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 V - Surabaya, 26 April 2025
Fakultas Teknik Elektro dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya

Informasi Artikel:
DOI : 10.31284/p.snestik.2025.6998

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

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

Yodanis E. Sutantio, Rachman Arief, dan Alfin Sugestian

Program Studi Sistem Informasi, Institut Teknologi Adhi Tama Surabaya
e-mail: 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. The testing results based on the ISO 25010:2023 standard indicate that the software achieved a score of 91.3% for the functional suitability, categorized as Highly Accurate. Furthermore, for the interaction capability, the software scored 84.2%, classified as Very Easy to use.

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. Hasil pengujian berdasarkan standar ISO 25010:2023 menunjukkan bahwa perangkat lunak memperoleh skor sebesar 91,3% untuk faktor ketepatan fungsional, yang dikategorikan sebagai Sangat Tepat. Selain itu, untuk faktor kapabilitas interaksi, perangkat lunak mendapatkan skor sebesar 84,2%, yang diklasifikasikan sebagai Sangat Mudah digunakan.

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

PENDAHULUAN

Tingginya angka kecelakaan lalu lintas di Indonesia, yang mencapai 94.617 kasus dalam sembilan bulan pertama tahun 2022 [1], menunjukkan pentingnya pelatihan mengemudi sebelum seseorang turun ke jalan. Sebanyak 61% kecelakaan disebabkan oleh faktor manusia, termasuk kurangnya keterampilan dan perilaku berkendara yang lalai [2]. Untuk menjawab permasalahan ini, penelitian ini menghadirkan solusi berupa aplikasi web yang dirancang untuk membantu penyedia kursus mengemudi dalam mengelola administrasi, peserta, dan edukasi secara terpadu. Diharapkan, sistem ini dapat meningkatkan efektivitas layanan dan turut menurunkan angka kecelakaan di masa mendatang.

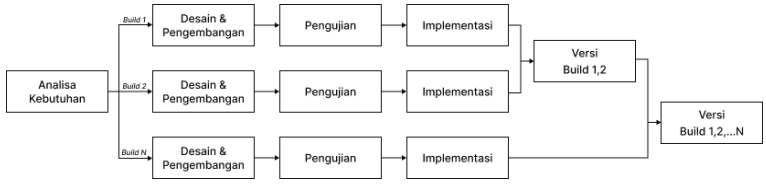
Penelitian yang dilakukan oleh Kurnia et al. (2023) menekankan bahwa aplikasi berbasis web mampu meningkatkan fleksibilitas dan efisiensi dalam proses registrasi siswa kursus mengemudi [3]. Sementara itu, Elianti et al. (2022) mengembangkan aplikasi Android yang membantu menyederhanakan manajemen bisnis kursus, mencakup fitur pendaftaran, penjadwalan, dan pelaporan siswa [4]. Terakhir, penelitian Wibowo et al. (2024) menunjukkan bahwa platform berbasis web tidak hanya berguna bagi siswa, tetapi juga meningkatkan keterlibatan dan produktivitas instruktur dan pemilik kursus, dengan respons positif dari seluruh pihak [5].

Berbeda dari penelitian sebelumnya yang terbatas pada satu lembaga, aplikasi ini dirancang untuk digunakan oleh seluruh penyedia kursus mengemudi di Surabaya. Penelitian ini juga membuktikan bahwa model incremental lebih efektif untuk pengembangan perangkat lunak berskala besar dibandingkan model konvensional seperti waterfall. Dengan pendekatan ini, sistem dibangun dan diuji secara bertahap, sehingga lebih fleksibel dan mudah disesuaikan.

METODE

Incremental Model

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 awal digunakan dalam tahap Analisa Kebutuhan untuk menentukan fitur yang dikembangkan pada masing-masing *build*. Tahap Desain & Pengembangan mencakup perancangan alur kerja, antarmuka, serta struktur database. Pengujian dilakukan untuk memastikan kualitas dan fungsionalitas aplikasi. Pada tahap Implementasi, setiap build digabungkan secara bertahap sesuai model incremental, sehingga seluruh fitur dapat saling terintegrasi tanpa gangguan [6].

