

Panduan Konfigurasi Pengenalan Wajah pada Robot Pepper untuk Mengingat Wajah Baru

Laporan teknis ini menyajikan panduan komprehensif untuk mengkonfigurasi kemampuan pengenalan wajah pada robot Pepper, yang dikembangkan oleh SoftBank Robotics. Fokus utama adalah pada mekanisme yang memungkinkan Pepper untuk mempelajari, menyimpan, dan kemudian mengenali wajah individu yang baru ditemui, sehingga robot dapat memanggil nama individu tersebut pada interaksi berikutnya. Robot Pepper dirancang untuk interaksi sosial yang alami dan intuitif dengan manusia¹, dan kemampuan untuk mengenali serta mengingat individu merupakan aspek fundamental dalam mencapai interaksi yang personal dan bermakna.

I. Pendahuluan Mengenai Robot Pepper dan Kemampuan Pengenalan Wajahnya

Robot Pepper adalah robot humanoid yang dirancang khusus untuk interaksi dengan manusia, dilengkapi dengan berbagai sensor dan kemampuan untuk memahami serta merespons lingkungan dan orang-orang di sekitarnya. Salah satu fitur unggulannya adalah kemampuan untuk mendeteksi dan mengenali wajah manusia.³ Kemampuan ini tidak hanya memungkinkan Pepper untuk mengidentifikasi kehadiran manusia tetapi juga untuk membedakan antara individu yang berbeda setelah melalui proses pembelajaran.

Pengenalan wajah pada Pepper berfungsi melalui serangkaian kamera, termasuk dua kamera HD dan satu kamera 3D yang terletak di kepalanya. Data visual dari kamera ini diproses oleh perangkat lunak pengenalan bentuk yang mampu mengidentifikasi wajah dan objek.¹ Lebih lanjut, Pepper dapat mempelajari wajah baru dan menyimpannya dalam basis data untuk pengenalan di masa mendatang. Ketika Pepper mendeteksi wajah yang sudah dikenalnya, ia dapat memanggil nama orang tersebut, yang secara signifikan meningkatkan kualitas interaksi dan menjadikannya lebih personal. Proses ini merupakan inti dari permintaan pengguna untuk membuat Pepper "mengingat" wajah yang baru saja dideteksi dan dipelajari.

Dokumentasi dan sumber daya yang tersedia, seperti portal pengembang SoftBank

Robotics dan berbagai panduan pengguna, menyediakan informasi mengenai cara kerja dan konfigurasi fitur ini.⁴ Laporan ini akan mengkonsolidasikan informasi tersebut untuk memberikan panduan yang jelas dan terstruktur.

II. Memahami Mekanisme Pengenalan Wajah pada Pepper

Inti dari kemampuan pengenalan wajah Pepper terletak pada modul perangkat lunak yang disebut ALFaceDetection.⁵ Modul ini bertanggung jawab tidak hanya untuk mendeteksi keberadaan wajah dalam bidang pandang robot, tetapi juga untuk mengenali siapa individu tersebut jika wajahnya telah dipelajari sebelumnya.⁶

A. Modul Inti: ALFaceDetection

ALFaceDetection adalah layanan NAOqi yang menjadi tulang punggung sistem persepsi wajah Pepper.⁵ Layanan ini menggunakan solusi deteksi dan pengenalan wajah yang dilengkapi dengan lapisan tambahan untuk meningkatkan akurasi hasil pengenalan.⁷ Modul ini dapat diakses dan dikontrol baik melalui antarmuka pemrograman grafis Choregraphe maupun secara programatik menggunakan Software Development Kit (SDK) Python.

B. Proses Deteksi dan Pengenalan

Proses pengenalan wajah pada Pepper umumnya melibatkan dua tahap utama:

1. **Deteksi Wajah (Face Detection):** Tahap awal di mana robot mengidentifikasi adanya satu atau lebih wajah dalam input video dari kameranya. Sistem akan menandai area yang dianggap sebagai wajah. Kinerja deteksi dipengaruhi oleh faktor seperti lebar wajah minimal dalam piksel (sekitar 20 piksel untuk deteksi dasar, yang setara dengan jarak sekitar 2 meter untuk gambar QVGA pada orang dewasa).⁶
2. **Pengenalan Wajah (Face Recognition):** Setelah wajah terdeteksi, jika fitur pengenalan diaktifkan, robot akan mencoba mencocokkan fitur wajah yang terdeteksi dengan wajah-wajah yang tersimpan dalam basis datanya.⁶ Setiap nama yang tersimpan dikaitkan dengan skor kecocokan antara 0 dan 1. Skor yang lebih tinggi menunjukkan kepastian yang lebih besar. Hanya hasil dengan skor di atas ambang batas tertentu yang akan dipertimbangkan.⁶

Penting untuk dipahami bahwa algoritma pengenalan wajah pada Pepper tidak

menggunakan representasi 3D dari orang yang akan dikenali, melainkan mengandalkan informasi seperti jarak antar titik-titik kunci pada wajah (fitur 2D).⁶ Keterbatasan ini menjelaskan mengapa pengenalan wajah lebih sensitif terhadap variasi pose kepala (pan, tilt, rotasi) dibandingkan dengan deteksi wajah semata.

