

Sistem Informasi Checklist Pada Suatu Bandara Menggunakan *Framework Flutter*

David Setiaraja¹, Hernawati², Nopi Ramsari*³

^{1,2,3}Program Studi Teknik Informatika, Universitas Nurtanio, Bandung

Email:korespondensi: *nopiramsarihatta@gmail.com

Abstrak. Bandara atau dalam Bahasa Inggris disebut *Airport* merupakan sebuah fasilitas dimana Pesawat Terbang seperti Pesawat Udara dapat lepas landas. Bandara tentunya dilengkapi kendaraan utama yang dilengkapi dengan peralatan pendukung operasional dan kendaraan pendukung lainnya untuk melakukan tugas-tugas operasional dalam menjaga keselamatan para pengguna jasa bandara. Penerapan sistem informasi operasional seperti *checklist* data pada bandara untuk kendaraan operasional tentunya diperlukan agar pencarian data secara periode waktu tertentu lebih mudah. Kemajuan teknologi pada sistem operasi *android* untuk aplikasi *mobile* dapat digunakan untuk pengembangan alat kerja yang memberi kemudahan bagi *user* terutama pada akses waktu, tempat, dan keamanan data. *Framework flutter* yang memiliki kelebihan yaitu dapat dijalankan di beberapa platform yang berbeda dan dapat membuat aplikasi *android* atau disebut juga *cross-platform*. Sistem informasi *Checklist* pada suatu bandara dengan menggunakan *Framework Flutter* menjadi alat bantu bagi *user* untuk mendapatkan informasi secara cepat apabila ada kondisi kerusakan kendaraan operasional, sehingga membantu para pengguna dalam menganalisis tindakan *maintenance* yang akan dilakukan terhadap kendaraan operasional tersebut. Pada Penelitian ini telah berhasil dilakukan pengujian menggunakan metode *blackbox* dengan skala terbatas yaitu hanya pada administrator aplikasi.

Kata kunci: Kendaraan, *Checklist*, *Android*, *Mobile*, *Flutter*, *Maintenance*.

1. Pendahuluan

Bandar Udara adalah kawasan di daratan dan/atau perairan dengan batas-batas tertentu yang digunakan sebagai tempat pesawat udara mendarat dan lepas landas, naik turun penumpang, bongkar muat barang, dan tempat perpindahan intra dan antarmoda transportasi, yang dilengkapi dengan fasilitas keselamatan dan keamanan bagi pengguna jasa bandara. Untuk keamanan tentunya bandara memiliki kendaraan utama yang dilengkapi dengan peralatan pendukung operasional. Data kendaraan yang ada masih dikelola secara manual yaitu dengan menggunakan lembar kertas sebagai laporan data kendaraan operasional sehingga data mudah sekali hilang dan sulit melakukan pengecekan akses riwayat data kondisi kendaraan operasional untuk memverifikasi apakah serangkaian langkah yang diperlukan telah dilakukan atau belum sehingga dapat dijadikan sebagai laporan kerja kepada atasan oleh karena itu diperlukan pembuatan sistem informasi. Sistem informasi juga merupakan suatu alat yang digunakan untuk memudahkan pekerjaan agar dapat dijalankan dengan lebih efektif dan efisien[1]. Disamping itu staf dan asisten manajer menggunakan perangkat akses yang berbeda-beda, sehingga dalam penggunaan aplikasi perlu ada kemudahan agar aplikasi yang dibuat dapat digunakan oleh berbagai macam perangkat yaitu dapat berjalan di *device* iOS dan Android, oleh karena itu maka penulis membangun aplikasi yang dapat berguna dengan menggunakan sistem operasi berbasis *android* dengan *framework flutter*. Sistem operasi *android* yang akan digunakan karena penggunaan teknologi Android merupakan keunggulan tersendiri untuk proses pengembangan aplikasi karena tidak

perlu memiliki akun tertentu (gratis)[2].

