Jurnal Teknologi Sistem Informasi dan Aplikasi **ISSN: 2654-3788**
Penerbit: Program Studi Teknik Informatika Universitas Pamulang **e-ISSN: 2654-4229**
Vol. 6, No. 3, Juli 2023 (283-290) **DOI: 10.32493/jtsi.v6i3.30303**

Perancangan Sistem Pendukung Objek Deteksi untuk Permainan Kartu *Cardfight!Vanguard* Menggunakan Aplikasi *Roboflow* dan *Flask*

Cornelius Arvel Pratama Tungady[1], Hindriyanto Dwi Purnomo[2]

Teknik Informatika, Universitas Kristen Satya Wacana, Jl. Dr. O. Notohamidjojo No.1, Salatiga, Jawa Tengah, Indonesia, 50715
e-mail: [1]672019254@student.uksw.edu, [2]hindriyanto.purnomo@uksw.edu

Submitted Date: May 30th, 2023 Reviewed Date: June 06th, 2023
Revised Date: June 09th, 2023 Accepted Date: June 16th, 2023

Abstract

The covid-19 pandemic that emerged in early 2020 has affected our activities with various limitations, for this reason the government has implemented several protocols starting from using masks, maintaining cleanliness, and social distancing. It is also undeniable that humans are social creatures, in this context boredom is the main enemy. There are some activities to be able relieve stress such as reading, listening to music, watching, or playing a game. TCG (Trading Card Game) is an artificial game that build in with such various interesting themes. Card games are generally played with other people, but what would happen if the card game was played in the covid-19 pandemic. Of course, the biggest obstacle we will be facing are distance and time. Konami as a card game manufacturer and developer has a brilliant idea by implementing Remote Duel where tournaments and other official matches can be held virtually. Cardfight!Vanguard is no exception, which has a gameplay that really depends on the interaction between players. Remote play requires players to use a camera that will take pictures of the card field while playing. This study uses the Waterfall Model which will be taken from dataset preparation, train the data, perform the Annotate process, and carry out a web-based implementation using the flask framework so that it can be used and tested for its functionality using one of tensorflow's product technologies, Roboflow, which is an application that specifically designed to be able to assist in the process of creating and recognizing objects. The results obtained by using flask as a web base can be seen to perform card object recognition properly so that it can display data from the detected cards.

Keywords: TCG; Remote play; Vanguard; Roboflow; Flask

Abstrak

Pandemi *covid-19* yang muncul di awal tahun 2020 ini telah mempengaruhi berbagai aktivitas kita dengan berbagai keterbatasan, untuk itu pemerintah telah menerapkan beberapa protokol mulai menggunakan masker, menjaga kebersihan, hingga *social distancing.* Tidak bisa dipungkiri pula bahwasannya manusia merupakan makhluk sosial, dalam konteks tersebut kejenuhan merupakan musuh utama. Untuk dapat melepas stres seperti membaca, mendengarkan music, menonton, maupun melakukan sebuah permainan. TCG (*Trading Card Game*) merupakan permainan buatan dengan berbagai tema yang menarik. Permainan kartu umumnya dimainkan di tempat yang sama bersama dengan orang lain, namun apa jadinya apabila permainan kartu tersebut dimainkan dalam kondisi pandemi *covid-19.* Tentunya hambatan terbesar yang akan didapat adalah jarak dan waktu. *Konami* selaku produsen dan pengembang permainan kartu telah melakukan inovasi dengan menerapkan *Remote Duel* di mana turnamen dan pertandingan lainnya secara resmi dapat diselenggarakan secara *virtual*. Tak terkecuali *Cardfight!Vanguard* yang memiliki pola permainan yang sangat tergantung terhadap interaksi antar pemainnya. *Remote play* mengharuskan pemainnya untuk menggunakan kamera yang akan mengambil gambar *field* kartu ketika sedang bermain. Penelitian ini menggunakan *Model Waterfall* yang akan dimulai dari persiapan *dataset*, melakukan *Train* terhadap *data*, melakukan proses *Annotate*, hingga melakukan

implementasi berbasis *web* menggunakan *framework flask* untuk dapat digunakan dan dicoba fungsionalitasnya dengan menggunakan salah satu teknologi produk dari *tensorflow* yaitu *Roboflow,* merupakan sebuah aplikasi yang khusus dirancang untuk dapat membantu dalam proses pembuatan dan pengenalan objek. Hasil yang didapatkan adalah dengan menggunakan *flask* sebagai basis *web* dapat melakukan pengenalan objek kartu dengan baik hingga dapat menampilkan data dari kartu yang terdeteksi.

