

Pemanfaatan Image Collector berbasis Mobile dan Desktop dalam Pembentukan Dataset Citra

Miftakhurrokhmat¹, Fajar Donny Bachtiar²

¹Prodi Magister Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia

Jl. Kaliurang No.Km. 14,5, Sleman, Daerah Istimewa Yogyakarta 55584

²Program Studi Informatika, Fakultas Teknologi Informasi & Elektro, UTY

Jl. Siliwangi (Ring Road Utara), Sleman, Daerah Istimewa Yogyakarta 55285

Email : 18917214@students.uui.ac.id *, ² fajardonnybachtiar@gmail.com

Abstract

The rapid development of information technology has brought changes both in the form of applied technology applications, changing human behavior, and indirectly working culture. One of them is the use of face recognition technology applied in things such as class attendance [1], presence detection using masks [2], and checking children's health from the face [3]. Utilization of face recognition as a solution requires a learning action (training) using a dataset. The dataset itself is formed from collecting images of both faces or certain parts, through manual methods by requesting images or by directing using certain applications. This study aims to build a mobile-based application prototype using the Pascal language, the Delphi FireMonkey FMX framework in the hope of facilitating image collection, and downloading images as a dataset.

Keywords: face recognition, dataset, image, pascal, firemonkey

Abstraksi

Perkembangan teknologi informasi yang begitu pesat telah membawa perubahan baik di bentuk aplikasi terapan teknologinya, mengubah perilaku manusia, dan budaya kerja secara tidak langsung. Salah satunya adalah pemanfaatan teknologi face recognition (pengenalan wajah) diterapkan dalam hal seperti presensi kelas [1], pendeteksian presensi menggunakan masker [2], dan cek kesehatan anak dari wajah [3]. Pemanfaatan face recognition sebagai suatu solusi mensyaratkan tindakan pembelajaran (training) dengan menggunakan dataset. Dataset sendiri dibentuk dari pengumpulan citra baik wajah atau beberapa bagian tertentu, melalui cara manual dengan meminta citra atau dengan mengarahkan menggunakan aplikasi tertentu. Penelitian ini bertujuan membangun purwarupa aplikasi berbasis mobile menggunakan bahasa Pascal, framework Delphi FireMonkey FMX dengan harapan mempermudah pengumpulan citra, dan mengunduh citra sebagai suatu dataset.

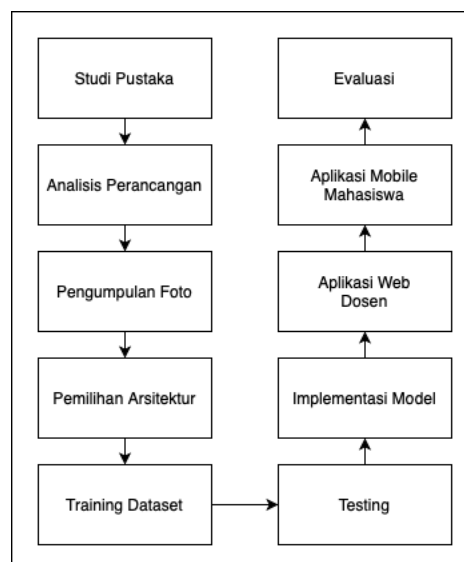
Kata Kunci: pengenalan wajah, dataset, citra, pascal, firemonkey

1. PENDAHULUAN

Deteksi wajah (*face detection*) merupakan tahapan awal sebelum proses pengenalan wajah (*face recognition*). Bidang-bidang terkait dengan pemrosesan wajah (*face processing*) diantaranya [4]:

1. Pengenalan wajah (*face recognition*) yaitu membandingkan input citra wajah dengan database wajah yang paling cocok.
2. Otentikasi wajah (*face authentication*) yaitu menguji keaslian / kesamaan suatu wajah dengan data wajah yang telah diinputkan.
3. Lokalisasi wajah (*face localization*) yaitu pendeteksian wajah dengan asumsi hanya ada satu wajah di dalam citra.
4. Penjejukan wajah (*face tracking*) yaitu memperkirakan lokasi suatu wajah di dalam video secara real time.
5. Pengenalan ekspresi wajah (*facial expression recognition*) yaitu mengenali kondisi emosi manusia.

Pada Gambar 1, terlihat urutan tahapan salah satu penelitian sistem pengenalan wajah sebelum diimplementasikan menjadi suatu aplikasi, diawali dengan tahapan pengumpulan foto dilanjutkan *training dataset*.