C. Tahap Pembelajaran (Learning Stage)

Agar Pepper dapat mengenali seseorang dengan nama, wajah orang tersebut harus terlebih dahulu "dipelajari" oleh robot. Proses pembelajaran ini melibatkan pengambilan satu atau lebih citra wajah individu tersebut, yang kemudian disimpan dan dikaitkan dengan sebuah nama dalam basis data internal robot.⁶ Selama proses pembelajaran, kondisi pencahayaan yang baik dan pose wajah yang netral menghadap kamera sangat disarankan untuk hasil yang optimal.⁶

D. Penyimpanan dan Pengambilan Informasi Wajah

Informasi wajah yang telah dipelajari disimpan dalam basis data pada robot. Ketika wajah terdeteksi dan dikenali, modul ALFaceDetection akan memicu sebuah event bernama FaceDetected dalam sistem memori robot (ALMemory).⁶ Event ini membawa berbagai informasi, termasuk label nama dari wajah yang dikenali (

faceLabel) dan skor kepercayaan pengenalan (scoreReco).⁶

Robot juga memiliki memori jangka pendek terkait wajah yang baru saja terdeteksi dan dikenali. Informasi ini akan dipertahankan selama wajah tersebut masih terdeteksi. Namun, jika tidak ada wajah yang terdeteksi selama lebih dari 4 detik, memori jangka pendek ini akan dihapus.⁶ Mekanisme ini penting untuk interaksi yang berkelanjutan, memastikan robot tidak terus-menerus mengumumkan nama seseorang yang baru saja ia sapa.

III. Konfigurasi Pengenalan Wajah Melalui Choregraphe

Choregraphe adalah lingkungan pengembangan terintegrasi (IDE) berbasis grafis yang memungkinkan pengguna untuk membuat perilaku dan aplikasi untuk robot Pepper tanpa perlu menulis kode secara ekstensif. Perangkat lunak ini menyediakan cara yang intuitif untuk mengimplementasikan fungsionalitas pengenalan wajah.

A. Pengenalan Choregraphe dan Box Library

Choregraphe menggunakan sistem "box" yang merepresentasikan fungsionalitas tertentu. Box-box ini dapat dihubungkan bersama untuk menciptakan alur perilaku robot. Untuk pengenalan wajah, beberapa box standar dalam Library menjadi sangat relevan.

B. Langkah-langkah Mengajarkan Wajah Baru ke Pepper

1. Menggunakan Box Learn Face:

- Temukan box Learn Face dalam Box Library Choregraphe, biasanya di bawah kategori "Vision" atau "Human Detection".
- Seret dan letakkan box Learn Face ke dalam diagram alur kerja.
- Ketika box ini dieksekusi, Pepper akan meminta nama orang yang akan dipelajari. Pengguna dapat memasukkan nama melalui antarmuka Choregraphe atau, jika diprogram, melalui input suara atau tablet Pepper.
- Setelah nama dimasukkan, orang tersebut memiliki waktu sekitar 5 detik untuk memposisikan wajahnya di depan kamera Pepper.⁶
- Proses pembelajaran dimulai, yang ditandai dengan perubahan warna mata LED Pepper menjadi biru.⁶
 - Jika wajah terdeteksi dengan baik dalam kondisi yang sesuai (pencahayaan cukup, tidak ada bayangan parsial, tidak terlalu jauh), mata Pepper akan berubah menjadi hijau dalam kurang dari satu detik, menandakan pembelajaran berhasil.⁶
 - Jika mata tetap biru selama beberapa detik, orang tersebut disarankan untuk sedikit bergerak atau mengubah posisi untuk memperbaiki kondisi pembelajaran.⁶
- Antarmuka Choregraphe menyediakan cara yang ramah pengguna untuk melakukan tahap pembelajaran ini.⁶

C. Langkah-langkah Membuat Pepper Mengenali dan Menyapa dengan Nama

1. Menggunakan Box Face Reco atau Face Detected Event:

- Setelah wajah dipelajari, box seperti Face Recognition (atau nama serupa yang berfungsi untuk pengenalan) dapat digunakan. Box ini akan memicu output ketika wajah yang dikenal terdeteksi.
- Alternatifnya, dan seringkali lebih fleksibel, adalah memantau event FaceDetected dari ALMemory. Event ini menyediakan informasi lebih detail, termasuk nama wajah yang dikenali (faceLabel) dan skor kepercayaan (scoreReco).⁶

2. Mengintegrasikan dengan Box Say atau Animated Say:

- Hubungkan output dari box pengenalan wajah (atau logika yang memproses event FaceDetected) ke input box Say atau Animated Say.
- Konfigurasi box Say untuk mengucapkan salam yang menyertakan nama orang yang dikenali. Misalnya, jika faceLabel adalah "Budi", Pepper dapat diprogram untuk mengatakan, "Halo, Budi!".
- Sebuah contoh video tutorial menunjukkan Pepper menyapa "Bob" dan "Kate" setelah wajah mereka dikenali, mengindikasikan keberhasilan proses ini.⁸

D. Menggunakan Basic Awareness dan Dialog

- Fitur Basic Awareness pada Pepper secara otomatis membuat robot mencari manusia dan bereaksi terhadap stimulus dasar seperti suara, gerakan, sentuhan, dan kehadiran manusia.⁹
- Ketika Basic Awareness aktif dan Pepper melihat seseorang, aplikasi Dialog dapat dimulai secara otomatis, memungkinkan interaksi berbasis topik.⁹ Mengintegrasikan pengenalan wajah ke dalam sistem dialog ini dapat menciptakan pengalaman yang lebih personal, di mana Pepper menyapa pengguna dengan nama sebelum memulai percakapan.
- Warna mata LED Pepper memberikan umpan balik visual selama interaksi: merah muda saat melacak orang baru, biru berputar saat mendengarkan, dan hijau saat berpikir.⁹

Meskipun Choregraphe menyederhanakan banyak aspek, ia mungkin tidak selalu menawarkan tingkat kontrol granular yang sama seperti pemrograman langsung menggunakan SDK. Namun, untuk banyak kasus penggunaan, pendekatan berbasis Choregraphe sudah cukup dan lebih mudah diakses oleh pengguna non-programmer. Kemudahan penggunaan ini, ditambah dengan umpan balik visual langsung dari robot (seperti perubahan warna mata), menjadikan Choregraphe alat yang efektif untuk mengimplementasikan fungsi pembelajaran dan pengenalan wajah dasar.

