



# SNESTIK

Seminar Nasional Teknik Elektro, Sistem Informasi,  
dan Teknik Informatika

<https://ejurnal.itats.ac.id/snestik> dan <https://snestik.itats.ac.id>



## Informasi Pelaksanaan :

SNESTIK I - Surabaya, 26 Juni 2021

Ruang Seminar Gedung A, Kampus Institut Teknologi Adhi Tama Surabaya

## Informasi Artikel:

DOI : .....

Prosiding ISSN 2775-5126

Fakultas Teknik Elektro dan Teknologi Informasi-Institut Teknologi Adhi Tama Surabaya  
Gedung A-ITATS, Jl. Arief Rachman Hakim 100 Surabaya 60117 Telp. (031) 5945043  
Email : [snestik@itats.ac.id](mailto:snestik@itats.ac.id)

## Rancang Bangun Aplikasi Untuk Penyedia Kursus Mengemudi Berbasis Web Dengan Incremental Model

Yodanis E. Sutantio<sup>1</sup> dan Rachman Arief<sup>2</sup>

Program Studi Sistem Informasi Institut Teknologi Adhi Tama Surabaya<sup>1 2</sup>  
e-mail: [yodanise66@gmail.com](mailto:yodanise66@gmail.com)

### ABSTRACT

*The rising number of four-wheeled vehicle owners has prompted authorities, including the Department of Transportation and the National Police of the Republic of Indonesia, to encourage new drivers to attend driving courses before operating a vehicle. In Surabaya, the number of driving course providers has grown to several hundred. The increase in new drivers and heightened competition among service providers underscores the need for an information system to address emerging challenges. This research focuses on developing an application through software engineering to support all parties involved in the driving course process, including students, instructors, and course administrators or owners. The software development process follows the Incremental Model, which allows the development team to address the specific needs of each user group effectively. The iterative nature of the Incremental Model ensures the creation of a high-quality application designed to overcome the challenges faced by stakeholders within the driving course ecosystem.*

**Keywords:** driving course, information system, software engineering, incremental model

### ABSTRAK

Seiring meningkatnya jumlah pemilik kendaraan bermotor roda empat, pihak berwenang, dalam hal ini, Dinas Perhubungan dan Satuan Kepolisian Republik Indonesia menghimbau para pengemudi baru untuk mengikuti Kursus Mengemudi sebelum mulai mengendarai mobil. Jumlah pengelola kursus mengemudi di Surabaya saat ini sudah menyentuh angka ratusan, dengan semakin banyaknya pengemudi-pengemudi baru dan persaingan antar penyedia jasa kursus mengemudi seperti saat ini, dibutuhkan sebuah Sistem Informasi untuk mengatasi potensi masalah tersebut. Dengan melakukan Rekayasa Perangkat Lunak, diharapkan penelitian ini akan menghasilkan aplikasi yang dapat membantu semua pihak yang terlibat dalam proses

kursus mengemudi, baik bagi Siswa / Pelajar, Instruktur Kursus, dan Pemilik / Admin Kursus. Pengembangan Perangkat Lunak menggunakan Incremental Model memberikan tim pengembang kemampuan untuk berfokus terhadap kebutuhan setiap pengguna yang sudah disebutkan sebelumnya. Karakteristik iteratif yang dimiliki incremental model diyakini akan menghasilkan aplikasi berkualitas dan mampu menyelesaikan masalah yang dihadapi oleh pihak-pihak dalam rangkaian proses kursus mengemudi.

**Kata kunci:** Kursus Mengemudi, Sistem Informasi, Rekayasa Perangkat Lunak, Incremental Model

## PENDAHULUAN

Saat ini, banyak penyedia jasa yang dapat membantu kita memahami teori dan mengajarkan kita kemampuan untuk berkendara dengan baik melalui kursus mengemudi. Sering sekali kita menemui pengemudi yang sedang belajar di jalan raya yang biasanya mereka didampingi dengan mentor/instruktur untuk memandu dan memberikan arahan. Sebab, melalui data yang dibagikan oleh Korps Lalu Lintas Kepolisian Republik Indonesia, pada Januari tahun 2022 hingga September 2022 terjadi setidaknya 94.617 kasus kecelakaan lalu lintas di seluruh wilayah Indonesia [1]. Lebih lanjut lagi Korlantas Polri menjelaskan faktor-faktor yang memiliki andil pada 94.617 kasus kecelakaan tersebut, faktor-faktor tersebut diantaranya 61% kasus kecelakaan diakibatkan oleh faktor kesalahan manusia [2] seperti kurang terampilnya pengemudi dalam mengendalikan kendaraannya, serta kelalaian-kelalaian yang lainnya seperti mengemudi dalam kondisi mengantuk, ugal-ugalan, dan lain lain.

