Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3424~3432
e-ISSN: 2548-6861

Real-Time Heart Rate Pattern Analysis During Computer-Based Work Activities

Muhammad Fatiha Assyfa [1]*, Muhammad Fikry [2]*, Zara Yunizar [3]*
* Informatics Engineering, Department of Engineering, Malikussaleh University, Aceh, Indonesia
muhammad.220170129@mhs.unimal.ac.id [1], muh.fikry@unimal.ac.id [2], zarayunizar@unimal.ac.id [3]

Article Info

Article history:

Received 2026-05-29
Revised 2026-07-24
Accepted 2026-08-08

Keyword:

Internet of Things (IoT),
Stress Detection,
Pulse Sensor,
Kurtosis,
Real-Time Monitoring

ABSTRACT

The development of Internet of Things (IoT) technology provides opportunities for real-time health monitoring systems, including stress detection based on users' physiological conditions. This study aims to develop an IoT-based heart rate monitoring and stress detection system using a Pulse Sensor and ESP8266 microcontroller. The system is designed to read heart rate signals in real-time and transmit the data to a computer through serial USB communication for further processing using the Python programming language. The data processing stages include signal preprocessing, Beats Per Minute (BPM) calculation, sliding window processing, and kurtosis analysis as an indicator of user stress levels. The processed data are visualized through a Streamlit-based monitoring dashboard in the form of time-series graphs, gauge meters, and real-time user condition status. The study involved 20 Informatics Engineering students performing computer-based work activities within a certain duration. The results show that the system is capable of performing real-time heart rate monitoring and stress analysis effectively. The kurtosis values indicate changes in heart rate signal distribution patterns that can be used as indicators of normal and stress conditions. The developed system is expected to provide a simple, affordable, and extensible health monitoring solution.

I. INTRODUCTION

The development of Internet of Things (IoT) technology in recent years has brought major transformations across various sectors, particularly in the field of digital health [3]. IoT enables devices to be interconnected and exchange data in real time via the internet, thereby supporting more efficient, adaptive, and sustainable monitoring systems. This technology opens opportunities for continuous monitoring of an individual's physiological conditions without spatial or temporal limitations, making it suitable for developing early detection systems for various health conditions, including stress and work-related fatigue [10]. Along with the increasing dependence on digital technology, especially in work activities that involve prolonged computer use, individual stress levels have also increased significantly. This condition not only affects psychological aspects but also physical health, such as sleep disorders, reduced concentration, chronic fatigue, and an increased risk of cardiovascular diseases [5]. Stress is often not detected early due to the lack of physiological monitoring systems capable of providing real-time and continuous information to users [7].

Heart rate is one of the primary physiological parameters commonly used to identify stress. An approach based solely on beats per minute (BPM) is insufficient to fully describe autonomic nervous system activity. Therefore, Heart Rate Variability (HRV) is used as a more comprehensive indicator because it reflects the balance between the sympathetic and parasympathetic nervous systems [1]. One of the most widely used HRV parameters in research is the Root Mean Square of Successive Differences (RMSSD), which is highly sensitive to changes in parasympathetic activity and effective for detecting short-term stress, a finding consistently supported across recent wearable-based HRV studies [13]. Heart rate measurement can be performed using photoplethysmography, such as a Pulse Sensor, which works by detecting blood volume changes through variations

in light intensity on the surface of the skin [2]. This sensor can be integrated with a microcontroller such as the ESP8266, which is equipped with Wi-Fi connectivity, enabling real-time data transmission to a processing system. The collected data is then processed using the Python programming language to perform signal extraction, BPM calculation, and HRV (RMSSD) analysis as the basis for determining the user's physiological condition [11].

Several studies have developed heart rate monitoring systems based on the Internet of Things (IoT) using photoplethysmography (PPG) sensors and microcontrollers such as the ESP8266 for real-time heart rate measurement. However, most of these studies are still limited to analyzing heart rate or beats per minute (BPM) parameters without considering heart rate variability [6]. On the other hand, some research has implemented the ESP8266 for health monitoring, but it still focuses on data transmission without further physiological analysis such as RMSSD calculation [14]. Meanwhile, other studies have utilized HRV for stress detection through controlled laboratory protocols validated against standardized psychological instruments and stress-induction tests; however, these approaches are generally still conducted offline and have not been integrated into an IoT system capable of operating in real time and continuously [9].

Although previous studies have developed IoT-based heart rate monitoring systems, most of them still focus on BPM measurement without conducting in-depth HRV analysis for stress detection [4]. This indicates limitations in the depth of physiological analysis as well as suboptimal system integration. Such conditions highlight the need for developing a system that is simpler and more affordable, yet still capable of providing more accurate and comprehensive physiological analysis [8]. Recent work on HRV-based wearable prototypes has also shown that cross-validating sensor-derived stress predictions against psychological questionnaires or other standard validation instruments substantially strengthens the credibility of stress-detection claims [15]. Based on these challenges, this study proposes the development of an IoT-based stress detection system that integrates a Pulse Sensor and ESP8266 as real-time heart rate data acquisition devices. The obtained data is then processed using Python to calculate BPM and HRV (RMSSD) parameters as the main indicators for stress identificatioN [12]. The novelty of this research lies in the integration of a real-time IoT-based heart rate monitoring system with HRV (RMSSD) analysis for stress detection, which is expected to produce more accurate, continuous, and responsive physiological monitoring compared to previous approaches.

