

Perancangan Aplikasi Pin In Menggunakan Appsheet Dan Google Sheets

Achmad Aditya Ashadul Ushud*

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, Petukangan Utara, Jakarta Selatan, 12260
Email: achmad.aditya@budiluhur.ac.id

* Penulis korespondensi

Diajukan: 21 November 2023

Direvisi: 3 Januari 2024

Diterima: 4 Januari 2024

Dipublikasikan: 24 Januari 2024

Abstract

This mobile application is designed to facilitate users in the check-in process at a particular location. PT. Nutrisi Juara Asia has sales representatives scattered in various regions and locations. With the main goal of facilitating these sales representatives to easily and efficiently report attendance, the application's main features include the ability to automatically identify the user's location, allowing them to check-in, referred to as Pin-In, with a single touch. The interface design is kept simple to make the Pin-In process easy, even for users unfamiliar with technology. This application is created using Appsheet, with the database utilizing Google Sheets. The Waterfall method is employed in the development of this application. After using the application, the expected results are visible, making attendance tracking easier and real-time. With a focus on user-friendliness and accessibility, this mobile application serves as a practical solution for PT. Nutrisi Juara Asia, aiming to optimize the work of sales representatives in terms of time and reporting when visiting various locations.

Keywords : *mobile application, AppSheet, Google Sheets*

Abstrak

Aplikasi mobile ini dirancang untuk mempermudah pengguna dalam melakukan proses check-in pada suatu lokasi. PT. Nutrisi Juara Asia memiliki sales yang tersebar di berbagai daerah dan lokasi. Dengan tujuan utama memfasilitasi sales-sales tersebut, agar dapat melakukan pelaporan absensi secara mudah dan efisien. Fitur utama aplikasi mencakup kemampuan untuk mengidentifikasi lokasi pengguna secara otomatis, memungkinkan mereka melakukan check-in dalam hal ini disebut Pin-In dengan sekali sentuh. Desain antarmuka dibuat sederhana sehingga proses Pin-In menjadi mudah, bahkan bagi pengguna yang tidak terbiasa dengan teknologi. Aplikasi ini dibuat dengan menggunakan Appsheet dan database menggunakan Google Sheets. Metode Waterfall digunakan dalam pengembangan aplikasi ini. Setelah aplikasi ini digunakan, terlihat hasil yang diharapkan, yaitu melakukan absensi menjadi lebih mudah dan real-time. Dengan fokus pada kemudahan penggunaan dan aksesibilitas, aplikasi mobile ini menjadi solusi praktis bagi PT. Nutrisi Juara Asia yang ingin mengoptimalkan kerja sales dalam hal waktu dan pelaporan saat mengunjungi berbagai lokasi.

Kata kunci : *aplikasi mobile, AppSheet, Google Sheets*

I. PENDAHULUAN

1.1. Latar Belakang

PT. Nutri Juara Asia yang bergerak di bidang penjualan memiliki banyak sales yang tersebar di Indonesia. Kantor pusatnya berada di daerah Tangerang Selatan. Para sales tersebut selama ini menggunakan aplikasi WhatsApp untuk melaporkan absen. Nantinya dari situ admin akan merekap laporan-laporan yang masuk tersebut dan menyatukannya ke dalam *spreadsheet*. Aplikasi *mobile* yang dirancang bertujuan untuk mengoptimalkan proses *check-in* sales di berbagai lokasi. Dengan tujuan utama memfasilitasi sales dalam melaporkan absensi secara efisien, aplikasi ini menawarkan solusi yang mudah diakses dan dapat diandalkan. Sales yang memiliki mobilitas tinggi akan merasakan manfaat signifikan dari fitur-fitur yang terintegrasi dalam aplikasi ini.

Salah satu keunggulan utama aplikasi ini adalah kemampuannya untuk mengidentifikasi lokasi pengguna secara otomatis. Dengan satu sentuhan, atau yang diistilahkan sebagai Pin-In, sales dapat melakukan Pin-In dengan cepat dan akurat. Desain antarmuka yang sederhana menjadikan aplikasi ini ramah pengguna, memungkinkan bahkan pengguna yang kurang berpengalaman dengan teknologi dapat dengan mudah mengakses dan memanfaatkannya.