Kata Kunci: TCG; Remote play; Vanguard; Roboflow; Flask

1 Pendahuluan

Pandemi *covid-19* yang muncul di awal tahun 2020 ini telah mempengaruhi berbagai aktivitas kita dengan berbagai keterbatasan, untuk itu pemerintah telah menerapkan beberapa protokol mulai menggunakan masker, menjaga kebersihan, hingga *social distancing* (Jahangir , et al, 2020). Tidak bisa dipungkiri pula bahwasannya manusia merupakan makhluk sosial sehingga aktivitas sosial merupakan sesuatu yang sangat krusial dalam kehidupan kita. Dalam konteks tersebut kejenuhan merupakan musuh utama tiap individu dikala sedang menjalani aktivitas dari rumah. Stres yang dirasakan dapat mencangkup kedalam tiga hal diantaranya adalah terkait dengan kecemasan terkait dengan dampak dan bahaya *covid-19* yang terus diberitakan, yang kedua mengenai kehidupan sosial di mana hal ini disebabkan dari kurangnya interaksi sosial, lalu yang ketiga adalah keterkatiannya dengan aktivitas dikarenakan banyak kegiatan yang harus terhambat atau tidak bisa dilakukan sama sekali (Afini dan Hanifah, 2021). Dari beberapa hal tersebut diperlukan sebuah media untuk dapat melepas stres seperti membaca, mendengarkan musik, menonton, maupun melakukan sebuah permainan.

Bila membahas mengenai sebuah permainan tentunya akan sangat luas, namun pada topik ini terdapat satu permainan yang menarik untuk dibahas yaitu TCG (*Trading Card Game*), Permainan kartu ini dapat dibilang sejenis permainan buatan sebuah korporasi atau pihak tertentu dengan berbagai tema yang menarik. Permainan kartu umumnya dimainkan di tempat yang sama bersama dengan orang lain seperti teman maupun kerabat sehingga permainan akan semakin asik dan dapat menambah kemampuan bersosial kita, namun apa jadinya apabila permainan kartu tersebut dimainkan dalam kondisi pandemi *covid-19*. Tentunya hambatan terbesar yang akan didapat adalah jarak dan waktu, Dalam hal ini *Konami* selaku produsen dan pengembang permainan kartu *Yu-Gi-Oh* telah melakukan inovasi dengan menerapkan *Remote Duel* atau umumnya disebut *Remote Play* di mana turnamen dan pertandingan lainnya secara resmi dapat diselenggarakan secara *virtual*, Hal terlihat pada pemberitahuan yang dapat diakses pada halaman resmi dari *Yu-Gi-Oh* (Bicheno, 2022). Meskipun ide ini secara resmi dicetuskan oleh *Konami* namun hal ini rupanya juga dapat diterapkan kedalam permainan kartu lain tak terkecuali *Cardfight!Vanguard* yang memiliki pola permainan yang sangat tergantung terhadap interaksi antar pemainnya.

Konsep pertemuan secara *Virtual* telah menjadi solusi untuk mengatasi permasalahan sosial yang berhubungan dengan dengan jarak dan waktu, Di mana secara tidak langsung telah memberikan kemudahan serta menjadi potensi yang bagus untuk dapat dikembangkan kedalam berbagai macam kepentingan. Namun pada era saat ini konsep ini masih terbilang baru untuk beberapa orang, Karenanya terdapat beberapa kelemahan yang dapat dilihat mulai dari kelengkapan alat yang digunakan hingga berbedaan yang paling signifikan adalah sikap dan perilaku fisik yang ditimbulkan jelas berbeda (Hurst, et al, 2022). Sebuah permainan kartu biasanya membutuhkan dua atau lebih pemain untuk dapat memulai permainan seperti contoh klasik yang dapat dilihat adalah permainan kartu domino, poker, dan uno. Dalam hal ini dapat dilihat dalam sebuah permainan kartu dibutuhkan interaksi langsung antar pemainnya sehingga permainan dapat lebih terasa seru dan menegangkan.

