ANALYSIS OF THE AUTO CONVERT SYSTEM IN THE SMT PRODUCTION LINE AT XYZ COMPANY USING THE RAD AND PIECES APPROACHES

Noval Zahwan Fhalma[1], Noper Ardi[2S]

[1] Batam Polytechnics
Software Engineering study Program
Ahmad Yani Street, Batam Centre, Batam 29461, Indonesia
E-mail: novalzf29@gmail.com

[2] Batam Polytechnics
Informatics Study Program
Ahmad Yani Street, Batam Centre, Batam 29461, Indonesia
E-mail: noperardi@polibatam.ac.id

Abstrak

Proses changeover pada mesin Printing DEK NeoHorizon di lini produksi Surface Mount Technology (SMT) di perusahaan XYZ masih dilakukan secara manual, sehingga membutuhkan waktu relatif lama dan berisiko menimbulkan kesalahan operator dalam pemilihan project. Kondisi ini berdampak pada efisiensi dan produktivitas produksi. Penelitian ini bertujuan merancang dan menganalisis sistem auto convert menggunakan ASP.NET untuk mengotomatisasi proses changeover pada mesin Printing DEK dan Solder Paste Inspection (SPI). Metode pengembangan menggunakan Rapid Application Development (RAD) dan evaluasi menggunakan kerangka PIECES (Performance, Information, Economy, Control, Efficiency, Service). Sistem mengintegrasikan aplikasi web ASP.NET, automation engine, barcode scanner, dan server SQL. Hasil implementasi membuktikan sistem mampu menekan waktu changeover, mengurangi intervensi manual, meningkatkan efisiensi, dan menurunkan potensi human error.

Kata kunci: Auto Convert, Surface Mount Technology (SMT), Printing DEK Machine, SPI Machine, Rapid Application Development (RAD), PIECES.

Abstract

The changeover process on the DEK NeoHorizon Printing machine in the Surface Mount Technology (SMT) production line at company XYZ is still carried out manually, requiring a relatively long time and risking operator errors in project selection. This condition impacts overall efficiency and productivity. This study aims to design and analyze an auto convert system using ASP.NET to automate the changeover process on DEK Printing and Solder Paste Inspection (SPI) machines. Development is conducted using Rapid Application Development (RAD) and evaluation through PIECES (Performance, Information, Economy, Control, Efficiency, Service) framework. The system integrates ASP.NET web applications, an automation engine, barcode scanner, and SQL server. Implementation results show the system significantly reduces changeover time, minimizes manual intervention, increases efficiency, and reduces human error potential.

Keywords: Auto Convert, Surface Mount Technology (SMT), Printing DEK Machine, SPI Machine, Rapid Application Development (RAD), PIECES.

1. Introduction

XYZ Company is a multinational Original Equipment Manufacturer (OEM) with an electronic component assembly line using Surface Mount Technology (SMT). Automation in production systems—particularly in electronics and SMT—plays a crucial role in improving efficiency and reducing manufacturing operating costs [7].

Currently, the changeover process (switching between projects) on the DEK NeoHorizon SMT printing machine is still manual. Operators must stop the machine, locate the files, and select the program profile—a process that is time-consuming and prone to human error. Manual systems in the digital manufacturing environment tend to reduce operational accuracy and efficiency [16]. Therefore, automation that can be managed through a centralized application is required [17].

The proposed auto-convert application is built using Active Server Pages with the .NET framework (ASP.NET) [19]. The ASP.NET and Razor Pages framework approach provides high security [6] and solid performance in internal corporate control systems [2], [21]. In addition, this system was developed using the Rapid Application Development (RAD) method [8], [11], and performance improvements were evaluated using the PIECES (Performance, Information, Economy, Control, Efficiency, Service) framework [1], [14].