Flutter adalah *cross-platform mobile app SDK (Software Development Kit)* untuk membuat aplikasi *android* dan *iOS* dari satu *codebase* dengan performa tinggi. *Framework* ini dipilih dengan tujuan agar mudah dalam pengembangan nantinya dikarenakan dengan menggunakan *Flutter* dengan membuat satu kali aplikasi tetapi dapat digunakan untuk perangkat *mobile* dengan berbagai sistem operasi seperti sistem *Android* dan *IOS* dengan kata lain dapat digunakan oleh perangkat *Android* dan juga *Apple*. Selain itu *framework Flutter* ini memiliki berbagai macam fitur dan juga *widget-widget* yang dapat digunakan guna mempermudah para *developer* untuk melakukan pengembangan aplikasi[3]. Pada Penelitian ini penulis menggunakan implementasi *framework flutter* secara maksimal dan memiliki fungsionalitas yang tinggi untuk proses bisnis pelaporan *checklist* data kendaraan operasional dengan menggunakan konsep arsitektur *CRUD (Create, Read, Update, Delete)* sehingga menghasilkan pelaporan kelola data dan laporan secara cepat dan tepat.

2. Metodologi Penelitian

Di dalam metodologi penelitian terdapat beberapa tahapan yang akan digunakan dalam pembuatan sistem informasi *Checklist* pada suatu bandara menggunakan *Framework Flutter*, diantaranya:

2.1 Metode Pengumpulan Data

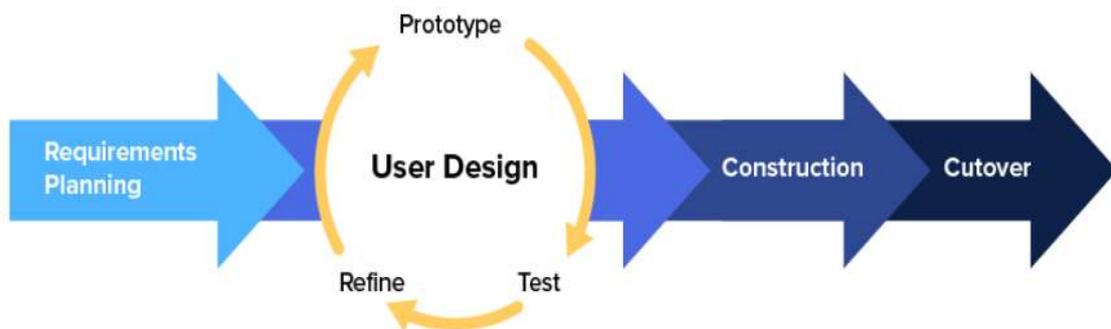
Merupakan metode yang digunakan peneliti, dalam melakukan analisis data dan menjadikannya informasi yang akan digunakan untuk mengetahui permasalahan yang dihadapi.

- 1) Identifikasi yaitu untuk mengetahui apa saja yang menjadi kendala dalam proses *checklist* data kendaraan operasional pada suatu bandara, sehingga dapat dibandingkan apakah pengimplementasian aplikasi dapat berguna untuk para karyawan.
- 2) Observasi yaitu kegiatan pengamatan dengan meneliti dokumen-dokumen produk yang ada untuk pembuatan aplikasi.
- 3) Studi Pustaka adalah pengumpulan bahan-bahan beberapa literatur yang berkaitan dengan judul tugas akhir melalui membaca buku-buku dari perpustakaan dan mencari manfaat referensi dari internet.

2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam pembuatan aplikasi *checklist* data kendaraan operasional menggunakan model *RAD (Rapid Application Development)*. *RAD* adalah metode yang berfokus pada pengembangan aplikasi secara cepat, melalui pengulangan dan *feedback* berulang-ulang. *RAD* atau *rapid prototyping* adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental* (bertingkat). Metode *RAD* menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. [4]

Adapun tahapan-tahapan pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi dapat di lihat pada gambar 1.