IV. Memprogram Pengenalan Wajah Menggunakan Python SDK

Bagi pengembang yang membutuhkan kontrol lebih besar dan fungsionalitas yang lebih canggih, SoftBank Robotics menyediakan Software Development Kit (SDK) yang mendukung berbagai bahasa pemrograman, termasuk Python. Menggunakan Python SDK memungkinkan interaksi langsung dengan NAOqi, sistem operasi yang

menjalankan Pepper, dan modul-modulnya seperti ALFaceDetection.

A. Pengenalan NAOqi Framework dan Python SDK

NAOqi adalah kerangka kerja perangkat lunak yang mendasari operasi Pepper. Python SDK menyediakan pustaka untuk berinteraksi dengan layanan NAOqi dari skrip Python. Untuk memulai, pengembang perlu menghubungkan skrip Python ke robot Pepper. Ini biasanya dilakukan dengan membuat sebuah sesi qi.Session dan menghubungkannya ke alamat IP dan port robot (port default adalah 9559).¹⁰

Python

```
import qi
import time

# Ganti dengan alamat IP Pepper Anda
pepper_ip = "your_pepper_ip_address"
pepper_port = 9559

session = qi.Session()
try:
    session.connect("tcp://" + pepper_ip + ":" + str(pepper_port))
    print("Berhasil terhubung ke Pepper.")
except RuntimeError:
    print("Tidak dapat terhubung ke NAOqi pada {}:{}".format(pepper_ip, pepper_port))
    exit(1)
```

B. Mengakses Layanan ALFaceDetection

Setelah sesi berhasil dibuat, layanan ALFaceDetection dapat diakses sebagai proksi objek, yang memungkinkan pemanggilan metode-metodenya.⁵

Python

```
face_detection_service = session.service("ALFaceDetection")
```

```
memory_service = session.service("ALMemory")
tts_service = session.service("ALTextToSpeech") # Untuk membuat Pepper berbicara
```

C. Fungsi Kunci Python untuk Manajemen Wajah

Berikut adalah beberapa fungsi penting dari ALFaceDetectionProxy untuk mengelola proses pembelajaran dan pengenalan wajah:

1. Mempelajari Wajah Baru (learnFace):

- Metode `face_detection_service.learnFace(name)` digunakan untuk mengajarkan wajah baru kepada Pepper dan mengaitkannya dengan `name` (string) yang diberikan.¹² Metode ini mengembalikan `True` jika pembelajaran berhasil.

```
Python
def learn_new_face(name_to_learn):
    print("Mempelajari wajah untuk: {}".format(name_to_learn))
    if face_detection_service.learnFace(name_to_learn):
        print("Wajah '{}' berhasil dipelajari.".format(name_to_learn))
    else:
        print("Gagal mempelajari wajah '{}".format(name_to_learn))
```

2. Menerima Data Pengenalan (Berlangganan Event FaceDetected):

- Untuk merespons secara dinamis ketika wajah dikenali, skrip perlu berlangganan (subscribe) ke event `FaceDetected` yang dipublikasikan oleh `ALMemory`.⁶
- Data yang diterima dalam callback event ini sangat kaya, mencakup `TimeStamp`, array `FaceInfo` (yang berisi `faceID`, `scoreReco`, `faceLabel`, `leftEyePoints`, `rightEyePoints`, dan informasi sudut kamera).⁶ Kekayaan data ini memungkinkan aplikasi yang lebih canggih daripada yang biasanya mudah diimplementasikan hanya dengan `Choregraphe`.

```
Python
face_detected_subscriber = None

def on_face_detected(value):
    # Struktur data 'value' dijelaskan dalam dokumentasi ALFaceDetection
    # Format umumnya: FaceInfo_array, CameraPose_Robot, Camera_Id]
    if value and len(value) >= 2:
        face_info_array = value
        if not face_info_array: # Tidak ada wajah yang terdeteksi
            return

    for face_info in face_info_array:
```

```
# face_info:
# FaceExtraInfo:
extra_info = face_info
face_label = extra_info
score_reco = extra_info
```

```
if face_label and score_reco > 0.4: # Ambang batas contoh
    print("Wajah dikenali: {} dengan skor: {:.2f}".format(face_label, score_reco))
    tts_service.say("Halo, " + face_label)
    # Hentikan langganan setelah sapaan pertama untuk menghindari sapaan berulang
    # Atau implementasikan logika untuk tidak menyapa lagi dalam waktu singkat
```

```
def subscribe_to_face_detection():
    global face_detected_subscriber
    face_detected_subscriber = memory_service.subscriber("FaceDetected")
    face_detected_subscriber.signal.connect(on_face_detected)
    print("Berlangganan event FaceDetected.")
```

```
def unsubscribe_from_face_detection():
    if face_detected_subscriber:
        face_detected_subscriber.signal.disconnect_all() # Cara yang lebih aman untuk memutuskan koneksi
    print("Berhenti berlangganan event FaceDetected.")
```

Penting untuk dicatat bahwa bekerja dengan event seperti FaceDetected memerlukan pemahaman tentang pemrograman asinkron (menggunakan fungsi callback). Ini adalah paradigma umum dalam robotika tetapi mungkin memerlukan penyesuaian bagi pemula.