2. Research Method

A. Research Flow

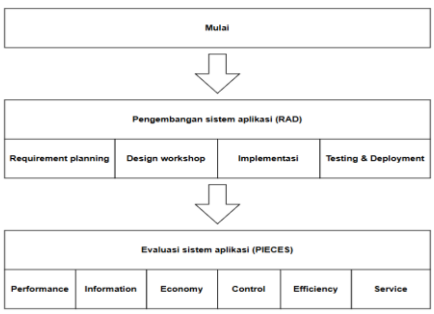

Figure 1. Research Flow

The research began with identifying issues in the manual changeover process for the DEK NeoHorizon and SPI machines, which had the potential to lead to incorrect project selection, take a long time, and result in minimal centralized control over files. The results served as the foundation for formulating objectives and designing the auto-convert system. Further development utilized the Rapid Application Development (RAD) method through four main stages:

a. Requirement Planning to define requirements through observation, interviews, and document review.
b. Design Workshop involving collaborative design of diagrams and user interfaces.
c. implementation using C# and the ASP.NET framework.
d. Testing & Development through black-box testing, production trials, and user training. The final stage of the research was the evaluation of the system using the PIECES framework to measure six dimensions of operational success (Performance, Information, Economy, Control, Efficiency, and Service).

B. Application Development Stage

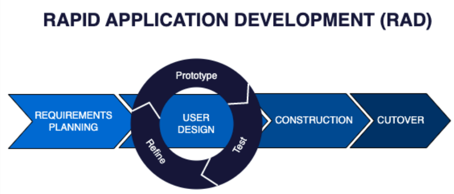

Figure 2. Rapid ApplicationDevelopment (RAD)

The RAD method was chosen because it emphasizes an adaptive, collaborative iterative cycle for software developers [9]. According to Hidayat & Hati [11] and Hariyanto et al. [8], the implementation of this model is highly relevant for industry-standard UI-driven web systems that require rapid alignment. The stages include:

a. Requirements Planning: Observations and interviews with engineers (including Field Mentors and Managers) to define the scope of the auto-conversion system.
b. Design Workshop: Creation of a visual logical design, including activity diagrams, database architecture, and web interfaces.
c. Implementation/Construction: Implementation ASP.NET code based on C#

and SQL Server to assemble web modules and automation algorithms.

d. Testing & Deployment: Deployment to the production environment via black-box testing and a pilot test.

C. System Overview

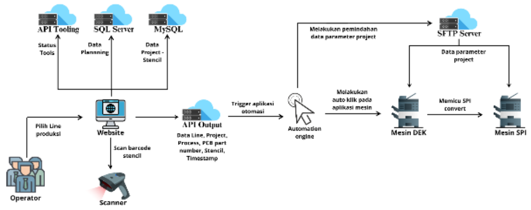

Figure 3. System Overview

The auto-convert system modernizes manual operational procedures. The system's workflow is designed so that when an operator scans a barcode (barcode stencil), the web application validates the project data against the SQL Server and MySQL databases. Once validated, the system sends an Application Programming Interface (API) output that instructs the Automation Engine to retrieve parameter files from the SFTP server to the local machine. Finally, the program executes automated click (auto-click) instructions on the DEK and SPI machine interface software in an integrated manner [13].

D. Functional Requirement

Table 1. Fungsional Requirement

ID	Fungsional Requirement
FR-01	Operators can select a production line from the main page.
FR-02	The system can retrieve planning data (Seq, Line, Work Order, Project, Process (TOP/BOT/COM), and Qty) from SQL Server.
FR-03	The system displays planning data (Seq, Line, Work Order, Project, Process, and Qty) on the main page.
FR-04	The system can retrieve tool status data from the tooling API.
FR-05	The system can display tool status data from the tooling API.
FR-06	Operators can scan a stencil to trigger the project conversion process.
FR-07	The system validates the stencil-project match between planning data (SQL Server) and stencil data (MySQL).
FR-08	The system generates API output as a trigger to run the automation engine.
FR-09	The automation engine retrieves project data from the SFTP server and transfers it to the local machine.
FR-10	The automation engine runs the auto-convert process and operates the DEK printing machine.
FR-11	The operator can run the SPI machine automation with a single action (button click).
FR-12	The automation engine runs the auto-convert process and activates the Solder Paste Inspection (SPI) machine after the DEK printing machine has started running.
FR-13	The system records activity logs for each machine automation process (DEK and SPI).

