

# GHC Area Core Machine Learning

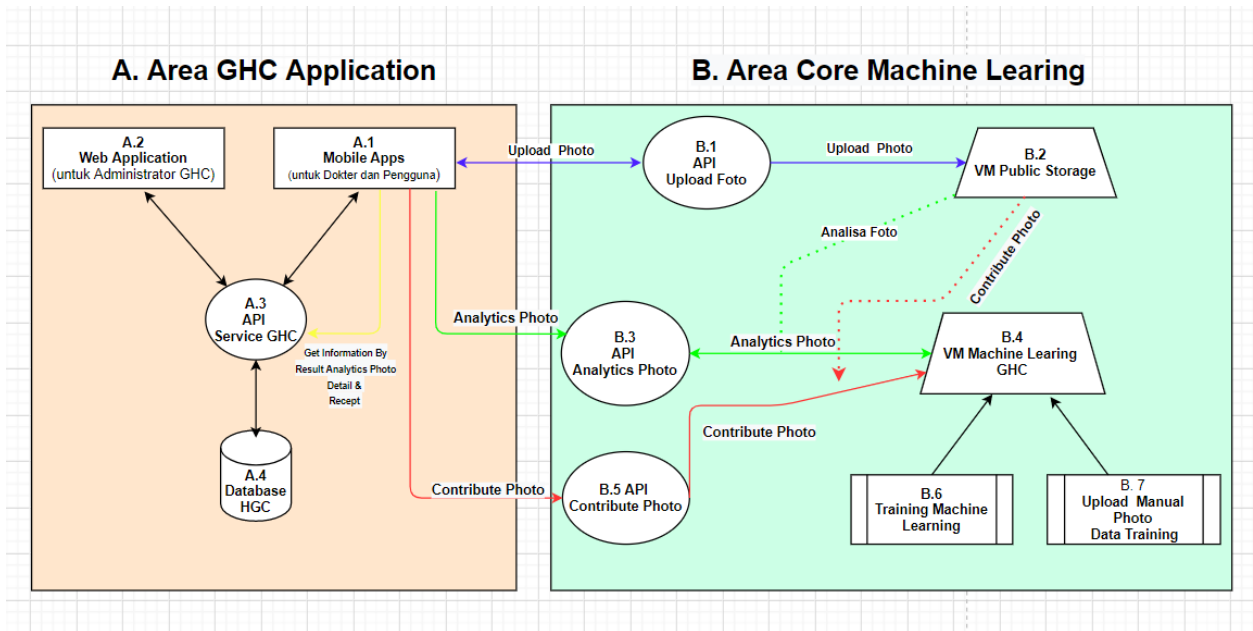
Mendeteksi Penyakit Hewan dari Foto Faces Hewan

Menggunakan Open Source ML - Tensor Flow

## Daftar Isi

|  |    |
|--|----|
| A. General Skema dan Flow Application .....                              | 2  |
| B. Pendekatan Arsitekur .....  | 3  |
| C. Service GHC Machine Learning (Teknologi & Spesifikasi Hardware) ..... | 4  |
| D. Machine Learning Training Data & Kontribusi Data / Foto .....         | 5  |
| E. Flow Akses API di Frontend / Mobile Apps .....                        | 6  |
| F. Dokumentasi API .....   | 7  |
| G. Source Code Repository .....  | 10 |
| H. Task List Area Core Machine Learning .....                            | 10 |

## A. General Skema dan Flow Application



1. Tahap 1 – Garis Biru, Mobile Apps upload foto bisa single atau multiple foto ke B.1 Upload Foto kemudian foto-foto yang di upload akan di simpan di VM Public Storage, kemudian B.1 Upload Foto akan memberikan return alamat public berupa URL (http / https) dari foto tersebut
2. Tahap 2 – Garis Hijau, Setelah Mobile Apps mendapatkan return URL (http / https) dari alamat public foto tersebut (hasil dari tahap pertama), kemudian URL Foto tesebut di kirim ke endpoint API B.3 API Analytis Photo untuk dianalisa untuk medapatkan return probabilitas dan clasifcation image (berupa kode\_clasifikasi\_penyakit)
3. Tahap 3 – Garis Kuning, Setelah Mobile Apps mendapatkan return URL probabilitas dan clasifcation image (berupa kode\_clasifikasi\_penyakit) dari API B.3 API Analytis Photo, kemudian kode\_clasifikasi\_penyakit di teruskan ke endpoint API A.3 API Service GHC untuk mendapatkan Detail Informasi penyakit dan resep obat yang disarankan
4. Tahap 4 – Garis Merah (Optional), adalah service apabila dokter akan memberikan kontribusi foto. Berdasarkan foto yang di upload dari point no 1, dokter dapat memberikan tag,url\_foto dan memilih kode\_clasifikasi\_penyakit dan diteruskan ke endpoint API B.5 Api Contribute Photo yang akan melakukan download dari dari B.2 VM Public Storage untuk di masukan ke bahan Training Data Machine Learning agar semakin system ML semakin cerdas

