International Journal of Information System & Technology
Akreditasi No. 36/E/KPT/2019 | Vol. 4, No. 1, (2020), pp. 417-427

Face Recognition Using Tiny Yolo V2 Algorithm as Attendance System

Hafidz Sanjaya[1], Dony Susandi[2], Sandi Fajar Rodiyansyah[3]

[123]Universitas Majalengka – Jawa Barat – Indonesia

hafidz@unma.ac.id

Abstract

Nowadays many websites use the usual online attendance system which does not pay attention to safety and comfort factors so that attendance activities still have a gap of cheating. Therefore, in this study the study of the application of face recognition systems in real-time using the Tiny Yolo V2 algorithm in the online attendance system. The study was conducted with several stages starting from collecting face images, the process of image improvement (preprocessing), face detection, face recognition, and data integration using web service. The test results of 10 students, each of whom has a face image facing forward as a dataset with 4 variations of distance, each of which performs 10 different face position scenarios. Based on the test results it can be concluded that the farther the distance of the face image with the webcam, the success rate decreases, it is shown at a distance of 0.5 meters the percentage of success reaches 97% and at a distance of 2 meters 88% where 2 faces are not detected and identified at the distance is due to wearing glasses and having rather dark skin.

Keywords: *biometrics, face recognition, attendance system, web services, Tiny Yolo V2*

1. Introduction

Safety and comfort are two factors that exist in an attendance system so that attendance activities can run smoothly and optimally. The security factor is more closely correlated with the interests of the university to get accurate and fast student attendance data. While the convenience factor is more directed to what is felt by students in making attendance more easily, quickly, and minimally obstacles. Therefore, it is needed the application of supporting media in the form of attendance systems that adopt biometric technology to support the safety and comfort of the attendance system, such as attendance systems using face recognition. The second factor might be able to solve the problems that occur in the attendance system, but the relationship between the number of people and biometric datasets are interrelated. The more people, the more datasets are needed. A regular attendance system with standard capacity cannot solve these problems, so another factor that is thought to be able to solve the data integration is needed, namely data service-based data integration using web services.

Web service is a set of program functions to do certain work in this case, namely data manipulation, retrieving, adding, or changing data [1]. In web services, the relationship between client and server is bridged by a web service with a certain format so that access to the database is not handled by the server. One web service architecture that is widely used in dealing with large amounts of data is the Representational State Transfer (REST). As a protocol, REST uses HTTP (Hypertext Transfer Protocol) for communication between data. Architecturally, REST displays data using Universal Resource Identifiers (URIs) that can be accessed using the internet or intranet. The use of simple, concise, and fast REST makes data integration between the attendance system interface with information service providers easier and faster so that the system interface is not burdened by the growing amount of data on the attendance system. The use of web services for student biometric information will be more open because it is based on data services so

ISSN : 2580-7250

that it is easily accessed anywhere, anytime, and can be used by any application program, it is supported by the presence of internet technology.

Internet access cannot be separated from devices used to access the internet itself, such as desktop computers, laptops, smartphones or cellphones, and tablets. In Indonesia, the scale of internet users every day using desktop computers is quite high, it has a percentage of 68.9%, using laptop devices has a percentage of 56.6%, using a smartphone or mobile devices has a percentage of 93.9%, and using devices tablets have a percentage of 85.2% [2].

Checking the internet using various devices shows that the needs of each internet user are different, which causes the distribution and availability of information or data on the internet more and more easily accessible. Devices that are widely used by users are devices that have a camera. The camera itself has various types, one of which is a digital camera.

Digital cameras are tools for making images of objects for refraction through lenses on CCD sensors and more recently on BSI-CMOS (Back Side Illuminated) sensors that are more efficient for more sophisticated cameras whose results are then recorded in digital format to in digital storage media [3]. Each image or video obtained from the camera has metadata or pattern that can be processed to produce information that can be used for a particular purpose, one of which is to recognize objects that are in the image data.

Recognizing an object in the image, in general, is one of the problems in computer vision (computer vision). Computer vision has not been able to mimic the ability of humans to understand image information, recognize objects in an image such as recognizing a person's face. For humans, it is a simple and easy thing, but in computer science, it becomes difficult to process one of them because there is no relationship between the representation of image data (pixels) stored and read inside the computer with images obtained from the real world. The face recognition problem becomes increasingly difficult for computers with the influence of the face such as perspective, distance, lighting, resolution, aging, and expression. With the variations of these factors, the image can have the same meaning with different classifications.