3. Mengambil Daftar Wajah yang Telah Dipelajari (getLearnedFacesList):

- Metode `face_detection_service.getLearnedFacesList()` mengembalikan daftar (list) string yang berisi nama-nama semua wajah yang telah dipelajari oleh Pepper.¹²

```
Python
def show_learned_faces():
    learned_faces = face_detection_service.getLearnedFacesList()
    if learned_faces:
        print("Daftar wajah yang telah dipelajari:")
        for face_name in learned_faces:
            print("- {}".format(face_name))
    else:
        print("Tidak ada wajah yang tersimpan dalam database.")
```

4. Meningkatkan Pengenalan untuk Orang yang Ada (reLearnFace):

- Metode `face_detection_service.reLearnFace(name)` digunakan untuk menambahkan gambar identifikasi baru ke catatan pengguna yang sudah

ada. Ini dapat meningkatkan hasil pengenalan, terutama jika wajah dipelajari dari berbagai sudut atau kondisi pencahayaan. Hingga 10 gambar dapat disimpan per orang.¹²

Python

```
def relearn_existing_face(name_to_relearn):
    print("Mencoba meningkatkan pengenalan untuk {}".format(name_to_relearn))
    if face_detection_service.reLearnFace(name_to_relearn):
        print("Pengenalan untuk '{}' berhasil ditingkatkan.".format(name_to_relearn))
    else:
        print("Gagal meningkatkan pengenalan untuk {}".format(name_to_relearn))
```

5. Menghapus Orang Tertentu atau Semua Wajah:

- `face_detection_service.forgetPerson(name)` menghapus semua wajah yang dipelajari yang terkait dengan name tertentu dari basis data.¹²
- `face_detection_service.clearDatabase()` menghapus semua wajah yang dipelajari dari basis data.¹²

Python

```
def forget_specific_person(name_to_forget):
    if face_detection_service.forgetPerson(name_to_forget):
        print("Wajah '{}' berhasil dihapus dari database.".format(name_to_forget))
    else:
        print("Gagal menghapus wajah {}".format(name_to_forget))
```

```
def clear_all_faces():
```

```
    if face_detection_service.clearDatabase():
        print("Semua wajah berhasil dihapus dari database.")
    else:
        print("Gagal menghapus semua wajah dari database.")
```

6. Mengatur Ambang Batas Kepercayaan Pengenalan (`setRecognitionConfidenceThreshold`):

- `face_detection_service.setRecognitionConfidenceThreshold(threshold)` mengatur nilai ambang batas baru untuk pengenalan wajah. Nilai ini harus antara 0 dan 1 (default 0.4).¹² Pengaturan ini memberikan kontrol eksplisit atas keseimbangan antara kinerja, akurasi, dan kekokohan, yang tidak selalu mudah disesuaikan di Choregraphe.

Python

```
# Mengatur ambang batas menjadi lebih ketat (misalnya, 0.5)
# face_detection_service.setRecognitionConfidenceThreshold(0.5)
```

D. Contoh Skrip Python: Aplikasi Sederhana untuk Belajar dan Menyapa

Berikut adalah contoh skrip yang menggabungkan beberapa fungsi di atas:

Python

```
# (Pastikan kode koneksi sesi dan inisialisasi layanan sudah ada di sini)
```

```
def main_application():
```

```
    # Contoh: Pelajari wajah baru bernama "Tamu"
```

```
    # learn_new_face("Tamu")
```

```
    # time.sleep(5) # Beri waktu untuk proses pembelajaran
```

```
    # Tampilkan semua wajah yang telah dipelajari
```

```
    # show_learned_faces()
```

```
    # Mulai berlangganan event deteksi wajah
```

```
    subscribe_to_face_detection()
```

```
    print("Robot sekarang akan mencoba mengenali wajah. Tekan Ctrl+C untuk berhenti.")
```

```
    try:
```

```
        while True:
```

```
            time.sleep(1)
```

```
        except KeyboardInterrupt:
```

```
            print("Interupsi diterima, menghentikan aplikasi.")
```

```
    finally:
```

```
        unsubscribe_from_face_detection()
```

```
        # (Optional: Putuskan koneksi sesi jika aplikasi selesai)
```

```
        # session.close()
```

```
if __name__ == "__main__":
```

```
    # Pastikan Anda sudah terhubung ke Pepper sebelum memanggil main_application
```

```
    if session.isConnected():
```

```
        main_application()
```

E. Mengaktifkan dan Menonaktifkan Pengenalan/Pelacakan

- `face_detection_service.setRecognitionEnabled(bool_value)` mengaktifkan atau menonaktifkan proses pengenalan wajah. Menonaktifkan pengenalan dapat

mempercepat proses deteksi wajah jika hanya deteksi yang dibutuhkan.¹²

- `face_detection_service.setTrackingEnabled(bool_value)` mengaktifkan atau menonaktifkan pelacakan wajah. Pelacakan biasanya memungkinkan robot untuk terus mengikuti wajah meskipun orang tersebut tidak lagi menghadap kamera, namun dapat meningkatkan kemungkinan deteksi palsu.¹²

Penggunaan Python SDK memberikan fleksibilitas dan kontrol yang jauh lebih besar atas sistem pengenalan wajah Pepper, memungkinkan pengembang untuk membuat aplikasi yang lebih kompleks dan disesuaikan dengan kebutuhan spesifik.

