

IMPLEMENTATION OF A PYTHON-BASED SCHEDULED AUDIO ALARM SYSTEM FOR LIBRARY LITERACY SUPPORT

Khalifah Audya Eka Putri¹, Endah Setyowati^{1*}

¹Telecommunication System, Universitas Pendidikan Indonesia

email: *endahsetyowati@upi.edu

Abstract: Libraries function not only as information centers but also as literacy spaces that require an orderly and communicative service environment. One supporting service in fostering such an environment is the delivery of literacy greetings to visitors. In practice, greetings are commonly delivered manually or through conventional bells, leading to inconsistency and dependence on staff availability. This study was conducted at the Amir Machmud Library, Ministry of Home Affairs, Jakarta, Indonesia, aiming to design and evaluate a Python-based scheduled audio alarm system for automated literacy greetings. An applied experimental method was employed, including system design, Python script development, scheduling configuration using Windows Task Scheduler, and direct system testing on a library computer connected to ceiling speakers. The system requires initial execution via Command Prompt (CMD) when the computer is powered on, after which it operates automatically according to predefined schedules. Testing results demonstrate that the system performs scheduled audio playback accurately and operates stably without further manual intervention. The findings indicate that the proposed system provides a practical and efficient solution to enhance service consistency and support a structured and conducive literacy environment in the library.

Keywords: scheduled audio alarm; library automation; python; literacy greeting.

Abstrak: Perpustakaan tidak hanya berfungsi sebagai pusat informasi, tetapi juga sebagai ruang literasi yang memerlukan suasana layanan yang tertib dan komunikatif. Salah satu bentuk dukungan layanan tersebut adalah penyampaian sapaan literasi kepada pengunjung. Dalam praktiknya, penyampaian sapaan masih dilakukan secara manual atau menggunakan bel konvensional sehingga kurang konsisten dan bergantung pada petugas. Penelitian ini dilaksanakan di Perpustakaan Amir Machmud, Kementerian Dalam Negeri, Jakarta, Indonesia, dengan tujuan merancang dan menguji sistem alarm audio terjadwal berbasis Python sebagai media penyampaian sapaan literasi. Metode yang digunakan adalah metode eksperimental terapan melalui tahapan perancangan sistem, pengembangan skrip Python, konfigurasi penjadwalan menggunakan Windows Task Scheduler, serta pengujian langsung pada komputer perpustakaan yang terhubung dengan speaker plafon. Sistem bekerja dengan mekanisme inisialisasi awal melalui Command Prompt (CMD) saat komputer dinyalakan, kemudian selanjutnya berjalan otomatis sesuai jadwal yang telah ditentukan. Hasil pengujian menunjukkan bahwa sistem mampu memutar audio secara konsisten dan stabil pada waktu yang telah diatur tanpa intervensi lanjutan dari petugas. Dengan demikian, sistem ini dapat menjadi solusi sederhana dan efisien untuk mendukung terciptanya suasana literasi yang lebih terstruktur dan kondusif di lingkungan perpustakaan.

Kata kunci: alarm audio terjadwal; otomasi perpustakaan; python; sapaan literasi.



PENDAHULUAN

The development of modern library services requires space and service management that is not solely collection-oriented but also focused on user experience. The library environment significantly influences users' comfort, concentration, and reading interest; therefore, it must be supported by services capable of delivering messages consistently and effectively [1][2]. One form of such supporting service is the delivery of greetings or literacy announcements as an indirect communication medium between the library management and visitors.

According to previous research [5], the delivery of greetings in many libraries is still conducted conventionally, either through simple bells or manual audio playback by staff. Conventional bells generally function only as time markers without conveying informative verbal messages, thus failing to provide direct literacy content to visitors. Meanwhile, manual audio playback heavily depends on staff availability and consistency, potentially leading to irregularities in timing and frequency of message delivery. As a result, literacy messages are not always delivered in a structured and continuous manner within the library environment.