Various methods in computer science can be used to recognize faces in images. A convolutional neural network (CNN) is a development of a multilayer perceptron (MLP) which is designed to process two-dimensional image data. Convolutional neural networks are specifically designed to process data in the form of multidimensional arrays such as pixel data in color images that are two-dimensional arrays for each color channel [4]. CNN is included in the type of Deep Neural Network because of the high network depth and is widely applied to images. In 2012, Alex Krizhevsky with his CNN implementation succeeded in winning the ImageNet Large Scale Visual Recognition Challenge 2012. This achievement showed that the CNN method proved to be superior to machine learning methods such as SVM in the case of image data object classification.

2. Related Works

Analyzing the implementation of the convolutional neural network algorithm on face recognition using The Extended Yale Face B testing data with 75.79% accuracy results using a dropout system of 86.71% with filter size 3x3 has an accuracy of 89.73% and 5x5 of 86.71% [5].

Meanwhile, research which makes a face detection system for student attendance at the PGRI Kediri archipelago university by using the eigenface method and combined with the Manhattan algorithm (city block) with 10 training data, bright lighting and a distance of 50cm which results in 80% face matching [6].

A study entitled Implementation of Convolutional Neural Network Using Hard Tomato Images which implemented a convolutional neural network using hard packages in the R programming language used to classify decent and improper tomatoes using

tomato images. This study tested 100 tomato image samples and resulted in an accuracy rate of 90% [7].

Examines facial recognition systems based on facial recognition in classrooms automatically using the convolutional neural network method with two categories of training data i.e. 1200 face image data see the camera and 1200 image data do not see the camera. The research resulted in a face clarification with an accuracy of 93.33% depending on the conditions of input image capture, face detection, and the clarification process [8].

3. Research Methodology

3.1. Data Collection and Data Processing

The collection of one image in the form of a sample photo appears straight ahead from 10 people taken from various sources, including from smartphone cameras and digital cameras. After 10 sample photos are obtained, the next is image preprocessing in the form of cropping and resizing the image. The first stage is cropping, the process of cropping is done to select sample photos so that it is more focused on all areas of the face only, the result of this process is a square image that contains the face concerning aspect ratios. Resizing of the image (resizing) is done on images that have been cropped because they have different resolutions caused by images obtained from various types of cameras. In this resizing process, the image is uniform in size to 416x416 pixels according to the Tiny Yolo v2 algorithm architecture.

3.2. Design of Face Detection Systems

The algorithm used to detect faces in this study is Tiny Yolo v2, where the algorithm was created to be used as object detection or object recognition. Unlike the face detection algorithm in general, that performs image processing, the Tiny Yolo v2 algorithm in this study in detecting faces acts as an object recognition with 1 class of objects, namely faces.

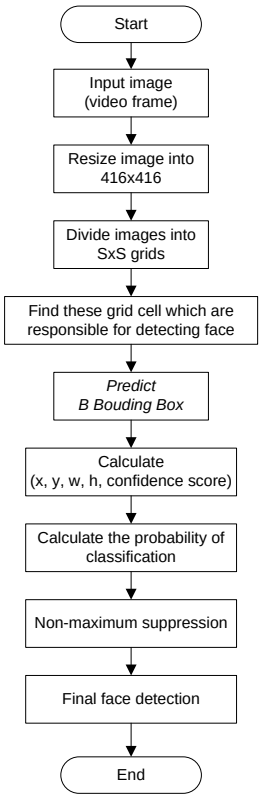

Figure 1. Flowchart Face Detection System

The input image used in face object detection is from video per frame with a video resolution of 360x270 pixels.

The video resolution used for streaming webcam is 360x270 in pixels, so the size of each frame is the same as the video resolution. In deep learning, each pixel of an image is a feature. For one frame with 360x270 resolution, the vector features are (x_{11}, x_{12}, . . . , x_{1n}, x_{21}, x_{22}, . . ., x_{2n}, . . ., x_{n1}, x_{n2}, . . ., $x_{360,270}$). So, the frame size of 360x720 is resized by face-api.js to 416x416 according to Tiny Yolo v2 architecture.