E. Non-Functional Requirement

Table 2. Non Fungsional Requirement

ID	Non Fungsional Requirement
NFR-01	System is web-based and can run on modern browsers without any additional installation.
NFR-02	System is capable of communicating with the company's API in real time.
NFR-03	System is integrated with an SQL Server database for storing validation data.
NFR-04	System is capable of controlling DEK and SPI printing machines via an automation engine with a 90% success rate.
NFR-05	Machine data must be uploaded to an SFTP server to ensure synchronization.
NFR-06	Auto-conversion process must be faster than the manual process.
NFR-07	A simple, user-friendly interface with a single main page.
NFR-08	The system must be stable.
NFR-09	The system interface (dashboard) uses a black background (dark mode) in accordance with visual ergonomics standards for production control rooms to reduce operator eye strain caused by prolonged exposure to monitor light.

F. Use case Diagram

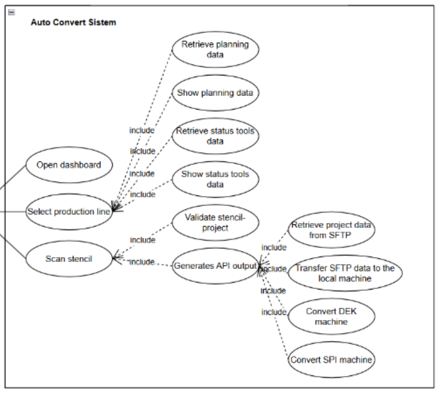

Figure 4. Use case diagram

The use case design centers on the “Operator” actor. The “Select Production Line” activity is linked to asynchronous data retrieval procedures (“Retrieve planning data,” “Show planning data”). The critical activity is “Scan Stencil,” where the system performs a series of validations that culminate in “Generate API Output.” This action triggers functions that are invisible to the user, such as downloading via SFTP, saving to local storage, and automatically converting DEK/SPI machine files.

G. ER Diagram

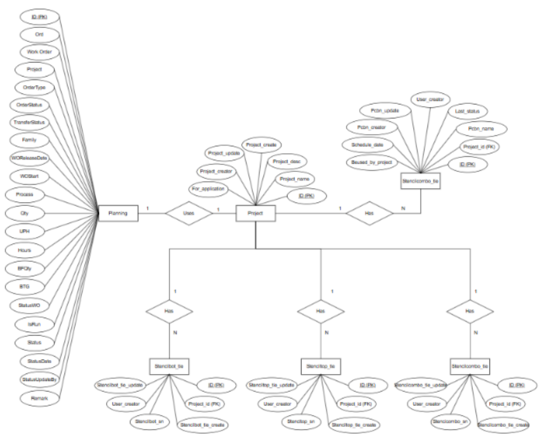

Figure 5. ER Diagram

The data structure design involves two databases. The Planning table in Microsoft SQL Server stores the line, job, and quantity entities. This data is logically related to the Project table in the MySQL system, which serves as the central entity. Derived entities such as Stencilbot_tie, Stenciltop_tie, Stencilcombo_tie, and PCBPN (component number) have one-to-many (1:N) and one-to-one (1:1) relationships that ensure all data is linked (tied) without redundancy [18].

H. Analysis Plan

This system analysis applies the PIECES Framework by Wetherbe [20] using a descriptive-comparative quantitative approach to evaluate operational performance before and after the implementation of an ASP.NET-based auto-convert system on DEK and SPI printing machines in the SMT production line. Referring to previous research [1], [12], [14], this framework is considered effective for an in-depth evaluation of information system development. All collected objective numerical data were validated against the six dimensions of PIECES, namely:

a. Performance to assess changeover duration, throughput, and API response time.

b. Information to measure data accuracy, completeness, and consistency using SSMS to synchronize planning data with actual machine data.

c. Economy to assess operational efficiency and reduction in operator working time through the Full-Time Equivalent (FTE) workload approach.

d. Control to evaluate the reliability of data validation and the reduction of error rates through digital audit trails from ASP.NET logging and SQL Server log files.

e. Efficiency to measure the reduction in manual work steps using a stopwatch.

f. Service to assess service quality, stability, and the ease of use of the application for operators through a Likert-scale questionnaire (1–5) administered directly in the production environment.

3. Result and Discussion

A. Implementation Result

1. Description of the System Implementation Process

The implementation of the Auto Convert SMT Program system in the production area is carried out through three main integrated stages. The installation stage involves installing the main application on production computers along with the SFTP data transfer automation application and click automation on the Decal Printing and Solder Paste Inspection (SPI) machines. This step is followed by the configuration stage, which includes setting up the SFTP connection, mapping the program file directories to local machines, configuring production lines (such as Line09A), adjusting automation coordinates, and synchronizing machine software versions. Finally, during the deployment phase, the system is operated directly by workers during the real-time program changeover process based on active Work Order data. The implementation of this automated workflow has proven successful in replacing manual methods, improving work efficiency, accelerating changeover times, and minimizing the risk of human error on the production line.