Dalam pengembangan aplikasi ini, digunakan platform AppSheet yang mempermudah proses pembuatan tanpa mengorbankan fungsionalitas. Basis data yang diintegrasikan dengan Google Sheets memberikan keleluasaan dalam manajemen data absensi. Selain itu, sistem ini memastikan keamanan data yang tinggi untuk melindungi informasi sensitif dari sales.

Dengan fokus pada kemudahan penggunaan dan aksesibilitas, aplikasi ini tidak hanya menjadi alat praktis bagi sales, tetapi juga merupakan langkah progresif dalam meningkatkan efisiensi operasional PT. Nutrisi Juara Asia. Dengan memaksimalkan penggunaan teknologi, perusahaan dapat merespon lebih cepat terhadap perubahan pasar dan memastikan bahwa tim sales mereka dapat bekerja dengan lebih efektif dan efisien di setiap lokasi yang mereka kunjungi. Secara keseluruhan, inovasi ini diharapkan dapat memberikan kontribusi positif dalam meningkatkan produktivitas dan responsivitas bisnis PT. Nutrisi Juara Asia.

1.2. Rumusan Masalah

Adapun perumusan masalah dalam penelitian perancangan aplikasi pin in menggunakan appsheet dan google sheets Sebagai berikut :

1. Bagaimana cara mengefisienkan pelaporan absen sales?
2. Apakah ada aplikasi yang mudah dibuat dan digunakan untuk pelaporan absen sales?
3. Apakah aplikasi yang dirancang menjadikan pelaporan absen sales jadi lebih mudah dan efisien?

2. METODOLOGI PENELITIAN

Perancangan aplikasi ini menggunakan metode *Waterfall*. Metode *Waterfall* adalah suatu model pengembangan perangkat lunak yang mengikuti pendekatan sekuensial atau linear dalam proses pengembangan. Model ini mengartikan pengembangan perangkat lunak sebagai suatu proses yang dilakukan secara berurutan dan terstruktur, dengan setiap fase bergantung pada penyelesaian fase sebelumnya. Berikut adalah penjelasan mengenai metode *Waterfall* dengan tahap-tahapnya:

1. Analisis dan Perencanaan (*Requirements*): Tahap ini melibatkan identifikasi kebutuhan perangkat lunak dari perspektif pengguna. Seluruh kebutuhan ini dikumpulkan dan dokumen-dokumen, seperti spesifikasi fungsional dan desain sistem, disusun.
2. Perancangan (*Design*): Pada tahap ini, berdasarkan kebutuhan yang telah diidentifikasi, perancangan sistem dan struktur program dihasilkan. Desain ini mencakup arsitektur perangkat lunak, rancangan antarmuka pengguna, dan perencanaan pengujian.
3. Implementasi (*Implementation*): Tahap ini melibatkan proses pengkodean perangkat lunak berdasarkan rancangan yang telah dibuat. Programmer mengimplementasikan rancangan ke dalam bahasa pemrograman yang sesuai.
4. Pengujian (*Testing*): Setelah tahap implementasi, dilakukan pengujian untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditentukan. Pengujian ini dapat mencakup pengujian unit, pengujian integrasi, dan pengujian sistem secara menyeluruh.
5. Penggabungan (*Deployment*): Setelah melewati tahap pengujian, perangkat lunak siap untuk di-*deploy* atau diimplementasikan ke lingkungan produksi. Ini berarti perangkat lunak dapat digunakan oleh pengguna akhir.
6. Pemeliharaan (*Maintenance*): Pada tahap ini, perangkat lunak dikelola untuk memastikan bahwa ia tetap dapat beroperasi sebagaimana mestinya. Pemeliharaan dapat melibatkan perbaikan *bug*, peningkatan fitur, atau adaptasi terhadap perubahan kebutuhan.

Metode *Waterfall* mewakili suatu paradigma klasik yang menyoroti rangkaian langkah-langkah berurutan dalam perancangan perangkat lunak menurut Sholikhah dkk (2017). Model *Waterfall* memperlihatkan suatu pendekatan sistematis dan berurutan dalam pengembangan perangkat lunak, di

mana setiap tahap dijalankan secara langkah demi langkah dengan ketertiban yang ketat menurut Hamid (2020).