II. METHODOLOGY

The research method used in this study is the experimental research method, which is conducted by designing, building, and testing a system to evaluate its performance and results. In this study, a real-time heart rate monitoring and stress level detection system based on the Internet of Things (IoT) was designed using the ESP8266 microcontroller and a Pulse Sensor to acquire the user's heart rate data during computer-based work activities. The acquired data were processed to calculate the Beats Per Minute (BPM) value and the kurtosis-based Heart Rate Variability (HRV) as the primary indicator of the user's stress level. The research stages included literature study, hardware and software design, device assembly, data collection, data processing, system testing, and analysis. In accordance with the research scope, this study does not employ machine learning methods or perform clinical validation of stress detection results; rather, the system serves as a preliminary monitoring tool.

A. Research Scheme

The research scheme illustrates the structured stages carried out from the initial phase to the final phase of the study. It provides a systematic overview of how each phase is interconnected, starting from problem identification, system design, implementation, testing, and evaluation, leading to the final conclusion. The general research flow is presented in Figure 1.

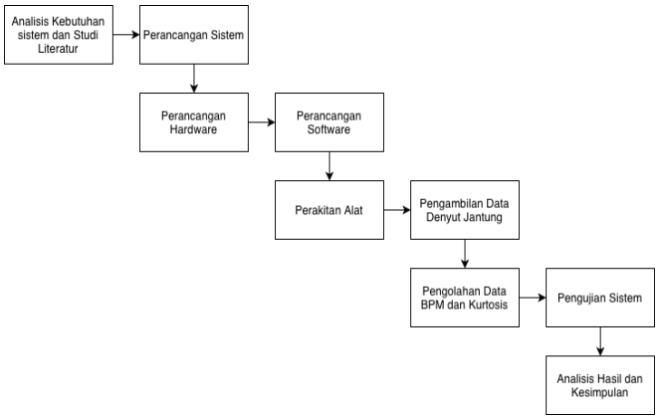

Figure 1. Research Scheme

System requirements analysis was conducted to understand the requirements in designing and developing an Internet of Things (IoT)-based heart rate monitoring and stress detection system. Based on the analysis results, the hardware used included a MacBook Air M1 with 8 GB RAM and 256 GB SSD, ESP8266, Pulse Sensor, jumper cables, USB cable, breadboard, and a USB to Type-C adapter. These components were selected to ensure reliable data acquisition, efficient processing, and stable communication between sensors and the processing device. Meanwhile, the software used consisted of macOS, Arduino IDE, Visual Studio Code, Python, as well as several supporting libraries such as Pandas, NumPy, Matplotlib, and Streamlit, which played roles in programming, data processing, visualization, and real-time system monitoring. The integration of these software tools enables smooth data flow from hardware acquisition to analytical processing and

interactive visualization, supporting the overall development of the IoT-based monitoring system.

B. System Scheme

The system scheme is a general illustration of the workflow of the heart rate monitoring and stress level detection system designed in this study. The system begins with reading heart rate signals using a pulse sensor that detects changes in blood flow and generates analog signals. These signals are then read by the ESP8266 and transmitted to the computer through USB serial communication. The received data are processed using the Python programming language in Visual Studio Code for analysis, data storage, and real-time monitoring visualization.

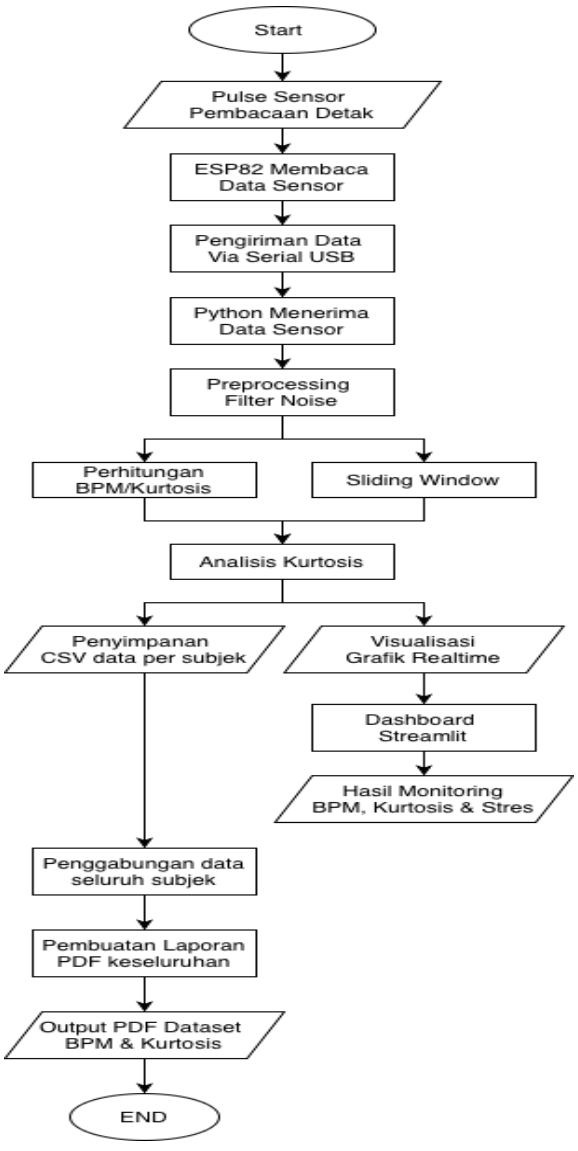

Figure 2. System Scheme

The data processing stage is carried out by continuously reading heart rate signals using the pulse sensor, then the obtained data undergo preprocessing to reduce noise and eliminate unstable data so that the data quality becomes better. Furthermore, the system calculates the Beats Per Minute (BPM) value and applies the sliding window method to group heart rate data within certain time intervals. The sliding window results are then analyzed using the kurtosis method to identify the user's stress level based on the characteristics of the obtained heart rate signal distribution. The processed data results are stored in CSV file format for each research subject and used as the basis for real-time monitoring graph visualization. All CSV files from each subject are then combined into a research dataset used in preparing the research report in PDF format containing BPM measurement results and kurtosis analysis from all subjects. In addition, the system provides a Streamlit-based monitoring dashboard that displays heart rate graphs, BPM values, kurtosis values, and real-time stress level detection results so that the user's condition can be monitored directly.