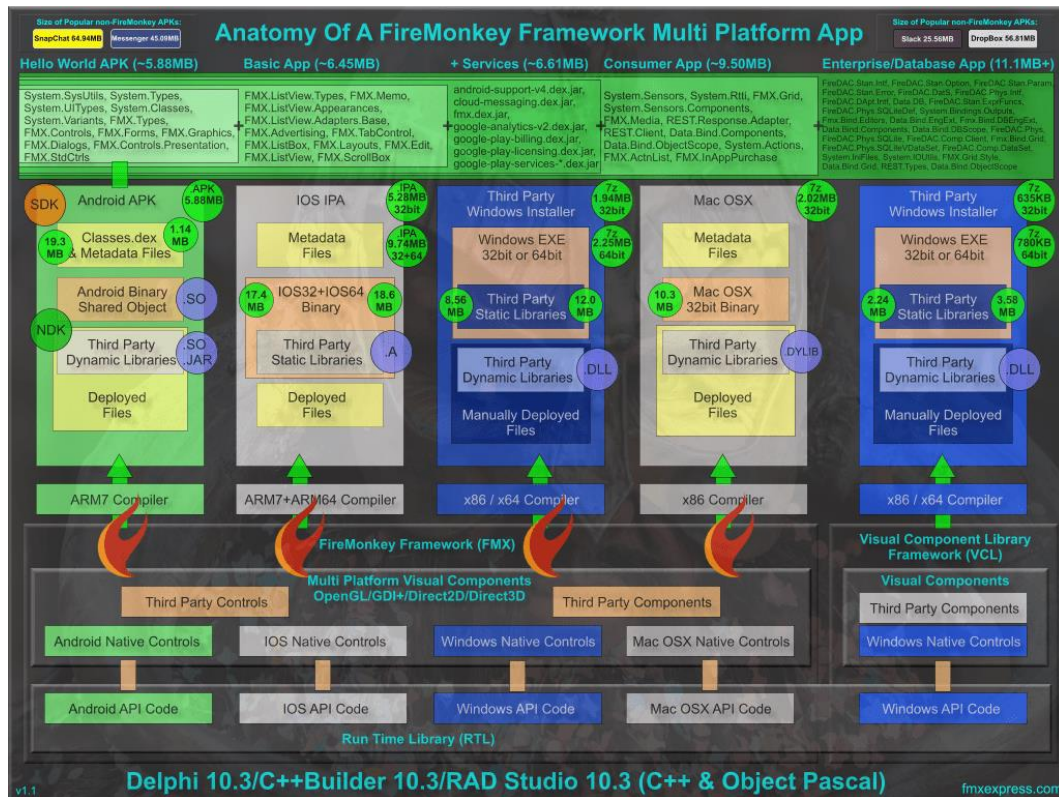
Gambar 1. Alur Penelitian Sistem Pengenalan Wajah [1]

Dataset menjadi sangat penting dalam sistem pengenalan wajah (*face recognition*), karena dari dataset inilah proses pembelajaran akan pola-pola wajah

dilakukan. Jumlah dataset yang besar akan meningkatkan kemampuan pola pengenalan seperti pada penelitian [5] menggunakan algoritma Quickprop. Semakin beragamnya jumlah citra wajah akan mempengaruhi hasil training model sehingga hasil yang didapat tidak terlalu overfitting (bagus saat *training*, tapi jelek saat *testing*) [1].

Sistem pengenalan wajah yang baik memerlukan fasilitas pendukung yaitu pengumpulan (*upload*) data user untuk memudahkan pengumpulan dataset wajah oleh pengguna [6]. Fasilitas tersebut dapat dikembangkan dalam platform mobile dengan menggunakan framework Delphi FireMonkey FMX seperti pada penelitian lain [7] yang memanfaatkan framework ini untuk mengembangkan aplikasi mobile. Firemonkey juga fleksibel juga untuk mengembangkan platform berbasis desktop dengan menggunakan source yang sama.

Framework FireMonkey FMX adalah platform pengembangan dan runtime aplikasi pada Delphi, C++ Builder, dan Appmethod. FireMonkey adalah platform aplikasi pertama yang asli berbasis GPU (Graphic Processing Unit). FireMonkey merupakan terobosan baru dan teknologi kunci dari Embarcadero untuk menjawab kebutuhan pengembangan aplikasi multiplatform dan multidevice. FireMonkey didesain untuk pengembangan aplikasi native untuk platform MS Windows, Mac OS X, Android, dan iOS. Pengembang dapat membuat satu aplikasi dan mengkompile aplikasi tersebut utk berbagai platform yang diinginkan [7]. Jika terjadi perubahan, hanya dengan menggunakan source yang sama, menjadi lebih fleksibel untuk diubah, tentu saja kesalahan dapat lebih di minimalisir. Gambar 2 menunjukkan bagaimana anatomi FireMonkey sebagai *multiplatform* dikembangkan untuk beberapa keperluan.