2. Visualization of System Design

Figure 6. Visualization System

The visualization of the system workflow begins when the operator scans the stencil barcode. The system will immediately validate whether the data matches the current project. If valid, a series of automated processes will be triggered sequentially: data output via API, retrieval of program files from the SFTP server to the machine's local storage, execution of automation on the DEK printing machine, and finally, forwarding the automation workflow to the SPI machine.

3. System Implementation Result

a. Dashboard Auto Convert

Figure 7. Dashboard

The Auto Convert SMT Program main page displays the company logo, application title, and real-time date and time at the top. For operations, the Select Line feature (e.g., Line09A) and the Scan Stencil feature are available to scan stencil barcodes, which are validated against the active project. This page also includes a Work Order (WO) table containing information on sequence, line, WO, process (TOP/BOT/COM), project, WO start, quantity, and tool status as a reference for changeovers. Monitoring is made easier with a color-coded legend for WO status: green for running, yellow for next running, and gray for pending.

b. Scan Successful

Figure 8. Scan Successful

If the scanned barcode stencil matches the current work order, the system displays a green "SCAN OK" success notification. After that, the input fields are automatically cleared for the next scan, and the system immediately executes the auto-conversion process to the machine automatically without manual operator intervention.

c. Scan Failed

Figure 9. Scan Failed

If the barcode stencil does not match the current work order, the system displays a red error notification stating "Stencil does not match." In this case, the conversion process is halted, and the operator must rescan using the correct stencil.

d. Output Website Data

Figure 10. Output Website Data

After a successful scan, the system generates validation data that includes production line information (e.g., Line09A), the SMT project name, process type (TOP/BOT/COM), PCB part number, stencil code, and execution timestamp. This data serves as a reference for subsequent processes as well as a system log.

e. Console Download Data from SFTP to Local Machine

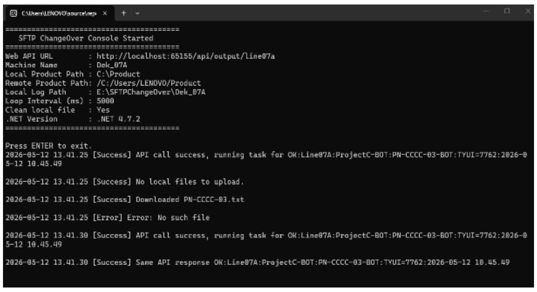

Figure 11. Console SFTP to Local Mesin

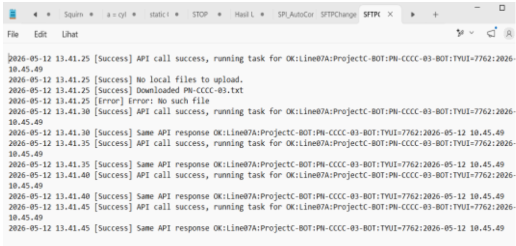

Figure 12. Log SFTP to Local Mesin

Immediately after the scan is validated, the system automatically initiates a connection to the SFTP server to download and save the SMT program file to the machine's local memory. All of these activities are automatically recorded in the system log for tracking and audit purposes.

f. Console Automation Printing DEK Machine

Once the program file is available in the machine's local storage, the system will initiate the automation process on the DEK printing machine.

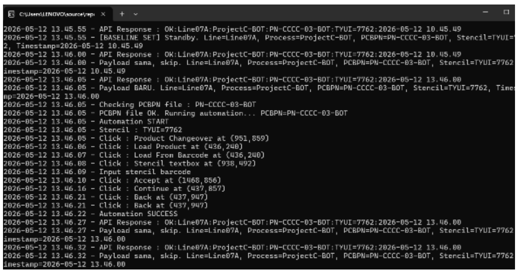

Figure 13. Console automation Printing DEK

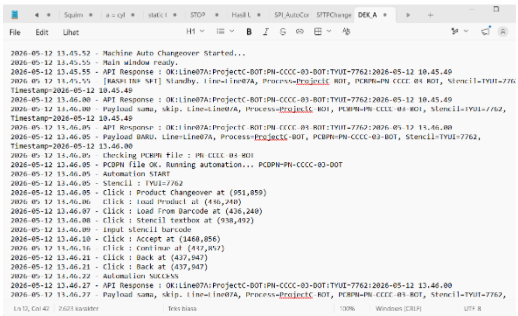

Figure 14. Log automation Printing DEK

Using an automation script, the system automatically selects and loads the appropriate program onto the DEK printing machine. This automation minimizes manual intervention to improve precision and reduce human error, with all activities recorded in the system log for auditing purposes.