Alasan setiap tahap dijadikan sebuah service teripisah adalah sebagai berikut :

1. Memastikan setiap tahap sudah valid sebelum ke tahapan selanjutnya
2. Membagi proses setiap tahap berada di resource yang berbeda sehingga masing-masing service akan dedicate menyelesaikan tugas nya lebih cepat
3. Membuat service menjadi lebih mandiri / fleksible / tidak dependency dengan service yang lain. Contoh apabila App Mobile menyediakan fitur user langsung insert URL Foto maka bisa langsung ke endpoint API B.3 API Analytis Photo (tidak perlu melalui Tahap 1, atau apabila dokter ingin memberikan kontribusi foto maka bisa langsung ke endpoint API B.5 Api Contribute Photo
4. Merupakan pendekatan arsitektur Microservice (penjelasan lebih lengkap tentang ini ,ada sub bab B. Pendekatan Arsitektur)

## B. Pendekatan Arsitektur

Area B. Area Core Machine Learning development disarankan menggunakan pendekatan microservice dengan tujuan :

- Mandiri, Setiap Service bisa mandiri atau depecency tidak tergantung yang lain, Apabila ada salah satu service yang tidak available maka tidak akan membuat seluruh service mati
- Isolasi, sehingga apabila salah satu service dilakukan penyesuaian source code maka tidak akan mengganggu service yang lain
- Reduce Complexity, setiap service bisa dikembangkan dikemudian hari lebih mudah dan di lanjutkan oleh programmer lain karna setiap service memiliki satu flow yang spesifik dan tidak ada relasi secara langsung dengan service yang lain (ini akan memudahkan dalam mengalokasi SDM dan tidak bergantung pada seorang programmer tertentu, dan biaya SDM akan lebih mudah diukur)
- Efisien Hardware, setiap service akan dapat dialokasikan resource hardware sesuai dengan kebutuhan (dan yang tentunya ini akan berdampak pada efisiensi biaya hardware di level production)

## C. Service GHC Machine Learning (Teknologi & Spesifikasi Hardware)

| Service                                | Keterangan  | Teknologi  | Recomendtion Hardware Minimum di Production Starting  |
|--|---|--|---|
| Api Upload Photo                       | <ul style="list-style-type: none"> <li>• Api yang memilik tugas untuk menerima multiple upload foto dan menyimpan di public store</li> <li>• Kemudian API akan memberikan return alamat url (http:// https://) dari lokasi foto tersebut berada</li> <li>• Sebelum di disimpan di store publicakan di rezie terlebih dahulu</li> <li>• Untuk menghemat storage scara otomatis foto akan dihapus secara berkala</li> </ul> | <ul style="list-style-type: none"> <li>• PHP 7.X - Native</li> <li>• PHP GDLibe for Resize Photo</li> <li>• OS Centos</li> </ul>   | <ul style="list-style-type: none"> <li>• Bisa server on premies atau cloud</li> <li>• Procesor Xeon dual Core</li> <li>• Memory 4 GB</li> <li>• Storage 100 GB</li> </ul> |
| Api Analytics Photo + ML Training Data | <ul style="list-style-type: none"> <li>• Api yang memilik tugas menganalisa foto</li> <li>• Kemudian API akan memberikan return berupa prediksi sakit dan probabilitasnya</li> </ul>  | <ul style="list-style-type: none"> <li>• Pyhton 3.9</li> <li>• Python Flask Microframework</li> <li>• Opensource ML Tensor Flow</li> <li>• OS Linux Ubuntu Server</li> </ul> | <ul style="list-style-type: none"> <li>• Bisa server on premies atau cloud</li> <li>• Procesor Xeon 8 Core</li> <li>• Memory 16 GB</li> <li>• Storage 30 GB</li> </ul>    |
| Api Contribute Photo                   | <ul style="list-style-type: none"> <li>• Api yang memilik tugas untuk menerima kontribusi foto dari dokter dengan tujuan untuk meningkat akurasi ML</li> <li>• Ketika API di akses menyertakan alamat URL foto dari public storage dan tagging/kode foto_penyakit dan foto tersebut akan disimpan di server ML bahan Training Data ML</li> </ul>  | <ul style="list-style-type: none"> <li>• Python Flask Microframework</li> <li>• OS Linux Ubuntu Server</li> </ul>  |   |

