JPPIPA 11(1) (2025)

Jurnal Penelitian Pendidikan IPA

Journal of Research in Science Education

http://jppipa.unram.ac.id/index.php/jppipa/index

Design and Build IoT Smart Home System with Blocks Based App Inventor Programming

Agus Suharto[1*]

[1] Information Systems Study Program, Faculty of Computer Science, Universitas Pamulang, Tangerang Selatan, Indonesia

Received: April 12, 2025
Revised: May 7, 2025
Accepted: July 25, 2025
Published: July 31, 2025

Corresponding Author:
Agus Suharto
dosen01539@unpam.ac.id

DOI: 10.29303/jppipa.v11i7.12028

Abstract: IoT adoption and integration has increased over the past few years due to the increasing consumer demand for convenience, security, and energy savings, creating opportunities for us to interact with our surroundings by integrating physical devices through IoT. Fundamentally, the Internet Society cares about IoT because it is an evolving aspect of how people and institutions are likely to interact with the Internet in their personal, social, and economic lives. MIT App Inventor is an intuitive visual programming environment that allows anyone to build applications including IoT for Android/iOS phones or tablets. Although a number of block-based approaches are available to program some IoT hardware, it is still an area that has not been widely explored. In this study, we will discuss how to implement IoT Smart Home with a block-based programming approach using MIT App Inventor. The application development method uses the ADDIE method with stages, namely Analysis, Design, Development, Implementation, Evaluation. The results of the research evaluation using the EUCS method with dimensions of Content, Accuracy Ease of Use, Format, Timeliness, the results obtained in the study average indexes of 79.80%, the hope for future research is how blocks based can inspire people to be creative with IoT so that beginners can build mobile applications integrated with IoT technology.

Keywords: Blocks based programming; IoT (Internet of Things); Smart home**:** System design

Introduction

MIT App Inventor is an easy-to-use drag-and-drop visual programming tool for designing and building mobile applications for both Android and iOS. This tool can turn one's ideas into functional applications without the need for coding or programming skills (Żyła et al., 2024). MIT App Inventor has a graphical user interface, built-in component blocks, and logical blocks. Users need to arrange several blocks to produce some behavioral actions. One can develop mobile applications with little knowledge of programming and coding (Kaddipujar et al., 2022). There are many tools available that can help develop mobile applications. MIT App Inventor is an intuitive puzzle-like block-based programming so that even novice programmers can build functional applications for smartphones and tablets. MIT App Inventor can also design and create IoT applications that interact with physical devices such as MCUs and other sensors. With over one million unique visitors each month from 195 countries who have collectively created nearly 30 million apps (Aïmeur et al., 2023; Kumar et al., 2021). MIT App Inventor is changing the way the world builds apps and the way kids learn about computing, Technology App Inventor provides easy-to-use tools for students to develop Android mobile apps. MIT App Inventor is an alternative for developing Android mobile apps (Pancane & Zahir, 2024; Sivarajah et al., 2017).

MIT App Inventor is ideal for developing IoT apps with its visual interface and blocks-based programming (Priawardana & Surriani, 2025). The easy and simple programming interface is able to create mobile apps quickly (Jayatilleke et al., 2019). The Development Interface in App Inventor is divided into two parts, namely Design and the blocks editor. Design shows the

How to Cite:
Suharto, A. (2025). Design and Build IoT Smart Home System with Blocks Based App Inventor Programming. *Jurnal Penelitian Pendidikan IPA*, *11*(7), 1016–1026. https://doi.org/10.29303/jppipa.v11i7.12028

user the layout of the application, the components contained in it, and the options available to customize the properties of the components. The block editor provides a drag-and-drop interface to connect blocks like a puzzle that represents actions to create a program. Unlike traditional Android application development, which usually requires programming such as Java, flutter. App Inventor encapsulates the Android development model and hides the implementation details through graphical blocks, so that the Java programming code is not visible to the user (Gharaat et al., 2021). Internet of Things (IoT), is a concept that describes the connection of smart devices to each other via the Internet (Choudhary, 2024; Lombardi et al., 2021). The Internet of Things is a significant first step towards a connected society.

The Internet of Things offers great social and economic opportunities, while at the same time also presents significant challenges, especially in terms of providing and managing the technical infrastructure that supports it and the flood of data generated from it, ensuring privacy and security, and capturing value from it (Quach et al., 2022). IoT does not only cover high-tech devices such as smartphones or computers, but also extends to common household appliances such as light bulbs, air conditioners, refrigerators, and even home security systems (Nižetić et al., 2020; Vardakis et al., 2024). Fundamentally, the Internet Society cares about IoT because IoT is an ever-evolving aspect of how people and institutions tend to interact with the Internet in their personal, social, and economic lives. IoT also has different consequences in different economies and regions, presenting a diverse set of opportunities and challenges around the world. Smart Home is one of the prominent IoT applications. Smart homes use IoT technology to allow users to remotely control, monitor, and automate home devices via smartphones, tablets, or virtual assistants such as Alexa, Google Assistant (Stolojescu-Crisan et al., 2021; Strzelecki et al., 2024).