Interaksi yang dimaksud dalam sebuah permainan kompetitif seperti *Yu-Gi-Oh*, *Pokemon*, dan *Vanguard* adalah komunikasi antar pemain yang dapat meliputi pergerakan kartu, respon pemain terhadap suatu kartu, dan beberapa kondisi lain yang terjadi terhadap pemain selama permainan berlangsung. Dalam kaitannya dengan "*Remote Duel*" yang menggunakan konsep *Virutal* serupa dapat dikatakan beberapa poin interaksi telah terpenuhi namun hal ini masih belum dapat

memaksimalkan pengalaman bermain, Oleh karena itu diperlukan sebuah sistem pendukung yang dapat menunjang jalannya permainan.

Salah satu interaksi yang cukup penting di kala permainan sedang berlangsung adalah ketika pemain meminjam kartu lawan guna melihat dan memahami kemampuan maupun kondisi dari sebuah kartu, Hal ini akan dapat menjadi masalah apabila diterapkan dalam pertandingan *virtual* dikarenakan beberapa hal diantaranya adalah pengaruh kamera yang kurang jelas, tangan pemain yang gemetar saat sedang menunjukkan kartu, dan lain sebagainya yang membuat pemain tidak dapat mengenali ataupun memahami isi dari kartu terkait. Dalam hal ini teknologi objek deteksi dapat menjadi sebuah solusi yang tepat dengan membantu pemain untuk dapat mengenali situasi kartu seperti warna, tulisan, gambar, dan atribut lainnya. Hal ini diperkuat dengan implementasi objek deteksi yang dapat digunakan secara *real time* menggunakan pendekatan *deep learning* (Ditrih, et al, 2021).

Penelitian dilakukan dengan bantuan aplikasi *Roboflow* untuk merancang objek deteksi mulai dari pembuatan *dataset, training*, hingga *deploy* hasil deteksi akan dilakukan oleh *Roboflow* yang akan diimplementasikan menggunakan *framework flask*. *Flask* sediri dipilih karena memiliki fleksibilitas pengembangan yang luas, penggunaannya yang gampang, serta menggunakan bahasa *python* sehingga dapat memanfaatkan *library* dari *python* yang cocok untuk mengolah data (Grinberg, 2018).

2 Penelitian Terdahulu

Masalah evaluasi objek deteksi yang terbatas pada data yang didapat dari *web* cenderung lebih banyak daripada yang didapatkan di dunia nyata sehingga ditakutkan akan mempengaruhi impelentasi objek deteksi dengan situasi aslinya di dunia nyata (Ciaglia, et al, 2022). Memperkenalkan projek *Roboflow-100* (RF100) yang disusun berdasarkan 100 *dataset* terpilih dari 90.000 *dataset* yang tersedia pada *Roboflow Universe*. Dengan membaginya menjadi beberapa katagori lalu dilakukan *training* menggunakan tiga metode berbeda yaitu *YOLOv5*, *YOLOv7*, dan *GLIP*. Didapat hasil perbandingan dari tiga metode tersebut dapat dijadikan bukti reabilitas *roboflow* dalam melakukan objek deteksi.

Unmanned Aerial Vehicles (UAVs) atau biasa disebut *drones* yang dilengkapi dengan perlengkapan untuk menangkap hasil visual dengan skala yang lebih luas telah populer dan menjadikannya dapat diaplikasikan kedalam beragam kepentingan, Namun terdapat beberapa kelemahan yang ditemukan dalam teknologi ini yaitu kemampuan menyimpan dan proses data yang terbatas, Terlebih lagi bila dihadapkan dengan kondisi untuk melakukan tangkapan visual secara *real time* (Zhang, et al, 2019). Dengan menggunakan *SlimYOLOv3* didapatkan hasil *SlimYOLOv3* dapat menyaingi akurasi *YOLOv3* dengan *FLOP* yang lebih rendah dan bekerja lebih cepat.