III. RESULT AND DISCUSSION

A. Research Results

The research results demonstrate the implementation and testing of an IoT-based heart rate monitoring and stress level detection system developed using the ESP8266 microcontroller and a Pulse Sensor. The system is capable of reading the user's heart rate in real-time and transmitting the acquired signal to a computer through USB serial communication at a baud rate of 115200 bps. The transmitted data are subsequently processed using the Python programming language to compute Beat Per Minute (BPM) values and perform kurtosis-based statistical analysis as an indicator of physiological stress level. The processed results are displayed in real-time on a Streamlit-based monitoring dashboard, presenting heart rate signal graphs, BPM values, kurtosis values, and stress classification status. In addition, each subject's data are automatically stored in CSV format, which are subsequently merged into a unified research dataset and utilized for generating research reports in PDF format. The physical prototype of the Pulse Sensor-based stress detection device was successfully assembled and verified to function properly, as shown in Figure 3.

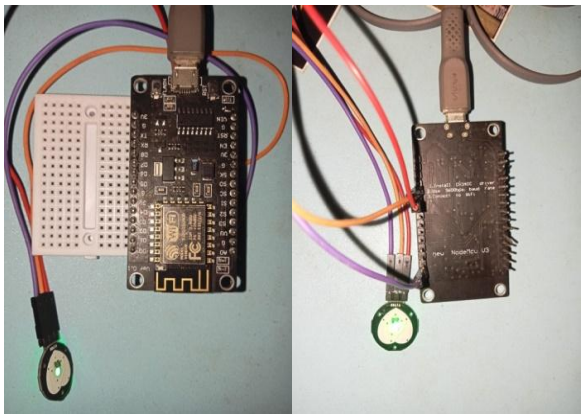

Figure 3. Front View (Left) and Back View (Right) of the Stress Sensor

Based on Figure 3, the Pulse Sensor is connected to the ESP8266 by linking the AO pin to the analog input pin for heart rate signal acquisition, the GND pin to the ground reference, and the 3V pin to the 3.3V power supply. The active green LED indicator on the sensor confirms that the sensor is operational and ready for use. In this study, the ESP8266 was programmed using the Arduino IDE to read heart rate signals via the analogRead() function and transmit the data to the computer through USB serial communication using Serial.println(). The data acquisition and transmission cycle runs continuously at 10-millisecond intervals, ensuring real-time heart rate data availability for downstream processing and analysis.

B. BPM and RMSSD Data Collection Results

Heart rate data collection was performed using a Pulse Sensor connected to the ESP8266 as the initial stage of the study. The ESP8266 continuously reads analog heart rate signals from the sensor and transmits them to the computer through USB serial communication at a baud rate of 115200 bps. Subjects were instructed to place their index finger on the sensor surface and minimize physical movement to maintain signal integrity throughout the measurement session. The raw signal received from the sensor contains considerable noise arising from motion artifacts and environmental interference. Therefore, a preprocessing stage was applied using a moving average filter with a window size of 5 samples to smooth the signal and reduce short-term fluctuations. This preprocessing step is critical to improving the reliability of subsequent feature extraction, as unfiltered noise can significantly distort BPM and RMSSD calculations. The sampling rate of the system was set at 100 Hz, meaning the sensor acquires 100 data points per second, which provides sufficient temporal resolution for detecting individual heartbeat events and computing inter-beat intervals accurately.

Following preprocessing, the system computes three primary features: BPM (Beats Per Minute), IBI (Inter-Beat Interval), and RMSSD (Root Mean Square of Successive Differences). BPM quantifies the average heart rate intensity, IBI represents the time interval between consecutive heartbeats in milliseconds, and RMSSD serves as a time-domain measure of short-term heart rate variability (HRV) that reflects parasympathetic modulation of the autonomic nervous system. Physiologically, a reduction in RMSSD is commonly associated with increased sympathetic nervous system activity, which corresponds to heightened stress responses [1]. All computations are executed automatically by the Python program, and the results including timestamp, raw signal value, BPM, IBI, and RMSSD are stored in individual CSV files for each subject to facilitate subsequent analysis. Table I presents a sample of raw data obtained from Subject 9 (Faris) during the monitoring session.

TABLE I
RAW DATA OF SUBJECT 9 (FARIS)

timestamp	signal	bpm	ibi	rmssd
2026-05-20 21:25:11	555	0	0	0
2026-05-20 21:25:12	471	0	0	0
2026-05-20 21:25:13	18	76	788	0
2026-05-20 21:25:14	578	77	777	0
2026-05-20 21:25:15	596	88	676	0
2026-05-20 21:25:16	545	91	655	10
2026-05-20 21:25:17	610	111	535	23
2026-05-20 21:25:18	529	111	535	23
2026-05-20 21:25:19	543	111	535	23
2026-05-20 21:25:20	585	111	535	23

As shown in Table I, the initial rows contain zero values for BPM, IBI, and RMSSD, which represent the system stabilization phase during which the sensor has not yet detected a sufficient number of valid heartbeat events to perform computation. Once the system detects consecutive heartbeats, BPM and IBI values begin to populate, followed by RMSSD values after at least two successive IBI measurements are available. The signal column reflects natural cardiovascular variability, with fluctuations corresponding to the systolic and diastolic phases of the cardiac cycle. Table II summarizes the data collection duration and total number of data points recorded for each of the 20 research subjects.