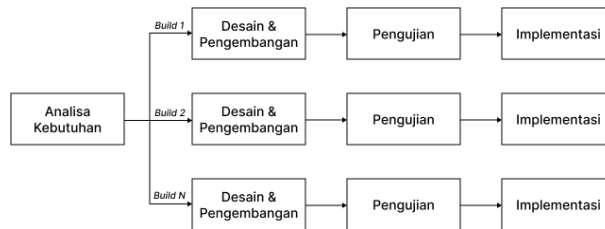
Singkatnya, kami ingin masyarakat memahami bahaya dan risiko yang ditemui apabila mereka mengemudikan kendaraan tanpa mengikuti suatu pelatihan untuk mengasah pemahaman dan kemampuannya terlebih dahulu tentang topik tersebut. Kami ingin menjadi wadah bagi penyedia jasa kursus mengemudi untuk memberikan edukasi ke masyarakat luas bahwa mengikuti kursus mengemudi tidak memerlukan biaya yang tinggi, dan meskipun biaya yang dibutuhkan masih dirasa cukup tinggi, setidaknya mereka mengerti betapa pentingnya untuk mengikuti kursus mengemudi sebelum berkendara langsung di jalan raya. Dengan Rekayasa Perangkat Lunak, kami ingin mengurangi beban dari penyedia kursus mengemudi konvensional yang masih kebingungan dalam melakukan kegiatan promosi untuk layanan mereka, dan kebutuhan untuk harus mencetak dokumen-dokumen administrasi sehingga semuanya dapat terintegrasi di satu aplikasi, dimana hasil akhir yang ingin kami capai adalah peningkatan layanan penyedia kursus mengemudi yang lebih baik dari sebelumnya.

## METODE

Pada penelitian ini, kami menerapkan tiga metode pengumpulan data yang dipilih berdasarkan tingkat efektivitasnya. Berikut adalah metode-metode tersebut:

(1) Observasi, dilakukan secara sistematis dan mencatat semua aspek penting yang secara jelas terlihat selama proses berlangsungnya kursus mengemudi; (2) Studi Literatur, dilakukan sebagai langkah awal dalam memahami aspek-aspek penting yang muncul selama proses observasi, yang kemudian akan dikonfirmasi lebih lanjut melalui wawancara; (3) Wawancara, dilakukan untuk memperoleh informasi dari pihak yang terlibat secara langsung pada proses kursus mengemudi [3].

Dengan data yang diperoleh, proses penelitian berlanjut ke tahap pengembangan aplikasi yang dikembangkan menggunakan *incremental model*. Model pengembangan perangkat lunak tersebut dikenal karena proses iteratifnya, dimana setiap iterasi yang dilakukan selanjutnya dikembangkan satu per satu. Dengan membagi proyek menjadi bagian-bagian kecil, peneliti mampu memprioritaskan fitur dan mengadaptasi perubahan-perubahan kecil dengan lebih efisien. Gambar 1 dibawah ini adalah siklus pengembangan aplikasi menggunakan *incremental model*.



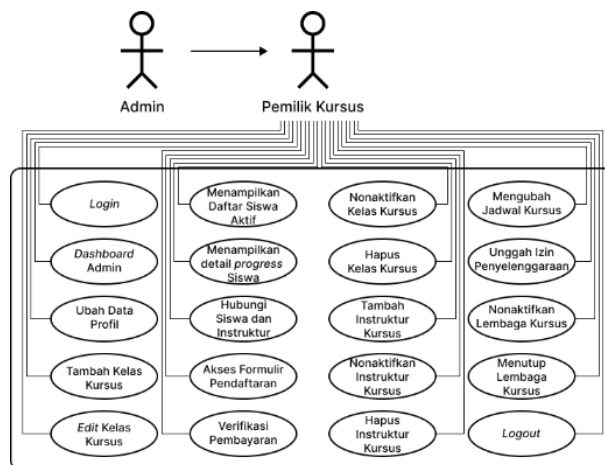
Gambar 1. Siklus Pengembangan Perangkat Lunak *Incremental Model*