Mesin pengenalan visual telah mengalami perkembangan yang pesat (Bahaghighat, et al, 2019). Salah satu kepentingannya adalah dalam kaitannya dengan *Quality Control* (QC) dan *Quality Assurance* (QA) untuk dapat melakukan sortir dan pengecekan terhadap produk yang diproduksi sebagaimana telah digunakan pada produksi makanan, dan barang, Tak terkecuali obat yang biasanya dikemas kedalam bentuk kumpulan blister dan dimasukkan kedalam kardus tersendiri. Dengan menggunakan *Haar Cascade* dan *Template Matching* (TM) peneliti membuat sebuah sistem untuk melakukan pengecekan terhadap blister obat yang telah disusun kedalam sebuah kardus. Hasilnya adalah sistem dapat melakukan pengenalan objek dengan akurasi paling rendah 68,63% dan akurasi paling tinggi sebesar 88,33% menggunakan *Haar Cascade*, Hal ini berbanding terbalik dengan *Template Matching* yang mempunyai tingkat akurasi paling tinggi sebesar 52,50%.

Sejak pandemi covid-19 dimulai pemerintahan Indonesia telah mengharuskan warganya untuk menggunakan masker, namun pada kenyataannya masih banyak warga yang tidak menggunakan masker terlebih dalam tempat umum (Prasetia, et al, 2022). Maka dari itu peneliti membuat sebuah sistem pengenalan objek untuk dapat mengenali dan membedakan warga yang menggunakan masker dan tidak dengan cepat. Peneliti menggunakan metode *YOLO* dan *Darknet* yang diterapkan pada *Nvidia Jetson Nano*. Pengetesan menggunakan masker yang berbeda dan dilakukan sebanyak dua kali, Pada tes pertama pada jarak 150m tingkat akurasi pengenalan berada pada 94% sedangkan pada tes kedua dengan jarak

150m tingkat akurasi yang didapat adalah 96% dan tidak menggunakan masker sama sekali mendapat akurasi sebesar 99% pada jarak 150m.

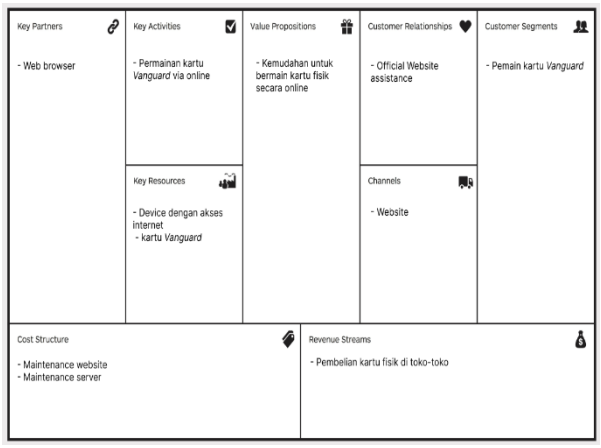
3 Metode Penelitian

3.1 User Requirements

Berdasarkan hasil yang didapat dari penelitian terdahulu didapatkan beberapa poin yang dapat dijadikan acuan untuk dapat menjawab kebutuhan *user* yang tertera pada rumusan masalah, yaitu

1. Proses pengenalan kartu secara *real-time*
2. Fitur yang dapat mendukung jalannya permainan

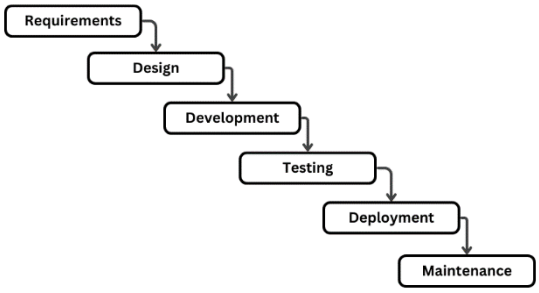
4 Business Model



Gambar 1. *Business Model Canvas*

Business model yang digunakan memiliki tujuan menggunakan media *website* untuk melakukan permainan sehingga mendorong maupun membangkitkan kembali minat dalam mengumpulkan dan memainkan kartu-kartu tersebut. Pemilihan penggunaan website diharapkan dapat menjangkau pasar secara luas dengan kompabititasnya dengan berbagai jenis perangkat keras.