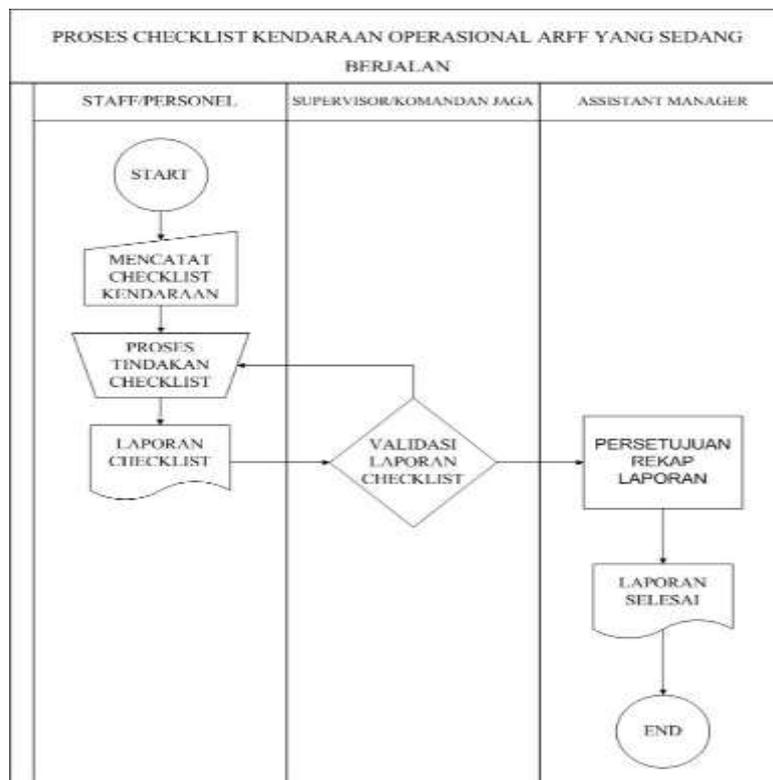
Gambar 1.Model RAD (Rapid Application Development)

Tahapan analisa masalah, bertujuan untuk mengidentifikasi kebutuhan, batasan dan objektivitas dari sistem yang akan dibangun, dengan mengumpulkan data dari *stakeholder*. Tahapan ini dilakukan dengan melakukan identifikasi pada operasional kerja di bandara mengenai kebutuhan sistem yang dibutuhkan. Realisasi dari tahap analisa masalah, yaitu dengan membangun rancangan sistem yang berbasis aplikasi *android* dengan menggunakan bahasa pemrograman *Dart* dan *Framework flutter*. Setelah tahap implementasi selesai, tahap selanjutnya adalah melakukan uji coba dengan mengawasi jalannya aktivitas aplikasi yang terjadi pada tahap awal *input checklist* data, sampai pelaporan hasil *checklist* data kendaraan operasional.

3. Analisa dan Perancangan

3.1 Proses Bisnis Yang Sedang Berjalan

Proses bisnis yang sedang berjalan saat ini masih menggunakan checklist kendaraan secara manual yaitu dengan menggunakan lembar kertas saat melakukan pengecekan pada kendaraan operasional. Pengecekan kendaraan bertujuan untuk perawatan lanjutan/perbaikan yang memerlukan penggantian *sparepart* dan lain-lain dilakukan oleh teknisi.