Data yang diperoleh di awal penelitian digunakan untuk merancang kebutuhan aplikasi pada tahapan Analisa Kebutuhan, dimana pada tahap tersebut, peneliti menetapkan kebutuhan yang akan dikembangkan pada *build 1*, *build 2*, dan seterusnya. Tahapan Desain & Pengembangan ditujukan untuk merancang alur kerja fitur-fitur aplikasi, tampilan antarmuka, serta skema dan relasi *database*. Tahapan Pengujian dilakukan untuk memastikan baik kualitas dan fungsionalitas aplikasi telah optimal. Sementara itu, tahapan Implementasi pada siklus pengembangan *incremental model* melibatkan penggabungan aplikasi dari *build* satu dengan *build* yang lainnya. Dengan demikian, fungsi-fungsi yang ada dapat dioperasikan satu sama lain, tanpa menimbulkan gangguan [4].

## HASIL DAN PEMBAHASAN

### Pengembangan Aplikasi *Build 1* – Pemilik / Admin Kursus

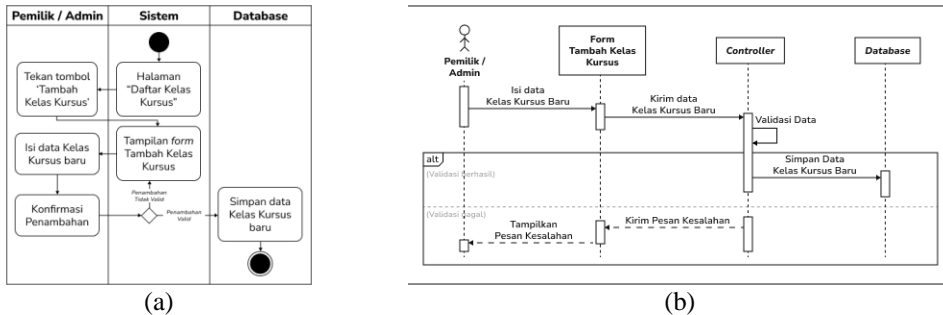
Sebagaimana yang sudah dijelaskan pada Gambar 1 mengenai siklus pengembangan perangkat lunak dengan menggunakan *Incremental Model*, tahap pengembangan untuk *build 1* diawali dengan proses desain dan pengembangan. Desain yang dimaksud meliputi desain arsitektur atau struktur perangkat lunak, desain interaksi perangkat lunak, serta desain tampilan perangkat lunak. Dalam penelitian ini, kami menerapkan *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram* sebagai metode dokumentasi perancangan perangkat lunak. Gambar 2 dibawah ini adalah *use case diagram* untuk user Pemilik / Admin Kursus.



Gambar 2. *Use Case Diagram* untuk Pemilik / Admin Kursus

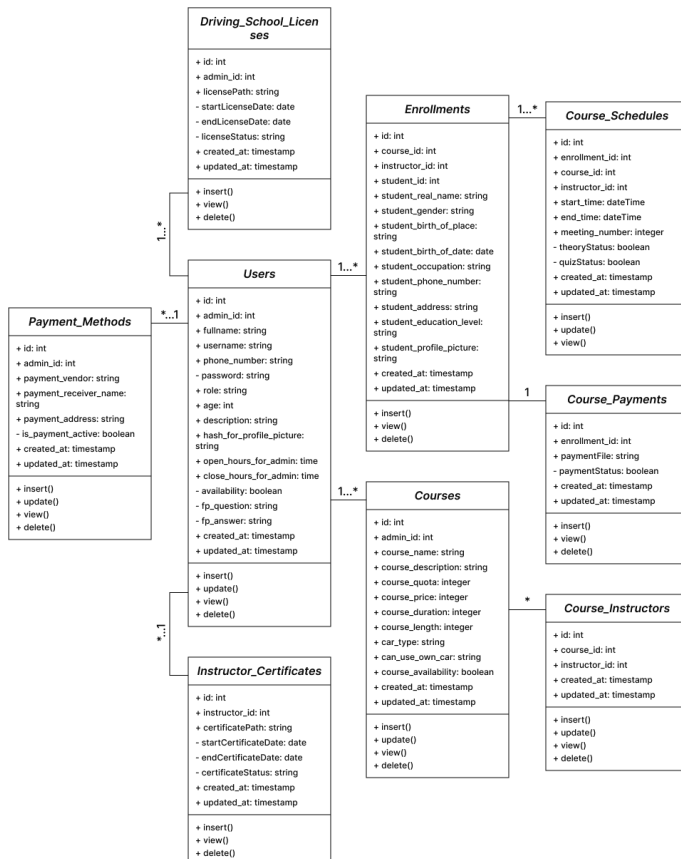
Dari masing-masing *use case* diatas, selanjutnya akan diuraikan menjadi dua diagram tambahan untuk memahami alur aktifitas [5] dan alur data yang terjadi saat sebuah proses