Pengembangan Perangkat Lunak Berbasis Web

Aplikasi ini dikembangkan menggunakan Laravel sebagai *framework backend* karena mendukung arsitektur MVC dan menyediakan fitur seperti *routing*, *middleware*, serta *Eloquent* ORM untuk mempermudah pengelolaan data dan struktur aplikasi [11]. MySQL digunakan sebagai DBMS karena kestabilan, performa tinggi, dan dukungan komunitasnya yang luas [12]. Di sisi *frontend*, kami menggunakan Tailwind CSS, *framework utility-first* yang memungkinkan pengembangan UI yang cepat, responsif, dan konsisten. Kombinasi teknologi ini dipilih untuk mempercepat pengembangan sekaligus menjaga kualitas, skalabilitas, dan kemudahan pemeliharaan aplikasi.

Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan dengan dua metode utama, yaitu *Black Box Testing* dan evaluasi berdasarkan standar ISO 25010:2023. Pada *Black Box Testing*, tim pengembang membuat skenario pengujian untuk setiap *build*. Setiap skenario diuji satu kali pada tahap pengujian setiap *build* dan setelah integrasi antar *build*; jika pengujian berhasil, maka tidak perlu diuji ulang. Namun, jika ditemukan *bug*, skenario tersebut akan diuji kembali setelah perbaikan hingga hasilnya sesuai harapan [9].

Setelah seluruh *build* selesai dan sistem telah terintegrasi, evaluasi dilakukan menggunakan standar ISO 25010:2023 [10] dengan pendekatan kuisioner menggunakan skala Likert. Responden yang merupakan gabungan antara Admin Kursus, Instruktur Kursus, Calon Siswa, Alumni, dan pengguna umum akan diberikan total 3 pertanyaan terkait faktor *functional suitability* dan 8 pertanyaan terkait faktor *interaction capability*. Setiap pertanyaan mewakili satu sub-karakteristik kualitas perangkat lunak.

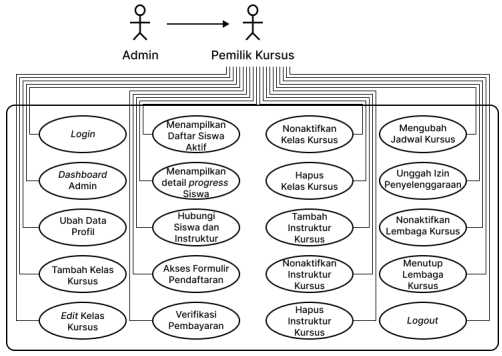
Nilai akhir dihitung berdasarkan total skor aktual yang diperoleh dari jawaban responden (misalnya 3x5 + 1x4 + 0x3 + 1x2 + 0x1 = 21) dibagi dengan skor maksimal (misalnya 5 responden × nilai maksimum 5 = 25), kemudian dikalikan 100 untuk mendapatkan skor dalam bentuk persentase, dimana untuk contoh tersebut skor yang diperoleh adalah 84%. Proses ini memberikan gambaran kuantitatif terhadap kualitas aplikasi berdasarkan persepsi pengguna akhir.

Untuk mendapatkan nilai akhir dari setiap faktor, skor yang didapatkan untuk setiap sub-karakteristik nantinya akan dirata-rata, sebagai contoh, skor yang didapat untuk faktor *functional suitability* adalah sebagai berikut, 90%, 92%, dan 92%, yang kemudian nilai akhir untuk faktor tersebut adalah 91,3%.

HASIL DAN PEMBAHASAN

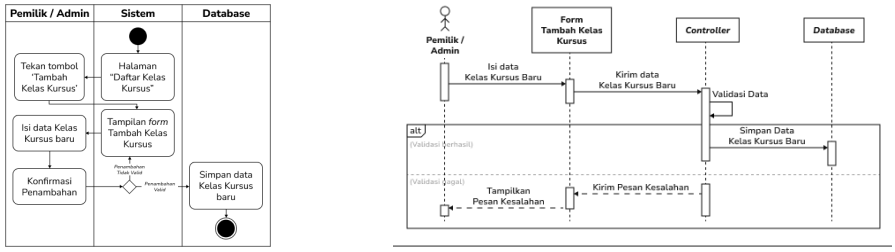
Pengembangan Aplikasi *Build* 1 – Pemilik / Admin Kursus