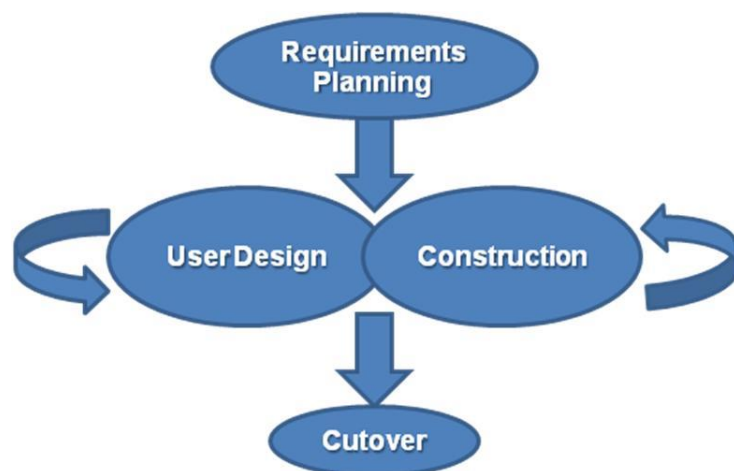
Gambar 2. Anatomi Framework FireMonkey [8]

Idealnya suatu aplikasi mobile membutuhkan *web service* atau biasa disebut API (*Application Programming Interface*) sebagai jembatan antara aplikasi mobile dengan *backend* (database). Salah satu format / protokol yang digunakan adalah REST. REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data. REST API terdiri dari beberapa komponen yaitu [9]:

1. *URL API* (endpoint), yaitu alamat API, diakses menggunakan protokol HTTP. Penamaan dan struktur yang baik akan memudahkan dalam penggunaan. Contoh pemanggilan URL API sebagai berikut: `users`, `users/123`, `users/123/photos`
2. *HTTP Verbs*, yaitu metode yang dilakukan *client* ketika melakukan *request* ke *server* sehingga diketahui apa yang diinginkan. Metode ini ada banyak, namun yang sering dipakai adalah: GET, POST, PUT, DELETE.

3. *HTTP Response Code*, yaitu kode yang menjadi standar dalam menginformasikan hasil *request* kepada klien. Secara umum terdapat 3 kelompok kode yang paling sering digunakan yaitu:
 - a) 2XX, yang menandakan *request* yang dilakukan berhasil.
 - b) 4XX, yang menandakan *request* mengalami kesalahan pada sisi klien.
 - c) 5XX, yang menandakan *request* mengalami kesalahan pada sisi server.
4. *Format Response*, yaitu bentuk format keluaran. Setiap *request* yang dilakukan klien akan menerima *response* dari server, biasanya berbentuk format XML ataupun JSON. Setelah mendapatkan data *response*, *client* dapat mengolah dengan cara *parsing* (mengurai) dan selanjutnya disesuaikan dengan kebutuhan.

Pengembangan aplikais mobile selain memerlukan REST API juga memerlukan suatu SDLC (*System Development Life Cycle*), salah satunya RAD. *Rapid Aplication Development* (RAD) adalah sebuah metode pengembangan software yang diciptakan untuk menekan waktu yang sehingga dihasilkan siklus pengembangan yang sangat pendek. Model RAD ini merupakan adaptasi dari model sekuensial linier. Apabila kebutuhan sistem dipahami dengan baik, proses RAD memungkinkan developer menciptakan sistem fungsional yang utuh dalam periode waktu yang sangat pendek (kurang lebih 60 sampai 90 hari). James Martin (1990) mendefinisikan pendekatan terhadap RAD menjadi empat fase [9] seperti yang diilustrasikan pada Gambar 3.



Gambar 3. Fase Pengembangan dengan RA

2. METODE PENELITIAN

Kegiatan penelitian dilakukan dengan mengadopsi metode RAD. Adapun beberapa tahapan yang dilakukan adalah sebagai berikut:

1. Requirements Planning

Requirements planning meliputi perencanaan dan analisa sistem. Pada tahap ini, peneliti melakukan studi literatur, dan observasi untuk mendapatkan kebutuhan dan spesifikasi yang dibutuhkan oleh sistem.