Salah satu kelemahan dari model *Waterfall* adalah bahwa perubahan kebutuhan sulit diakomodasi setelah tahap analisis dan perencanaan. Pressman (2014) menyatakan bahwa model *Waterfall* memiliki kelemahan berikut:

1. Ketidakmampuan untuk Menanggapi Perubahan (*Inflexibility*): *Waterfall* sulit menanggapi perubahan kebutuhan pelanggan atau perubahan lainnya yang mungkin terjadi selama siklus pengembangan. Karena setiap fase harus diselesaikan sebelum melanjutkan ke fase berikutnya, perubahan yang dibutuhkan setelah fase dimulai dapat sulit atau mahal untuk diimplementasikan.
2. Ketidakpastian Akhir Proyek (*Uncertainty at the Project's End*): Karena pengujian umumnya dilakukan pada tahap akhir siklus pengembangan, risiko terkait kualitas dan fungsionalitas tidak selalu terdeteksi sampai tahap ini. Hal ini dapat menyebabkan penundaan signifikan jika masalah muncul pada tahap akhir, dan perubahan pada tahap ini dapat menjadi sangat mahal.
3. Tidak Sesuai untuk Proyek yang Besar dan Kompleks (*Not Suitable for Large and Complex Projects*): Metode *Waterfall* mungkin kurang cocok untuk proyek-proyek yang besar dan kompleks karena sulit untuk merinci semua kebutuhan dan desain pada awal proyek. Perubahan yang diperlukan kemudian dalam siklus dapat mengarah pada kesulitan implementasi dan penundaan.
4. Keterbatasan Keterlibatan Pelanggan (*Limited Customer Involvement*): Keterlibatan pelanggan dalam model *Waterfall* terbatas pada awal dan akhir proyek, yang dapat mengakibatkan solusi yang tidak sepenuhnya memenuhi harapan pelanggan.

2.1. AppSheet

AppSheet adalah satu di antara platform untuk membuat aplikasi tanpa kode, menurut Suharna (2021). AppSheet berbasis web, sehingga tidak perlu diinstal ke dalam computer untuk menggunakannya. AppSheet mengadopsi prinsip "*what you see is what you get*" (WYSIWYG). Dimana seorang programmer bisa langsung mendesain tampilan *website* dengan hanya mengklik beberapa tombol yang disediakan oleh *software* tersebut, menurut Ushud (2019). Dengan menggunakan AppSheet, pengguna dapat merancang, mengonfigurasi, dan mengimplementasikan aplikasi berbasis *cloud* tanpa menulis kode atau skrip secara manual.

Berikut adalah beberapa fitur dan aspek penting dari AppSheet:

1. Tanpa Kode (*No-Code*): AppSheet menekankan konsep "*no-code*" atau "*low-code*," memungkinkan pengguna yang tidak memiliki latar belakang pengembangan perangkat lunak untuk membuat aplikasi dengan mudah. Ini dilakukan melalui antarmuka pengguna grafis yang intuitif.
2. Integrasi Data: AppSheet memungkinkan pengguna untuk mengintegrasikan data dari berbagai sumber, termasuk Google Sheets, Excel, SQL databases, dan layanan *cloud* lainnya. Hal ini mempermudah akses dan pengelolaan data untuk aplikasi yang dibuat.
3. Antarmuka Pengguna (UI) Adaptif: Aplikasi yang dibuat dengan AppSheet memiliki antarmuka pengguna yang responsif dan dapat disesuaikan. Hal ini memungkinkan pembuat aplikasi untuk mendesain pengalaman pengguna sesuai dengan kebutuhan bisnis mereka.
4. Otomatisasi Bisnis: AppSheet memungkinkan pengguna untuk mengotomatiskan berbagai tugas bisnis dengan menambahkan logika dan aturan bisnis ke dalam aplikasi. Ini dapat mencakup alur kerja (*workflow*) otomatis, validasi data, dan fungsi otomatis lainnya.
5. Aplikasi *Cross-Platform*: Aplikasi yang dibuat dengan AppSheet dapat diakses dan digunakan melalui berbagai *platform*, termasuk perangkat seluler (Android dan iOS) serta *browser web*, meningkatkan fleksibilitas dan keterjangkauan pengguna.
6. Keamanan dan Izin: AppSheet menyediakan fitur keamanan dan pengaturan izin yang memungkinkan pengguna untuk mengontrol akses ke data dan fungsi aplikasi, memastikan bahwa informasi sensitif tetap terlindungi.