dieksekusi [6]. Sebagai contoh, pada Gambar 3 disajikan *Activity Diagram* dan *Sequence Diagram* yang menggambarkan proses Tambah Kelas Kursus.



Gambar 3. a) *Activity Diagram* untuk proses Tambah Kelas Kursus, b) *Sequence Diagram* untuk proses Tambah Kelas Kursus

Sebelum memulai tahapan pengembangan perangkat lunak, tim pengembang terlebih dahulu menyusun sebuah *Class Diagram* yang berfungsi sebagai pedoman dalam merancang skema *database*. Karakteristik *Class Diagram* yang mencerminkan elemen-elemen dalam skema *database* menjadikannya sangat relevan untuk memastikan konsistensi antara perancangan dan implementasi sistem [5]. Untuk lebih jelasnya Gambar 4 berikut menyajikan *Class Diagram* yang dirancang untuk kebutuhan *build 1*.



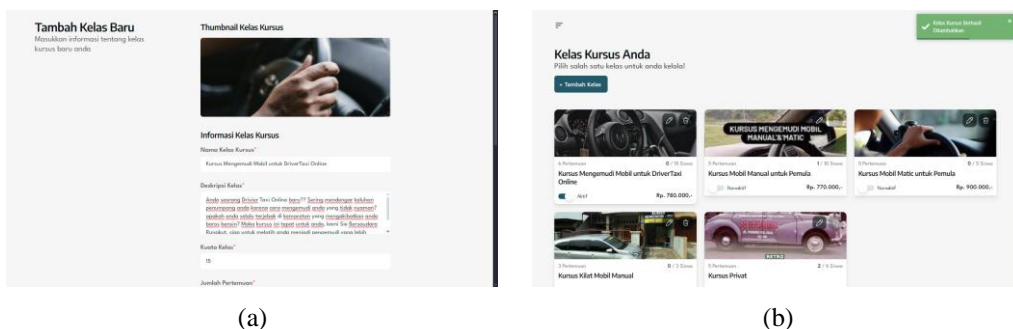
Gambar 4. *Class Diagram* untuk *Build 1* - Pemilik / Admin Kursus

Setelah tahapan desain dan pengembangan selesai, langkah berikutnya adalah proses pengujian. Metode pengujian perangkat lunak yang diterapkan adalah *Black Box Testing*, yang bertujuan untuk memastikan bahwa sistem yang dikembangkan telah berfungsi dengan benar dan sesuai dengan spesifikasi yang diharapkan [7]. Berikut ini disajikan Tabel 1 contoh skenario dan hasil *Black Box Testing* untuk proses Tambah Kelas Kursus.

Tabel 1. Skenario dan hasil *Black Box Testing Build 1* untuk proses Tambah Kelas Kursus

ID	Skenario Pengujian	Hasil yang diharapkan	Tingkat Pengujian	Hasil Pengujian
...	...	...	...	...
B1-T39	Memasukkan harga di bawah Rp. 1.000,- di kolom “Harga Kelas” kemudian menekan tombol “Tambah”	Sistem menampilkan pesan kesalahan kepada pengguna bahwa harga kursus terlalu kecil	Minor	Valid
B1-T40	Mengisi semua kolom data dengan sesuai dan benar kemudian menekan tombol “Tambah”	Sistem menyimpan data kelas baru ke database dan mengarahkan pengguna ke daftar kelas kursus untuk menampilkan kelas kursus yang baru ditambahkan ada di urutan pertama	Major	Valid
...	...	...	...	...