Adapun kebutuhan yang diperlukan oleh aplikasi adalah:

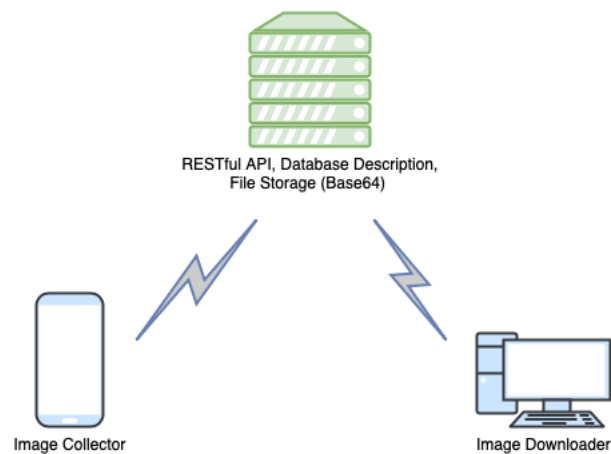
- a. Pengumpulan foto berbasis mobile
- b. Pengunduhan foto sebagai dataset citra berbasis desktop
- c. Penyimpanan secara deskripsi melalui tabel di database
- d. Penyimpanan konten file image atau citra ke dalam file fisik (*storage*)
- e. Penyimpanan file dan database dipusatkan di server
- f. Baik aplikasi mobile atau web berkomunikasi dengan server melalui REST API
- g. Kategori untuk image atau citra menggunakan SMILE dan NO_SMILE (untuk keperluan face recognition dengan senyuman).

Sedangkan spesifikasi aplikasi yang dikembangkan adalah:

- a. Framework: FireMonkey FMX (Pascal)
- b. Language: Borland Pascal (mobile, desktop), PHP (REST API)
- c. Platform: Android, Windows
- d. Database: MySQL

2. System Design

Dari spesifikasi dan kebutuhan sistem yang telah didapatkan kemudian dilakukan implementasi dalam bentuk desain sistem. Desain yang dibuat antara lain adalah arsitektur dari aplikasi yang dibuat. Pada Gambar 4, terlihat komponen sistem ini terdiri dari 3 komponen besar yaitu *server*, *image collector*, dan *image downloader*. Server sebagai pusat database dan penyimpanan file, serta REST api sendiri. *Image collector* adalah aplikasi berbasis mobile, dan *image downloader* adalah aplikasi berbasis desktop.



Gambar 4. Arsitektur Aplikasi Image Collector

3. Construction

Tahap ini berfokus pada pengembangan layanan atau aplikasi. Selama tahap *construction* ini dilakukan *programming* dengan mengikuti desain sistem yang telah ditentukan. Setiap fungsi yang telah selesai, lalu diuji untuk mencari bug atau kesalahan yang ada. Pengujian dapat dilakukan dengan menggunakan *smartphone* untuk menguji fitur *image collector*, dan menggunakan komputer atau laptop untuk menguji fitur *image downloader*. Pada penelitian ini sebagai batasan *smartphone* menggunakan *platform* Android, dan laptop yang digunakan menggunakan *platform* Windows.

4. Implementation

Apabila layanan sudah selesai dikembangkan, maka layanan atau aplikasi sudah siap untuk digunakan. Apabila nantinya ditemukan adanya temuan yang tidak terduga, akan menjadi acuan pengembangan selanjutnya.

3. HASIL DAN PEMBAHASAN

Pada pemaparan hasil dari penelitian ini, seperti pada *design system* yang sudah dirancang sebelumnya, terlihat struktur dari project aplikasi menggunakan Delphi FireMonkey FMX seperti pada Gambar 5, dan potongan isi salah satu file (kode utama) yang ditulis menggunakan bahasa Borland Pascal terlihat pada Tabel 1.