In this study, how IoT applications and MIT App Inventor monitor humidity, temperature, motion sensors, relays, and proximity sensors, with a block programming-based approach. Preparations made in the implementation of this IoT in addition to MIT App Inventor, other software, namely Arduino Ide as writing code or sketches to microcontrollers and sensor tools (Jihadi et al., 2024), then firebase as a real-time database that regulates the status of receiving and sending data, Among the available real-time databases, several have become dominant. Firebase is Google's web and mobile application development platform, which provides a real-time database as its service. Several JavaScript frameworks such as AngularJS (MIT) and Backbone.js (MIT) facilitate database access for the Firebase real-time database. This database can also be accessed via the REST API: Representational State Transfer Application Program Interface which uses HTTP requests to Get, Put, Post and Delete data (Za'bah & Mohamad, 2021), while the hardware preparation such as MCU Wemos Esp 8266 (microcontroller), ultrasonic sensor, temperature sensor, distance sensor and relay. At the application interface design stage with App Inventor, the creation is quite easy, namely by dragging components to the design screen, then the blocks stage to give commands to the components that have been designed (Baena-Perez et al., 2024; Cen et al., 2023).

The blocks programming environment, where programs are created by connecting fragments that are formed like puzzle pieces. Blocks are a way for beginners to get to know programming and hobbyists, scientists or programmers in writing programs. Examples of programming include App Inventor, Scratch, Blockly, Snap! Pencil Code, and Alice. Block-based programming has become a popular way to introduce programming and computational thinking to tens of millions of people of all ages and backgrounds.

Method

The method used in this study uses the ADDIE model, one of the parts of the Research and Development (R&D) Method. The ADDIE model is a widely used framework for designing and developing learning materials, especially in instructional and educational design. This model is widely used to develop e-learning applications, allowing educators and instructional designers to create structured and effective learning experiences, and also allowing for continuous revision during the educational material design process (Norouzkhani et al., 2025). This model is known for its flexibility and adaptability, allowing instructional designers to adjust each phase to suit the specific needs of learners and learning contexts. However, with the rapid evolution of technology, these phases require a harmonious blend of digital tools and methodologies. This model can also be effectively applied in building website applications and other applications such as IoT, because the stages provide structured guidance. The acronym ADDIE comes from five stages, namely Analysis, Design, Development, Implementation, Evaluation. The ADDIE model can be adjusted with several flow modifications according to research needs.

The following are the stages in research using the ADDIE method: First analysis: This stage is the initial stage carried out by researchers. From the problems that have been identified and the literature studies conducted, researchers are interested in developing an innovation that can be carried out on Smart Homes by utilizing the Internet of Things (IoT) control system as a remote control. This stage is also a description of the research needs analysis of the problems that need to be carried out, which can be described in Figure 1 above and determining specific needs for IoT applications, collecting requirements. Software used, MCU hardware and what sensors need to be integrated. Second design: The Design stage uses the results of the analysis stage to plan a product design development strategy. At this Design stage, it describes how the design can achieve the goals that have been determined based on the results of the analysis. The things that are emphasized include user targets, In addition, system instructions. Planning the architecture and features of the IoT application. including determining the user interface, data flow, connectivity protocols. Third development: Model Development Stage, in this model development, a Mobile-Based Model is prepared that is ready to be validated and revised based on validator input, then a limited trial is carried out on the product being developed.

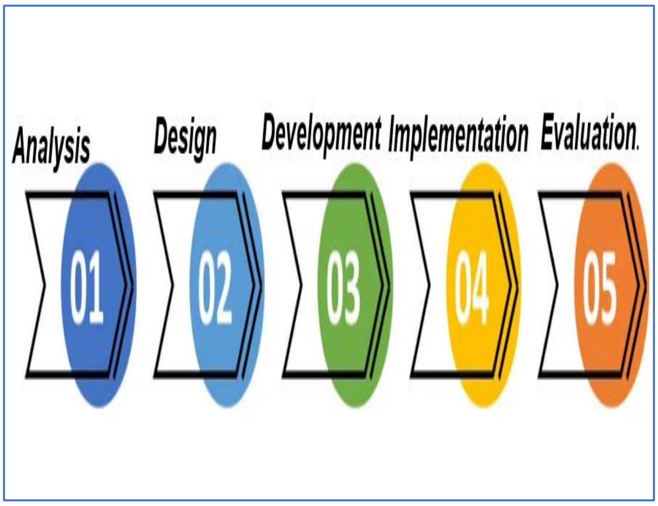


Figure 1. ADDIE Method Stages

Building IoT applications, which can involve programming, device integration testing, and ensuring connectivity. during this stage a Prototype is created and can be used to test functionality. Fourth implementation: During this phase, the Researcher translates the design specifications into actual code. The researcher writes the scatch code on the Arduino IDE Idea. Fifth evaluation: Assess the performance, usability, and scalability of the application. Collect feedback from users and monitor data analysis to make improvements. In this study, the evaluation used the EUCS (End User Computing Satisfaction) Model, the research variables used in the EUCS model instrument are: Content, Accurate, Format, Ease of Use, and Timeliness. Respondent questionnaire data is adjusted to the questions in the EUCS model. The results of these answers are used as guidelines for researchers. Table 3 explains the description of the indicator questions for each variable in the EUCS model. Measurement Interval using the Likert Scale the Likert Scale is a sequence of ordered ranking categories, where only nonparametric tools can be used. Concluding that the scale has the same distance between the categories allows the use of more powerful and precise parametric tools on possible mathematical operations.