g. Console Otomation SPI (Solder Paste Inspection) Machine

Once the printing process is complete, the system automatically moves the workflow to the Solder Paste Inspection (SPI) machine.

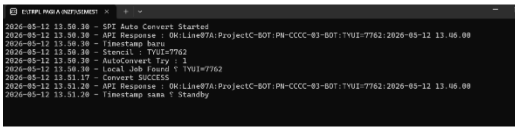

Figure 15. Console automation SPI

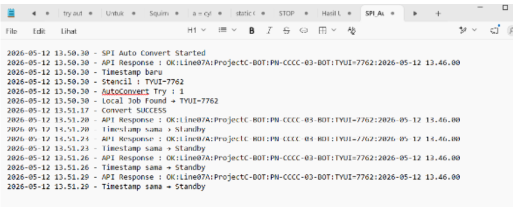

Figure 16. Log automation SPI

The system automatically selects and runs inspection programs on the SPI machine without manual input. This process is synchronized with the results from the previous printing run to ensure data consistency and SMT production quality. Every activity is automatically recorded in the system log.

B. Analysis Result

1. Performance

Performance testing was conducted through 10 trials to measure project conversion speed, throughput, and API response time. Process stability was analyzed using standard deviation as the primary quality control metric [10] and the coefficient of variation (CV) to assess the reliability of relative data dispersion [5].

Standard deviation formula:

$$\sigma = \sqrt{\frac{\sum(x_{i-\bar{x}})^2}{n-1}}$$

Formula for the coefficient of variation (CV):

$$CV = \frac{\text{Standard deviation}}{Average} \times 100\%$$

Table 3. Convert Project Time

Conditions	Total Time (Seconds)	Average (seconds)	deviation	CV
Before	4378	437,8	42,86	9,79
After	3025	302,5	5,99	1,98

Percentage Increase

$$\frac{437{,}8 - 302{,}5}{437{,}8} \times 100\% = 30{,}90\%$$

The system is able to speed up the project conversion process by 30.90% compared to before. Throughput refers to the number of project conversions that can be completed in one hour.

formula:

$$Throughput = \frac{3600\ second}{Processing\ time\ (second)}$$

Table 4. Thoughput

Conditions	Time (second)	Thoughput (process/hours)
Before	437,8	8,22
After	302,5	11,90

Persentage Inrease

$$\frac{11{,}90 - 8{,}22}{8{,}22} \times 100\% = 44{,}73\%$$

The system is able to speed up the project conversion process by 44.73%

Table 5. API Respon Time

Conditions	Average Respon Time	Total Tests
Before	Not available	Not available
After	51,38 ms	10

The API response output was tested using the curl command, resulting in an average response time of 51.38 ms. Overall, this demonstrates the success of the Auto Convert system in optimizing data access and accelerating workflows in the main phase of the SMT line changeover process.

2. Information

Evaluasi pada dimensi *Information* mengukur kualitas data yang mengalir dalam sistem berdasarkan tiga indikator utama: akurasi, kelengkapan, dan konsistensi data dengan memvalidasi parameter *line, work order* (WO), *project, PCB part number, stencil*, dan *timestamp*.

Formula:

$$Accuracy = \frac{Valid\ Data}{Total\ Tests} \times 100\%$$

Table 6. Accuracy Data

Conditions	Valid Data	Total Tests	Accuracy
Before	7	10	70
After	10	10	100

The data-matching accuracy rate prior to the system's implementation was only 70% due to reliance on operators' visual observations (human-based validation), which were prone to subjective errors. After the Auto Convert system was implemented, validation was automatically handled by the program (system-based validation) through an API output mechanism, resulting in a dramatic increase in data accuracy to 100%.