Name	Date Modified	Size	Kind
Android64	Today 07:54	18,2 MB	Folder
Release	Today 07:54	18,2 MB	Folder
assets	Today 07:53	1,6 MB	Folder
ico	18 September 2020 01:44	7 KB	Folder
img	Today 07:54	1,5 MB	Folder
key	18 September 2020 14:01	2 KB	Folder
style	18 September 2020 01:44	12 KB	Folder
Desktop	Today 07:54	36 KB	Folder
frMain.fmx	27 December 2021 20:52	26 KB	Document
frMain.pas	27 December 2021 20:18	11 KB	Pascal...rce File
source	27 December 2021 20:52	56 KB	Folder
uFunc.pas	27 December 2021 20:19	46 KB	Pascal...rce File
uOpenUrl.pas	6 August 2019 16:32	1 KB	Pascal...rce File
uRest.pas	12 August 2020 20:34	9 KB	Pascal...rce File
Win64	Today 07:54	30,7 MB	Folder
Release	Today 07:54	30,7 MB	Folder
assets	27 December 2021 20:36	7,9 MB	Folder
HasilExport	27 December 2021 20:36	10 KB	Folder
DMasmivDownload.exe	18 September 2020 14:28	22,9 MB	Windo...Archive
AccessCameraApp.identcache	12 August 2020 21:47	423 bytes	Document
AccessCameraApp.res	10 August 2020 07:41	32 bytes	Document
DMasmivCam.deployproj	11 August 2020 10:14	37 KB	Document
DMasmivCam.dpr	27 December 2021 20:45	866 bytes	Document
DMasmivCam.dproj	11 August 2020 14:18	120 KB	Document
DMasmivCam.dproj.local	27 December 2021 20:36	869 bytes	Document
DMasmivCam.identcache	27 December 2021 20:21	429 bytes	Document
DMasmivCam.res	27 December 2021 20:43	32 bytes	Document
DMasmivCamGroup.dpr	27 December 2021 20:18	334 bytes	Document
DMasmivCamGroup.dproj	12 August 2020 19:02	88 KB	Document
DMasmivCamGroup.dproj.local	27 December 2021 20:18	3 KB	Document
DMasmivCamGroup.identcache	27 December 2021 20:21	442 bytes	Document
DMasmivCamGroup.res	27 December 2021 20:18	112 KB	Document
frMain.fmx	27 December 2021 20:32	3,9 MB	Document
frMain.LgXhdpIh.fmx	11 August 2020 09:29	16 KB	Document
ProjectGroup1.groupproj	12 August 2020 18:59	2 KB	Document
ProjectGroup1.groupproj.local	12 August 2020 18:59	485 bytes	Document
uMain.LgXhdpIh.fmx	11 August 2020 09:29	688 KB	Document
readme.html	14 November 2019 16:00	130 KB	HTML
frMain.pas	27 December 2021 20:42	17 KB	Pascal...rce File
AndroidManifest.template.xml	20 February 2021 06:55	2 KB	XML Source File

Gambar 5. Struktur Aplikasi Image Collector

Tabel 1. Potongan Source Code Utama

```

unit frMain;
interface
uses
  System.SysUtils, System.Types, System.UITypes, System.Classes,
  System.Variants, System.Permissions,
  FMX.Types, FMX.Controls, FMX.Forms, FMX.Dialogs, FMX.StdCtrls,
  System.Actions, FMX.VirtualKeyboard,
  FMX.ActnList, FMX.StdActns, FMX.MediaLibrary.Actions,
  FMX.Objects, FMX.Graphics,
  FMX.Controls.Presentation, FMX.Media, FMX.Effects,
  FMX.TabControl, FMX.Layouts, System.Math,
  FMX.Edit, System.ImageList, FMX.ImgList, FMX.Ani,
  IdBaseComponent, System.Threading,
  IdComponent, IdTCPConnection, IdTCPClient,
  IdExplicitTLSClietServerBase,
  IdFTP, UI.Toast, UI.Base, UI.Standard, System.Net.URLClient,

```



```

    System.Net.HttpClient, System.Net.HttpClientComponent,
    FMX.ScrollBox, FMX.Memo,
    IdCustomTCPServer, IdTCPServer, IdCmdTCPServer, IdFTPServer,
    FMX.LoadingIndicator, FMX.Memo.Types
    {$IF DEFINED (ANDROID)}
    , FMX.MediaLibrary
    , FMX.Platform
    , System.Messaging
    , Androidapi.JNI.Os //TJBuild
    , Androidapi.Helpers
    {$ENDIF}
;

    procedure btnCameraClick(Sender: TObject);
    procedure edNamaTyping(Sender: TObject);
    procedure edNamaChange(Sender: TObject);
    procedure btnBackClick(Sender: TObject);
    procedure TakePhotoFromCameraAction1DidFinishTaking(Image:
TBitmap);
    procedure btnUploadClick(Sender: TObject);
    procedure imgLastClick(Sender: TObject);
    procedure btnCloseFotoClick(Sender: TObject);

    {Keyboard}
    procedure fnUpload;
    procedure fnSave(transID, jenis : String);
    procedure fnShowE(str : String);
    procedure fnSaveData(str, jenis : String);

    procedure TakePicturePermissionRequestResult(Sender:
TObject; const APermissions: TArray<string>; const
AGrantResults: TArray<TPermissionStatus>);
    public
    end;
var
    FMain : TFMain;
implementation