Results and Discussion

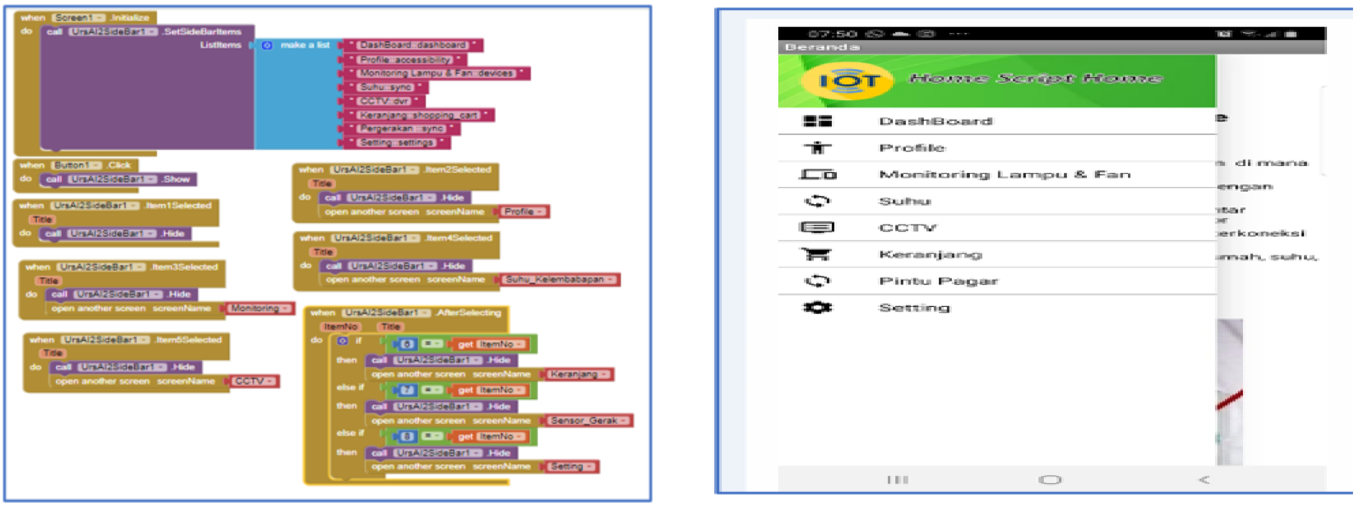
Result

The result of this study is an IoT Smart Home application in APK format that can be installed on Android-based mobile devices. This application consists of seven screens or main pages, each of which represents the main features of the designed smart home system. Details of the seven displays are presented in Table 1 below:

Table 1. List of IOT Smart Home application screens

Application Display Text	Screen Name	Icon
Dashboard	Home	Home
Profile	Profile	Accebility
Light and Fan Monitoring	Monitoring	Device
Temperature	Temperature Humidity	Sync
Trash Bin	Basket	Shopping_Cart
Motion Sensor	Movement	Sync
Setting	Setting	Settings

To organize navigation between screens, researchers used the UrsAI2SideBar extension, an extension to MIT App Inventor that allows developers to integrate sidebar menus into applications. This extension is very useful in developing applications with many features because it is able to organize navigation efficiently and supports the use of icons taken from Google Font Material and Google Icon (Abiri et al., 2023; Arafat et al., 2023). However, there is a technical limitation to this extension, namely the maximum number of items in the sidebar menu is only five items. To overcome this limitation, researchers added the AfterSelecting event command in the programming block so that it can display all seven menu items according to the list in Table 1. The following is a snippet of the programming blocks used to organize the home page (Dashboard) in the IoT Smart Home application.



(a) (b)

Figure 1. IoT Smart Home: (a) Home Page Block Program After the block design and programming stages are carried out on the home screen, the following are the results of the IoT Smart Home project installation on a mobile device; (b) Home page menu

Discussion

This study uses the ADDIE development model approach, which consists of five main stages, namely Analysis, Design, Development, Implementation, and Evaluation. In the first stage, namely analysis, researchers focus on collecting system requirements from both software and hardware sides. This stage is very crucial because it is the foundation of the entire development process. As stated by Aldoseri et al. (2023); Bernardo et al. (2024), in this analysis phase, all basic information must be collected carefully because errors in identifying initial needs can have a major impact on the final results of the system. In the context of this study, needs analysis includes identifying the features of the IoT application to be developed, determining the type of hardware and software to be used, and integration between components. From the software side, there are three main components that are analyzed and selected (Kerdoudi et al., 2022; Tariq et al., 2023). First, the Arduino IDE is used as a platform for programming the Wemos D1 R1 ESP8266 microcontroller and various sensors connected to it. This platform is open-source and highly supports the needs of microcontroller-based system development because it provides high flexibility in library management and writing program code (Chandra et al., 2023).

Second, MIT App Inventor was chosen as a tool for developing application interfaces based on blocks programming. This platform allows researchers to design the main pages of the application, such as the monitoring pages for lights and fans, temperature and humidity, as well as trash can and gate controls. Third, the use of Google's Firebase Realtime Database is a solution for the need for direct data synchronization between mobile applications and IoT devices. Firebase offers a cloud system that allows instant data storage and exchange, so that any changes to sensors or commands to the microcontroller can be processed in real-time. Using the REST API, Firebase supports up to 20.000 data writes and 50.000 data reads per day on the basic package (Za'bah & Mohamad, 2021), making it an ideal choice for application prototypes like this. Meanwhile, hardware requirements are also analyzed in detail to ensure system compatibility and efficiency. For the light and fan monitoring feature, the devices used include NodeMCU ESP8266 Wemos D1 mini series, 4 channel relay, lights, fan (fan cooler), jumper cables, and project board. For the temperature and humidity feature, researchers integrated the DHT11 sensor with NodeMCU ESP8266, accompanied by a project board and connecting cables.

