

## ABSTRAK

Presensi perkuliahan merupakan suatu proses pencatatan kehadiran dalam mengikuti kegiatan belajar mengajar dalam perkuliahan sehingga membutuhkan sistem presensi yang baik, salah satunya dengan sistem presensi otomatis. Sistem presensi otomatis dapat mempermudah sebuah kegiatan untuk mencatat kehadiran. Pada sistem presensi otomatis ini menggunakan Haar Cascade sebagai metode utamanya serta menggunakan *smart-phone* sebagai kamera untuk media *face recognition*. Metode Haar Cascade digunakan sebagai sistem pendeteksi wajah dan mata serta penggunaan fungsi *Local Binary Pattern Histograms* (LBPH) sebagai pengenalan wajah (*face recognition*) Sistem presensi otomatis ini dibuat menggunakan bahasa program python dengan aplikasi IDLE. Kamera *smart-phone* terhubung dengan program menggunakan alamat IP kamera yang didapatkan dari aplikasi droidcam. Selain itu sistem presensi otomatis ini menggunakan sistem GUI yang dibuat sederhana agar mempermudah dalam melakukan proses presensi serta menggunakan microsoft excel dan *folder* sebagai penyimpanan data yang telah diinputkan. Pengujian yang dilakukan dengan menguji kinerja sistem secara keseluruhan. Sistem ini dapat mendeteksi lebih dari 1 orang dengan memanfaatkan fungsi dari metode Haar Cascade, selain itu pada pengujian lainnya seperti jarak ideal, sudut ideal, cahaya ideal, parameter penghalang dan sistem penyimpanan. Berdasarkan data hasil pengujian yang telah dilakukan, jarak ideal melakukan presensi otomatis ini pada rentang 30cm-70cm, untuk sudut ideal pada rentang 45 derajat-90 derajat, untuk cahaya ideal pada level cahaya 20-100, untuk parameter penghalang terdapat benda yang mengganggu kinerja dari metode Haar Cascade seperti plastik dan buku, serta sistem penyimpanan yang menyimpan data diri tersimpan pada microsoft excel dan gambar hasil *capture* latih wajah tersimpan pada *folder*. Pengujian ini dilakukan dari penghubungan IP kamera sampai data yang tersimpan pada penyimpanan.

Kata kunci: Droidcam, *Face Recognition*, GUI, Haar Cascade, *Local Binary Pattern Histograms*

## **ABSTRACT**

*Lecture attendance is the process of recording participation in teaching and learning activities during lectures. This process necessitates a reliable attendance system, one of which is an automatic attendance system. An automatic attendance system streamlines attendance recording by employing the Haar Cascade method as its core technique and utilizing a smartphone camera for face recognition. The Haar Cascade method serves as both a face and eye detection system, while the Local Binary Pattern Histograms (LBPH) function is employed for face recognition. This automatic attendance system has been developed using the Python programming language through the IDLE application. The smartphone camera connects to the program via the camera's IP address obtained from the Droidcam application. Furthermore, this automatic attendance system incorporates a user-friendly GUI system to simplify the attendance process. Data collected through this system is stored in Microsoft Excel and folders. Testing has been conducted to evaluate the overall performance of the system, including its ability to detect multiple individuals using the Haar Cascade method. Additional tests have assessed various factors such as optimal distance, angle, lighting conditions, potential barriers, and the efficiency of the storage system. Based on the test results, the ideal distance for using this automatic attendance system falls within the range of 30cm to 70cm, the optimal angle is within 45 to 90 degrees, and the ideal lighting level ranges from 20 to 100. The system may encounter issues when there are objects obstructing the Haar Cascade method's performance, such as plastic or books. Personal data is stored in Microsoft Excel, and facial recognition training images are saved in a designated folder. Testing encompasses the entire process, from connecting the camera's IP to accessing data stored in the system.*

*Keywords: Droidcam, Face Recognition, GUI, Haar Cascade, Local Binnary Pattern Histograms*