Previous studies indicate that scheduled systems in computer environments can be utilized to execute repetitive processes automatically with high reliability [8]. However, most library automation research still focuses on administrative and collection management aspects [9]. While the implementation of automation in supporting services such as literacy greetings remains relatively limited. In

contrast, artificial intelligence-based technologies and automation play a significant role in supporting faster, more consistent, and responsive information services [10].

The implementation of automated systems in digital libraries has been proven to improve efficiency and reduce dependence on manual processes [3][4]. Software advancements also create opportunities for implementing simple automation to support information delivery within library environments [6][7]. The Python programming language is widely used in library system development due to its flexibility and ease of implementation [11][12], as well as its effectiveness in executing scheduled and repetitive tasks automatically [13].

In addition, audio media serves as an effective communication tool in public spaces because it can reach users without requiring direct visual attention. In the library context, structured and consistently delivered audio messages can enhance user attention and responsiveness to the information provided [14][15].

Based on these conditions, a research gap exists in developing a simple, affordable automated literacy greeting system that can be implemented using existing infrastructure. This study aims to design and test a Python-based scheduled audio alarm system as a medium for delivering automated literacy greetings in a library environment. The system is designed to operate on a library computer by utilizing the scheduling mechanism of the Windows operating system, enabling literacy greetings to be played semi-automatically and consistently according to a predetermined schedule. This research is expected to

produce an applicable and efficient supporting service solution to foster a conducive and literacy-friendly library atmosphere.

METHOD

This study employed an applied experimental method by directly testing a scheduled audio alarm system as a medium for delivering automated literacy greetings in a library environment. The system was tested on a library computer connected to ceiling-mounted speakers, ensuring that the results represent actual operational conditions.

The hardware and software specifications used in this study are presented in Table 1.

Tabel 1. Components and System Specifications

Component	Specification
Operating System	Windows 10/11
Programming Language	Python 3.13
Program Editor	Notepad
Scheduler	Task Scheduler
Audio Format	MP3/WAV
Output	Library Ceiling Speakers

The system was developed using the Python programming language, with Task Scheduler functioning as the automatic scheduling mechanism. Audio files were stored in MP3 format and played through the library’s ceiling-mounted speakers as the primary system output.

Research Stages

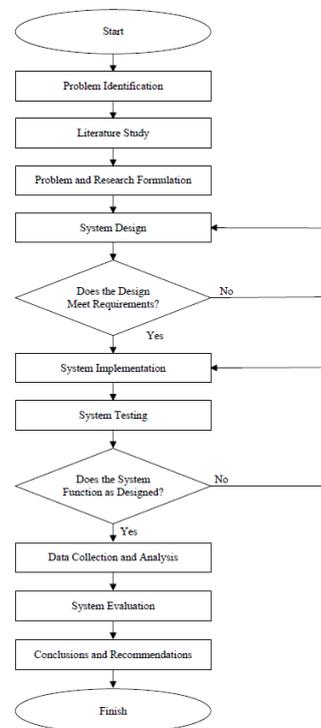


Figure 1. Research Stages

The research stages began with problem identification through observation of the literacy greeting delivery process in the library environment, which was still conducted manually and not consistently scheduled. A literature review was subsequently carried out as the theoretical foundation for system development and for formulating the research framework and problem statement.

Based on this formulation, system design was conducted, including workflow design, program structure, and scheduling mechanisms. The proposed design was evaluated to ensure its suitability with operational requirements. The next stage involved system implementation, namely translating the design into a Python-based program and

configuring the Task Scheduler. The implemented system was then tested to ensure that all functions operated according to the design. If the system did not function properly, revisions and retesting were performed. Once the system was declared functional, data collection and analysis were conducted, followed by evaluation to draw conclusions and research recommendations.

System Workflow

The workflow of the scheduled audio alarm system begins when the system is active and in standby mode. In this state, the system periodically monitors the system clock and matches it with the scheduled playback times for literacy greeting audio configured within the program.

If the system time matches the predetermined schedule, the system executes the playback of the literacy greeting audio through the library speakers. Otherwise, the system remains in standby mode until the scheduled time is reached. After the audio playback is completed, the system returns to standby mode to wait for the next scheduled playback. This process repeats continuously as long as the system remains active. The overall workflow of the scheduled audio alarm system is illustrated in Figure 2.