Tabel IV.1: Metode Kunci API Python ALFaceDetection untuk Manajemen Wajah

Method Signature	Deskripsi Singkat	Parameter Kunci	Nilai Kembali	Contoh Penggunaan (Konseptual)
<code>learnFace(name)</code>	Mempelajari wajah baru dan mengaitkannya dengan name.	name (string)	True jika sukses	<code>face_proxy.learnFace("Alice")</code>
<code>reLearnFace(name)</code>	Menambahkan gambar identifikasi baru ke profil name yang sudah ada (hingga 10 gambar).	name (string)	True jika sukses	<code>face_proxy.reLearnFace("Alice")</code>
<code>getLearnedFacesList()</code>	Mengambil daftar semua nama wajah yang telah dipelajari.	Tidak ada	List string	<code>names = face_proxy.getLearnedFacesList()</code>
<code>forgetPerson(name)</code>	Menghapus semua data wajah yang terkait dengan name.	name (string)	True jika sukses	<code>face_proxy.forgetPerson("Alice")</code>
<code>clearDatabase()</code>	Menghapus semua wajah yang telah	Tidak ada	True jika sukses	<code>face_proxy.clearDatabase()</code>

	dipelajari dari basis data.			
setRecognitionConfidenceThreshold(threshold)	Mengatur ambang batas kepercayaan untuk pengenalan (0.0 - 1.0, default 0.4).	threshold (float)	Tidak ada	face_proxy.setRecognitionConfidenceThreshold(0.5)
setRecognitionEnabled(enable)	Mengaktifkan/menonaktifkan proses pengenalan wajah.	enable (boolean)	Tidak ada	face_proxy.setRecognitionEnabled(False)
setTrackingEnabled(enable)	Mengaktifkan/menonaktifkan pelacakan wajah.	enable (boolean)	Tidak ada	face_proxy.setTrackingEnabled(True)
(melalui ALMemory) subscriber("FaceDetected")	Berlangganan event yang dipicu saat wajah terdeteksi/dikenali.	Nama event, nama subscriber unik, metode callback	Objek subscriber	memory.subscriber("FaceDetected").signal.connect(on_face_detected)

Tabel ini berfungsi sebagai referensi cepat yang krusial bagi pengembang yang menggunakan Python, memusatkan panggilan API terpenting dari dokumentasi ¹² untuk memfasilitasi implementasi permintaan pengguna.

V. Mengelola Basis Data Wajah yang Telah Dipelajari Pepper

Manajemen basis data wajah yang telah dipelajari oleh Pepper adalah aspek penting untuk memastikan kinerja pengenalan yang optimal dan relevan dari waktu ke waktu. Serangkaian panggilan API yang tersedia ¹² menunjukkan bahwa basis data wajah ini bukanlah entitas statis, melainkan dirancang untuk dikelola secara aktif.

A. Pentingnya Manajemen Basis Data

Basis data wajah yang tidak terkelola dengan baik dapat menimbulkan beberapa masalah. Akumulasi data wajah yang tidak lagi relevan atau berkualitas buruk dapat berpotensi menyebabkan kesalahan identifikasi. Meskipun tidak ada batasan keras yang secara eksplisit disebutkan mengenai jumlah maksimum wajah yang dapat disimpan¹⁴, performa pengenalan secara teoritis dapat menurun jika basis data menjadi sangat besar karena waktu pencarian yang lebih lama. Setiap wajah yang dipelajari, terutama dengan beberapa gambar pendukung per individu¹², juga akan mengonsumsi sebagian dari kapasitas penyimpanan internal robot (Pepper memiliki 32GB eMMC flash memory¹⁴).

B. Melihat Daftar Orang yang Dikenal

Untuk mengetahui siapa saja yang telah dipelajari oleh Pepper, metode `ALFaceDetectionProxy.getLearnedFacesList()` pada Python SDK dapat digunakan.¹² Metode ini mengembalikan daftar nama semua individu yang tersimpan. Hingga saat ini, Choregraphe tampaknya tidak menawarkan cara langsung untuk melihat daftar ini tanpa menggunakan box Python kustom yang memanggil API tersebut.

C. Strategi untuk Meningkatkan Akurasi Pengenalan untuk Individu Tertentu

Jika Pepper kesulitan mengenali seseorang yang telah dipelajari, atau jika akurasi pengenalan perlu ditingkatkan, metode `reLearnFace(name)` adalah solusinya.¹² Metode ini memungkinkan penambahan hingga 10 gambar berbeda untuk satu orang, yang dapat membantu robot membangun profil wajah yang lebih kuat dan tahan terhadap variasi. Sangat disarankan untuk melakukan proses

`reLearnFace` dalam kondisi pencahayaan yang baik dan dengan sedikit variasi sudut wajah (tetap dalam batas toleransi yang diizinkan, lihat batasan pan/tilt pada⁶). Mengajarkan wajah dengan ekspresi netral juga penting, karena wajah netral berada di antara kesedihan dan kebahagiaan, sehingga memudahkan pengenalan saat ekspresi berubah.⁶

D. Menghapus Wajah: Individual dan Keseluruhan

Ada kalanya diperlukan untuk menghapus data wajah dari Pepper. Untuk menghapus data individu tertentu, gunakan `ALFaceDetectionProxy.forgetPerson(name)`.¹² Jika seluruh basis data wajah perlu dihapus—misalnya, saat robot akan digunakan di lingkungan baru, untuk mengatasi masalah privasi, atau untuk memulai dari awal jika kualitas pengenalan menurun secara umum—metode

`ALFaceDetectionProxy.clearDatabase()` dapat digunakan.¹² Kemampuan untuk

menghapus data biometrik ini sangat penting, mengingat implikasi privasi yang melekat pada penyimpanan informasi wajah dan nama.