As for the trash can monitoring feature, the system is equipped with an HCSR04 ultrasonic sensor that functions to detect distance and position, which is then connected to the microcontroller via jumper cables and media board. This entire series forms the initial infrastructure that is the basis for implementing an efficient and responsive smart home control system. Entering the design stage, researchers compiled a smart home architecture design that is an integrative framework for the entire system being built. This architecture reflects the process of collecting, processing, and analyzing information carried out by various devices connected via the internet network. In this system, small devices such as sensors and actuators that are spread throughout the house play an active role in responding to changes in environmental conditions dynamically. When a sensor is activated, the system immediately processes the signal and provides a response according to the programmed command. According to Goessler & Kaluarachchi (2023); Reis & Serôdio (2025), the ideal smart home architecture includes an adaptive and responsive reaction system to real-time data from sensors, thus creating intelligent and efficient automation.

The design of the application architecture design that has been prepared in this study illustrates how the interconnectedness between components works harmoniously in a well-managed IoT ecosystem (Vishwakarma et al., 2019; Yaïci et al., 2021). The following is an overview of the design of the smart home application architecture design**:**

Figure 2. Smart Home Design Architecture

Figure 3. Smart Home Interface Design

Figure 4. Design of Monitoring Interface for Lights and Fans

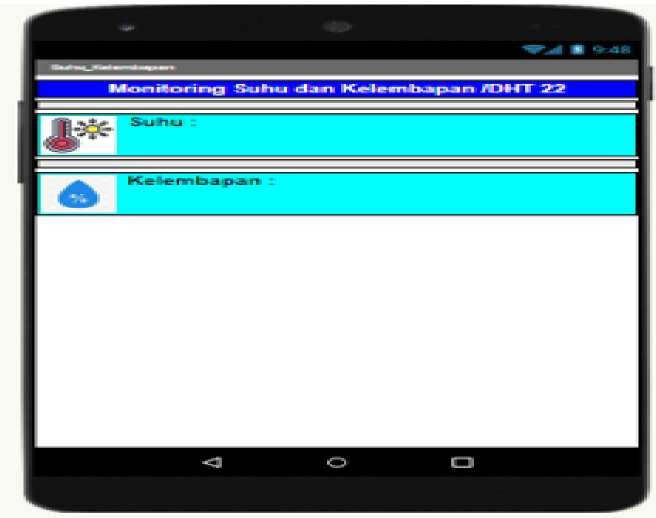


Figure 5. Temperature and Humidity Interface Design

Figure 6. Recycle Bin Interface Design

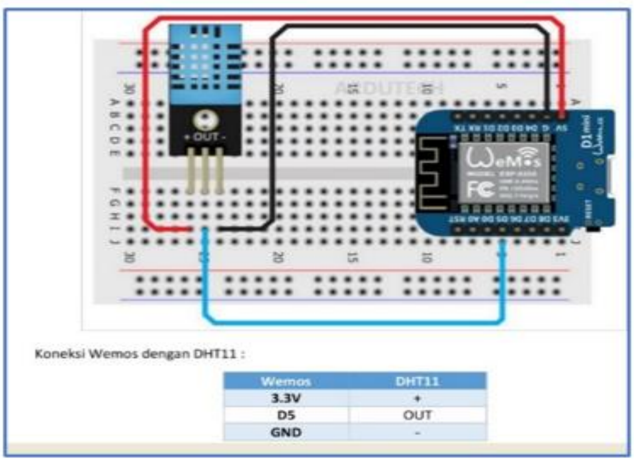


Figure 7. Hardware Wiring Design

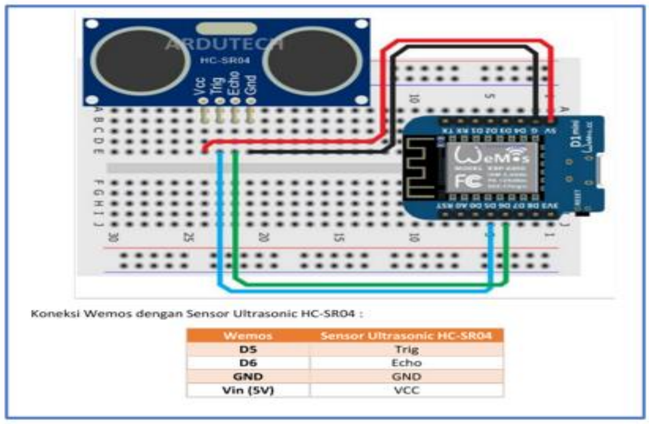


Figure 8. Wiring Ultra Sonic HC-SR04 Temperature and Humidity DHT 11 Distance Sensor

Entering the design stage, researchers conducted development to build IoT applications, which involved programming, namely after the interface design was created, programming was carried out with MIT App Inventor blocks. The following is a block image for the Lamp and Fan Monitoring design.