AppSheet telah menjadi solusi yang populer untuk banyak organisasi yang ingin mengembangkan aplikasi bisnis tanpa melibatkan tim pengembang perangkat lunak yang khusus. Ini mempercepat siklus

pengembangan dan memungkinkan berbagai departemen dalam suatu perusahaan untuk mandiri membuat dan mengelola aplikasi yang sesuai dengan kebutuhan mereka menurut AppSheet (2023).

2.2. Google Sheets

Google Sheets adalah aplikasi *spreadsheet* berbasis *cloud* yang dikembangkan oleh Google. Sebagai bagian dari Google Workspace (sebelumnya dikenal sebagai G Suite), Google Sheets memberikan kemampuan untuk membuat, mengedit, dan berbagi *spreadsheet* secara *online*. Google Sheets dan Microsoft Excel memiliki kesamaan namun memiliki kelebihan dan kekurangan masing-masing, menurut Yudhy (2023). Berikut adalah beberapa fitur dan aspek penting dari Google Sheets:

1. Akses Berbasis *Cloud*: Google Sheets di-*hosting* di *cloud*, memungkinkan pengguna untuk mengakses dan mengedit *spreadsheet* dari berbagai perangkat yang terhubung ke internet. Ini memfasilitasi kolaborasi tim secara *real-time*.
2. Kolaborasi *Real-Time*: Salah satu keunggulan utama Google Sheets adalah kemampuan untuk berkolaborasi secara *real-time*. Pengguna dapat mengundang orang lain untuk melihat atau mengedit *spreadsheet* yang sama, memungkinkan tim bekerja bersama-sama tanpa konflik versi.
3. Antarmuka Pengguna yang Ramah Pengguna: Google Sheets menawarkan antarmuka pengguna yang ramah pengguna dengan banyak fitur seperti rumus, grafik, dan alat analisis data. Antarmuka yang mirip dengan *spreadsheet* tradisional memudahkan pengguna yang sudah terbiasa dengan perangkat lunak sejenis.
4. Integrasi dengan Aplikasi Google Lainnya: Google Sheets terintegrasi dengan berbagai aplikasi Google lainnya, seperti Google Docs dan Google Slides. Ini memungkinkan pengguna menggabungkan data dari berbagai sumber dan membuat presentasi atau dokumen berbasis data.
5. Fungsi dan Rumus: Google Sheets menyediakan berbagai fungsi dan rumus untuk melakukan perhitungan dan analisis data. Fungsi ini mencakup fungsi matematika, statistik, logika, dan banyak lagi, memungkinkan pengguna membuat *spreadsheet* yang lebih kompleks.
6. *Autosave* dan Revisi: Google Sheets secara otomatis menyimpan perubahan yang dilakukan oleh pengguna, meminimalkan risiko kehilangan data. Selain itu, pengguna dapat melihat revisi atau versi sebelumnya dari *spreadsheet* dan mengembalikan ke titik waktu tertentu.
7. Keamanan dan Pengaturan Izin: Google Sheets menyediakan kontrol akses tingkat tinggi dengan pengaturan izin yang dapat disesuaikan. Pengguna dapat menentukan siapa yang dapat melihat, mengedit, atau berkolaborasi pada *spreadsheet*.

Dengan kelebihan-kelihatannya, Google Sheets menjadi pilihan populer untuk pengelolaan data, analisis, dan kolaborasi tim. Keberadaannya sebagai layanan *cloud* memudahkan akses dan penggunaan tanpa memerlukan instalasi perangkat lunak tambahan menurut Google (2023).