E. Pertimbangan Ukuran Basis Data dan Performa

Seperti yang telah disebutkan, dokumentasi tidak secara eksplisit menyatakan batas maksimum jumlah wajah yang dapat dipelajari Pepper.⁶ Namun, kapasitas penyimpanan internal sebesar 32GB eMMC¹⁴ adalah sumber daya yang terbatas dan dibagi dengan semua data dan aplikasi robot lainnya. Setiap entri wajah, terutama jika menggunakan fitur

reLearnFace untuk menyimpan beberapa gambar per orang (hingga 10 gambar¹²), akan memakan ruang. Oleh karena itu, meskipun tidak ada batasan yang dikodekan secara keras, ada batasan praktis yang ditentukan oleh sumber daya sistem dan potensi dampak pada kecepatan pencarian dalam basis data yang sangat besar. Dianjurkan untuk melakukan peninjauan dan pembersihan basis data secara berkala, terutama jika robot berinteraksi dengan banyak orang berbeda dari waktu ke waktu atau jika kualitas pengenalan untuk individu tertentu menurun.

Manajemen basis data yang efektif memastikan bahwa kemampuan pengenalan wajah Pepper tetap akurat, efisien, dan relevan dengan konteks penggunaannya.

VI. Praktik Terbaik dan Pemecahan Masalah

Untuk memaksimalkan efektivitas sistem pengenalan wajah Pepper dan mengatasi potensi masalah, penting untuk memahami kondisi optimal serta cara penanganan isu yang umum terjadi. Kinerja sistem pengenalan wajah, meskipun canggih, sangat bergantung pada kualitas input visual yang diterima oleh sensor robot.

A. Mengoptimalkan Kondisi untuk Pembelajaran dan Pengenalan

1. **Pencahayaan:** Faktor ini sangat krusial. Kondisi pencahayaan kantor (sekitar 100 hingga 500 lux) dianggap sebagai standar dasar.⁶ Hindari pencahayaan dari belakang (backlight) yang kuat atau bayangan parsial pada wajah, karena dapat mengganggu deteksi dan pembelajaran.⁶ Jika deteksi tidak berjalan baik, pertimbangkan untuk mengaktifkan fitur auto gain pada kamera atau menyesuaikan kontras kamera secara manual melalui antarmuka Monitor.⁶
2. **Jarak:**

- **Deteksi:** Lebar wajah minimal dalam gambar adalah sekitar 20 piksel. Untuk orang dewasa, ini setara dengan jarak sekitar 2 meter dalam gambar QVGA dan 4 meter dalam gambar VGA.⁶ Sumber lain menyebutkan minimal ~45 piksel dalam gambar QVGA, yang setara dengan sekitar 3 meter dengan kamera VGA v3.x dan lebih dari 2 meter pada kamera HD v4.⁷
 - **Pengenalan:** Lebar wajah minimal juga 20 piksel, namun ukuran minimal 40 piksel sangat direkomendasikan untuk hasil pembelajaran dan pengenalan yang lebih baik.⁶
3. **Pose Wajah:**
- **Kemiringan (Tilt/Pan):** Untuk deteksi, toleransi kemiringan kepala adalah sekitar +/- 15 derajat⁶ hingga +/- 20 derajat⁷ dari posisi menghadap lurus ke kamera. Pengenalan wajah lebih sensitif terhadap variasi ini.⁶
 - **Rotasi dalam Bidang Gambar:** Toleransi rotasi wajah dalam bidang gambar adalah maksimal +/- 45 derajat untuk deteksi⁶, atau +/- 20 derajat menurut sumber lain.⁷
 - **Ekspresi Wajah:** Saat proses pembelajaran, subjek diharapkan menghadap kamera dengan ekspresi netral. Ini karena ekspresi netral dianggap sebagai titik tengah antara berbagai emosi seperti sedih dan senang, sehingga memudahkan pengenalan di kemudian hari ketika subjek mungkin menunjukkan ekspresi berbeda.⁶
4. **Lingkungan:** Meskipun sistem dirancang untuk berfungsi di lingkungan dunia nyata, latar belakang yang terlalu ramai atau bergerak cepat dapat berpotensi mengganggu.

B. Mengatasi Masalah Umum

1. **Gagal Mempelajari Wajah Baru:**
 - **Penyebab:** Pencahayaan buruk, jarak tidak optimal, pose wajah tidak sesuai, wajah terhalang sebagian.
 - **Solusi:** Periksa dan perbaiki kondisi pencahayaan. Pastikan orang tersebut berada pada jarak yang tepat dan wajahnya terlihat jelas oleh kamera Pepper. Minta orang tersebut untuk memasang ekspresi netral dan menghadap lurus ke robot. Selama proses pembelajaran melalui Choregraphe atau API, mata robot seharusnya berubah menjadi hijau sebagai indikasi keberhasilan.⁶ Jika mata tetap biru, kondisi pembelajaran perlu diubah.
2. **Kesalahan Identifikasi / Orang yang Dikenal Tidak Dikenali:**
 - **Penyebab:** Kondisi pencahayaan atau ekspresi saat pengenalan berbeda signifikan dari saat pembelajaran, orang berada pada sudut atau jarak yang sulit, skor pengenalan di bawah ambang batas yang ditetapkan.
 - **Solusi:** Optimalkan kondisi lingkungan. Minta orang tersebut untuk melihat

langsung ke Pepper. Gunakan metode learnFace (untuk pembelajaran baru) atau lebih baik lagi reLearnFace (untuk memperkuat data orang yang sudah ada) untuk menambahkan lebih banyak sampel wajah orang tersebut dalam berbagai kondisi (namun tetap baik).⁶ Jika menggunakan Python SDK, periksa scoreReco dari event FaceDetected dan pertimbangkan untuk menyesuaikan ambang batas kepercayaan melalui setRecognitionConfidenceThreshold().¹²