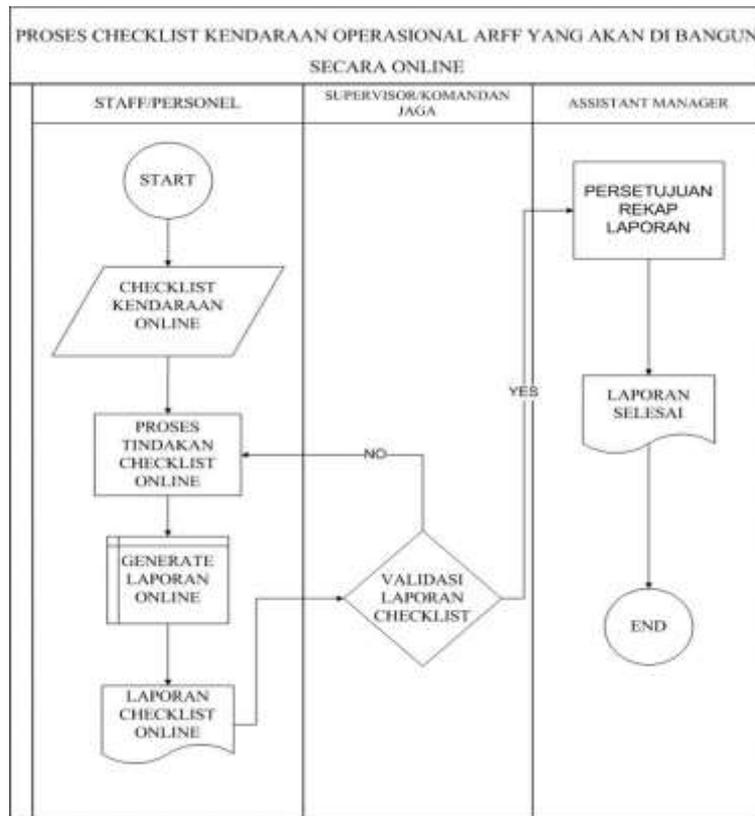


Gambar 2. Flowmap Proses Checklist Yang Sedang Berjalan

Pada flow map diatas proses pencatatan dan pembuatan laporan *checklist* kendaraan masih dilakukan secara manual, yaitu menggunakan kertas.

3.1.1 Analisa Sistem Yang Di Usulkan

Analisa sistem yang diusulkan berupa laporan, formulir dan tembusan serta prosedur-prosedur, yang disajikan dalam *diagram* alir dokumen,



Gambar 3.Flowmap Sistem yang diusulkan.

Pada gambar flow map diatas proses pencatatan *checklist* kendaraan yang dilakukan oleh staf/personel sudah menggunakan aplikasi berbasis *online* termasuk juga pembuatan laporan *checklist* kendaraan, validasi laporan *checklist* yang dilakukan oleh supervisor/koman dan jaga dan juga proses persetujuan rekap laporan oleh *assistant manager*.

3.1.2 Kebutuhan Fungsional Perangkat Lunak

Analisis Kebutuhan Fungsional adalah suatu proses apa saja yang nantinya harus disediakan oleh sistem, mencakup bagaimana sistem harus bereaksi pada *input* tertentu dan bagaimana perilaku sistem pada situasi tertentu. Spesifikasi kebutuhan fungsional perangkat lunak *software requirement specification/SRS*, sebagai berikut:

Tabel 1 Daftar Kebutuhan Fungsional Perangkat Lunak.

Kode Identifikasi	Deskripsi
SRS-01	Sistem dapat <i>login</i> ke aplikasi
SRS-02	Sistem dapat <i>register</i> ke aplikasi
SRS-03	Sistem dapat <i>input</i> data, meliputi: <ul style="list-style-type: none"> a. <i>Checklist</i> kendaraan utama <i>OshkoshStriker</i> b. <i>Checklist</i> kendaraan pendukung <i>Ambulance</i> c. <i>Checklist</i> kendaraan pendukung <i>ComandoCar</i>