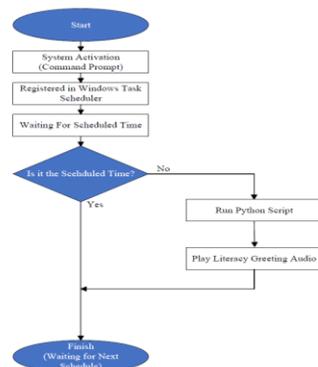


Figure 2. System Workflow

System Implementation

The implementation of the scheduled audio alarm system was carried out by translating the workflow design into a Python-based program executed in a Windows operating system environment. The system was implemented on a library computer connected to ceiling-mounted speakers, enabling literacy greeting audio to be broadcast directly within the service area.

The initial implementation stage involved writing the program code using Visual Studio Code. After preliminary testing, the program was transferred to a simple text editor such as Notepad and saved as a file with a .py extension (e.g., filename.py). The program file was then placed in the Python 3.13 installation directory to facilitate system access.

To simplify system operation, Task Scheduler was used as the initial execution mechanism. It was configured using the “When I log on” option, so that the Python program would automatically run when the user logs into the Windows operating system.

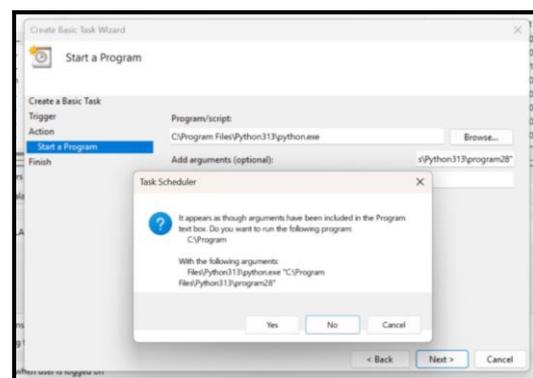


Figure 3. Task Scheduler Configuration

In the Action menu, the system was configured to execute the Python interpreter by specifying the path:

C:\ProgramFiles\Python313\python.exe.

In the “Add arguments” field, the path to the Python program file was entered. This configuration ensures that the Python interpreter executes the specified script independently of the operating system’s default file association settings.

As a verification stage, the program was also executed manually through Command Prompt using the command: “python.exe filename.py”. This step ensured that the program could run properly and that core functions—such as system time monitoring and audio playback—operated without errors before full deployment via the scheduling mechanism.

RESULTS AND DISCUSSION

Based on the testing results, the Python-based scheduled audio alarm system was successfully operated on the library computer connected to ceiling-mounted speakers. The program can be executed through Command Prompt as shown in Figure 4. After initialization, the system enters standby mode to monitor system time according to the scheduling logic embedded in the program code. The audio playback schedule is fully controlled by the Python program, while Windows Task Scheduler functions as the initial trigger during the operating system login process.

```
C:\Program Files\Python313>python.exe program30.py
C:\Users\hratt\AppData\Roaming\Python\Python313\site-packages\py
game\pkgdata.py:25: UserWarning: pkg_resources is deprecated as
an API. See https://setuptools.pypa.io/en/latest/pkg_resources.h
tml. The pkg_resources package is slated for removal as early as
2025-11-30. Refrain from using this package or pin to Setuptools
<81
  from pkg_resources import resource_stream, resource_exists
pygame 2.6.1 (SDL 2.28.4, Python 3.13.5)
Hello from the pygame community. https://www.pygame.org/contribu
te.html
[SYSTEM] Program berjalan...
[ALARM] 00:25 - Bel Berbunyi!
[INFO] Memastikan YouTube dipause...
[INFO] Mencoba fokus ke Chrome...
[WARNING] Chrome tidak ditemukan.
[INFO] Tidak ada Chrome aktif. Skip pause.
[INFO] Memutar alarm...
[INFO] Memainkan kembali YouTube...
[INFO] Alarm selesai.
```

Figure 4. Command Prompt Display

When Python Is Executed

When the system time reaches the predetermined schedule, the literacy greeting audio is played automatically. After one playback cycle is completed, the system returns to standby mode to wait for the next schedule. This mechanism repeats continuously while the system remains active.