3. Pengenalan Lambat:

- **Penyebab:** Potensi basis data yang sangat besar (meskipun tidak secara eksplisit dibuktikan, ini adalah kekhawatiran umum pada sistem berbasis pencarian), beban pemrosesan tinggi pada robot.
- **Solusi:** Kelola ukuran basis data secara berkala. Jika hanya deteksi wajah yang diperlukan untuk sementara waktu (bukan pengenalan nama), pertimbangkan untuk menonaktifkan proses pengenalan melalui setRecognitionEnabled(False) menggunakan Python SDK untuk mempercepat deteksi.¹²

C. Memahami Skor Pengenalan dan Ambang Batas

Sistem pengenalan wajah menghasilkan skor kepercayaan (scoreReco) antara 0 dan 1 untuk setiap potensi kecocokan.⁶ Skor yang lebih tinggi menunjukkan tingkat kepastian yang lebih besar bahwa wajah yang terdeteksi adalah orang yang dikenali. Ambang batas (

threshold), yang secara default bernilai 0.4 jika diakses melalui API Python¹², menentukan apakah suatu kecocokan dianggap valid.

- **Ambang Batas Tinggi:** Lebih ketat, mengurangi kemungkinan salah identifikasi (false positive), tetapi dapat meningkatkan kemungkinan gagal mengenali orang yang seharusnya dikenal (false negative).
- **Ambang Batas Rendah:** Lebih longgar, meningkatkan kemungkinan mengenali lebih banyak wajah, tetapi juga meningkatkan risiko salah identifikasi.

Penyesuaian ambang batas ini (melalui API Python) memungkinkan pengembang untuk menyeimbangkan antara kekokohan dan spesifisitas sesuai dengan kebutuhan aplikasi. Misalnya, aplikasi yang memerlukan keamanan tinggi mungkin menggunakan ambang batas yang lebih tinggi, sementara aplikasi interaksi sosial kasual mungkin dapat mentolerir ambang batas yang sedikit lebih rendah. Perlu dipahami bahwa algoritma pengenalan pada dasarnya adalah proses 2D yang membandingkan fitur-fitur wajah.⁶ Hal ini menjelaskan mengapa perubahan signifikan dalam pose 3D (seperti melihat dari samping jika hanya dipelajari dari depan) dapat sangat mempengaruhi kemampuan pengenalan, karena proyeksi 2D dari fitur-fitur tersebut

akan berubah.

Tabel VI.1: Pemecahan Masalah Umum Pengenalan Wajah pada Pepper

Masalah	Kemungkinan Penyebab	Solusi/Tindakan (Choregraphe)	Solusi/Tindakan (Python SDK)
Pepper gagal mempelajari wajah baru (mata tetap biru).	Pencahayaan buruk, wajah terlalu jauh/dekat, ekspresi tidak netral, wajah terhalang sebagian.	Sesuaikan posisi orang, perbaiki pencahayaan ruangan. Ulangi inisiasi box "Learn Face".	Periksa umpan kamera jika memungkinkan. Pastikan learnFace() dipanggil dalam kondisi baik. Mata robot harus berubah hijau.
Orang yang dikenal tidak dikenali.	Pencahayaan/ekspresi berbeda dari saat belajar, orang pada sudut/jarak sulit, skor pengenalan di bawah ambang batas.	Minta orang untuk melihat langsung ke Pepper, perbaiki pencahayaan. Pertimbangkan untuk mempelajari ulang wajah tersebut.	Gunakan reLearnFace() dengan gambar baru. Periksa scoreReco dari event FaceDetected; pertimbangkan menyesuaikan ambang batas via setRecognitionConfidenceThreshold().
Terjadi salah identifikasi (misidentification).	Ambang batas terlalu rendah, kemiripan antar wajah yang dipelajari, kualitas data pembelajaran buruk.	Pelajari ulang wajah yang sering salah diidentifikasi dengan kondisi lebih baik. Hapus entri yang salah jika perlu.	Tingkatkan setRecognitionConfidenceThreshold(). Gunakan reLearnFace() untuk wajah yang terlibat. Hapus entri yang tidak akurat dengan forgetPerson().

Dengan memperhatikan praktik terbaik ini dan memahami cara mengatasi masalah umum, pengguna dapat meningkatkan keandalan dan efektivitas fitur pengenalan wajah Pepper secara signifikan.

VII. Kesimpulan

Kemampuan robot Pepper untuk mempelajari wajah baru, mengingatnya, dan kemudian mengenali individu berdasarkan nama merupakan fitur kunci yang secara signifikan meningkatkan kualitas interaksi manusia-robot. Laporan ini telah menguraikan mekanisme teknis di balik fitur ini, serta langkah-langkah konfigurasi baik melalui antarmuka grafis Choregraphe maupun secara programatik menggunakan Python SDK.

A. Rekapitulasi Pengaturan Pengenalan Wajah Pepper

Proses ini melibatkan penggunaan modul ALFaceDetection yang menangani deteksi dan pengenalan wajah. Tahap pembelajaran, di mana Pepper merekam dan mengasosiasikan wajah dengan sebuah nama, adalah fundamental. Ini dapat dilakukan dengan mudah melalui box Learn Face di Choregraphe atau metode `learnFace()` dalam Python. Setelah dipelajari, Pepper dapat mengidentifikasi individu tersebut saat bertemu lagi, memicu respons seperti sapaan verbal yang dipersonalisasi. Pengelolaan basis data wajah yang berkelanjutan, termasuk memperbarui data orang yang sudah ada (`reLearnFace`) dan menghapus entri yang tidak lagi relevan (`forgetPerson`, `clearDatabase`), juga penting untuk menjaga kinerja optimal. Faktor lingkungan seperti pencahayaan, jarak, dan pose wajah subjek memainkan peran krusial dalam keberhasilan baik tahap pembelajaran maupun pengenalan.