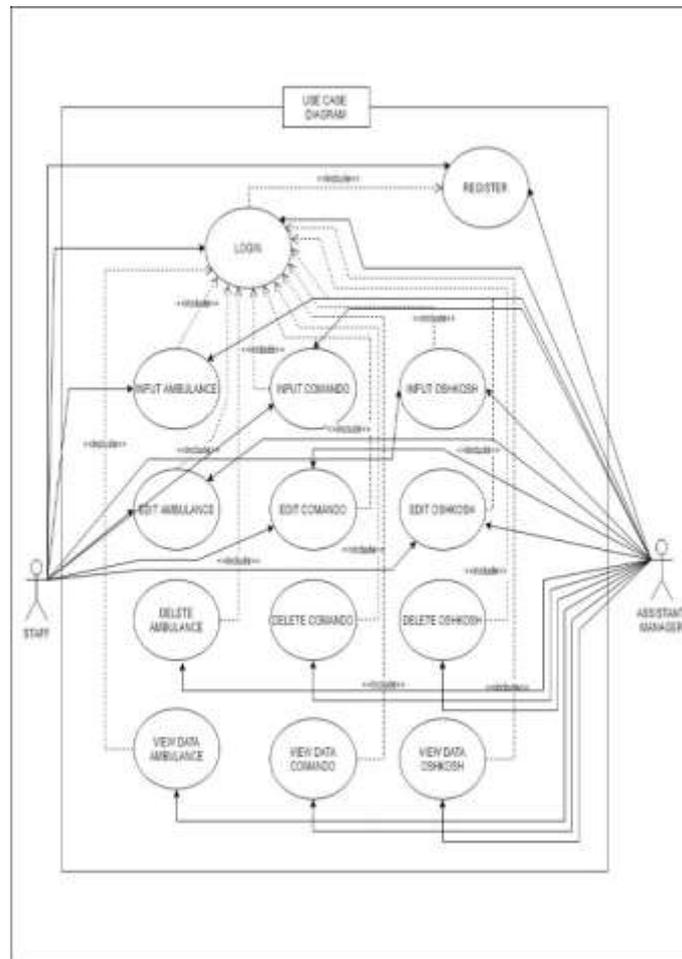
SRS-04	Sistem dapat <i>edit</i> data, meliputi: <ul style="list-style-type: none"> a. <i>Checklist</i> kendaraan utama <i>OshkoshStriker</i> b. <i>Checklist</i> kendaraan pendukung <i>Ambulance</i> c. <i>Checklist</i> kendaraan pendukung <i>ComandoCar</i>
SRS-05	Sistem dapat <i>delete</i> data, meliputi: <ul style="list-style-type: none"> a. <i>Checklist</i> kendaraan utama <i>OshkoshStriker</i> b. <i>Checklist</i> kendaraan pendukung <i>Ambulance</i> c. <i>Checklist</i> kendaraan pendukung <i>ComandoCar</i>
SRS-06	Sistem dapat menampilkan laporan data, meliputi: <ul style="list-style-type: none"> a. <i>Checklist</i> kendaraan utama <i>OshkoshStriker</i> b. <i>Checklist</i> kendaraan pendukung <i>Ambulance</i> c. <i>Checklist</i> kendaraan pendukung <i>ComandoCar</i>

3.2 Perancangan Sistem

Pada tahap ini dilakukan perancangan sistem menggunakan metode Unified Model Language (UML) bertujuan untuk memodelkan suatu sistem yang dapat mudah dipahami oleh manusia maupun mesin. UML yang digunakan yaitu : Use Case Diagram, Class Diagram dan activity diagram. Selain itu penulis juga membuat rancangan entitas Relationship Diagram (ERD) dan struktur menu.

3.2.1 Use Case Diagram

Berdasarkan spesifikasi kebutuhan yang sudah di uraikan, maka *usecase* yang peneliti usulkan untuk perancangan adalah sebagai berikut:

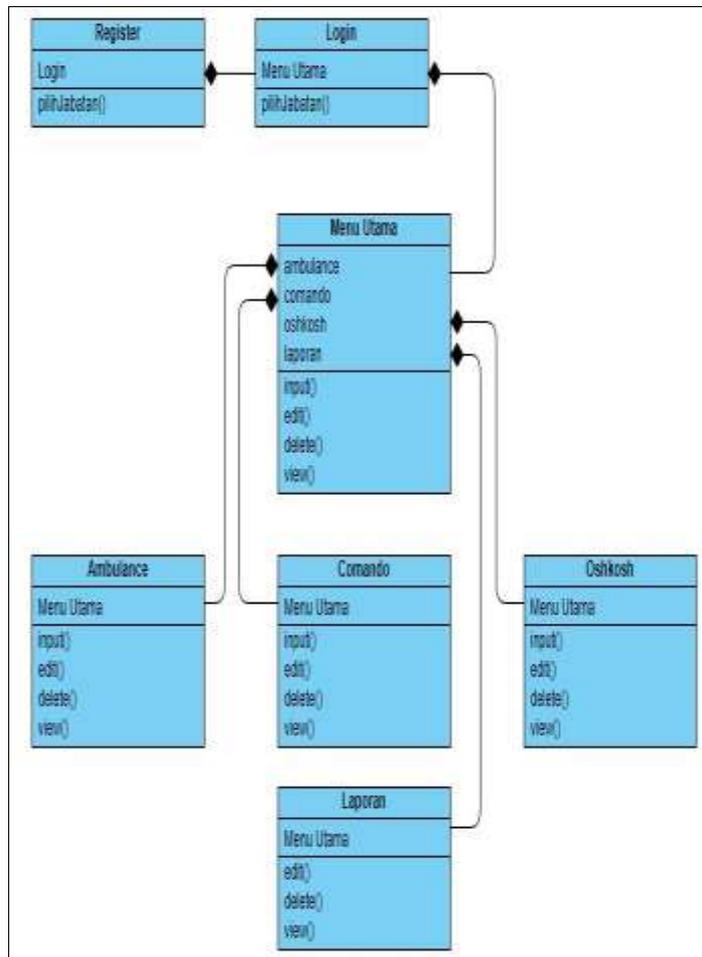


Gambar 4.Diagram *Use Case*

Pada gambar diagram *Use Case* diatas *actor* yang terlibat ada dua, yaitu *staff* dan *Assistant Manager*.

3.2.2 *Class Diagram*

Class Diagram merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute* atau *properties*. Sedangkan perilaku (*behavior*) suatu objek mendefinisikan bagaimana sebuah objek bertindak atau beraksi dan memberikan reaksi.

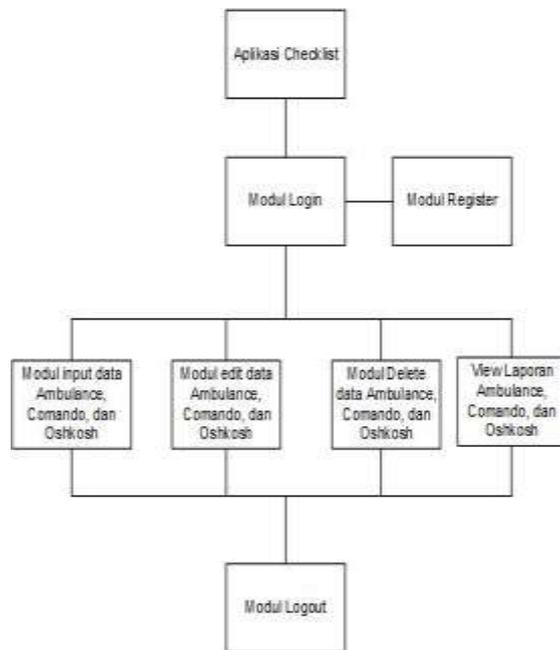


Gambar 5. *Class Diagram.*

Pada gambar *class diagram* diatas terdapa tbeberapa kelas yang digunakan, diantaranya kelas *register*, *login*, *ambulance*, *commando*, *Oshkosh* dan *laporan*.

3.2.3 Struktur Menu

Struktur *menu* program di akses oleh *staff&assistantmanager*, yaitu *menu utama assistantmanager* meliputi *input*, *delete*, *edit*, dan *view* laporan. Sedangkan *staff* meliputi *input* dan *edit*.



Gambar 6. Struktur *Menu* Utama

Untuk dapat login, pengguna harus daftar dulu ke menu *Modul register*, setelah mendaftar maka pengguna dapat memilih menu *input*, *edit*, *delete* dan *view* laporan pada *Ambulance*, *Commando* dan *Oshkosh*.

4. Hasil dan Pembahasan

Hasil dari sistem informasi Checklist pada suatu bandara menggunakan *Framework Flutter* dapat dilihat pada gambar 7 untuk tampilan antarmuka halaman register pada staff & Assistant Manager.

4.1 Implementasi Sistem

Setelah tahapan perancangan dilanjutkan dengan tahapan implementasi.

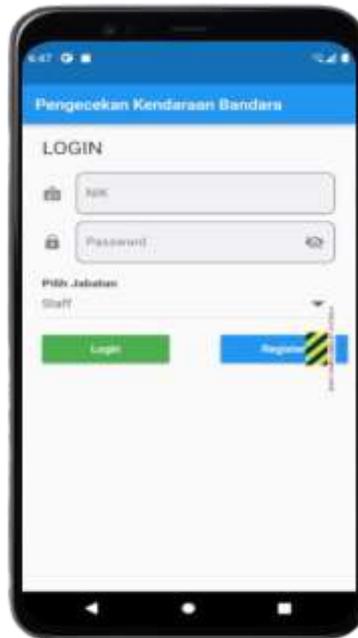
- a. Tampilan antarmuka halaman *register* pada *Staff&AssistantManager*



Gambar 7. Antarmuka Halaman *Register*.