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

Setelah *use case* untuk Pemilik / Admin Kursus diidentifikasi, langkah selanjutnya adalah menguraikannya ke dalam dua diagram utama, yaitu *Activity Diagram* dan *Sequence Diagram*, untuk memberikan pemahaman yang lebih mendalam terkait alur aktivitas dan alur data pada aplikasi. *Activity Diagram* digunakan untuk memvisualisasikan alur kerja secara rinci, termasuk langkah-langkah penting, interaksi, serta potensi hambatan dalam proses [7]. Sementara itu, *Sequence Diagram* menggambarkan alur pesan antar objek berdasarkan urutan waktu, sehingga membantu pengembang memahami bagaimana setiap komponen saling berinteraksi dalam menyelesaikan suatu proses [8]. Sebagai contoh, Gambar 3 menunjukkan kedua diagram tersebut pada fitur Tambah Kelas Kursus, yang mencakup proses pengisian data, validasi input, hingga penyimpanan ke dalam *database*.



(a) (b)

Gambar 3. a) *Activity Diagram* untuk proses Tambah Kelas Kursus*,* b) *Sequence Diagram* untuk proses Tambah Kelas 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 [9]. Total skenario untuk *build* 1 adalah 120 skenario. Tabel 1 berikut adalah contoh skenario dan hasil pengujian untuk proses Tambah Kelas Kursus.

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

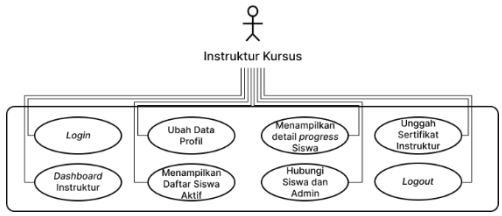
ID	Skenario Pengujian	Hasil yang diharapkan	Tingkat Pengujian	Hasil Pengujian
…	…	…	…	…
B1-T35	Mengetikkan angka "0" di kolom "Kuota Kelas" kemudian menekan tombol	Sistem menampilkan pesan kesalahan kepada	Minor	Valid

	"Tambah"	pengguna bahwa kuota minimal adalah 1		
…	…	…	…	…

Setelah pengujian, tahap implementasi dilakukan untuk memastikan integrasi *build* berjalan lancar tanpa *bug*. Karena *build* untuk Pemilik / Admin Kursus adalah *build* pertama dalam proses pengembangan, maka proses selanjutnya dapat langsung dilakukan, yaitu tahapan desain dan pengembangan *build* 2.

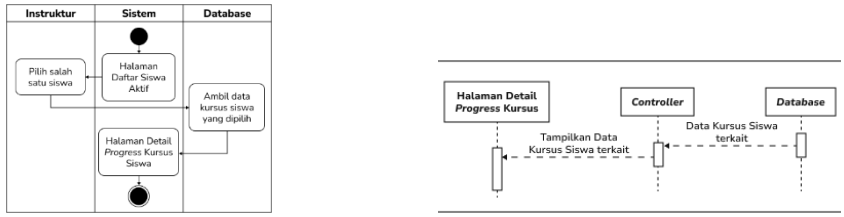
Pengembangan Aplikasi *Build* 2 – Instruktur Kursus

Melanjutkan proses *build* sebelumnya, tim pengembang akan mengidentifikasi masing-masing *use case* berdasarkan Analisa Kebutuhan yang dilakukan pada awal perancangan. Gambar 4 dibawah ini adalah *use case diagram* untuk *user* Instruktur Kursus.



Gambar 4. *Use Case Diagram* untuk Instruktur Kursus

Dengan mengidentifikasi masing-masing *use case*, langkah selanjutnya adalah menguraikan alur aktivitas kerja dan alir data antar objek lebih lanjut menggunakan *Activity Diagram* dan *Sequence Diagram* melalui Gambar 5 dibawah ini adalah proses Detail *Progress* Kursus untuk Instruktur Kursus dimana sistem akan menampilkan beberapa data seperti jadwal kursus, kelas atau paket kursus yang diikuti siswa, serta capaian capaian yang dilakukan siswa untuk setiap pertemuan.