## D. Machine Learning Training Data & Kontribusi Data / Foto

Untuk meningkat akurasi ML diperlukan bahan berupa foto yang jadi bahan untuk training ML akan semakin cerdas untuk mengklasifikasikan image, cara ada dua yaitu sebagai berikut :

1. Cara Pertama B.7 Upload Manual Photo ke VM Machine Learning melalui winscp atau ftp client, dengan ukuran resolusi yang sama (contoh 150pixel x proposal), satu nama directory mewakili nama atau kode dari penyakit hewan

| Name               | Date modified      | Type        | Size |
|--------------------|--------------------|-------------|------|
| foto_tinja_sakit_a | 10/17/2021 4:11 AM | File folder |      |
| foto_tinja_sakit_b | 10/17/2021 4:11 AM | File folder |      |
| foto_tinja_sakit_c | 10/17/2021 4:11 AM | File folder |      |
| foto_tinja_sakit_d | 10/17/2021 4:11 AM | File folder |      |
| foto_tinja_sakit_e | 10/17/2021 4:11 AM | File folder |      |
| foto_tinja_sakit_f | 10/17/2021 4:11 AM | File folder |      |

2. Cara Kedua melakukan kontribusi foto yang dikirim oleh dokter melalui Mobile Apps, dokter dapat memberikan tag,url\_foto dan memilih kode\_clasifikasi\_penyakit dan diteruskan ke endpoint **API B.5 Api Contribute Photo** yang akan melakukan download untuk di masukan ke bahan Training Data Machine Learning agar system ML semakin cerdas
3. Script Training ML yaitu **training.py** bisa jalan otomatis setiap waktu tertentu atau dijalankan secara manual secara berkala untuk meningkatkan kecerdasan ML Ketika ada bahan / data training yang baru

## E. Flow Akses API di Frontend / Mobile Apps

### #upload foto ke **api.b.1.upload\_foto**

If (upload\_foto ke **api.b.1.upload\_foto**) {

#### #jika upload foto berhasil ambil semua return url public foto

foreach (**api.b.1.upload\_foto** as data\_url\_foto ) {

#### #kirim url foto untuk di analisa ML

if (data\_url\_foto ke **api.b.3.analytic\_foto**) {

#### #kirim kode\_foto\_penyakit untuk mendapat detail sakit dan resep

if( akses ke api **api.a3.Server.ghc**) {

#### #simpan detail sakit dan resep, url foto, kode\_penyakit

var list\_success\_ml = **api.a3.Server.ghc**

}

}

}

### #jika aplikasi ingin mengirimkan kontribusi

foreach(list\_success\_ml as data) {

#### #kirim url foto url dan tag/kode penyakit

If (upload\_foto ke **api.b.5.contribute\_phot**) {

}

}

## F. Dokumentasi API

### API B.3 API Analytic Photo

The screenshot shows a REST client interface with the following details:

- Request Method:** GET
- Request URL:** `http://localhost:5000/analysis?key=HGC87624&url_foto=https://2.bp.blogspot.com/-3N_KfaNWxz4/V5oaqCgQX3I/AAAAAAAAACeA/XBTnU9ptuk0DfX8xwAxlERncZ1KuPsZgCLcB/s320/cacing.jpg`
- Query Params:**

| KEY  | VALUE  | DESCRIPTION |
|--|--|-------------|
| <input checked="" type="checkbox"/> key      | HGC87624   |             |
| <input checked="" type="checkbox"/> url_foto | https://2.bp.blogspot.com/-3N_KfaNWxz4/V5oaqCgQX3I/AAAAAAAAACeA/XBTnU9ptuk0DfX8xwAxlERncZ1KuPsZgCLcB/s320/cacing.jpg |             |
| Key  | Value  | Description |
- Response Status:** 200 OK, Time: 1079 ms, Size: 438 B
- Response Body (JSON):**

```
{  "code": 200,  "key": "HGC87624",  "message": "Foto berhasil di analisa",  "prediksi": "foto_tinja_sakit_b",  "probabilitas": 99.99507665634155,  "url_foto": "https://2.bp.blogspot.com/-3N_KfaNWxz4/V5oaqCgQX3I/AAAAAAAAACeA/XBTnU9ptuk0DfX8xwAxlERncZ1KuPsZgCLcB/s320/cacing.jpg"}
```