4.1 Waterfall Model



Gambar 2. *Waterfall Model*

Waterfall model merupakan salah satu teknik perencanaan pengembangan aplikasi dengan cara melewati tahapannya secara terarah dan terurut tanpa mengulangi ataupun kembali ke tahapan sebelumnya. Model ini cocok digunakan dalam penelitian ini dikarenakan alurnya yang terstruktur sehingga penelitian dapat diselesaikan tepat waktu dan dengan hasil yang telah disetujui sebelumnya.

4.2 Implementasi

Implementasi dilakukan dengan mengikuti beberapa tahapan:

1. *Upload and Organize*

   Pada tahap ini akan dilakukan proses *upload* data-data yang nanti akan diolah, Data tersebut diperoleh dari internet dan dilakukan pengambilan secara manual dengan berbagai kemungkinan kriteria yang nantinya akan membuat proses train akan semakin akurat. Data-data tersebut berupa gambar kartu *Vanguard* dengan berbagai sudut yang berbeda agar dapat mendefinisikan bentuk serta dimensi dari kartu terkait. Gambar melalui internet dapat diambil dari fandom *Vanguard* maupun melalui website resminya di *https://en.cf-vanguard.com/cardlist/cardsearch*

   Beberapa gambar tersebut akan digunakan sebagai standar gambar yang ingin diidentifikasi karena memiliki resolusi bagus sehingga akan memudahkan saat melakukan pengenalan, selanjutnya adalah gambar yang diambil langsung dengan kartu asli dengan berbagai sudut atas, bawah, kanan, kiri, dan seterusnya dengan menggunakan *background* yang berbeda-beda pula.
2. *Annotate*

Pada tahap ini akan dilakukan proses untuk melakukan pembenahan identifikasi berupa label kepada data-data terkait meskipun dari *Roboflow* sendiri telah menyediakan AI khusus untuk membantu dalam memberikan Annotate dan memasukkan berbagai kriteria yang cocok sehingga hasil yang didapatkan maksimal dan akurat. Dalam penerapannya akan digunakan label untuk memberikan keterangan maupun informasi terkait sebuah kartu seperti nama, *grade*, serta atribut lainnya dapat dimasukkan untuk mendefinisikan sebuah kartu tersebut.

3. *Train*
   Data yang telah melewati *Annotate* selanjutnya akan melewati proses training data di mana data-data tersebut akan diolah sehingga dapat diidentifikasi oleh sistem. Bentuk yang diharapkan adalah kemampuan sistem untuk dapat mengenali kartu apa yang dikeluarkan berserta dengan posisinya.
4. *Deploy and Display*
   Setelah *data* telah dilakukan proses *train*, maka sistem tersebut dapat dikatakan siap diuji coba dan digunakan untuk dilihat fungsionalitasnya apakah sudah memenuhi target ataupun tidak. Sistem tersebut akan dapat digunakan dengan berbagai media salah satunya yang sederhana merupakan via *website* dengan jangkauan luas pada berbagai *platform*.

Penggunaan *framework Roboflow* ini dapat dilihat pada *web* resminya yaitu "*https://roboflow.com/"*. Pada *website* tersebut telah dipaparkan banyak informasi mulai dari alasan terbentuknya *Roboflow*, berbagai dokumentasi fitur, teknis penggunaan, hingga masalah harga yang dapat diajukan untuk penggunaan proyek berskala besar. Contoh penggunaan *Roboflow* untuk *Support System* yang diterapkan terhadap sebuah TCG paling populer, yaitu *Magic The Gathering* dapat dilihat di "*https://spelltable.wizards.com/*", Pada *website* tersebut dapat dilihat bahwa dengan menggunakan *Roboflow* dapat membuat sebuah *support system* yang berguna dan menarik bagi para pemain TCG *Magic The Gathering.*