The next step in the Tiny Yolo v2 algorithm is dividing the input image into an SxS grid with a value of S = 13 that matches the final result of the convolution algorithm that divides the image into a 13x13 grid

Face models that have been trained or commonly called pre-trained datasets can automatically determine the center of the face object in the input image. If the center of the face object falls into the lattice cell, the lattice cell is responsible for detecting the face. If the cell contains a face, then the probability of a face is one or Pr (face) = 1, and vice versa or Pr (face) = 0.

Each lattice cell predicts a bounding box B (B bounding box) and the confidence score of each bounding box. The value of B can be determined dynamically according to the algorithm's optimal performance in detecting objects. In this study, the value of B is 5 which is following the algorithm architecture. The value is that the algorithm used can predict 5 bounding boxes for each face object that appears, the size of each bounding box can be different because it makes predictions on different layers.

Each bounding box contains five detection parameters, namely the coordinates (x, y), the width of the box, the height of the box, and the score of the detected object. The score value can be searched by the following formula (1).

$$score = Pr(face) * IOU \quad (1)$$

Where IOU is Intersection Over Union. IOU is defined as follows.

$$IOU = \frac{area(P \cap G)}{area(P \cup G)} \quad (2)$$

Where the variable G is the actual position of the face (ground truth bounding box), and the variable P is the predicted position (predicted bounding box). This IOU is used to calculate the correlation between reality and prediction. Where the greater the score, the better the performance in predicting. When IOU = 1, the detection performance is the best. However, in general, the performance can be said to be good when the IOU score> 0.5 in detection. Therefore, referring to the definition of Pr (face) and IOU, the distance of the score value is between 0 and 1 which shows how sure the box contains face objects and also how accurate the box is in making predictions.

In fact, in deep learning, after the target is detected, the detection system also needs to classify the target whether it is a bird, human, dog, car, etc., as in the PASCAL VOC dataset. Because this research only needs to detect one type of object, the target can only detect the object, namely the face. Then, the conditional probability is Pr (face | target) = Pr (face).

This algorithm uses NMS to maintain each bounding box B that has the best score. The first step, which is carried out by the NMS, is to select all boundary box predictions that have a probability score (threshold) that is less than the threshold and maintain more than that.

3.3. Face Recognition Architecture

The face recognition process is performed by matching the extracted features between the input face image with the image in the database. Therefore, architecture is implemented to get features or face descriptors that are used to describe the characteristics of a person's face. To calculate the face description in this study used the ResNet-34 architecture or so-called residual neural network that has 34 layers and produces vector features as many as 128 of each face that is recognized. This architecture was chosen because this architecture has good performance and can work on limited hardware or low

specifications, as used by researchers, then the architecture was chosen only has 34 layers. This architecture is not limited to face datasets used for training data, but can also be used for anyone's face recognition. With this architecture, it can determine two different faces by comparing the distance of the descriptors from each face.

In this study, to determine the distance between two descriptors is to use the Euclidean distance method. The following is a flowchart of the face recognition process.

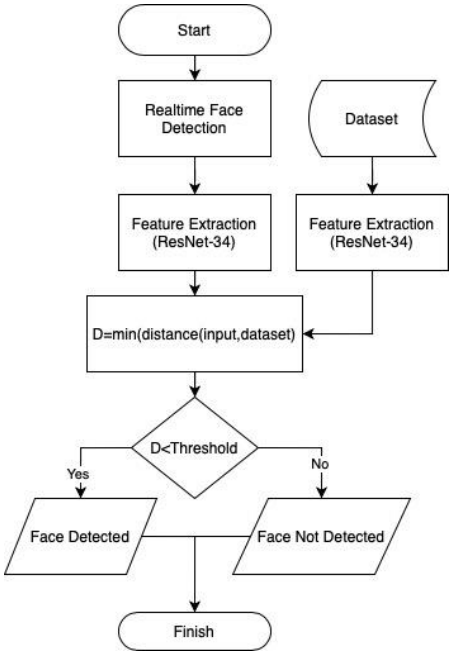

Figure 2. Flowchart Face Recognition System

From the results of face detection in real-time in the previous stage, detected faces are selected and extracted to get descriptors as input features. Each face image that is in the dataset is extracted to get the descriptor as a dataset feature.

In this process, the distance between the input image descriptors and the image descriptors in the dataset is calculated. Input image descriptors are compared one by one with the image descriptors in the dataset. In the process of measuring distance, descriptors are to use the Euclidean distance method using formula (3).

$$d_{xy} = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2} \quad (3)$$

d_{xy} is the obtained Euclidean distance value, while xi is the input image data vector and y_i is the image data vector in the dataset. After obtaining the Euclidean distance value in each image then look for the smallest value, in the flowchart above the smallest value stored in variable D.