(a) (b)

Gambar 5. a) *Activity Diagram* untuk proses Detail *Progress* Kursus untuk Instruktur, b) *Sequence Diagram* untuk proses Detail *Progress* Kursus untuk Instruktur

Untuk proses pengujian *build* 2 serupa pada *build* 1, pengembang akan memeriksa kesesuaian respon sistem. Total skenario yang dipersiapkan untuk *build* 2 adalah 42 skenario. Berikut ini disajikan Tabel 2 contoh skenario dan hasil *Black Box Testing* untuk proses Detail *Progress* Kursus untuk Instruktur.

Tabel 2. Skenario dan hasil *Black Box Testing Build* 2 untuk proses Detail *Progress* Kursus untuk Instruktur

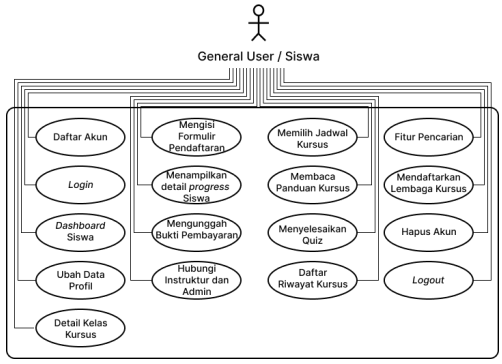
ID	Skenario Pengujian	Hasil yang diharapkan	Tingkat Pengujian	Hasil Pengujian
…	…	…	…	…

B2-T32	Untuk siswa yang belum mengunggah bukti pembayaran	Sistem menampilkan peringatan bahwa siswa belum mengunggah bukti pembayaran	Minor	Valid
…	…	…	…	…

Dengan proses pengujian selesai, tim pengembang dapat melanjutkan proses pengembangan ke tahapan berikutnya yaitu implementasi. Setelah dilakukan integrasi antara *build* 1 dan *build* 2, dan memastikan ulang bahwa tidak terjadi *error* atau *bug*, langkah selanjutnya adalah tahapan desain dan pengembangan untuk *build* 3.

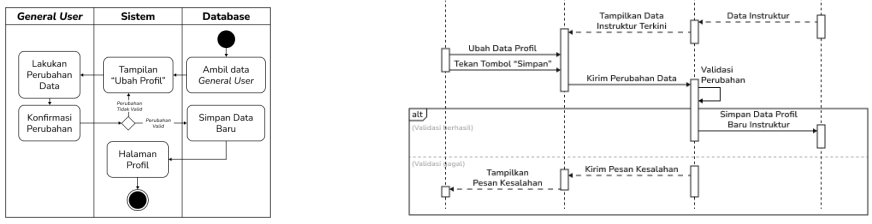
Pengembangan Aplikasi *Build* 3 – Siswa / Pelajar / Umum

Untuk *build* terakhir, pengembang akan merancang *use case diagram* untuk Siswa / Pelajar / Umum. Pengguna Umum yang dimaksud adalah pemilik akun yang mungkin sedang melakukan proses mendaftarkan lembaga kursus baru di aplikasi. Sesuai dengan hasil analisa kebutuhan pada proses sebelumnya, sama seperti pada *build-build* diatas. Gambar 6 dibawah ini adalah *use case diagram* untuk *user* Siswa / Pelajar / Umum.



Gambar 6. *Use Case Diagram* untuk Siswa / Pelajar

Setelah merancang *use case diagram*, langkah selanjutnya adalah menguraikannya lebih lanjut menggunakan *Activity Diagram* dan *Sequence Diagram* melalui Gambar 7 dibawah ini adalah proses Ubah Data Profil yang meliputi perubahan data yang dilakukan oleh Siswa, seperti mengubah *username*, mengunggah foto profil baru, mengubah *password*, dimana sistem selanjutnya akan memvalidasi data baru dan menyimpan perubahan pada *database* apabila validasi data baru berhasil.