Formula:

$$Completeness = \frac{Valid\ Data}{Total\ Tests} \times 100\%$$

Table 7. Completeness Data

Conditions	Valid Data	Total Test	Completeness
Before	10	10	100
After	10	10	100

Data completeness tests conducted before and after the system's implementation showed a consistency rate that remained at 100%. This demonstrates that the system is capable of maintaining the completeness of all required parameter attributes in real time and reliably.

Rumus:

$$Consistency = \frac{Consistency\ Data}{Total\ Tests} \times 100\%$$

Table 8. Consistency Data

Conditions	Consistency Data	Total Tests	Consistency
Before	3	10	30
After	10	10	100

Data consistency improved from 30% to 100% after the system was implemented. This demonstrates that the system is able to ensure that all data is uniform and eliminate duplicates, making the information used more consistent and reliable.

3. Economy

The economic analysis evaluates savings in operator working time, workload, and their impact on the plant's operating costs.

Table 9. Operator work time per

Conditions	Cycle Time (mean)	Activity/ Month	Total Tim (second)	Total Time
Before	437,8	88	38526,4	10,70
After	302,5	88	26620	7,39

The system saves operators 3.31 hours of work time per month.

Workload is calculated using the Full-Time Equivalent (FTE) approach by comparing total working hours to the available effective working hours per month (where 1 FTE is equivalent to 173 hours), in accordance with Rifki's methodology [15].

Formula:

$$FTE = \frac{Total\ Working\ Hours}{Effective\ Working\ Hour\ per\ Month}$$

Table 10. Workload (FTE)

Conditions	Total Hour	FTE
Before	10,70	0,062
After	7,39	0,043

The system is capable of reducing the workload by 0.019 FTE.

An analysis of operating costs using a quantitative approach that represents FTE values as a measure of labor cost efficiency. Thus, a decrease in the FTE index directly indicates savings in operating costs through the optimization of full-time workforce utilization.

Table 11. Operating costs (FTE)

Conditions	FTE
Before	0,062
After	0,043

Operating costs were reduced by 0.019 FTE.

4. Control
The Control dimension analysis evaluates the system's ability to track operational activities, reduce the error rate, and ensure the security of production data.

Table 12. Total Log Activity

Conditions	Recorded activities	Total Activity
Before	Not available	Not available
After	10	10

Before the system was implemented, errors could occur due to the lack of automatic validation and the absence of systematic process documentation. After the system was implemented, errors were minimized thanks to system validation and the recording of errors in logs.
Formula:

$$Error\ Rate = \frac{Total\ Error}{Total\ Process} \times 100\%$$

Table 13. Error Rate

Conditions	Total Error	Total Process	Error Rate
Before	3	10	30
After	0	10	0

Before the system was implemented, errors could occur due to the lack of automatic validation and the absence of systematic process documentation. After the system was implemented, errors were minimized thanks to system validation and the recording of errors in logs.

5. Efficiency
Before the system was implemented, the project conversion process was performed manually and involved several steps, such as locating program files, manually entering data into the machine, and manual validation. After the Auto Convert SMT Program system was implemented, this process is now performed automatically by scanning the stencil barcode, which is directly integrated with the system.

Table 14. Work Step

Conditions	Work Step
Before	29 Step
After	8 Step

Effisiensi:

$$\frac{29 - 8}{29} \times 100\% = 72{,}41\%$$

The implementation of the SMT Program Auto Convert system has significantly streamlined the process. The elimination of 21 steps has resulted in a 72.41% improvement in workflow efficiency. This demonstrates that the system is capable of eliminating non-value-added activities, thereby making the process more streamlined, faster, and less prone to human error.

Table 15. Processing Time

Conditions	Processing Time (Seconds)
Before	437,8
After	302,5

Based on the analysis results, the number of work steps was reduced from 28 to 7. In addition, the project conversion time also decreased significantly, from 437.8 seconds to 302.5 seconds.

This demonstrates that the SMT Program Auto Convert system is capable of improving process efficiency by reducing manual tasks and minimizing the potential for human error in the Auto Convert process.

6. Service

Data collection was conducted using a Google Forms questionnaire with a Likert scale (1–5). The method used was total sampling, with a total of 4 respondents who were direct users of the system during the pilot phase.