TABEL II
RAW DATA DETAILS

No	Subject	Data Collection Duration	Number of Data
1	ale	618 Seconds	614
2	amel	660 Seconds	656
3	arfa	625 Seconds	621
4	bagus	632 Seconds	628
5	bayu	584 Seconds	580
6	briliansyah	641 Seconds	637
7	dimas	626 Seconds	622
8	dini	693 Seconds	689
9	faris	622 Seconds	618
10	fatur	638 Seconds	633
11	hafiz	602 Seconds	598
12	ivandanizar	647 Seconds	643
13	nazrivan	619 Seconds	615
14	raja	705 Seconds	701
15	rapli	541 Seconds	537
16	rifky	630 Seconds	625
17	sultan	651 Seconds	647
18	teuku	635 Seconds	631
19	tio	611 Seconds	607
20	yunda	652 Seconds	648

As presented in Table II, data collection duration ranged from 541 to 705 seconds across subjects, with an average duration of approximately 632 seconds. The slight variation

in duration is attributed to differences in session management for each subject. The number of recorded data points is marginally lower than the duration in seconds due to occasional packet loss during USB serial transmission, which was handled by the Python program through data validation checks. Overall, the dataset totals 12,719 data records across all 20 subjects, providing a sufficient volume of physiological data for subsequent kurtosis-based stress analysis.

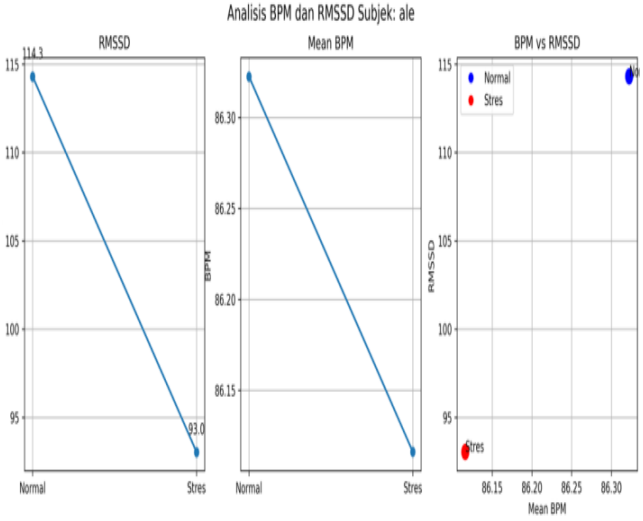

Figure 4. Raw Data of BPM and RMSSD for Subject 1 (ale)

The processing results were visualized in the form of time-series graphs displaying changes in BPM and RMSSD values throughout the monitoring session, as shown in Figure 4. The visualization of Subject 1 (Ale) in Figure 4 shows that the RMSSD value under normal conditions reached approximately 114 ms, while under stress conditions it decreased to approximately 93 ms. This reduction indicates that heart rate variability diminished during stress, consistent with established HRV literature that associates lower RMSSD with reduced parasympathetic activity and heightened sympathetic dominance [2]. Conversely, BPM values did not exhibit a substantial difference between normal and stress conditions, suggesting that average heart rate alone is insufficient as a standalone stress indicator. This finding supports the rationale for incorporating RMSSD and subsequently kurtosis as complementary features for stress detection.

C. Development of BPM and Kurtosis Methods

Building upon the initial BPM and RMSSD analysis, the research methodology was further developed by incorporating kurtosis as the primary statistical feature for stress level classification. Kurtosis is a fourth-order statistical moment that quantifies the degree of peakedness or tailedness of a probability distribution relative to a normal distribution. In the context of heart rate signal analysis, physiological perturbations such as stress-induced autonomic nervous system changes can alter the distributional characteristics of the signal, resulting in measurable changes in kurtosis values. Specifically, stress conditions tend to produce more irregular and impulsive signal patterns, which manifest as elevated kurtosis values exceeding those observed under normal resting states.

Prior to kurtosis computation, the heart rate signal underwent a two-stage preprocessing pipeline. First, a moving average filter with a window size of 5 samples was applied to reduce high-frequency noise. Second, invalid data points including zero values resulting from sensor initialization or momentary signal loss were removed to prevent distortion of statistical calculations. This filtering stage is essential because the presence of outlier values or invalid readings can disproportionately inflate kurtosis estimates and produce false stress classifications.

Following preprocessing, the cleaned signal was segmented using a sliding window approach implemented via Python's deque(maxlen=100) structure, corresponding to a window size of 100 samples. Given the system's sampling rate of 100 Hz, this window represents approximately one second of continuous heart rate data. The window size of 100 samples was selected based on a trade-off between temporal resolution and statistical stability: smaller windows may produce unstable kurtosis estimates due to insufficient sample size, while larger windows reduce sensitivity to rapid physiological changes. As new data points arrive, the oldest sample is automatically discarded, ensuring that kurtosis computation always reflects the most recent physiological state of the subject.

A kurtosis value of zero corresponds to a normal (mesokurtic) distribution, positive values indicate a leptokurtic distribution with a sharper peak and heavier tails, and negative values indicate a platykurtic distribution with a flatter peak and lighter tails. In cases where the computation returns a NaN value due to zero variance within the window, the value is converted to zero to maintain continuity of the analysis pipeline. The stress classification threshold was determined at $\kappa \geq 10$, established empirically based on observational analysis of kurtosis distribution patterns obtained throughout the data collection process. Based on observations of the kurtosis value distribution across all subjects, values exceeding 10 consistently appeared during periods exhibiting increased signal variation and peakedness, while values below 10 corresponded to relatively stable and regular heart rate signal patterns. The threshold of $\kappa \geq 10$ was therefore selected as the classification boundary between normal and stress conditions. Conditions with $\kappa < 10$ are classified as normal, while $\kappa \geq 10$ are classified as stress. This approach is consistent with prior studies employing higher-order statistical moments for physiological signal classification, where elevated kurtosis has been associated with increased signal irregularity under autonomic perturbation.