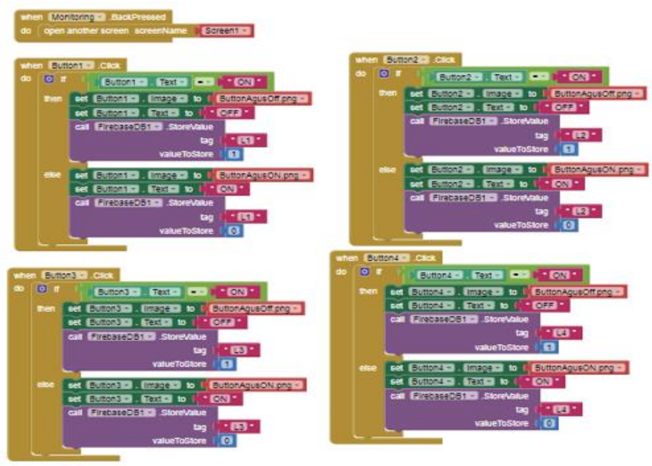


Figure 9. Programming the Lamp and Fan Monitoring Block

In the Lamp and Fan Monitoring Block there are 4 buttons, namely Button1 used for the porch light, Button2 for the living room light, Button3, for the bedroom light, Button4 for the fan where the event when clicked will test if the status button is On, then the text status becomes Off and vice versa if the text status Off becomes On, each status calls the realtime firebase which is read in the storevalue with a value of 0 lights off and 1 lights on. For the fan if the status is on the fan is on and vice versa.

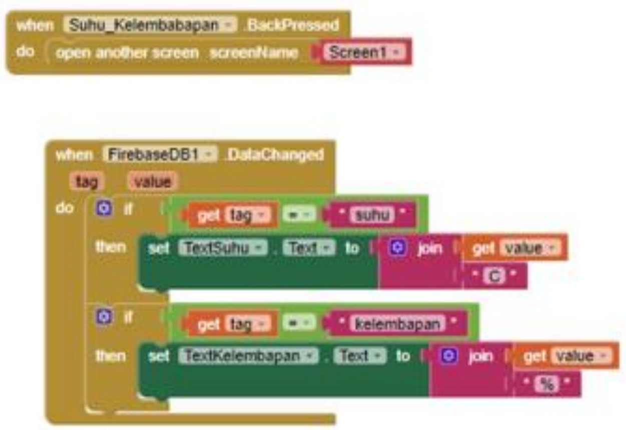


Figure 10. Temperature and humidity block using DHT sensor 11

In the Temperature and Humidity Block using the DHT 11 sensor with the firebase datachange event, if there is a realtime firebase change in temperature and humidity, the temperature text and humidity text will adjust. Steps to Send DHT11 Data to Firebase Prepare Firebase, Create a Firebase project, Enable Realtime Database or Firestore, Configure authentication (optional). Connect the ESP8266 to the DHT11 VCC → 3.3V or 5V, GND → Ground, DATA → GPIO pin in this study on pin D4 on the ESP8266.

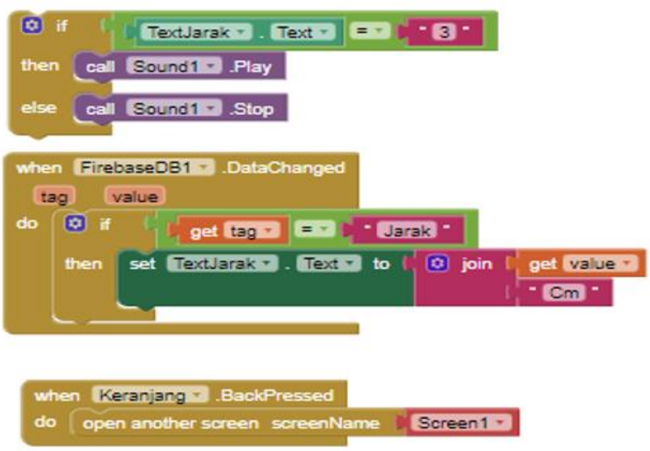


Figure 11. Block Application of trash bin using ultrasonic sensor HC04

In the Trash Bin Application Block using the HC04 ultrasonic sensor. The first algorithm if the text Distance = 3 cm then plays a sound that says the trash is full. Determining the distance is controlled by the Firebase datachange Event, if there is a change in distance tksjarak will provide a value according to the sensor working. You can send HC-SR04 ultrasonic sensor data to Firebase using ESP8266. Steps to Send HC-SR04 Data to Firebase, Prepare Firebase, Create a Firebase project. Enable Realtime Database or Firestore. Configure authentication (optional). Connect the ESP8266/ESP32 to the HC-SR04, VCC → 5V, GND → Ground, Trig → GPIO Pin (for example, D5 on the ESP8266, Echo → GPIO Pin in this study on pin D6 on the ESP8266. The implementation stage is a step where the entire system design that has been prepared in the design phase begins to be realized in the form of a functional application. In this study, the implementation was carried out by applying block-based programming through MIT App Inventor for the user interface side, and writing sketch code using the Arduino IDE to program the Wemos D1 Mini microcontroller based on the ESP8266.

One of the important components implemented is the temperature and humidity monitoring feature using the DHT11 sensor. This process begins with the configuration of the connection between the microcontroller and the Firebase Realtime Database which acts as cloud storage. In the program code uploaded to the microcontroller, the API Key and Firebase database URL are inserted as authentication identities and communication channels data (Azhari et al., 2023; Tang et al., 2020). Next the system is conditioned to be able to read temperature and humidity values every second. The sensor reading data is then sent to Firebase in real-time by utilizing the setFloat method provided by the Firebase library for Arduino. The response from Firebase, both success and failure of data delivery, is returned to the serial terminal for debugging purposes. This implementation proves that the system is not only able to read environmental data, but also synchronize it directly to the cloud server so that it can be accessed and monitored by users through the MIT App Inventor application (Jan et al., 2022). This programming structure shows that the logic design in the previous stage can be implemented effectively, and produces a stable integration between hardware and software (Roy & Abdul-Nour, 2024; Ryalat et al., 2024).