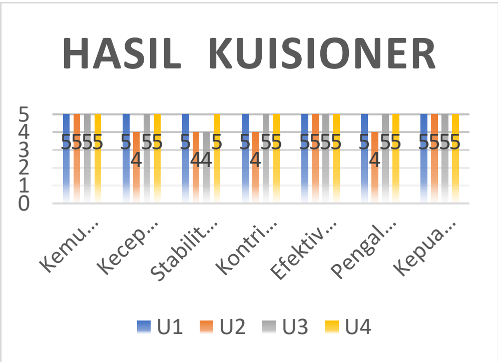

Figure 17. Survey Result

Table 16. Survey Results

Indicators	Mean
Kemudahan penggunaan sistem bagi operator	5
Kecepatan respon sistem (*Web & API Output*)	4,75
Stabilitas sistem selama proses produksi	4,5
Kontribusi sistem dalam mempermudah pekerjaan	4,75
Efektivitas sistem dalam mempercepat *changeover*	5
Pengalaman pengguna (*User Experience*)	4,75
Kepuasan keseluruhan terhadap sistem *Auto Convert*	5
Mean	4,82

The visualization shows that the rating trend is very positive and consistent. Although there were some indicators that did not reach a score of 5 from all respondents, not a single respondent gave a score below 4. This reinforces the argument that the Auto Convert system has met the performance standards expected by the Engineering team.

4. Conclusions

Based on the PIECES analysis and the implementation of the ASP.NET MVC-based Auto Convert SMT Program system using the RAD method, several conclusions were drawn:

a. System Integration: The application successfully automated the changeover process on the DEK NeoHorizon and SPI printing machines in sync via the automation engine.

b. Performance: Process duration was reduced by 30.90% (from 437.8 to 302.5 seconds) with very stable variability. Line throughput increased by 44.73% (from 8.22 to 11.90 processes/hour) with very fast API response times (51.38 ms).

c. Information: The transition to system-based validation boosted the accuracy of work plan data from 70% to an absolute 100%, and improved data consistency to 100%.

d. Economy: Saved 3.31 hours of operator work time per month and reduced the operational workload by 0.019 FTE (from 0.062 to 0.043 FTE).

e. Control: Reduced the operator error rate from 30% to 0% thanks to automated validation and periodic audit trail logging.

f. Efficiency: Simplifies work procedures from 29 manual steps to 8 systematic steps, with a workflow efficiency rate of 72.41%.

g. Service: Achieves an average user satisfaction score of 4.82 on a scale of 5.00, proving the system is highly reliable and easy to operate.

Although it is operating optimally, the current system is still limited to software-level automation, vendor-imposed configuration restrictions, the use of dummy test data, and the lack of self-service login authentication. To address these limitations, recommendations for further development include:

expanding the SMT line through system integration with other downstream machines such as Pick and Place, Reflow, and AOI so that automation can run fully along the production line.

a real-time monitoring dashboard to track error alarms and the development of auto-recovery scripts as a fail-safe mechanism in the event of a process failure.

system development can focus on integrating the Industrial Internet of Things (IoT) network to support the implementation of smart manufacturing, while also implementing modern login methods such as OAuth or JWT to strengthen access security controls for operators on the production floor.

References

[1] D. R. Adha, E. D. Wahyuni, and R. Hadiwiyanti, "Implementasi Analisis PIECES Untuk Pengembangan Sistem Informasi Task Management," Jurnal Informatika dan Teknik Elektro Kontemporer (JITET), vol. 13, no. 1, pp. 433–443, Jan. 2025, doi: 10.23960/jitet.v13i1.5726.

[2] A. D. I. Adji, A. Andriyanto, A. R. Syarifudin, A. Maulana, and R. Alit, "Redesain Website Pemerintah dengan Pengembangan Menggunakan Kerangka Kerja ASP.NET dan fitur Razor Pages," Jurnal Ilmu Teknik, vol. 1, no. 2, pp. 50–63, 2023.

[3] ArtPictures, "Autumn 2023 – ArtPictures," 2023. [Online]. Available: [tautan mencurigakan telah dihapus]. [Accessed: Oct. 18, 2025].

[4] N. Azis, M. S. Hartawan, and S. Amelia, "Rancang Bangun Otomatisasi Penyiraman dan Monitoring Tanaman Kangkung," Jurnal IKRA-ITH Informatika, vol. 4, no. 3, pp. 95–102, Nov. 2020.