Setelah tahapan pengujian selesai, langkah selanjutnya adalah proses implementasi. Tahap ini bertujuan untuk memastikan bahwa perangkat lunak yang dikembangkan di *build* pertama dapat berintegrasi dan berfungsi dengan *build* berikutnya tanpa menimbulkan kesalahan atau *bug* saat digunakan. Apabila selama proses implementasi ditemukan kesalahan atau *bug*, tim pengembang akan segera melakukan perbaikan agar masalah yang ditemukan pada satu *build* tidak menyebabkan kesalahan yang sama pada *build* selanjutnya. Gambar 5 dibawah ini adalah hasil implementasi dari *build 1* proses Tambah Kelas Kursus.



Gambar 5. a) Pemilik / Admin Kursus mengisi informasi Kelas Kursus Baru, b) Kelas Kursus baru berhasil di tambahkan

Setelah semua *build* berhasil dikembangkan, selanjutnya adalah melakukan penjaminan kualitas perangkat lunak dengan standar ISO 25010:2023. Tabel 2 dibawah ini adalah hasil pengukuran kualitas perangkat lunak berdasarkan standar ISO 25010:2023 untuk faktor

*functional suitability* yang ditujukan untuk memastikan apakah fitur-fitur yang dikembangkan sudah memenuhi kebutuhan pengguna dalam melakukan proses kursus mengemudi [8].

Tabel 2. Hasil Pengukuran Kualitas Perangkat Lunak dengan ISO 25010:2023 untuk faktor *functional suitability*

Sub-Faktor	Pertanyaan	Indeks (%)
<i>Functional Completeness</i>	Apakah KEMUDI menyediakan semua fungsi yang dibutuhkan oleh Anda untuk mengelola, mendaftar atau melakukan proses kursus mengemudi yang diinginkan?	90%
<i>Functional Correctness</i>	Apakah KEMUDI memberikan hasil yang benar dan konsisten sesuai dengan harapan Anda?	92%
<i>Functional Appropriateness</i>	Apakah fungsi-fungsi yang disediakan oleh KEMUDI membantu Anda menyelesaikan mengelola, mendaftar atau melakukan proses kursus mengemudi dengan efisien?	92%

Berdasarkan hasil pengujian yang telah dilakukan, diperoleh rata-rata skor sebesar 91,3%. Skor ini berada dalam rentang 81 – 100%, yang mengindikasikan bahwa perangkat lunak telah dinilai **Sangat Tepat** dalam menyediakan fitur-fitur yang mendukung proses kursus mengemudi. Selanjutnya, Tabel 3 dibawah ini memaparkan hasil pengujian kualitas perangkat lunak berdasarkan standar ISO 25010:2023 untuk faktor *interaction capability* yang bertujuan untuk memastikan bahwa perangkat lunak yang sudah dirancang memiliki antarmuka yang ramah pengguna dan mudah digunakan.

Tabel 3. Hasil Pengukuran Kualitas Perangkat Lunak dengan ISO 25010:2023 untuk faktor *interaction capability*

Sub-Faktor	Pertanyaan	Indeks (%)
<i>Appropriateness Recognizability</i>	Apakah Anda setuju bahwa KEMUDI membantu anda dalam mengelola, mendaftar atau melakukan kursus mengemudi?	94%
<i>Learnability</i>	Apakah Anda merasa bahwa Anda membutuhkan waktu untuk membiasakan diri dengan KEMUDI?	66%
<i>Operability</i>	Apakah Anda dapat mengoperasikan KEMUDI dengan mudah dan tanpa kesulitan?	86%
<i>User Error Protection</i>	Apakah KEMUDI membantu Anda mencegah kesalahan dan memberikan bantuan saat terjadi kesalahan?	84%
<i>User Engagement</i>	Apakah KEMUDI berhasil membuat Anda ingin kembali menggunakannya secara terus menerus?	82%
<i>Inclusivity</i>	Seberapa mudahkah Anda bernavigasi dan memahami berbagai fitur yang ditawarkan oleh KEMUDI, terlepas dari keahlian teknis Anda?	86%
<i>User Assistance</i>	Seberapa puaskah Anda dengan ketersediaan dan kualitas bantuan (tutorial, pertanyaan-pertanyaan umum tentang aplikasi, dll.) yang diberikan oleh KEMUDI?	90%
<i>Self-Descriptiveness</i>	Apakah label-label, petunjuk, instruksi, dan pesan kesalahan yang ditampilkan oleh KEMUDI sudah jelas dan mudah dipahami?	86%