After the implementation is complete, the evaluation stage is carried out to assess the overall performance of the system (Tang et al., 2020; Yulianti, 2023). The evaluation in this study focuses on three main aspects, namely system performance, application usability, and scalability of use. The evaluation process was carried out through beta testing involving 20 respondents as end users. The main purpose of this test is to obtain objective feedback from users who are not involved in the development process. The evaluation instrument used was a questionnaire based on the EUCS (End User Computing Satisfaction) method, which measures user satisfaction with five main dimensions, namely Content, Accuracy, Ease of Use, Format, and Timeliness. Each statement in the questionnaire is assessed using a five-point Likert scale, from “Strongly Disagree” to “Strongly Agree”. This scale is then converted into a percentage and categorized using value intervals to determine the level of user satisfaction.

The EUCS model was chosen because it has proven to be relevant and adaptive in measuring user satisfaction with information systems, including the latest technology-based systems such as IoT. Several studies have even modified or strengthened this model to accommodate various contexts and dimensions of technology (Munap et al., 2018). Based on the results of data processing, the highest value was obtained in the Format dimension of 91%, which indicates that users are very satisfied with the readability of icons and text in the application. Followed by Ease of Use of 87% and Accuracy of 76%, which confirms that users find the application easy to use and accurate in representing data from the real world. The Content dimension scored 75%, while Timeliness scored 70%, indicating that there is still room for improvement in the speed of the application's response to user commands.

Table 2. Likert Scale Interval

Assessment Index (%)	Category
0 – 19.99	Very Dissatisfied
20 – 39.99	Dissatisfied
40 – 59.99	Neutral
60 – 79.99	Satisfied
80 - 100	Very Satisfied

The average satisfaction index of the five EUCS dimensions is 79.80%. Based on the Likert scale interval, this figure is included in the “Satisfied” category, which means that the IoT Smart Home application developed has met the expectations of the majority of users. The results of this evaluation are an important indicator that the system design that has been implemented is able to function according to its initial purpose, although it still opens up room for quality improvement, especially in terms of synchronization speed and user interface efficiency. This finding also emphasizes the importance of participatory evaluation in the IoT-based technology development cycle so that the final product truly meets the needs and convenience of users.

Table 3. Results of the EUCS Questionnaire Evaluation

Statement	Linkert Scale					% Calculation Results	Index
	SS	S	CS	TS	STS		
Content:	5	5	10	0	0	(5x5) + (5x4) + (10x3) = 75/100 = 0.75 75%	75
Dashboard page display information, Relay, humidity temperature, humidity sensor are displayed clearly	6	4	10	0	0	(6x5) + (4x4) + (10x3) = 76/100 = 0.76 76%	76
Accuracy:	10	7	3	0	0	(10x5) + (7x4) + (3x3) = 87/100 = 0.87 87%	87
Application information is displayed according to the real world	12	7	1	0	0	(12x5) + (7x4) +(1x3) = 91/100 = 0.91 91%	91
Ease Of Use:	5	5	5	5	0	(15x5) +(5x4) +(5x3) +(5x2) = 70/100 = 0.70 70%	70
						Average Index	79.80

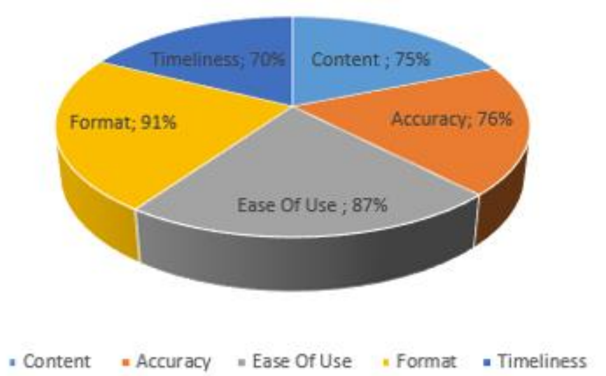


Figure 11. EUCS Evaluation Graph for each dimension

From the results of the EUCS dimension questionnaire, the average index result is 79.8%, which if referring to the Likert scale interval table is included in the "Satisfied" category. The following is the EUCS Evaluation graph for each dimension.

Conclusion

Based on the analysis and evaluation using the EUCS method of the IoT Smart Home Design and Construction research with Blocks Based Mit App Inventor Programming, the conclusions obtained for each dimension are Content 75% "Satisfied", Accuracy

76% "Satisfied", Ease Of Use 87% "Very Satisfied", Format 91% "Very Satisfied", Timeliness 70% "Satisfied", so that the final result of the average index is 79.8% in the "Satisfied" category, the results of the analysis are that improvements are needed to achieve the results of "Very Satisfied". The suggestion for this research in the future is to improve the category, especially in the category below 80%, namely content, accuracy and Timeliness, the hope of the research is how blocks based can inspire people to be creative with IoT so that beginners can build mobile applications integrated with IoT technology. in addition, subsequent research can be used with different methods.

Acknowledgments
Thanks to all parties who have supported the implementation of this research. I hope this research can be useful.

Author Contributions
Conceptualization; methodology; validation; formal analysis; investigation; resources; data curation; writing original draft: review and editing; visualization A. S, the author has read and approved the published version of the manuscript.

Funding
This research was funded by the Information Systems Study Program, Faculty of Computer Science, Pamulang University, South Tangerang.