Parameter Input :

- Key = key yang dibuat manual untuk mengakses API
- url\_foto = alamat URL public dari foto yang akan di analisa ML
- Method = Post

## API B.1 API Upload Photo

Untitled Request

POST http://localhost/integria/tensorflow-upload-foto/

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

|                                     | KEY     | VALUE                        | DESCRIPTION |
|-------------------------------------|---------|------------------------------|-------------|
| <input checked="" type="checkbox"/> | photo[] | nX0E6T.jpg X                 |             |
| <input checked="" type="checkbox"/> | photo[] | 12240303_80d87f77a3_n.JPG X  |             |
| <input checked="" type="checkbox"/> | photo[] | name-server.PNG X            |             |
| <input checked="" type="checkbox"/> | photo[] | flower-animated-gif-12.gif X |             |
| <input checked="" type="checkbox"/> | key     | HGC87624                     |             |

Body Cookies Headers (7) Test Results Status: 200 OK Time: 163 ms SI

Pretty Raw Preview Visualize JSON

```
1 {
2   "code": "200",
3   "key": "HGC87624",
4   "message": "Foto berhasil di upload",
5   "foto": [
6     {
7       "status": "success",
8       "source": "nX0E6T.jpg",
9       "message": "Foto yang di upload valid",
10      "url_public": "http://localhost/integria/tensorflow-upload-foto/storage/202110/20211024_031050_61755b5a904f2.jpg"
11    },
12  ]
13 }
```

Parameter Input :

- Key = key yang dibuat manual untuk mengakses API
- Photo[] = foto yang diupload
- Method = Post



## API B.2 API Contribute Photo

► <http://localhost/integria/tensorflow-contribute/contribute>

POST

http://127.0.0.1:5001/contribute

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

☐ none

☒ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

|                                     | KEY | VALUE   |
|-------------------------------------|-----|---|
| <input checked="" type="checkbox"/> | key | HGC87624  |
| <input checked="" type="checkbox"/> | tag | foto_tinja_sakit_a  |
| <input checked="" type="checkbox"/> | url | <a href="http://localhost/integria/tensorflow-upload-foto/storage/202110/20211024_031050_61755b5aa6aeb.jpg">http://localhost/integria/tensorflow-upload-foto/storage/202110/20211024_031050_61755b5aa6aeb.jpg</a> |
|                                     | Key | Value   |

Body

Cookies

Headers (4)

Test Results

Status: 200

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "code": 200,
3   "key": "HGC87624",
4   "message": "Foto berhasil di download dan disimpan",
5   "tag": "foto_tinja_sakit_a",
6   "url": "http://localhost/integria/tensorflow-upload-foto/storage/202110/20211024_031050_61755b5aa6aeb.jpg"
7 }
```

Parameter Input :

- Key = key yang dibuat manual untuk mengakses API
- tag = Directory tag atau kode Penyakit
- url = alamat URL dari lokasi public storage foto yang akan di kontribusikan
- Method = Post

## G. Source Code Repository

1. API B.3 API Analytic Photo  
<https://github.com/dendie-sanjaya/tensorflow-clasificaton-image>
2. API B.2 API Contribute Photo  
<https://github.com/dendie-sanjaya/tensorflow-upload-foto>
3. API B.2 API Contribute Photo  
<https://github.com/dendie-sanjaya/tensorflow-contribute.git>

## H. Task List Area Core Machine Learning

| No | Task  | STATUS   |
|----|---|--|
| 1  | Engine Python ML Training Data  | SELESAI  |
| 2  | Engine Python ML Clasification Image  | SELESAI  |
| 3  | B.3 API Analytic Photo  | SELESAI  |
| 4  | B.1 Api Upload Photo  | SELESAI  |
| 5  | B.5 Api Contribute Photo  | SELESAI  |
| 6  | B.2 VM Public Storage<br>(Server Production – Instalasi & Config OS, Webserver, PHP, SSH)                                   | SELANJUTNYA<br>(Status Menunggu Informasi<br>Mengenai Server Production) |
| 7  | B4. VM Machine Learning GHC<br>(Server Production – Instalasi & Config OS, Webserver, Pyhont, SSH, FTP Server, Tensor Flow) | SELANJUTNYA<br>(Status Menunggu Informasi<br>Mengenai Server Production) |