The smallest of D, then compared to the threshold value, the value of the hold used for face recognition in this study was 0.45. if the value of D is smaller than the holding, it will display the Face ID (person ID) as proof of the input face image stored in the dataset, if it is larger, it will display the unknown status. The smallest value is taken because the smaller the Euclidean distance, the more similar the face image is stored in the dataset.

3.4. Testing Algorithm Accuracy

The scenario of this accuracy testing is carried out to determine the ability of the system to recognize person faces that are based on distance and perspective influenced by normal room lighting.

a) Face Recognition Distance
In this test, the accuracy of facial recognition will be tested with different distances, namely 0.5 meters, 1 meter, 1.5 meters, and 2 meters. The following table is used in face recognition distance testing.

b) Facial Recognition Variations
This accuracy test will be carried out with 10 different face conditions namely facing left, right, up and down with a viewpoint of 25^{o} and 45^{o} from the camera and facing forward with closed eyes and facing the front of the camera with the eyes open. The following table tests the accuracy of the facial conditions.

4. Results and Discussion

4.1. Data Collection and Data Processing

This stage collects a photo of samples looking straight ahead (frontal face) from 10 samples. The results of collecting student photos can be seen in Figure 3.

Figure 3. Samples Image

The first stage in preprocessing is cropping the photos collected so that the photo is more focused on the student's face. The results of this cropping will be different because the photos collected are obtained from different sources, such as cellphone cameras, digital cameras, and web cameras (webcam). Besides, the results of cropping are influenced by the needs of researchers in selecting faces, resulting in different sizes that can be seen in Figure 4.

Figure 4. Cropping Image

In the Tiny Yolo v2 architecture that the input image has a size of 416x416 pixels, then in this study also for images in the dataset that have gone through the cropping process will be resized to 416x416 as shown in Figure 5.

Figure 5. Image with Size 416x416

4.2. Face Detection System

In Figure 6 is a display if there is a face in the video, the face detection system will provide a bounding box on the face and display the probability of detection of face objects according to the ratio between the weight of the face in the pre-trained data with those on the camera.

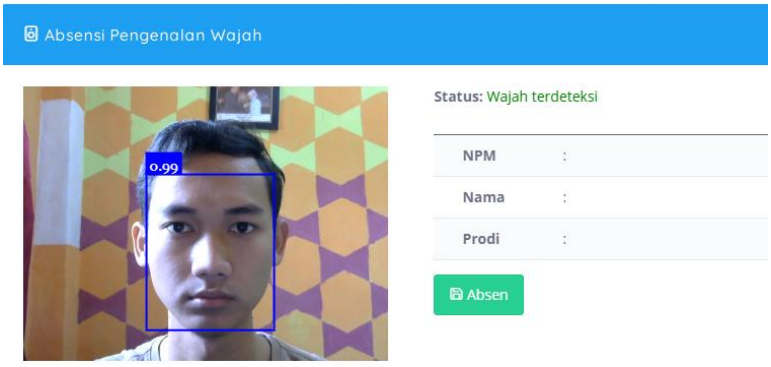

Figure 6. Face Detection System

In Figure 6 is a display if there is a face on the video, the face detection system will provide a bounding box on the face and display the probability score of the detection of face objects according to the ratio between the weight of the face in the pre-trained data with those on the camera.

4.3. Face Recognition System

In Figure 7 shows the interface when a face is detected by a face detection system, the face descriptor totaling 128 vectors in the form of a 1-dimensional array as shown in Figure 8.a which is the result of the ResNet-34 architecture of the detected face and then calculates the closest distance to the face already in the dataset as shown in Figure 8.b using Euclidean distance.