Table 2. Testing Results

Testing Parameter	Result
Scheduled audio playback	Successful
Playback time accuracy	On schedule
System activation via Task Scheduler	Successful
Operation while YouTube is active	Successful
Manual intervention during testing	Required once at Windows login
Errors during testing	None

Testing was conducted to evaluate the functional aspects of the system, including playback timing accuracy, successful activation via Task Scheduler, and system stability under actual operational conditions. The results summarized in Table 2 indicate that all parameters operated according to the design. Audio was played precisely on schedule, system activation at Windows login functioned properly, and no errors were detected during testing.

In an additional scenario, YouTube video playback was simulated through a web browser. The results showed that the system automatically paused the video before playing the literacy greeting audio and resumed

playback afterward. This mechanism demonstrates the system's ability to maintain clarity of information delivery without causing permanent disruption to users multimedia activities. Manual intervention was required only once during the initial stage when the computer was powered on and the user logged into the operating system. After that, the system operated semi-automatically by continuously monitoring time and executing scheduled audio playback without further staff intervention.

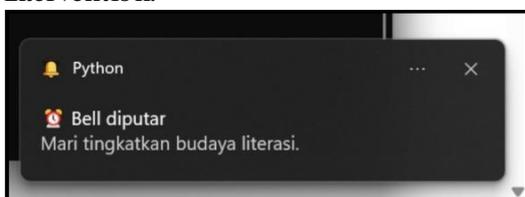


Figure 5. Audio Playback Notification

Overall, the research findings indicate that the system demonstrates sufficient reliability and stability for implementation in a library environment. The implementation not only fulfills the technical function of scheduled audio playback but also contributes to improving literacy services through consistent, structured, and context-adaptive information delivery.

CONCLUSION

This study resulted in the development of a Python-based scheduled audio alarm system that can be effectively implemented and operated in a library environment. The system is capable of playing literacy greeting audio according to predetermined schedules through a program-based scheduling mechanism. Based on testing results, the system operates semi-automatically, requiring one manual intervention during

the initial stage when the computer is powered on and the user logs into the operating system. After this process, the system operates independently in monitoring time and executing scheduled playback without further staff intervention.

The integration of the Python program and Windows Task Scheduler ensures system activation upon login and maintains operational stability, including during concurrent multimedia activities. The system can manage overlapping audio playback with other applications by temporarily pausing active media and resuming it after the literacy greeting finishes, thereby maintaining user comfort. All testing parameters performed according to the design, with no errors identified during evaluation.

For future development, the system may be enhanced toward full automation through autorun configuration without dependency on user login, as well as integration with a web-based dashboard for dynamic scheduling and content management. Further research may also incorporate artificial intelligence features for adaptive greeting content and conduct quantitative evaluation of the system's impact on user comfort and reading interest. Integration with Internet of Things (IoT) technology may further support real-time monitoring and management within a smart library framework.

ACKNOWLEDGMENTS

The author would like to express sincere gratitude to Mr. Moh. Ilham A. Hamudy, M.Soc.Sc., as internship supervisor, and Yosi Pratama, S.Kom., as internship mentor, as well as all staff of the Amir Machmud Library, Ministry of

Home Affairs, for their guidance, support, and assistance during the data collection and system testing processes.

The author also extends appreciation to the academic supervisor for providing constructive guidance, feedback, and support throughout the research and manuscript preparation process, enabling this study to be completed successfully.