Tampilan diatas adalah tampilan form untuk pengguna melakukan *register* agar dapat *login* keaplikasi. Pengguna wajib memasukkan data NIK, namalengkap, alamat-mail, nomor *handphone*, *password* dan memilih jabatan yang sesuai.

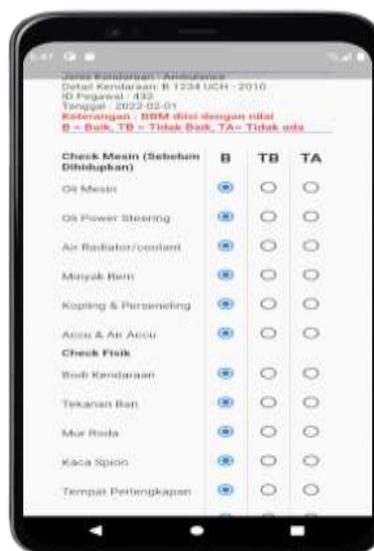
- b. Tampilan antarmuka halaman *login* pada *Staff & Assistant Manager*



Gambar 8. Antarmuka Halaman *Login Staff*

Setelah melakukan register maka pengguna dapat login sesuai dengan tampilan diatas.

- c. Tampilan antarmuka halaman *menu Input*



Gambar 9. Antarmuka Halaman *Menu Input*.

4.2. Pengujian Sistem

Dalam penelitian ini pengujian yang dilakukan menggunakan *black box testing* yaitu pengujian dari perangkat lunak yang dianalogikan bahwa hanya dapat dilihat dari tampilan luarnya (*interface*), tanpa mengetahui yang ada didalamnya atau detailnya. Fungsinya hanya mengetahui *input* sampai dengan *outputnya*. Pengujian aplikasi menggunakan metode black box, pengujian ini fokus pada persyaratan fungsional dari perangkat lunak (aplikasi) yang dibangun[5].

Tabel 2.Proses Pengujian *Staff*.

Kode Identifikasi	Kasus Uji	Proses pengujian	Hasil Pengujian	Hasilyang Diharapkan
SRS-01	<i>Register</i>	Memasukan <i>input</i> data nik, nama lengkap, <i>email</i> , <i>nohandphone</i> dan <i>password</i>	Validasi berhasil dan berhasil menyimpan data ke dalam <i>database</i>	Sesuai
SRS-02	<i>Login</i>	Memasukan nik dan <i>password</i> di halaman <i>login</i>	Validasi berhasil dan masuk ke halaman <i>dashboard</i>	Sesuai
SRS-03	<i>Input Data</i>	<i>Staff</i> input data, meliputi: a. <i>Checklist</i> kendaraan utama <i>Oshkosh Striker</i> b. <i>Checklist</i> kendaraan pendukung <i>Ambulance</i> c. <i>Checklist</i> kendaraan pendukung <i>Comando Car</i>	Sistem berhasil <i>input</i> data dan berhasil menyimpan data kedalam <i>database</i> .	Sesuai

SRS-04	<i>Edit Data</i>	<i>Staffedit data, meliputi:</i> a. <i>Checklist kendaraan utama Oshkosh Striker</i> b. <i>Checklist kendaraan pendukung Ambulance</i> c. <i>Checklist kendaraan pendukung Comando Car</i>	Sistem berhasil <i>edit data</i> dan berhasil <i>menyimpan data</i> kedalam <i>database</i> .	Sesuai
---------------	------------------	---	---	--------

Tabel 3. Proses Pengujian *Assistant Manager*.