Conflicts of Interest
This research was conducted on the mandate of the institution to improve the competence and capacity of lecturers. It is expected that the findings of this study can provide a significant positive impact on the development of human resources, especially in the academic environment, as well as support the creation of innovation and progress in the world of education.

References

Abiri, R., Rizan, N., Balasundram, S. K., Shahbazi, A. B., & Abdul-Hamid, H. (2023). Application of digital technologies for ensuring agricultural productivity. *Heliyon*, *9*(12), e22601. https://doi.org/10.1016/j.heliyon.2023.e22601

Aïmeur, E., Amri, S., & Brassard, G. (2023). Fake news, disinformation and misinformation in social media: A review. *Social Network Analysis and Mining*, *13*(1). https://doi.org/10.1007/s13278-023-01028-5

Aldoseri, A., Al-Khalifa, K. N., & Hamouda, A. M. (2023). Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges. *Applied Sciences*, *13*(12), 7082. https://doi.org/10.3390/app13127082

Arafat, M. Y., Alam, M. M., & Moh, S. (2023). Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges. *Drones*, *7*(2), 89. https://doi.org/10.3390/drones7020089

Azhari, Nasution, T. I., Sinaga, S. H., & Sudiati. (2023). Design of Monitoring System Temperature And Humidity Using DHT22 Sensor and NRF24L01 Based on Arduino. *Journal of Physics: Conference Series*, *2421*(1), 012018. https://doi.org/10.1088/1742-6596/2421/1/012018

Baena-Perez, R., Ruiz-Rube, I., Mota, J. M., Berns, A., & Balderas, A. (2024). Visual authoring of virtual reality conversational scenarios for e-learning. *Universal Access in the Information Society*, *23*(1), 227–244. https://doi.org/10.1007/s10209-022-00934-3

Bernardo, B. M. V., Mamede, H. S., Barroso, J. M. P., & Dos Santos, V. M. P. D. (2024). Data governance & quality management—Innovation and breakthroughs across different fields. *Journal of Innovation & Knowledge*, *9*(4), 100598. https://doi.org/10.1016/j.jik.2024.100598

Cen, C., Luo, G., Li, L., Liang, Y., Li, K., Jiang, T., & Xiong, Q. (2023). User-Centered Software Design: User Interface Redesign for Blockly–Electron, Artificial Intelligence Educational Software for Primary and Secondary Schools. *Sustainability*, *15*(6), 5232. https://doi.org/10.3390/su15065232

Chandra, Y. I., Irfan, I., Gustina, D., Purtiningrum, S. W., & Yuliani, N. (2023). Real-Time Prototype Electricity Monitoring and Forecasting System based on Wemos D1 R1 ESP8266 and IoT. *Logistic and Operation Management Research (LOMR)*, *2*(2), 1–13. https://doi.org/10.31098/lomr.v2i2.1551

Choudhary, A. (2024). Internet of Things: A comprehensive overview, architectures, applications, simulation tools, challenges and future directions. *Discover Internet of Things*, *4*(1). https://doi.org/10.1007/s43926-024-00084-3

Gharaat, M., Sharbaf, M., Zamani, B., & Hamou-Lhadj, A. (2021). ALBA: A model-driven framework for the automatic generation of android location-based apps. *Automated Software Engineering*, *28*(1). https://doi.org/10.1007/s10515-020-00278-3

Goessler, T., & Kaluarachchi, Y. (2023). Smart Adaptive Homes and Their Potential to Improve Space Efficiency and Personalisation. *Buildings*, *13*(5), 1132. https://doi.org/10.3390/buildings13051132

Jan, F., Min-Allah, N., Saeed, S., Iqbal, S. Z., & Ahmed, R. (2022). IoT-Based Solutions to Monitor Water Level, Leakage, and Motor Control for Smart Water Tanks. *Water*, *14*(3), 309. https://doi.org/10.3390/w14030309

Jayatilleke, B. G., Ranawaka, G. R., Wijesekera, C., & Kumarasinha, M. C. B. (2019). Development of mobile application through design-based research. *Asian Association of Open Universities Journal*, *13*(2), 145–168. https://doi.org/10.1108/aaouj-02-2018-0013

Jihadi, H., Kristantini, A., Baskoro, H., Maulana, D., & Muhammad, S. (2024). Developing Simple Android Applications On Websites With MIT App Inventor For Students Of SMAN 4 Bogor City. *International Journal Of Community Service*, *4*(3), 206–212. https://doi.org/10.51601/ijcs.v4i3.281

Kaddipujar, M., Rajan, J., & Kumbar, B. D. (2022). Mobile Application Development Using MIT App Inventor: An Experiment at Raman Research Institute Library. *Bulletin of the AAS*. https://doi.org/10.3847/25c2cfeb.d68a2a42

Kerdoudi, M. L., Ziadi, T., Tibermacine, C., & Sadou, S. (2022). A novel approach for Software Architecture Product Line Engineering. *Journal of Systems and Software*, *186*, 111191. https://doi.org/10.1016/j.jss.2021.111191

Kumar, R., Verma, A., Shome, A., Sinha, R., Sinha, S., Jha, P. K., Kumar, R., Kumar, P., Shubham, Das, S., Sharma, P., & Vara Prasad, P. V. (2021). Impacts of Plastic Pollution on Ecosystem Services, Sustainable Development Goals, and Need to Focus on Circular Economy and Policy Interventions. *Sustainability*, *13*(17), 9963. https://doi.org/10.3390/su13179963

Lombardi, M., Pascale, F., & Santaniello, D. (2021). Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information*, *12*(2), 87. https://doi.org/10.3390/info12020087