```

```
uses
    FMX.DialogService, uFunc, uRest;
{$R *.fmx}
const
    PermissionCamera = 'android.permission.CAMERA';
    PermissionReadExternalStorage =
'android.permission.READ_EXTERNAL_STORAGE';
    PermissionWriteExternalStorage =
'android.permission.WRITE_EXTERNAL_STORAGE';
    tIMG = 'temp.jpg';
    NO_SMILE = 'NO_SMILE';
    SMILE = 'SMILE';
    FACE_SHIELD = 'FACE_SHIELD';
    MASKER = 'MASKER';

procedure TFMMain.btnCameraClick(Sender: TObject);
begin
    {$IF DEFINED (MSWINDOWS)}
        tcMain.Next;
    {$ENDIF}
    PermissionsService.RequestPermissions(
        [PermissionCamera, PermissionReadExternalStorage,
PermissionWriteExternalStorage],
        TakePicturePermissionRequestResult,
        DisplayRationale)
end;

procedure TFMMain.fnSave(transID, jenis : String);
var
    req : String;
    arr : TStringArray;
    sl : TStringList;
begin
    sl := TStringList.Create;
    try
        sl.AddPair('tempID', transID);
```

```
sl.AddPair('nm', edNama.Text);
sl.AddPair('nim', edNIM.Text);
sl.AddPair('jenis', jenis);
sl.AddPair('model', model);
sl.AddPair('version', version);

tKet := 'Save Data';

req := 'insertDataV2';
arr := fnPostJSON(nHTTP, req, sl);

if arr[0,0] = 'null' then begin
    fnShowE(arr[1, 0]);
    Exit;
end;

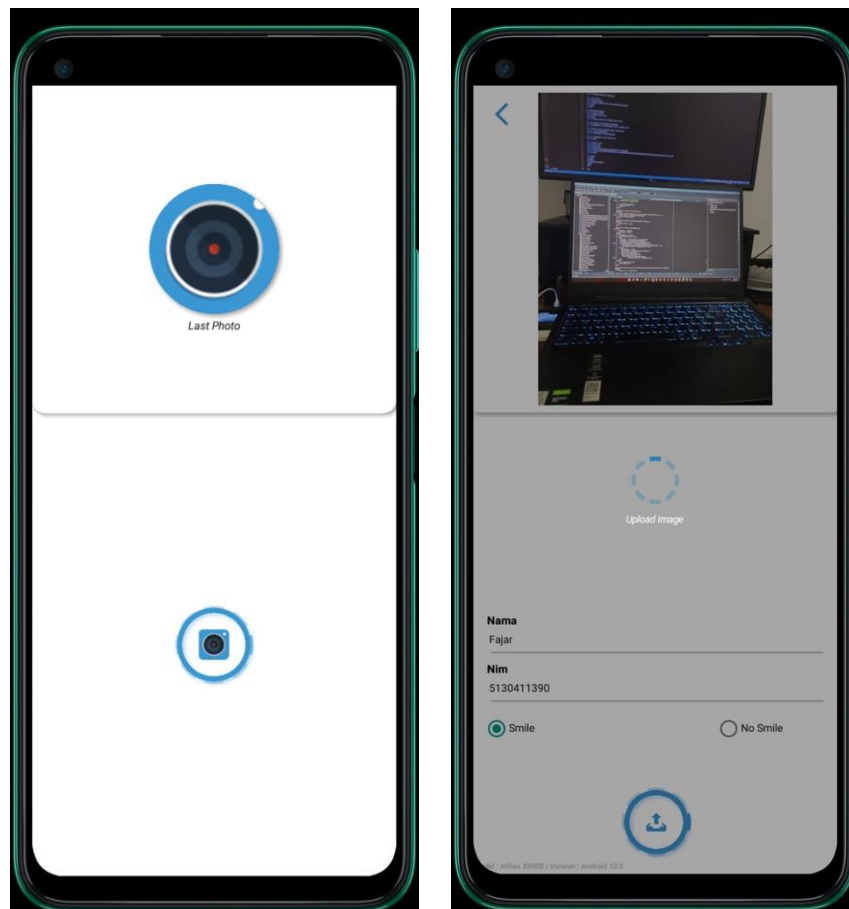
fnShowE('UPLOAD SUKSES');

SaveSettingString('config', 'lastfoto', transID + '.jpg');

TThread.Synchronize(nil, procedure begin
    fnLoadImage(imgCameraImage, imgLast);
    tcMain.First;
end);
finally
    sl.DisposeOf;
end;
end;

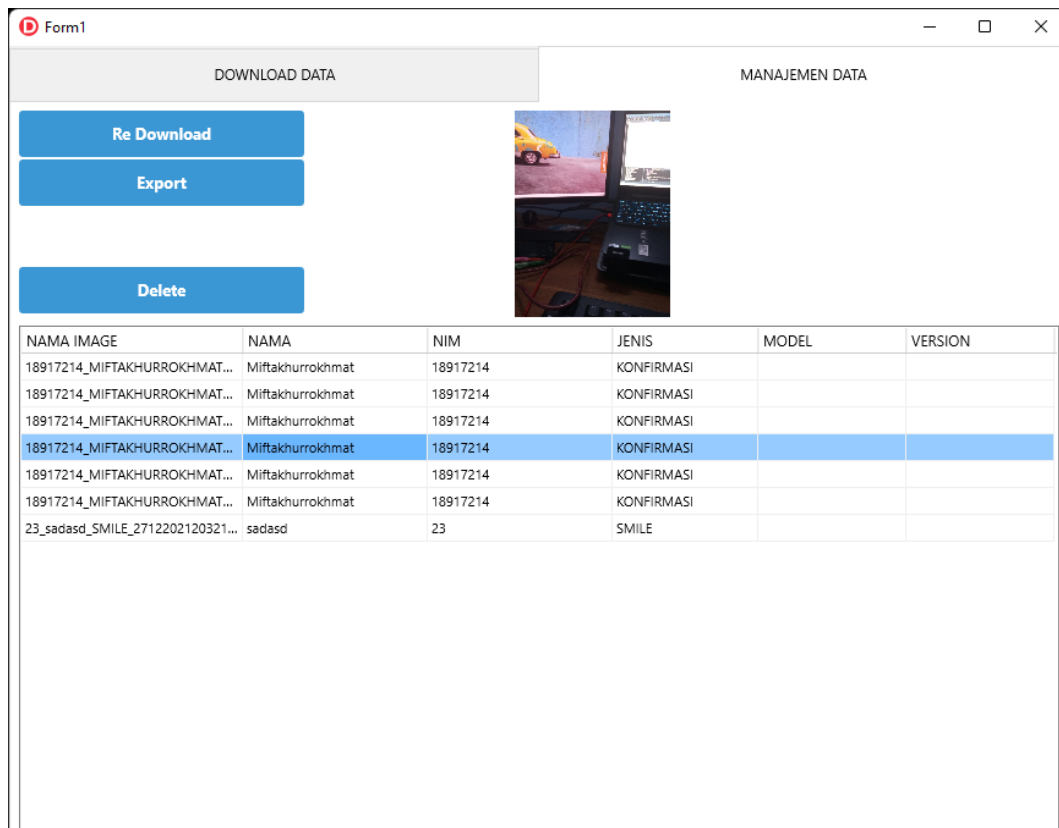
end.
```