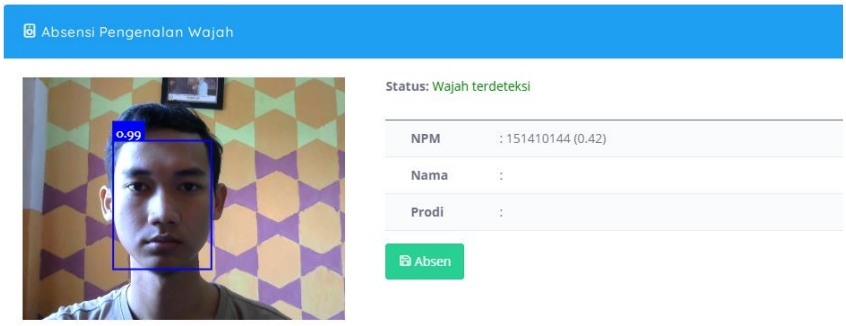

Figure 7. Face Recognition System

0: -0.10898739099502563 1: 0.17290399968624115 2: 0.05694536492228508 3: -0.03600268065929413 4: -0.09793945401906967 ... ... 124: 0.10206598043441772 125: -0.046007826924324036 126: 0.06636353582143784 127: 0.04814722761511803	0: -0.0840005874633789 1: 0.11700045317411423 2: 0.011817606166005135 3: -0.0808630064129829 4 4: -0.08169063925743103 ... ... 124: 0.10768411308526993 125: -0.024324333295226097 126: 0.08735903352499008 127: 0.028131850063800812
(a)	(b)

Figure 8. Face Descriptor from Webcam (a), Face Descriptor from Dataset (b)

To show that the entered image is identified with the image in the dataset and has a distance less than the specified face recognition value, then the distance value between the two image descriptors is calculated using Euclidean distance.

$$= \sqrt{\begin{array}{l}((-0.10898739099502563) - (-0.0840005874633789))^2 + \\ \left((0.17290399968624115) - (0.11700045317411423)\right)^2 + \cdots + \\ ((0.04814722761511803) - (0.028131850063800812))^2\end{array}}$$

$$= \sqrt{0.1809822223805851}$$

$$= 0.42542005404139693$$

Then, the value of the distance between the input image descriptor and the image descriptor in the dataset is 0.42542005404139693 or 0.42 proximity. This distance is the smallest (min) among other images in the dataset and can be identified by a face recognition system by displaying face ID because it has a distance less than the specified threshold (D), which is 0.45.

4.4 Algorithm Accuracy

a) The test results are based on variations in faces at a distance of 0.5 meters

Table 1. Experiment with a Distance of 0.5 Meters

Position	Number of experiments	number succeeded
Condition 1	10	10
Condition 2	10	10
Condition 3	10	10
Condition 4	10	9
Condition 5	10	10
Condition 6	10	9
Condition 7	10	10
Condition 8	10	9
Condition 9	10	10
Condition 10	10	10
	Average	97

On the results of testing a distance of 0.5 meters from 10 experiments (number of samples) were conducted, obtained a percentage of less than 100% for the condition of the face facing right 45°, facing up to 45° and facing down 45°.

b) The test results are based on variations in faces at a distance of 1.0 meter

Table 2. Experiment with a Distance of 1.0 Meter

Position	Number of experiments	number succeeded
Condition 1	10	10
Condition 2	10	10
Condition 3	10	10
Condition 4	10	9
Condition 5	10	10
Condition 6	10	10
Condition 7	10	9
Condition 8	10	8
Condition 9	10	10
Condition 10	10	10
	Average	96

In the test results of a distance of 1 meter from 10 experiments (number of samples) were conducted, obtained 2 facial conditions that have a percentage of 90%, namely the condition of the face facing right 45^o and facing up to 25^o, in this test, there is also a face condition that has a percentage of 80% is on the condition of the face facing upwards 45^o.

c) The test results are based on variations in faces at a distance of 1.5 meters

Table 3. Experiment with a Distance of 1.5 Meters

Position	Number of experiments	number succeeded
Condition 1	10	10
Condition 2	10	9
Condition 3	10	10
Condition 4	10	8
Condition 5	10	10
Condition 6	10	10
Condition 7	10	10
Condition 8	10	8
Condition 9	10	10
Condition 10	10	10
	Average	95

On the results of testing the distance of 1.5 meters from 10 experiments (number of samples) were carried out, obtained a percentage of 80% on the condition of the face facing right 45^o and facing down 45^o, and the percentage of 90% on the condition of the face facing left 45^o.

d) The test results are based on variations in faces at a distance of 2.0 meters

Table 4. Experiment with a Distance of 2.0 Meters

Position	Number of experiments	number succeeded
Condition 1	10	10
Condition 2	10	8
Condition 3	10	9
Condition 4	10	7
Condition 5	10	9
Condition 6	10	9
Condition 7	10	8

Position	Number of experiments	number succeeded
Condition 8	10	8
Condition 9	10	10
Condition 10	10	10
	Average	88

In the test results of a distance of 2 meters from 10 experiments (number of samples) were carried out, the lowest percentage was 70% on the condition of the face facing right 45°, the percentage of 80% on the condition of the face facing left 45°, facing down 25° and facing down 45° , and a percentage of 90% on the condition of the face facing right 25°, facing upwards 25°, and facing upwards 45°.