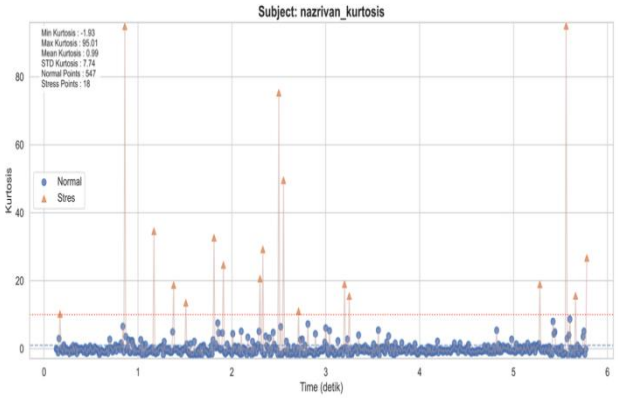

Figure 5. Kurtosis Time-Series Visualization of Subject 13 (Nazrivan)

The kurtosis time-series visualization for Subject 13 (Nazrivan) is presented in Figure 5. The horizontal axis represents time (timestamp) during the monitoring session, while the vertical axis represents the computed kurtosis value at each sliding window interval. The dashed horizontal line at $\kappa = 10$ indicates the stress classification threshold. Data points plotted below the threshold line are categorized as normal conditions, while data points exceeding the threshold line are categorized as stress conditions, distinguished by different markers on the graph. As shown in Figure 5, kurtosis values predominantly remain below the threshold throughout the monitoring session, indicating a baseline normal physiological state for this subject. However, several intermittent spikes exceeding $\kappa = 10$ are observed at specific timestamps, corresponding to transient stress events during the monitoring period. The intermittent rather than sustained nature of these spikes is physiologically plausible, as acute stress responses in healthy individuals are typically short-duration autonomic events rather than prolonged states. The sliding window mechanism ensures that these transient fluctuations are captured in real-time without delay, enabling immediate classification and dashboard update.

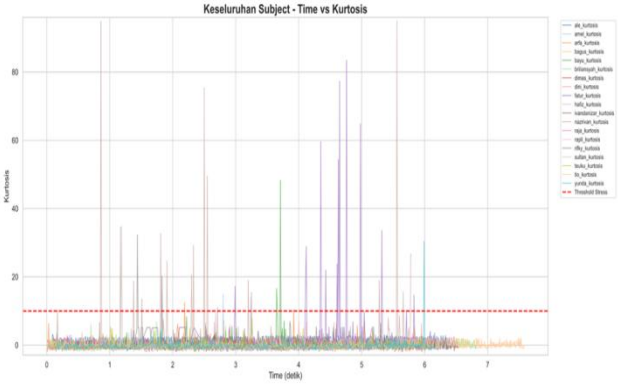

Figure 6. Combined Kurtosis Graph of All Subjects

The aggregated kurtosis results across all 20 subjects are presented in Figure 6. The horizontal axis represents the subject index, while the vertical axis represents the range of kurtosis values recorded for each subject throughout the monitoring session. The threshold line at $\kappa = 10$ is displayed across the graph to facilitate visual comparison of stress classification boundaries across subjects. As shown in Figure 6, individual subjects exhibit distinct kurtosis behaviors, reflecting inter-individual variability in physiological stress responses and baseline heart rate signal characteristics. Some subjects display relatively stable kurtosis values that remain consistently below the threshold, indicating predominantly normal physiological conditions during monitoring. Other subjects show more frequent spikes exceeding the threshold, suggesting greater signal variability and more transient stress events throughout the session. These inter-subject differences are consistent with the understanding that physiological stress responses vary considerably across individuals depending on baseline autonomic tone, cognitive workload sensitivity, and personal stress coping capacity. Compared to conventional HRV-based stress detection methods such as RMSSD and SDNN, the kurtosis approach offers a complementary perspective by capturing the shape of the signal distribution rather than solely its temporal variability. While RMSSD reflects parasympathetic modulation through successive beat differences, kurtosis captures the impulsive nature of signal irregularities that may not be fully represented by variance-based measures.

D. Real-Time Monitoring Dashboard

The real-time monitoring dashboard was developed using Streamlit as the main interface to display the results of heart rate monitoring and stress level detection directly. The dashboard receives pulse sensor reading data from the ESP8266 through USB serial communication, then processes it using Python to generate BPM values, kurtosis values, and user condition classifications. This information is displayed in the form of real-time BPM and kurtosis graphs, monitoring gauge meters, and user condition status that are automatically updated according to the latest sensor data. The implementation results show that the dashboard is capable of monitoring changes in heart rate and stress levels interactively, informatively, and in real-time during the monitoring process.

Figure 7. Real-Time Monitoring Dashboard Display

The monitoring dashboard displays the results of heart rate readings and stress level analysis in real-time in the

form of sensor signal values, BPM (Beats Per Minute), kurtosis, user condition status, gauge meters, as well as BPM and kurtosis time-series graphs over time. The heart rate data read by the pulse sensor is transmitted through the ESP8266 and processed using Python to generate BPM values, then analyzed using the kurtosis method to detect normal or stressed conditions. The BPM and kurtosis graphs are automatically updated on the Streamlit dashboard, equipped with a stress threshold line to facilitate the identification of user conditions. The implementation results show that the dashboard is capable of visualizing changes in heart rate and stress levels in real-time properly, making the monitoring process easier and more informative.