Secara aplikasi sendiri, untuk *Image Collector* berbasis mobile (menggunakan Android) ditunjukkan tampilan hasilnya pada Gambar 6, dan *Image Downloader* berbasis desktop (menggunakan Windows) ditunjukkan tampilan hasilnya pada Gambar 7.



Gambar 6. Tampilan Aplikasi Mobile Image Collector sebagai Pengumpul

Pada *Image Collector* setelah ikon kamera diklik, *smartphone* akan memanggil kamera bawaan untuk mengambil foto, selanjutnya jika sudah diambil akan diarahkan ke pengisian deskripsi yaitu Nama, NIM, dan label yang digunakan (disini Smile dan No Smile). Baik foto dan deskripsi akan dikirimkan ke server melalui REST API.



Gambar 7. Tampilan Aplikasi Desktop Image Collector sebagai Downloader

Pada Image Downloader terdapat 2 tab (menu) yaitu Download Data dan Manajemen Data. Pada tab Download, pengguna dapat mengunduh foto beserta deskripsinya, Manajemen Data berisi fitur-fitur modifikasi jika ada *record* yang salah. Fitur Re-download digunakan jika akan melakukan download ulang (mengambil dari server). Fitur Delete untuk menghapus citra atau image yang tidak diperlukan, dan Export dalam wujud excel untuk melakukan rekap *record*.