4.3 Object Detection

Terdapat berbagai macam cara dan alat yang dapat digunakan dan diimplementasikan untuk dapat melakukan objek deteksi, namun pada penilitian ini yang akan digunakan adalah YOLO (*You Only Look Once*) yang telah menggunakan pendekatan *Deep Learning* yang dapat mengahasilkan hasil prediksi lebih baik dari R-CNN (*Region Based Convolutional Neural Networks*), Hal ini dipermudah dengan bantuan alat dari *Roboflow* untuk melakukan label, membuat berbagai macam *filter* dan *training*. (Xia, et al, 2023)

5 Hasil dan Pembahasan

5.1 Analisis Kebutuhan Sistem

Program pada penelitian ini masih dalam tahapan *alpha* sehingga pengujian dilakukan dengan menggunakan *local server* dari *Framework Flask*. Adapun beberapa kebutuhan yang diperlukan disaat akan menjalankan program ini antara lain:

Kebutuhan perangkat lunak:
1. Sistem Operasi : Windows 11
2. *Code Editor* : Visual Studio Code
3. Bahasa Pemrograman : Python, JavaScript
4. *Third Party* : Roboflow

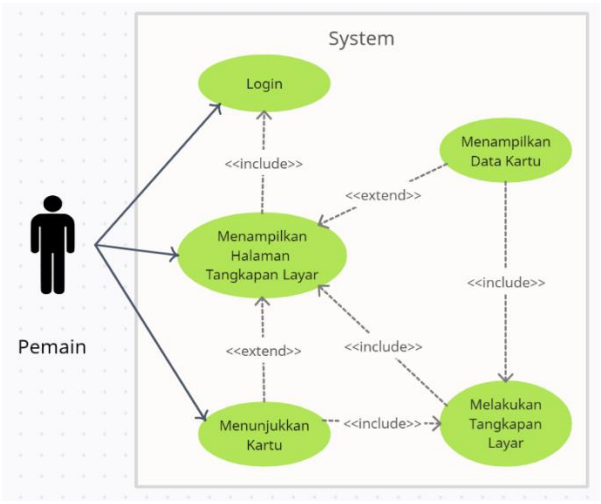
Kebutuhan perangkat keras:
1. Intel i5-10300H
2. Dedicated Video Graphics Adapter GTX 1650 4GB
3. Random Access Memory 8GB
4. Webcam

Kebutuhan lain:
1. Kartu *Vanguard Blaze Maiden deck*

5.2 Perancangan Sistem

Sistem program ini dibangun dengan menggunakan *model, view,* dan *controller*. Di mana *model* digunakan untuk mengambil *data* dari *database* maupun media luar lainya, *model* yang digunakan disini adalah dengan menggunakan '*api roboflow'* untuk keperluan pemrosesan gambar dari *user* menggunakan *cloud* yang telah disediakan oleh *roboflow*. Selanjutnya terdapat *view* di mana akan menampilkan *data* yang telah didapatkan oleh model sebelumnya, lalu yang terakhir merupakan *controller* yang memiliki peran

untuk menghubungkan antara *model* dan *view* sehingga proses *routing* yang terjadi dapat berjalan dengan terstruktur dan terorganisir.

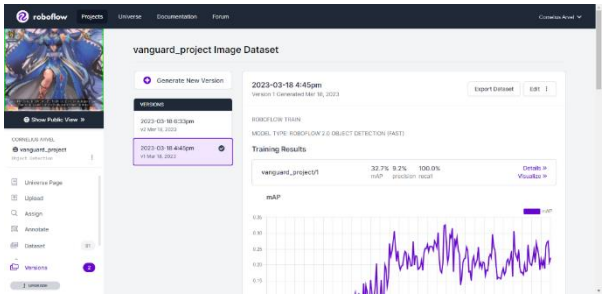


Gambar 3. *Use Case Diagram*

Pada Gambar 3 merupakan *Use Case Diagram* yang digunakan dalam penelitian ini, yaitu dengan menitikberatkan pada *user* sebagai *client* di mana sebagai pelaku yang akan melakukan sebuah *Object Detection.*