DAFTAR PUSTAKA

- [1] O. Rizkyantha, D. Tamalah, and R. Iswanto, “Pengaruh Fasilitas Perpustakaan terhadap Minat Baca Siswa di Perpustakaan MAN Curup Rejang Lebong,” *TADWIN J. Ilmu Perpust. dan Inf.*, vol. 3, no. 2, pp. 124–135, 2022.
- [2] N. Yoliadi, Dodi, “Pengaruh Pemanfaatan Automasi Perpustakaan (E- Library) Terhadap Kemampuan Literasi Informasi Mahasiswa Fakultas Ushuluddin Adab dan Dakwah UIN Mahmud Yunus Batusangkar,” vol. 1, no. 2, pp. 87–97, 2022.
- [3] M. Yusuf, “Peran Artificial Intelligence (AI) sebagai Pendukung Otomatisasi Perpustakaan,” *J. Kaji. Inf. dan Perpust.*, vol. 16, no. 1, pp. 41–57, 2024, doi: 10.21154/pustakaloka.v16i1.751.
- [4] B. E. Slam, F. Irawan, N. Efranda, and R. Herikson, “Implementasi Machine Learning Untuk Klasifikasi Buku Otomatis Pada Perpustakaan Digital,” *JIP (Jurnal Inform. Polinema)*, vol. 11, no. 3, pp. 305–310, 2025.
- [5] A. R. Saleh, D. M. D. Warouw, and A. Runtuwene, “Peran sistem otomasi dalam meningkatkan pelayanan pada pengguna perpustakaan dan kearsipan kota tidore kepulauan,” pp. 1–6.
- [6] N. N. Ashikin, A. Nur, L. Hanum, S. Perpustakaan, S. Perpustakaan, and P. S. Perpustakaan, “Penerapan Sistem Otomasi Perpustakaan Pada Layanan UPT Perpustakaan Universitas OSO,” vol. 3, no. 1, pp. 1–8, 2024.
- [7] B. A. Asid, “LIBRARY AUTOMATION SYSTEM OF ACADEMIC LIBRARIES : A MULTICULTURAL PARADIGM,” vol. 15, no. 2, pp. 236–262, 2020.
- [8] A. S. Ngozi, “Library Automation in University: A Literature Review,” vol. 13, no. 2, pp. 115–134, 2020.
- [9] K. R. Retno Wulan, Ni Gusti Ayu and I. B. Gede Purwa, “Optimalisasi Sistem Otomasi Sebagai Upaya Peningkatan Layanan Penelusuran Koleksi Buku Bagi Pengguna di Perpustakaan Universitas Pendidikan Ganesha,” *Acarya Pustaka*, vol. 8, no. 1, 2021.
- [10] F. Sentiana, B. Mustofa, Muhamad, and S. Wuryan, “Pemanfaatan artificial intelligence pada layanan informasi di perpustakaan,” *J. Ilm. Ilmu Perpust. dan Inf.*, vol. 12, no. 2, 2024.
- [11] S. Rudianto, A. A. Sapaatullah, S. Permana, and M. Darip, “Implementasi struktur data array dalam sistem perpustakaan berbasis web menggunakan Python Flask,” *Bul. Ilm. Inform. Teknol.*, vol. 3, no. 2, pp. 30–36, 2025.

- [12] H. Lowagie and V. Woensel, Julie, “Simplifying Subject Indexing: A Python-Powered Approach in KBR, the National Library of Belgium,” *Code 4 Lib J.*, no. 59, 2024.
- [13] A. Garg, R. D. Smith-unna, and P. Murray-rust, “pygetpapers: a Python library for automated retrieval of scientific literature,” *J. Open Source Softw.*, vol. 7, no. 75, pp. 1–7, 2022, doi: 10.21105/joss.04451.
- [14] K. Budiman, M. H. Adi, A. Munawar, and A. Purwinarko, “The Telegram notification system for improving library services Universitas Negeri Semarang,” vol. 10, no. 1, pp. 75–88, 2022.
- [15] F. Rizki Rizaldi, Moch, Z. Lapi, Imron, A. Akbar, Muhammad, and D. Widayaka, Parama, “Inventarisasi peminjaman buku secara otomatis pada perpustakaan menggunakan rfid berbasis iot via telegram,” vol. 8, no. 1, pp. 1–16, 2023.