From the table representation of the experimental results above, the more the distance increases, the success decreases. At a distance of 0.5 meters identification reached 97% while at a distance of 2 meters 88% success.
From the above experiments obtained some analysis as follows:

a) The distance of the face from the camera greatly influences the success of the system in detecting and identifying faces. This is seen on the graph the farther the distance the accuracy decreases.
b) Face position affects the identification process, if the face position is different or not the same when taking photos used as a dataset then the accuracy is reduced it is indicated by different Euclidian values, the position of the face straight ahead has high accuracy because the position is used in the process of taking photos for the dataset.
c) The use of accessories on the face and facial skin color influence in the process of detection and identification, it is seen in the test of a distance of 2 meters some faces are not detected and not identified because they use glasses and have a slightly darker facial skin.

4. Conclusion

a) Application of the Tiny Yolo V2 algorithm as a face detection system that uses the Central Processing Unit (CPU) as its processing by obtaining detected face descriptions in the form of 128 vectors using ResNet-34 which is then measured the distance between face descriptions from the camera and face descriptions in the dataset using Euclidian distance for the face recognition process. In this study, a face is recognized if the distance between 2 faces has a small distance or less than equal to 0.45 or 45%.
b) Distance, position, use of accessories, and skin color on the face greatly affect the success rate in detecting and identifying faces. In this study, the percentage of success at a distance of 0.5 meters reaches 97, a distance of 1 meter 96%, a distance of 1.5 meters 95% and a distance of 2 meters with a percentage of 88% where 2 faces are not detected and identified at that distance due to wearing glasses and has a rather dark skin color.

References

[1] Priyambodo, Tri Kuntoro, 2005, Implementasi Web Service Untuk Pengembangan Layanan Pariwisata Terpadu, Jurnal Fakultas Hukum UII Vol. 10 No. 2 . pp. 105-118.
[2] APPJI. 2018. Penetrasi & Profil Perilaku Pengguna Internet Indonesia. Jakarta: Asosiasi Penyedia Jasa Internet Indonesia.
[3] Wahana Komputer, 2005, *Pemanfaatan Kamera Digital dan Pengolahan Imagenya*, Penerbit Andi: Yogyakarta.

[4] LeeCun Y., Yoshua B., & Geoffrey H., 2015, Deep Learning, Nature Vol. 521, pp. 436-444.
[5] Abhirawa, H., J. & Arifianto, A., 2017. Pengenalan Wajah Menggunakan Convolutional Neural Network. e-Proceeding of engineering, 4(3), pp. 4907-4916.
[6] Subkhi, M. Bahrul, dkk., 2018. Sistem Pengenalan Wajah Untuk Presensi Kuliah Dengan Metode Eigenface PCA (Principal Component Analysis) dan City Block. Simki-Techsain, Vol. 2, No. 3, pp. 1-10.
[7] Shafira,Tiara. 2018. Implementasi Convolutional Neural Network Klasifikasi Citra Tomat Menggunakan Keras. Skripsi. Yogyakarta: FMIPA Universitas Islam Indonesia.
[8] Endrianti, Fenti, dkk., 2018. Sistem Pencatatan Kehadiran Otomatis di Ruang Kelas Berbasis Pengenalan Wajah Menggunakan Convolutional Neural Network (CNN). Jurnal Teori dan Aplikasi Ilmu Komputer (JATIKOM), Vol. 1, No. 1, pp. 40-44.

Authors

Graduate from Universitas Majalengka in informatics. Currently works as an IT staff at Universitas Majalengka. Handling several IT development projects in Universitas Majalengka.

Graduate from Universitas Jenderal Achmad Yani-Cimahi in industrial engineering and Master Degree from Institut Teknologi Bandung in industrial engineering. Currently work as lecturer at Universitas Majalengka

Graduate from Universitas Pendidikan Indonesia-Bandung in computer science education and Master Degree from Universitas Gadjah Mada-Yogyakarta in computer science. Currently work as lecturer at Universitas Majalengka