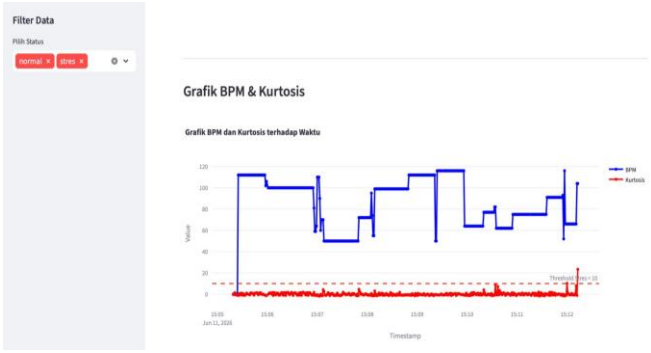

Figure 8. Real-Time BPM and Kurtosis Graphs

The real-time monitoring dashboard displays BPM and kurtosis graphs to monitor changes in heart rate and user stress levels during the monitoring process. The BPM graph shows changes in heart rate values, while the kurtosis graph is used to analyze the distribution pattern of heart rate signals equipped with stress threshold line as classification boundary between normal and stressed conditions. In addition to graphs, the dashboard also provides BPM and kurtosis gauge meter features that are automatically updated based on pulse sensor data processed through Python and ESP8266, making it easier for users to understand heart rate conditions and stress levels quickly and visually.

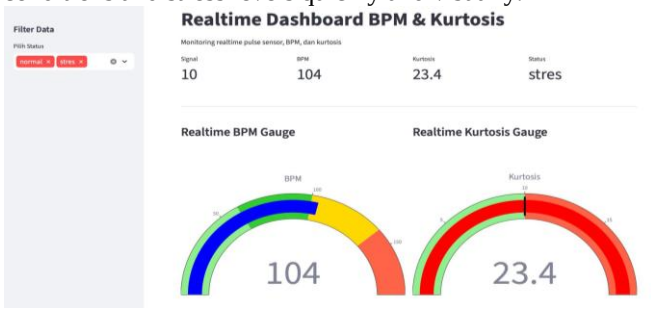

Figure 9. BPM and Kurtosis Gauge Meter

The gauge meter display is used to visually present BPM and kurtosis values during the monitoring process, allowing users to observe heart rate conditions and changes in stress levels in real-time. The dashboard also displays heart rate signal graphs in the form of time-series that are automatically updated based on pulse sensor data transmitted through ESP8266 and processed using Python. In addition, a real-time monitoring table is provided containing information on reading time, signal values, BPM, kurtosis, and user condition status. The combination of gauge meters, graphs, and tables makes the process of monitoring heart rate and stress level detection easier, more informative, and interactive during the research process.

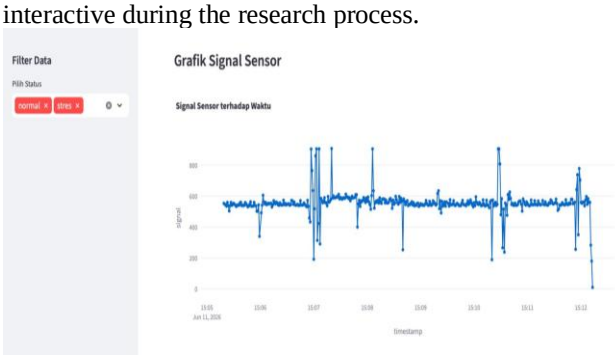

Figure 10. Sensor Signal Graph

The real-time sensor signal graph displays changes in heart rate signal values over time dynamically based on sensor reading results, while the dashboard also presents a real-time data table containing reading time, BPM values, kurtosis values, and user condition status. The implementation results show that the system successfully performs heart rate monitoring and stress level detection in real-time properly, capable of automatically updating data, displaying changes in BPM, kurtosis, as well as the results of normal and stressed condition classifications directly, thereby making the user condition monitoring process easier, interactive, informative, and running stably during the research process.

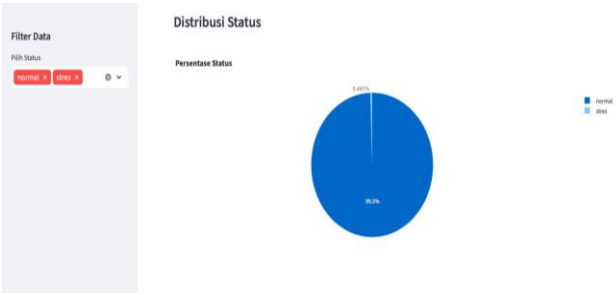

Figure 11. Sensor Signal Graph

The real-time monitoring results show that the dashboard is capable of displaying sensor data directly during the monitoring process. The dashboard displays monitoring graphs, BPM values, kurtosis values, user condition status, as well as data tables that are automatically updated. In addition, the dashboard also displays the distribution of normal and stressed statuses based on the classification results. This display helps the process of heart rate monitoring and stress analysis become easier to understand in real-time.

E. System Testing

System testing was conducted to ensure that the heart rate monitoring and stress level detection system could

operate according to the design. The testing included the pulse sensor, ESP8266, BPM calculation, kurtosis analysis, and Streamlit-based monitoring dashboard by observing the sensor reading process, data transmission, BPM and kurtosis processing, as well as real-time monitoring displays. The test results showed that the pulse sensor was able to read heart rate signals dynamically for all research subjects, the data could be transmitted and processed properly, and the BPM results, kurtosis values, and user condition status could be displayed interactively on the dashboard so that the system was able to support the real-time monitoring and stress detection process.

