRING"/>
         <property name="uri.var.imageFilename"
expression="//xformValues//imageFilename//text()" scope="default" type="STRING"/>
         <property name="uri.var.model" expression="//xformValues//model//text()"
scope="default" type="STRING"/>
         <property name="uri.var.locationID" expression="//xformValues//locationID//text()"
scope="default" type="STRING"/>
```

```
         <property name="uri.var.latLon" expression="//xformValues//latLon//text()"
scope="default" type="STRING"/>
         <enrich>
            <source type="body" clone="true"/>
            <target type="property" property="payloadRequest"/>
         </enrich>
         <class name="ac.id.amikom.mangopestapi.FileTransformer"/>
         <payloadFactory media-type="json">
            <format>{"responseCode":"00","status": "success","statusMessage" : "Image
succesfully Uploaded"}</format>
            <args/>
         </payloadFactory>
         <class name="ac.id.amikom.mangopestapi.core.ClassificationRunner"/>
         <payloadFactory media-type="json">
            <format>$1</format>
            <args>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('identificationResult')"/>
            </args>
         </payloadFactory>
         <log>
            <property xmlns:ns="http://org.apache.synapse/xsd" name="IDENTIFICATION RESULT"
expression="get-property('identificationResult')"/>
            <property name="PEST TYPE" expression="json-eval($.result.pest_type)"/>
            <property name="pestType" expression="json-eval($.result.pest_type)"/>
            <property name="probability" expression="json-eval($.result.probability)"/>
            <property name="identificationRespMessage" expression="get-
property('identificationResult')"/>
            <property name="payload" expression="get-property('payloadRequest')"/>
         </log>
         <property name="pestType" expression="json-eval($.result.pest_type)" scope="default"
type="STRING"/>
         <property name="probability" expression="json-eval($.result.probability)"
scope="default" type="STRING"/>
         <property name="procTime" expression="json-eval($.result.proc_time)" scope="default"
type="STRING"/>
         <payloadFactory media-type="xml">
            <format>
               <insertRequestLog>
                  <deviceID>$1</deviceID>
                  <fileName>$2</fileName>
                  <requestMessage>$3</requestMessage>
                  <responseMessage>$4</responseMessage>
                  <identificationResult>$5</identificationResult>
                  <identificationAccuracy>$6</identificationAccuracy>
                  <locationID>$7</locationID>
                  <latLon>$8</latLon>
               </insertRequestLog>
            </format>
            <args>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.deviceID')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.fileName')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('formattedReqPayload')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('identificationResult')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('pestType')"/>
```

```
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('probability')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.locationID')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.latLon')"/>
            </args>
         </payloadFactory>
         <send>
            <endpoint key="mangopestapi-data-service-EP"/>
         </send>
      </inSequence>
      <outSequence>
         <log>
            <property name="insertResponse" expression="$body"/>
            <property xmlns:ns="http://www.amikom.ac.id/MangoPestAPIDataService"
name="identificationID"
expression="//ns:RequestLogResponse/ns:RequestLog/ns:identificationID/text()"/>
         </log>
         <property xmlns:ns="http://www.amikom.ac.id/MangoPestAPIDataService"
name="identificationID"
expression="//ns:RequestLogResponse/ns:RequestLog/ns:identificationID/text()" scope="default"
type="STRING"/>
         <property xmlns:ns="http://www.amikom.ac.id/MangoPestAPIDataService" name="fileName"
expression="//ns:RequestLogResponse/ns:RequestLog/ns:fileName/text()" scope="default"
type="STRING"/>
         <property name="SC_ACCEPTED" value="false" scope="axis2" type="BOOLEAN"/>
         <property name="HTTP_SC" value="200" scope="axis2" type="STRING"/>
         <payloadFactory media-type="json">
            <format>
{"result":{"identification_id":"$1","file_name":"$2","pest_type":"$3","probability":"$4","proc
_time":"$5"}}                </format>
            <args>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('identificationID')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('fileName')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('pestType')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('probability')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('procTime')"/>
            </args>
         </payloadFactory>
         <respond/>
      </outSequence>
      <faultSequence/>
   </resource>
   <resource methods="POST" uri-template="/insertmasterlocation">
      <inSequence>
         <property name="uri.var.locationID" expression="//xformValues//locationID//text()"
scope="default" type="STRING"/>
         <property name="uri.var.displayName" expression="//xformValues//displayName//text()"
scope="default" type="STRING"/>
         <property name="uri.var.villageName" expression="//xformValues//villageName//text()"
scope="default" type="STRING"/>
         <property name="uri.var.municipality"
expression="//xformValues//municipality//text()" scope="default" type="STRING"/>
         <property name="uri.var.stateDistrict"
expression="//xformValues//stateDistrict//text()" scope="default" type="STRING"/>
```

```
         <property name="uri.var.state" expression="//xformValues//state//text()"
scope="default" type="STRING"/>
         <property name="uri.var.postCode" expression="//xformValues//postCode//text()"
scope="default" type="STRING"/>
         <property name="uri.var.country" expression="//xformValues//country//text()"
scope="default" type="STRING"/>
         <property name="uri.var.countryCode" expression="//xformValues//countryCode//text()"
scope="default" type="STRING"/>
         <payloadFactory media-type="xml">
            <format>
               <insertMasterLocation>
                  <placeID>$1</placeID>
                  <displayName>$2</displayName>
                  <villageName>$3</villageName>
                  <municipality>$4</municipality>
                  <stateDistrict>$5</stateDistrict>
                  <state>$6</state>
                  <postCode>$7</postCode>
                  <country>$8</country>
                  <countryCode>$9</countryCode>
               </insertMasterLocation>
            </format>
            <args>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.locationID')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.displayName')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.villageName')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.municipality')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.stateDistrict')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.state')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.postCode')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.country')"/>
               <arg xmlns:ns="http://org.apache.synapse/xsd" evaluator="xml" expression="get-
property('uri.var.countryCode')"/>
            </args>
         </payloadFactory>
         <send>
            <endpoint key="mangopestapi-data-service-EP"/>
         </send>
      </inSequence>
      <outSequence>
         <property name="SC_ACCEPTED" value="false" scope="axis2" type="BOOLEAN"/>
         <property name="HTTP_SC" value="200" scope="axis2" type="STRING"/>
         <payloadFactory media-type="json">
            <format>{"responseCode":"00","status": "success","statusMessage" : "Location
succesfully Saved"}</format>
            <args/>
         </payloadFactory>
         <respond/>
      </outSequence>
      <faultSequence/>
   </resource>
</api>
```