[5] B. S. Everitt and A. Skrondal, The Cambridge Dictionary of Statistics, 4th ed. Cambridge, UK: Cambridge University Press, 2010.

[6] S. S. Chinthalapudi, "Enhancing Security in ASP.NET Core Applications: Implementing Oauth, JWT, and Zero-Trust Models," International Journal of Innovative Science and Research Technology, vol. 10, no. 3, pp. 2561–2575, Mar. 2025, doi: 10.38124/ijisrt/25mar1677.

[7] Deepika C, Khamar Taj, and Parashuram Bedar, "Automation in Production Systems: Enhancing Efficiency and Reducing Costs in Mechanical Engineering," Nanotechnology Perceptions, vol. 5, no. 5, pp. 1436–1447, 2024, doi: 10.62441/nano-ntp.vi.3895.

[8] D. Hariyanto, R. Sastra, and F. E. Putri, "Implementasi Metode Rapid Application Pada Sistem Informasi Perpustakan," Jurnal Penelitian Ilmu Dan Teknologi Komputer, vol. 13, no. 1, pp. 3–10, 2021.

[9] J. Martin, Rapid Application Development. New York, USA: Macmillan Publishing, 1991.

[10] D. C. Montgomery, Introduction to Statistical Quality Control, 8th ed. New York, USA: John Wiley & Sons, 2020.

[11] N. Hidayat and K. Hati, "Penerapan Metode Rapid Application Development (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE)," Jurnal Sistem Informasi, vol. 10, no. 1, pp. 8–17, 2021, doi: 10.51998/jsi.v10i1.352.

[12] I. Oktaviani and S. Sumarlinda, "Penerapan Metode PIECES pada Analisis Sistem Informasi Manajemen Apotek," Infokes: Jurnal Ilmiah Rekam Medis Dan Informatika, vol. 11, no. 1, pp. 54–58, 2021.

[13] M. O. Onibonoje, O. O. Alegbeleye, and A. O. Ojo, "Control Design and Management of a Distributed Energy Resources System," International Journal of Technology, vol. 14, no. 2, pp. 236–245, 2023, doi: 10.14716/ijtech.v14i2.5884.

[14] A. Rahmi, S. Lastri, and H. Hasnur, "Pieces (Performance, Information, Economic, Control, Efficiency, Service) Dengan Pemanfaatan Sistem Informasi Manajemen Puskesmas (Simpus)," Jambura Journal of Health Sciences and Research, vol. 6, no. 2, pp. 146–154, 2024, doi: 10.35971/jjhsr.v6i2.23969.

[15] M. H. A. R. Rifki, "Analisis Beban Kerja Dengan Metode FTE (Full Time Equivalent) Sebagai Dasar Perencanaan Kebutuhan Tenaga Kerja Pada PT XYZ," Jurnal Industri, pp. 433–443, 2024.

[16] H. Sheth, "The Impact of Automation on Business Process Efficiency and Accuracy: Enhancing Operational Performance in the Digital Age," Institute of Radio Engineers Journal, vol. 4, no. 12, pp. 317–321, 2021.

[17] M. Simar Kurniawan, K. Wijaya, and H. Hardi Purba, "Implementasi Sistem Kendali Otomatis Pada Industri Manufaktur: Kajian Literatur," Meruya Selatan, vol. 11, no. 1, p. 3, 2025.

[18] P. Sistem, I. Persediaan, B. Web, and D. Analisa, "Perancangan Sistem Informasi Persediaan Barang Berbasis Web Dengan Analisa Pieces," Swadharma (jris), vol. 2, no. 1, pp. 25–31, 2022.

[19] M. A. Talekattu, "Design of Web Applications in ASP.NET," International Journal for Research in Applied Science and Engineering Technology, vol.

12, no. 6, pp. 536–541, Jun. 2024, doi: 10.22214/ijraset.2024.63128.

[20] J. C. Wetherbe, Systems Analysis and Design: Traditional, Structured, and Object-Oriented Approaches. Boston, USA: West Publishing Company, 1994.

[21] W. Williem and A. L. Adila, "Local Web Based Internal Corporate Meals Request System With Email Approval Using ASP.Net MVC," IT for Society, vol. 8, no. 1, pp. 28–33, 2023, doi: 10.33021/itfs.v8i1.4727.