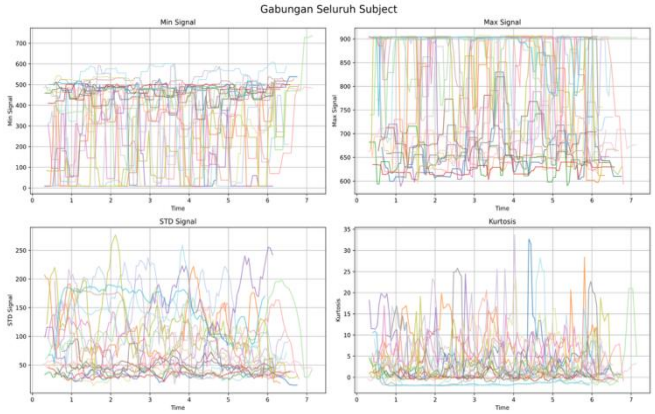

Figure 12. Pulse Sensor Testing Results for All Subjects

The test results showed that the pulse sensor was able to produce sufficiently stable heart rate data, indicated by changes in the min signal, max signal, STD signal, and kurtosis values that could be visualized during the monitoring process. Heart rate data were read by the ESP8266 and transmitted continuously to the computer through USB serial communication without interruption, so that the data could be received and processed properly for BPM calculation and stress level analysis using the kurtosis method. The test results also proved that the process of transmitting, processing, and visualizing real-time data on the monitoring dashboard operated stably and according to the system design.

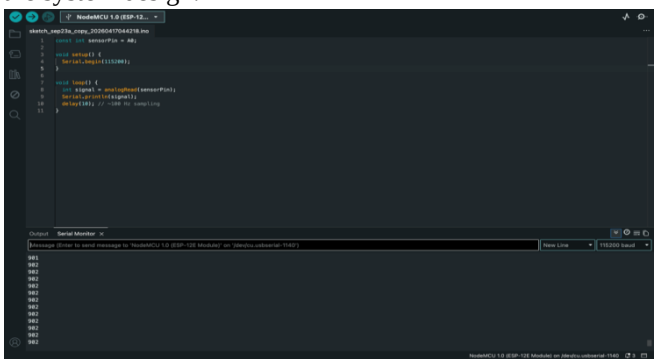

Figure 13. ESP8266 Testing

The test results showed that the ESP8266 was able to receive data from the pulse sensor and transmit it to the system in real-time through USB serial communication. The received heart rate data were then processed using Python to calculate BPM values based on the time interval between heartbeats. The system successfully displayed BPM values continuously and followed changes in user conditions during the monitoring process, both in numerical form, time-series graphs, and gauge meters on the dashboard. These results indicate that the processes of data acquisition, transmission, and BPM calculation operated properly so that they could be used for heart rate monitoring and real-time stress analysis.

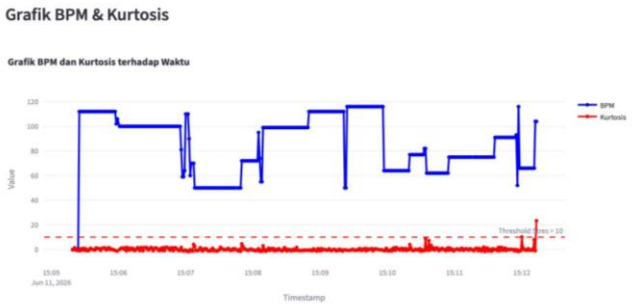

Figure 14. Real-Time BPM Testing

The test results showed that the system was able to calculate and display BPM values and kurtosis values in real-time during the monitoring process. BPM values changed according to the user's heart rate condition, while kurtosis values were used to analyze the distribution pattern of heart rate signals as an indicator of stress level based on predetermined threshold values. All data obtained from the pulse sensor were processed using Python and displayed directly through the monitoring dashboard, so that user conditions could be monitored continuously. The implementation results showed that the system was able to perform heart rate monitoring and stress level analysis effectively using the kurtosis method.

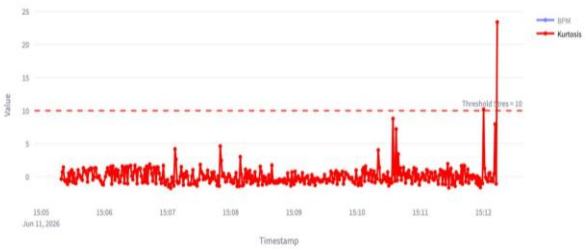

Figure 15. Real-Time Kurtosis Testing

The test results showed that the monitoring system was able to calculate and display BPM and kurtosis values in real-time during the monitoring process. The dashboard built using Streamlit was able to automatically update graphs, gauge meters, data tables, and user condition status based on heart rate data received from the pulse sensor through the ESP8266 and processed using Python. Kurtosis values that changed according to the distribution pattern of heart rate signals could be used as indicators of normal and stress conditions, so that the system was able to present monitoring

information directly, interactively, and informatively throughout the research process.

IV. CONCLUSION

Based on the results of the design, implementation, and testing processes, the Real-Time Heart Rate Pattern Analysis System during Computer-Based Work Activities was successfully developed and implemented. The Internet of Things (IoT)-based heart rate monitoring system using ESP8266 and a Pulse Sensor was able to perform real-time heart rate acquisition effectively. The ESP8266 functioned successfully as the main microcontroller to receive sensor data and continuously transmit the data to a computer through stable serial USB communication during the monitoring process. Furthermore, the system successfully processed heart rate data using Python to calculate BPM values and perform kurtosis analysis as an indicator of the user's stress level. The preprocessing and sliding window methods were also successfully applied to improve signal stability and support real-time kurtosis analysis.