Dari hasil pengujian perangkat lunak untuk faktor *interaction capability*, diperoleh rata-rata skor sebesar 84,2%. Skor ini berada dalam rentang 81 – 100%, yang menunjukkan bahwa perangkat lunak memiliki kemampuan interaksi yang sangat baik. Kemampuan ini mencakup kemudahan penggunaan, perlindungan terhadap kesalahan pengguna, serta penyediaan fitur-fitur bantuan seperti tutorial dan *Frequently Asked Questions* (FAQ). Berdasarkan hasil tersebut, perangkat lunak dinilai **Sangat Mudah** digunakan oleh pengguna.

## KESIMPULAN

Dengan menerapkan pengembangan perangkat lunak menggunakan *Incremental Model*, tim pengembang dapat lebih fokus menyelesaikan permasalahan yang dihadapi pengguna pada setiap *increment* atau *build*. Peneliti berharap, dengan kemajuan teknologi *Artificial Intelligence* (AI), beberapa proses seperti pemilihan jadwal kursus, pembelajaran teori kursus, dan berbagai fungsionalitas lainnya dapat diintegrasikan dengan AI. Dengan demikian, penjadwalan kursus dapat dilakukan secara otomatis, dan pemahaman siswa kursus terhadap teori berkendara dapat ditingkatkan dengan signifikan.

## DAFTAR PUSTAKA

- [1] Kementerian Perhubungan Republik Indonesia, “Tekan angka kecelakaan lalu lintas, Kemenhub ajak masyarakat beralih ke transportasi umum dan utamakan keselamatan berkendara,” [dephub.go.id](https://dephub.go.id), 19-Sep-2023. [Online]. Available: <https://dephub.go.id/post/read/%E2%80%8Btekan-angka-kecelakaan-lalu-lintas,-kemenhub-ajak-masyarakat-beralih-ke-transportasi-umum-dan-utamakan-keselamatan-berkendara>. [Diakses: 08-Feb-2024]
- [2] Kementerian Komunikasi dan Informatika, “Rata-rata tiga orang meninggal setiap jam akibat kecelakaan jalan,” [komdigi.go.id](https://komdigi.go.id), 22-Agu-2017. [Online]. Available: <https://www.komdigi.go.id/berita/artikel/detail/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan>. [Diakses: 14-Jan-2024]
- [3] F. Adhiva Kurnia, “Driving course and driving license service information system web-based (study case Kurnia Jaya),” *Journal of Computer Science and Big Data*, vol. 1, no. 1, 2023. [Online]. Available: <http://jcosbida.com/index.php/index/login>. [Diakses: 20-Feb-2024].
- [4] D. Sachan, “Incremental model in software engineering,” *Scaler*, Apr. 4, 2024. [Online]. Available: <https://www.scaler.com/topics/incremental-model-in-software-engineering/>. [Diakses: 6-Jun-2024].
- [5] L. Jacobson and J. R. G. Booch, *The Unified Modeling Language Reference Manual*, 2nd ed., 2021.
- [6] L. Letaw, “Unified modeling language class and sequence diagrams,” in *Handbook of Software Engineering Methods*. [Online]. Available: <http://debracollege.dspaces.org/bitstream/123456789/404/1/UML%20Reference%20Manual%20by%20James%20Rambaugh.pdf>. [Accessed: 4-Apr-2024].
- [7] R. Setiawan, “Black box testing untuk menguji perangkat lunak,” *Dicoding*, 2021. [Online]. Available: <https://www.dicoding.com/blog/black-box-testing/>. [Accessed: Mar. 29, 2024].
- [8] International Organization of Standardization, *ISO/IEC 25010:2023(en)*. 2023. [Online]. Available: <https://www.iso.org/obp/ui/en/#iso:std:iso:iec:25010:ed-2:v1:en>. [Accessed: 16-Jul-2024].