Kode Identifikasi	Kasus Uji	Proses Pengujian	Hasil Pengujian	Hasil yang Diharapkan
SRS-01	<i>Register</i>	Memasukan input data nik, nama lengkap, <i>email, no handphone</i> dan <i>password</i>	Validasi berhasil dan berhasil menyimpan data ke dalam <i>database</i>	Sesuai
SRS-02	<i>Login</i>	Memasukan nik dan <i>password</i> di halaman <i>login</i>	Validasi berhasil dan masuk ke halaman <i>dashboard</i>	Sesuai
SRS-03	<i>Input Data</i>	<i>Assistant Managerinput data, meliputi:</i> a. <i>Checklist kendaraan utama Oshkosh Striker</i> b. <i>Checklist kendaraan pendukung Ambulance</i> c. <i>Checklist kendaraan pendukung Comando Car</i>	Sistem berhasil <i>input data</i> dan berhasil <i>menyimpan data</i> kedalam <i>database</i> .	Sesuai

SRS-04	Edit Data	Assistant Manageredit data, meliputi: a. Checklist kendaraan utama Oshkosh Striker b. Checklist kendaraan pendukung Ambulance c. Checklist kendaraan pendukung Comando Car	Sistem berhasil edit data dan berhasil menyimpan data kedalam database.	Sesuai
SRS-05	Delete Data	Assistant Managerdelete data, meliputi: a. Checklist kendaraan utama Oshkosh Striker b. Checklist kendaraan pendukung Ambulance c. Checklist kendaraan pendukung Comando Car	Sistem berhasil delete data dan berhasil menyimpan data kedalam database.	Sesuai
SRS-06	Menampilkan Laporan	Assistant Manager menampilkan laporan data, meliputi: a. Checklist kendaraan utama Oshkosh Striker b. Checklist kendaraan pendukung Ambulance c. Checklist kendaraan pendukung Comando Car	Sistem berhasil menampilkan laporan data dari database.	Sesuai

5. Kesimpulan

Berdasarkan hasil implementasi dan pengujian terhadap sistem informasi Checklist pada suatu bandara menggunakan Framework Flutter maka dapat disimpulkan sebagai berikut:

1. Sistem informasi Checklist kendaraan telah berhasil melakukan proses manipulasi data dan menampilkan laporan, meliputi:
 - a. Checklist kendaraan utama Oshkosh Striker.
 - b. Checklist kendaraan pendukung Ambulance.
 - c. Checklist kendaraan pendukung Comando Car.

2. Sistem informasi Checklist kendaraan dapat menyimpan data dengan aman di dalam *database*, sehingga jika diperlukan dikemudian hari, data masih tersimpan dan tidak mudah rusak.

6. Daftar Pustaka

- [1] K. Ramadhan and A. Utami, "Sistem Informasi Kelulusan dan Kriptografi Ijasah pada Lembaga Pendidikan Penerbangan," *Manaj. Inform.*, vol. 9, no. 2, pp. 136–143, 2019, [Online]. Available: <https://ejournal.unesa.ac.id/index.php/jurnal-manajemen-informatika/article/view/29453>.
- [2] A. R. Hakim, K. Harefa, and B. Widodo, "Pengembangan Sistem Informasi Akademik Berbasis Android Menggunakan Flutter Di Politeknik," *SCAN - J. Teknol. Inf. dan Komun.*, vol. 14, no. 3, 2019, doi: 10.33005/scan.v14i3.1684.
- [3] A. P. Pratama and M. Kamisutara, "Pengembangan Sistem Informasi Akademik Berbasis Mobile Menggunakan Flutter Di Universitas Narotama Surabaya," *Netw. Eng. Res. Oper.*, vol. 6, no. 2, p. 145, 2021, doi: 10.21107/nero.v6i2.238.
- [4] J. R. Sagala, "Model Rapid Application Development (Rad) Dalam Pengembangan Sistem Informasi Penjadwalan Belajar Mengajar," *J. Mantik Penusa*, 2018, [Online]. Available: <http://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/454>.
- [5] E. Haryanto, M. Arif, and M. Setiawan, "Perancangan Sistem Informasi Pembelajaran Praktikum Online Multiplatform Berbasis Framework Flutter," *Pros. Semin. Nas. ...*, 2021, [Online]. Available: <http://e-journal.janabadra.ac.id/index.php/PSN/article/view/1568>.