Munap, R., Ahmad, S. N. B., Hamid, S. A., & Talib Beg, M. F. B. M. (2018). The Influence of End User Computing System (EUCS) on User Satisfaction: The Case of a Logistic and Courier Service Company. *International Journal of Social Sciences and Humanities Invention*, *5*(12), 5103–5110. https://doi.org/10.18535/ijsshi/v5i12.03

Nižetić, S., Šolić, P., López-de-Ipiña González-de-Artaza, D., & Patrono, L. (2020). Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*, *274*, 122877. https://doi.org/10.1016/j.jclepro.2020.122877

Norouzkhani, N., Norouzi, S., Faramarzi, M., Bahari, A., Shirvani, J. S., Eslami, S., & Tabesh, H. (2025). Developing and evaluating a gamified self-management application for inflammatory bowel disease using the ADDIE model and Sukr framework. *BMC Medical Informatics and Decision Making*, *25*(1). https://doi.org/10.1186/s12911-024-02842-3

Pancane, I. W. D., & Zahir, A. S. S. A. (2024). Rancang Bangun Sistem Kontrol dan Monitoring Beban Listrik Pada Unit Produksi Bakmi Gm Denpasar Dengan Internet of Think Menggunakan App Inventor. *Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro Dan Informatika*, *3*(1), 84–102. https://doi.org/10.61132/jupiter.v3i1.660

Priawardana, S. G., & Surriani, A. (2025). Design of a Monitoring and Nutrient Management System Based on Internet of Things (IoT) for Hydroponic Method Using MIT App Inventor. *Jurnal Listrik, Instrumentasi, Dan Elektronika Terapan*, *6*(1), 1. https://doi.org/10.22146/juliet.v6i1.99431

Quach, S., Thaichon, P., Martin, K. D., Weaven, S., & Palmatier, R. W. (2022). Digital technologies: Tensions in privacy and data. *Journal of the Academy of Marketing Science*, *50*(6), 1299–1323. https://doi.org/10.1007/s11747-022-00845-y

Reis, M. J. C. S., & Serôdio, C. (2025). Edge AI for Real-Time Anomaly Detection in Smart Homes. *Future Internet*, *17*(4), 179. https://doi.org/10.3390/fi17040179

Roy, M.-A., & Abdul-Nour, G. (2024). Integrating Modular Design Concepts for Enhanced Efficiency in Digital and Sustainable Manufacturing: A Literature Review. *Applied Sciences*, *14*(11), 4539. https://doi.org/10.3390/app14114539

Ryalat, M., Franco, E., Elmoaqet, H., Almtireen, N., & Al-Refai, G. (2024). The Integration of Advanced Mechatronic Systems into Industry 4.0 for Smart Manufacturing. *Sustainability*, *16*(19), 8504. https://doi.org/10.3390/su16198504

Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*, *70*, 263–286. https://doi.org/10.1016/j.jbusres.2016.08.001

Stolojescu-Crisan, C., Crisan, C., & Butunoi, B.-P. (2021). An IoT-Based Smart Home Automation System. *Sensors*, *21*(11), 3784. https://doi.org/10.3390/s21113784

Strzelecki, A., Kolny, B., & Kucia, M. (2024). Smart Homes as Catalysts for Sustainable Consumption: A Digital Economy Perspective. *Sustainability*, *16*(11), 4676. https://doi.org/10.3390/su16114676

Tang, X., Tan, C., Chen, A., Li, Z., & Shuai, R. (2020). Design and implementation of temperature and humidity monitoring system for small cold storage of fruit and vegetable based on Arduino. *Journal of Physics: Conference Series*, *1601*(6), 062010. https://doi.org/10.1088/1742-6596/1601/6/062010

Tariq, U., Ahmed, I., Bashir, A. K., & Shaukat, K. (2023). A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review. *Sensors*, *23*(8), 4117. https://doi.org/10.3390/s23084117

Vardakis, G., Hatzivasilis, G., Koutsaki, E., & Papadakis, N. (2024). Review of Smart-Home Security Using the Internet of Things. *Electronics*, *13*(16), 3343. https://doi.org/10.3390/electronics13163343

Vishwakarma, N. K., Singh, R. K., & Sharma, R. R. K. (2019). Internet of things architectures: Do organizational strategies matters? *Business Process Management Journal*, *26*(1), 102–131. https://doi.org/10.1108/bpmj-03-2018-0092

Yaïci, W., Krishnamurthy, K., Entchev, E., & Longo, M. (2021). Recent Advances in Internet of Things (IoT) Infrastructures for Building Energy Systems: A Review. *Sensors*, *21*(6), 2152. https://doi.org/10.3390/s21062152

Yulianti, Y. (2023). Implementation of Computer-Based Summative Assessment. *Jurnal Penelitian Pendidikan IPA*, *9*(11), 10439–10445. https://doi.org/10.29303/jppipa.v9i11.5494

Za'bah, N. F., & Mohamad, A. (2021). Mobile Application Monitoring System Using Internet of Things (IOT) and Firebase. *Journal of Information Systems and Digital Technologies*, *3*(1), 16–30. https://doi.org/10.31436/jisdt.v3i1.168

Żyła, K., Chwaleba, K., & Choma, D. (2024). Evaluating Usability and Accessibility of Visual Programming Tools for Novice Programmers—The Case of App Inventor, Scratch, and StarLogo. *Applied Sciences*, *14*(21), 9887. https://doi.org/10.3390/app14219887