B. Kekuatan Interaksi yang Dipersonalisasi

Kemampuan untuk mengenali dan memanggil nama seseorang mengubah interaksi dengan Pepper dari sekadar fungsional menjadi lebih personal dan menarik. Hal ini sejalan dengan tujuan desain Pepper sebagai robot sosial yang mampu menjalin hubungan yang lebih alami dengan manusia. Ketika Pepper dapat mengingat individu, interaksi terasa lebih seperti percakapan dengan entitas yang sadar akan kehadiran spesifik pengguna, bukan hanya respons generik.

Implementasi yang berhasil dari pembelajaran dan pengenalan wajah bukan hanya merupakan tujuan akhir, tetapi juga fondasi untuk aplikasi interaksi manusia-robot yang lebih kompleks dan kaya. Mengetahui *siapa* yang berinteraksi dengan Pepper memungkinkan dialog yang disesuaikan, kemampuan untuk mengingat interaksi sebelumnya (jika diprogram demikian), dan adaptasi perilaku terhadap preferensi individu.

C. Dorongan untuk Eksplorasi Lebih Lanjut

Proses mengajarkan wajah kepada Pepper dan menyempurnakan pengenalannya juga merupakan bentuk adaptasi bersama: pengguna belajar cara terbaik menyajikan informasi kepada robot, dan robot (melalui basis datanya) belajar tentang pengguna. Dinamika ini menunjukkan bahwa kemampuan interaktif robot dapat berkembang seiring dengan lebih banyak data (dan data berkualitas lebih baik).

Pengguna didorong untuk menjelajahi modul NAOqi terkait lainnya untuk membangun aplikasi interaktif yang lebih kaya. Misalnya, ALFaceCharacteristics dapat digunakan untuk mendeteksi usia, jenis kelamin, dan ekspresi wajah ⁵, sementara

ALEngagementZones dapat membantu memahami tingkat keterlibatan pengguna dengan robot.¹² Kombinasi fitur-fitur ini membuka potensi besar untuk menciptakan pengalaman yang benar-benar cerdas dan adaptif. Untuk dokumentasi lebih lanjut dan pembaruan, portal pengembang SoftBank Robotics tetap menjadi sumber daya utama.⁴

Karya yang dikutip

1. Pepper Robot Online Training - RobotLAB, diakses Juni 13, 2025, <https://www.robotlab.com/store/pepper-robot-online-training>
2. Pepper - Aldebaran robotics, diakses Juni 13, 2025, <https://aldebaran.com/en/pepper/>
3. Pepper Robot Features - Why Pepper? - GWS Robotics, diakses Juni 13, 2025, <https://www.gwsrobotics.com/why-pepper-robot>
4. Pepper — Robot House User Documentation 1.0rc1 documentation, diakses Juni 13, 2025, <https://robohouse.herts.ac.uk/userdocs/robots/pepper/index.html>
5. aerdem4/robot-photo-studio: A simple Pepper application ... - GitHub, diakses Juni 13, 2025, <https://github.com/aerdem4/robot-photo-studio>
6. ALFaceDetection — Aldebaran 2.4.3.28-r2 documentation, diakses Juni 13, 2025, <http://doc.aldebaran.com/2-4/naoqi/peopleperception/alfacedetection.html>
7. ALFaceDetection — NAO Software 1.12 documentation, diakses Juni 13, 2025, <https://www.cs.cmu.edu/~cga/nao/doc/reference-documentation/naoqi/vision/alfacedetection.html>
8. NAO Face Recognition Tutorial - YouTube, diakses Juni 13, 2025, <https://www.youtube.com/watch?v=nF2suPjTF9g>
9. RobotLAB Choregraphe Training - CCSF, diakses Juni 13, 2025, https://hills.ccsf.edu/~grwoo/cs110a_sp20/pepper_robot.pdf
10. Pepper-Controller/pepper/robot.py at main · incognite-lab/Pepper-Controller - GitHub, diakses Juni 13, 2025, <https://github.com/incognite-lab/Pepper-Controller/blob/main/pepper/robot.py>
11. Pepper API for Python - Cvut, diakses Juni 13, 2025, https://people.ciirc.cvut.cz/tesarm11/pepper_docs/pepper_doc.html
12. ALFaceDetection API — Aldebaran 2.5.11.14a documentation, diakses Juni 13, 2025, <http://doc.aldebaran.com/2-5/naoqi/peopleperception/alfacedetection-api.html>
13. ALFaceDetection API — Aldebaran 2.5.11.14a documentation, diakses Juni 13, 2025, <http://doc.aldebaran.com/2-5/naoqi/peopleperception/alfacedetection-api.html?highlight=learn%20face>
14. PEPPER- Technical Specifications : - Support : - United Robotics Group, diakses Juni 13, 2025, <https://support.unitedrobotics.group/en/support/solutions/articles/80000958735-pepper-technical-specifications>
15. Pepper Robot Can Scan Your Face and Ask you to Wear a Face Mask - RobotLAB, diakses Juni 13, 2025, <https://www.robotlab.com/blog/pepper-robot-can-scan-your-face-and-ask-you-to-wear-a-face-mask>
16. What is the Pepper robot? Features and uses of a humanoid bot - Standard Bots, diakses Juni 13, 2025, <https://standardbots.com/blog/pepper-robot>
17. Assessment of Pepper Robot's Speech Recognition System through the Lens of Machine Learning - MDPI, diakses Juni 13, 2025,

<https://www.mdpi.com/2313-7673/9/7/391>

18. Face Characteristics - Get Expression — Aldebaran 2.4.3.28-r2 documentation, diakses Juni 13, 2025,

http://doc.aldebaran.com/2-4/software/choregraphe/tutos/get_expression.html