4. KESIMPULAN

Berdasarkan hasil yang didapatkan, beberapa kesimpulan yang dapat diambil dari penelitian ini yaitu :

1. *Image Collector* ini telah berhasil diselesaikan dan diwujudkan dalam bentuk mobile app dan desktop

2. Batasan *field-field* pada *dataset* ini adalah NIM (mahasiswa), Nama, dan label (*Smile* dan *No Smile*)
3. Platform pengembangan dibatasi, yaitu mobile menggunakan Android, dan desktop menggunakan Windows. Apabila diperuntukkan untuk *platform* lain belum bisa (iPhone, MacOS, Linux).
4. Hasil ambil foto dari kamera *smartphone* dipengaruhi oleh spesifikasi dari *smartphone* itu sendiri, dan pengaturan resolusi.

5. SARAN

Adapun penelitian ini adalah penelitian awal yang diharapkan dapat dikembangkan di kemudian hari karena keterbatasan waktu dan batasan penelitian. Untuk ke depan harapan peneliti, penelitian ini bisa dikembangkan sebagai berikut :

1. Pengembangan ke platform yang belum dilakukan seperti di Linux dan MacOS, jika memungkinkan dengan teknologi yang sama, jika belum, dapat menggunakan teknologi yang *support*.
2. Penambahan *field-field* pada dataset yang lebih global sehingga mendukung beberapa aplikasi *face recognition* lain.
3. Pengambilan foto bisa dilakukan lebih dari satu kali, jika nantinya akan membentuk variasi citra seperti tampak depan, tampak samping, dan lain-lain.

DAFTAR PUSTAKA

- [1] Miftakhurrokhmat, R. A. Rajagede, and R. Rahmadi, "Presensi Kelas Berbasis Pola Wajah, Senyum dan Wi-Fi Terdekat dengan Deep Learning," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 31–38, 2021, doi: 10.29207/resti.v5i1.2575.
- [2] I. S. K. Wardhana, M. D. Lusita, and D. R. Irawati, "PEMANFAATAN INTERNET OF THINGS UNTUK CEGAH PENYEBARAN COVID-19," *SeNTIK*, vol. 4, no. September, pp. 125–130, 2020, [Online]. Available: <https://ejournal.jak-stik.ac.id/index.php/sentik/article/view/288/174>.
- [3] A. D. A. N. P. Check-ap, E. Ogi, I. Pratiwi, A. Khamid, F. Hidayanti, and S. R. Nudin,

- “ANALISA DAN PERANCANGAN CHECK-AP : APLIKASI PEMERIKSAAN KESEHATAN ANAK DENGAN FITUR FACE DETECTOR,” vol. 1, pp. 25–28, 2019.
- [4] A. Darmawan, “Aplikasi Mobile Pengenalan Wajah Secara Real-Time Berbasis Principal Component Analysis,” *Ubiquitous Comput. its Appl. J.*, vol. 2, pp. 57–66, 2019, doi: 10.51804/ucaiaj.v2i1.57-66.
- [5] S. Nugroho and A. Harjoko, “Penerapan Jaringan Syaraf Tiruan Untuk Mendeteksi Posisi Wajah Manusia Pada Citra Digital,” *Semin. Nas. Apl. Teknol. Inf.*, vol. 2005, no. Snati, pp. 1–6, 2005.
- [6] F. Setiawan and D. A. R., “Sistem Pengenalan Wajah Dengan Metode Local Binary Pattern Histogram Pada Firebase,” *SeNTIK*, vol. 4, no. 1, pp. 19–25, 2020.
- [7] H. D. Siswaja, “Pengembangan Aplikasi Android Menggunakan Firemonkey,” *Media Inform.*, vol. 14, no. 1, pp. 42–56, 2015.
- [8] Firemonkeyx.com, “Anatomy Of A Delphi 10.3 Rio Firemonkey App On Android, IOS, Windows, And macOS,” 2019. <http://www.firemonkeyx.com/anatomy-of-a-delphi-10-3-rio-firemonkey-app-on-android-ios-windows-and-macos/> (accessed Sep. 30, 2019).
- [9] M. A. K. Perdana, “Pengembangan REST API Layanan Penyimpanan menggunakan Metode Rapid Application Development (Studi kasus PT. XYZ),” *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 3, no. 1, pp. 100–104, 2018, doi: 10.30743/infotekjar.v3i1.563.