(a) (b)

Gambar 7. a) *Activity Diagram* untuk proses Ubah Data Profil, b) *Sequence Diagram* untuk proses Ubah Data Profil

Selanjutnya adalah proses pengujian untuk *build* 3. Apabila ditemukan *bug* pada saat pengujian, hal ini adalah kesempatan bagi tim pengembang untuk melakukan perbaikan sebelum melanjutkan proses pengujian dengan ISO 25010:2023. Gambar 8 dibawah ini adalah *Class Diagram* untuk *build* 3 yang menjadi acuan bagi tim pengembang untuk pembangunan dan integrasi tabel yang ada didalam *database*.



Gambar 8. *Class Diagram*

Selanjutnya tim pengembang akan menguji respons sistem berdasarkan skenario yang sudah ditentukan sebelumnya. Tabel 3 berikut adalah skenario dan hasil *Black Box Testing* untuk proses ubah data profil.

Tabel 3. Skenario dan hasil *Black Box Testing Build* 3 untuk proses Ubah Data Profil

ID	Skenario Pengujian	Hasil yang diharapkan	Tingkat Pengujian	Hasil Pengujian
…	…	…	…	…
B3-T39	Melakukan perubahan dengan benar dan sesuai kemudian menekan tombol “Simpan”	Sistem melakukan perubahan di database dan mengarahkan pengguna ke halaman profil	Major	Valid
…	…	…	…	…

Setelah proses pengujian untuk *build* 3 selesai, tim pengembang dapat melanjutkan proses pengembangan ke tahapan berikutnya yaitu implementasi. Gambar 11 dibawah ini adalah hasil implementasi untuk *build* 3 Ubah Data Profil.

Setelah semua *build* berhasil dikembangkan, selanjutnya adalah melakukan penjaminan kualitas perangkat lunak dengan standar ISO 25010:2023 [10]. Hasil pengukuran kualitas perangkat lunak berdasarkan standar ISO 25010:2023 untuk faktor *functional suitability* 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, faktor *interaction capability* yang bertujuan untuk memastikan bahwa perangkat lunak yang sudah dirancang memiliki antarmuka yang ramah pengguna dan mudah digunakan. 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

Melalui pengembangan aplikasi kursus mengemudi berbasis web menggunakan *incremental model*, penelitian ini berhasil memberikan solusi digital yang dapat digunakan oleh seluruh penyedia kursus di Surabaya. Aplikasi ini mengintegrasikan proses administrasi, manajemen Siswa dan Instruktur, hingga materi edukasi dalam satu platform terpadu, yang diharapkan mampu meningkatkan efisiensi operasional penyedia kursus sekaligus membantu mencetak pengemudi yang lebih terampil. Dengan demikian, penelitian ini turut menjawab permasalahan utama yang diangkat, yaitu pentingnya pelatihan mengemudi dalam upaya menurunkan angka kecelakaan lalu lintas akibat faktor kesalahan manusia.

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, 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, 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] N. M. D. Elianti, D. P. Githa, and A. A. N. H. Susila, "Android-based driving course information system," in Proceedings of the Information Technology Department, Faculty of Engineering, Udayana University, Bali, Indonesia, 2023.

[5] Wibowo, B. T., Dewi, F. K. S., & Citrayasa, V. (2024). "Pembangunan Sistem Informasi Manajemen Kursus Mengemudi Berbasis Website". Jurnal Informatika Atma Jogja, 5(1), 27-36.

[6] 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].

[7] L. Jacobson and J. R. G. Booch, The Unified Modeling Language Reference Manual, 2nd ed., 2021.

[8] 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].

[9] R. Setiawan, “Black box testing untuk menguji perangkat lunak,” Dicoding, 2021. [Online]. Available: https://www.dicoding.com/blog/black-boxtesting/. [Accessed: Mar. 29, 2024].

[10] International Organization of Standardization, ISO/IEC 25010:2023(en). 2023. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:isoiec:25010:ed-2:v1:en. [Accessed: 16-Jul-2024].

[11] Subecz, Z. (2021). Web-development with Laravel framework. Gradus, 8(1), 211-218. [Online]. Available: https://real.mtak.hu/125616/1/2021_1_CSC_006_Subecz.pdf. [Accessed: 03-Sep-2024].
[12] Hermiati, R., Asnawati, A., & Kanedi, I. (2021). Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database Mysql. Jurnal Media Infotama, 17(1).