In addition, the kurtosis analysis method successfully identified changes in heart rate signal distribution patterns and assisted in classifying normal and stressed conditions based on the obtained kurtosis values. The system was also capable of storing monitoring results in CSV format and combining all subject data into the main research dataset for visualization and statistical analysis purposes. The Streamlit-based dashboard successfully displayed real-time monitoring results interactively through BPM graphs, kurtosis graphs, gauge meters, sensor signal graphs, monitoring tables, and user condition status indicators. Moreover, the system automatically generated visualization and statistical analysis reports in PDF format, making the documentation and evaluation processes more efficient. Based on the testing results, all system components, including the Pulse Sensor, ESP8266, BPM calculation process, kurtosis analysis, and monitoring dashboard, performed properly and stably throughout the research process. Therefore, the developed system successfully achieved the research objectives and demonstrated strong potential as a real-time sensor-based health monitoring system.

REFERENCES

[1] D. Salsabila, A. T. Hanuranto, and A. I. Irawan, "Sistem monitoring denyut jantung berbasis IoT menggunakan protokol XMPP," vol. 2, no. 2, pp. 171–178, 2022, doi: 10.35313/jitel.v2.i2.2022.171-178.

[2] S. Edo, Sindy, Johan, Indi, "JITE (Journal of Informatics and Telecommunication Engineering)," vol. 9, no. 1, pp. 129–138, 2025, [Online]. Available: https://ojs.uma.ac.id/index.php/jite/article/view/15062?utm_source=chatgpt.com&__cf_chl_tk=wZXyLcqbZwiKHLjWhvAUrRhNqY_VCB1FCbbXo6mC6JI-1780081584-1.0.1.1-rp6zRGUqUgwH9hNZIxyTZYJ0yyQwkGxD9OlT1i8hc84

[3] S. Arafat, A. M. H. Pardede, and H. Khair, "Design and Development of a Realtime Human Heart Rate Measurement System Using IOT-Based Pulse Sensors," vol. 2, no. 3, pp. 83–88, 2023, doi: 10.59934/jaiea.v2i3.199.

[4] Y. Segono, A. Ridwan, E. Yose, M. Hutagalung, and H. G. Simbolon, "IoT-Based Stress Monitoring Using CNN for HRV-GSR Analysis," vol. 10, no. 1, pp. 621–637, 2026, doi: 10.33395/sinkron.v10i1.15671.

[5] D. Muhajir, F. Mahananto, N. A. Sani, D. Muhajir, F. Mahananto, and N. A. Sani, "Stress level measurements using heart rate variability analysis on android based application," *Procedia Comput. Sci.*, vol. 197, no. 2021, pp. 189–197, 2022, doi: 10.1016/j.procs.2021.12.200.

[6] S. M. Gondowijoyo, R. Setiawan, and N. F. Hikmah, "Applying artificial neural network on heart rate variability and electroencephalogram signals to determine stress," vol. 22, no. 4, pp. 910–920, 2024, doi: 10.12928/TELKOMNIKA.v22i4.25832.

[7] A. Hendryani, D. Gunawan, M. Rizkinia, R. N. Hidayati, and F. Y. Hermawan, "Real-time stress detection and monitoring system using IoT-based physiological signals," vol. 12, no. 5, pp. 2807–2815, 2023, doi: 10.11591/eei.v12i5.5132.

[8] B. Szakonyi, I. Vassányi, E. Schumacher, and I. Kósa, "Efficient methods for acute stress detection using heart rate variability data from Ambient Assisted Living sensors," *Biomed. Eng. Online*, vol. 20, no. 73, pp. 1–19, 2021, doi: 10.1186/s12938-021-00911-6.

[9] T. Sutikno, M. H. Ar-rasyid, T. Wahono, and W. Arsadiando, "Internet of things with NodeMCU ESP8266 for MPX-5700AP sensor-based LPG pressure monitoring," vol. 12, no. 3, pp. 257–264, 2023, doi: 10.11591/ijaas.v12.i3.pp257-264.

[10] T. S. Sunil, S. A. Santosh, T. A. Tanaji, and S. G. Sanjay, "Design and Implementation of an IoT-Based Smart Energy Meter for Real-Time Monitoring and Automated Billing," vol. 5, no. 4, pp. 169–184, 2025, doi: 10.48175/IJARSCT-29411.

[11] A. Arada, Raed M H Halim, "Development of IoT System for Weight, Height, Temperature, and Heart Rate Monitoring with Health Recommendations," vol. 10, no. 2, pp. 1–13, 2025, doi: 10.56873/jpkm.v10i2.6014.

[12] A. Ahad, M. Tahir, M. A. Sheikh, K. I. Ahmed, A. Mughees, and A. Numani, "Technologies Trend towards 5G Network for Smart Health-Care Using IoT : A Review," vol. 20, no. 14, pp. 1–22, 2020, doi: 10.3390/s20144047.

[13] K. M. D. and G. L. Masala, "HRV Features as Viable Physiological Markers for Stress Detection Using Wearable Devices," vol. 21, no. 8, pp. 1–18, 2021, doi: 10.3390/s21082873.

[14] C. Widowati and K. Purwantini, "Inovasi Sensor Wearable untuk Monitoring Kesehatan Mental melalui Variabilitas Denyut Jantung," vol. 2, no. 2, pp. 63–73, 2024, doi: 10.59031/jnts.v2i2.787.

[15] B. Szakonyi, I. Vassányi, E. Schumacher, and I. Kósa, "Efficient methods for acute stress detection using heart rate variability data from Ambient Assisted Living sensors," *Biomed. Eng. Online*, vol. 20, no. 73, pp. 1–19, 2021, doi: 10.1186/s12938-021-00911-6.