Gambar 4. *Activity Diagram*

Pada Gambar 4 dijabarkan sebuah skenario proses yang akan dilakukan oleh user ketika menggunakan program. Dimulai dari ketika *user* masuk kedalam program lalu memberikan akses kamera untuk dapat digunakan sistem, dan mendapatkan respon *Object Detection* dari *'Cloud Roboflow'.*



Gambar 5. *Roboflow Dataset*

Penelitian ini dimulai dengan membuat sebuah *dataset* baru yang berkaitan dengan kartu *vanguard*. Didalamnya dimasukkan berbagai *sample* kartu-kartu *vanguard* yang kemudian dilakukan proses *training* dengan menggunakan bantuan '*Roboflow train'* untuk menghemat *resource* dan mempermudah proses *training.*

Kode Program 1 Playing.html

```
<div>
      <a class="card-text" href="#">
        <img id="bg" class="camera" src="{{
        url_for('video_feed') }}">
      </a>
</div>
```

Dalam "*Playing.html"* memiliki *format* pada umumnya namun untuk membuat halaman *html* diperlukan bantuan dari *controller* untuk dapat menerima gambar dari *webcam* yang tersedia. Sehingga "*Playing.html"* disini memiliki peran sebagai *view.*

Jurnal Teknologi Sistem Informasi dan Aplikasi **ISSN: 2654-3788**
Penerbit: Program Studi Teknik Informatika Universitas Pamulang **e-ISSN: 2654-4229**
Vol. 6, No. 3, Juli 2023 (283-290) **DOI: 10.32493/jtsi.v6i3.30303**

Kode Program 2. model.py

```
from roboflow import Roboflow
      def infer(imgs):

          rf = Roboflow(api_key="YOUR-API-
          KEY")

          project =
          rf.workspace().project("WORKSPACE
          -NAME")

          model =
          project.version(VERSION).model

          # infer on a local image
            print(model.predict(imags,
        confidence=40, overlap=30).json())
```

"*Model.py"* memiliki peran sebagai model, sehingga yang dilakukan oleh package ini adalah untuk menerima request dari controller dan mendapatkan hasil proses deteksi menggunakan *Api Roboflow* yang kemudian dikirim kembali ke *Controller* untuk dapat diteruskan ke *View.*

Kode Program 3. app.py bagian 1

```
from Camera import Camera
import cv2
from flask import Response, Flask,
render_template
import model
import base64

      app = Flask(__name__)
      video = cv2.VideoCapture(0)

      @app.route('/play')
      def indexPlay():
          return
render_template('playing.html')
```

Kode Program 4. app.py bagian 2

Terakhir merupakan "app.py", di mana berfungsi sebagai *Controller* yang bertugas menghubungkan "*Playing.html* " dan "*model.py"*. Pada *package* ini akan dilakukan pula proses pengambilan gambar dari *webcam* secara realtime dengan menggunakan *Library Opencv.*

5.3 Pengujian

Berikut merupakan hasil percobaan implementasi menggunakan *flask* dimulai dari tampilan awal pada Gambar 6**,** lalu setelah user menekan *start* akan diarahkan menuju halaman di Gambar 7 yang akan menangkap gambar user

```
def gen(video):
   while True:
      try:
      if video.isOpened() == False:
                           break

      success, image = video.read()
      ret, buffer = cv2.imencode('.jpg',
      image)
      frame = buffer.tobytes()
      yield (b'--frame\r\n'
      b'Content-Type: image/jpeg\r\n\r\n' +
      frame + b'\r\n\r\n')

      except Exception as e:
        print(e)
   video.release()

@app.route('/video_feed')
def video_feed():
    return Response(gen(video),
        mimetype='multipart/x-mixed-
        replace;    boundary=frame')
```

menggunakan *webcam*, terakhir pada Gambar 8 ketika kartu yang ditunjukkan mempunyai respon dari *Api Roboflow* maka akan secara otomatis menampilkan Kartu yang dimaksud beserta dengan keterangan dan atribut yang dapat dibaca dengan cukup jelas.

Gambar 6. Tampilan Awal

Gambar 7. Tampilan pada tahap *standby*

Gambar 8. Tampilan saat *Object Detection*

6 Kesimpulan

Pada penelitian yang telah dilakukan dapat dilihat bagaimana *Roboflow* dapat diimplementasikan dengan sangat baik dan efisien untuk dapat melakukan *Object Detection*, mulai dari pembuatan *Dataset* hingga saat proses *Train* dan *Deploy* menggunakan berbagai macam *framework*, salah satunya adalah *Flask* Sehingga program objek deteksi ini dapat membantu pemain dalam mengidentifikasikan sebuah kartu.

Untuk kesulitan yang dihadapi saat penelitian ini adalah saat pengembangannya yang lumayan memakan *resource* terutama bagi mereka yang ingin melakukan *Training dataset* secara *manual* menggunakan *Framework* lainnya seperti *YOLO, Pytorch* dan sebagainya. Oleh karena itu sangat dianjurkan untuk mencoba menggunakan *Roboflow Train* yang dapat digunakan secara gratis hingga 3 kali.

7 Saran

Dari Penilitian ini masih banyak fitur-fitur yang dapat ditambahkan menggunakan bantuan dari hasil objek deteksi terutama dapat dilanjutkan dengan menghubungkannya ke *server realtime* sehingga dapat dikoneksikan dengan *user* lainnya.

Referensi

Jahangir, M. A., Muheem, A., & Rizvi, M. F. (2020). Coronavirus (COVID-19): history, current knowledge and pipeline medications. *Journal of Pharmaceutical Research Science & Technology [ISSN: 2583-3332], 4*(1), 1-9.

Afini, M., & Hanifah, H. (2021). Stresor dan Penanggulangan Stres Selama Masa Awal Pandemi Covid-19. *Psikostudia: Jurnal Psikologi, 10*(3), 294-305.

Bicheno, T. (2022). The Impact of the COVID-19 Pandemic on the International Marketing Strategies of MNE's: A Report on the Konami Company and 'YuGiOh!'. *Essex Student Journal*, 5-6.

Hurst, W., Withington, A., & Kolivand, H. (2022). Virtual conference design: features and obstacles. *Multimedia Tools and Applications*, 16908-16909.

Ditrih, H., Grgić, S., & Turković, L. (2021). Real-Time Detection and Recognition of Cards in the Game of Set. *International Symposium on Electronics in Marine*.

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python.* United States of America: O'Reilly Media, Inc.

Ciaglia, F., Zuppichini, F. S., Guerrie, P., McQuade, M., & Solawetz, J. (2022). Roboflow 100: A Rich, Multi-Domain Object Detection Benchmark. *arXiv preprint arXiv:2211.13523*.

Zhang, P., Zhong, Y., & Li, X. (2019). SlimYOLOv3: Narrower, faster and better for real-time UAV applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (pp. 0-0).

Bahaghighat, M., Akbari, L., & Xin, Q. (2019). A machine learning-based approach for counting blister cards within drug packages. *IEEE Access*, *7*, 83785-83796.

Prasetia, D. D., Yuswanto, A., & Wibowo, B. (2022). Design of Machine Learning Detection Mask Using Yolo and Darknet on Nvidia Jetson Nano. *Teknokom*, *5*(1), 88-95.

Xia, Y., Nguyen, M., & Yan, W. Q. (2023, February). A Real-Time Kiwifruit Detection Based on Improved YOLOv7. In *Image and Vision Computing: 37th International Conference, IVCNZ 2022, Auckland, New Zealand, November 24–25, 2022, Revised Selected Papers* (pp. 48-61). Cham: Springer Nature Switzerland.

Khoo, E. J., & Lantos, J. D. (2020). Lessons learned from the COVID-19 pandemic. *Acta Paediatrica (Oslo, Norway: 1992)*, *109*(7), 1323.

Ramadhany, A., Firdausi, A. Z., & Karyani, U. (2021). Stres pada mahasiswa selama pandemi covid-19. *Jurnal Psikologi Insight*, *5*(2), 65-71.

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, *11*(4), 193.