3. HASIL DAN PEMBAHASAN

3.1. Analisa Kebutuhan Perangkat Lunak

Rancangan perangkat lunak adalah proses yang mencakup perencanaan dan penentuan struktur dan karakteristik sistem perangkat lunak yang akan dikembangkan. Ini mencakup pemodelan arsitektur, antarmuka pengguna, struktur data, serta algoritma yang digunakan dalam perangkat lunak tersebut. Analisis sistem melibatkan pemeriksaan mendalam terhadap operasi sistem yang ada, mengevaluasi aspek positif dan negatif, serta merinci kebutuhan yang harus terpenuhi dalam pengembangan sistem yang baru (Rosa 2019). Analisis kebutuhan perangkat lunak adalah langkah kritis dalam perancangan aplikasi yang membantu dalam pemahaman dan dokumentasi kebutuhan fungsional dan non-fungsional dari aplikasi yang akan dikembangkan. Proses analisis ini mencakup identifikasi, dokumentasi, dan validasi kebutuhan, dan dapat melibatkan pihak-pihak terkait, seperti pengguna, pemilik bisnis, dan pengembang perangkat lunak. Berikut adalah beberapa aspek kunci dalam analisis kebutuhan perangkat lunak:

1. Identifikasi Pengguna dan *Stakeholder*:
 - a. Identifikasi semua pihak yang terlibat dan terpengaruh oleh aplikasi.
 - b. Memahami kebutuhan dan harapan pengguna akhir, serta kebutuhan bisnis dari pemilik atau pemangku kepentingan.

2. Wawancara dan Komunikasi:
 - a. Melakukan wawancara dengan pemangku kepentingan untuk mendapatkan pemahaman yang mendalam tentang kebutuhan mereka.
 - b. Membangun saluran komunikasi yang efektif antara tim pengembang dan pengguna akhir.
3. Pengumpulan dan Dokumentasi Kebutuhan:
 - a. Mengumpulkan kebutuhan fungsional, seperti fitur dan fungsi yang diinginkan.
 - b. Mengumpulkan kebutuhan non-fungsional, seperti keamanan, kinerja, dan persyaratan keandalan.
 - c. Membuat dokumen kebutuhan yang jelas dan dapat diukur.
4. Analisis Proses Bisnis:
 - a. Memahami proses bisnis yang ada dan menentukan bagaimana aplikasi akan berinteraksi dengan atau memperbaiki proses tersebut.
 - b. Menyusun alur kerja dan diagram untuk menggambarkan hubungan antara berbagai elemen dalam aplikasi.
5. Prioritisasi Kebutuhan:
 - a. Menetapkan prioritas kebutuhan berdasarkan urgensi dan dampak bisnis.
 - b. Mengidentifikasi kebutuhan yang krusial untuk versi pertama aplikasi dan yang dapat ditangani dalam rilis berikutnya.
6. Validasi Kebutuhan:
 - a. Mengkomunikasikan dan memvalidasi kebutuhan dengan pemangku kepentingan untuk memastikan pemahaman yang sama.
 - b. Mengidentifikasi dan menanggapi perubahan kebutuhan selama siklus pengembangan.
7. Prototyping (Opsional):
 - a. Membuat prototipe atau model awal untuk membantu pemangku kepentingan memvisualisasikan bagaimana aplikasi akan berfungsi.
 - b. Menggunakan prototipe untuk mengidentifikasi kebutuhan tambahan atau perubahan yang mungkin diperlukan.

Analisis kebutuhan perangkat lunak bertujuan untuk menciptakan dasar pemahaman yang kokoh untuk desain dan pengembangan aplikasi. Proses ini membantu mengurangi risiko kelalaian atau kesalahpahaman dalam pengembangan dan memastikan bahwa aplikasi yang dihasilkan memenuhi harapan dan kebutuhan pengguna dan pemilik bisnis.

3.2. Desain

1. Rancangan *Database*

Sebelum merancang aplikasi, hal pertama yang dilakukan adalah merancang *database*, dalam hal ini menggunakan Google Sheets. Salah satu pertimbangan menggunakan Google Sheets adalah, selain disimpan di *cloud* dan mudah dalam berkolaborasi, Google Sheets ini juga sudah terintegrasi dengan AppSheet, sehingga sangat memudahkan dalam perancangan aplikasi.

FormID	HASID	Nama	Waktu	Foto	Lokasi
0b8cdfba	HAS200276	Supriyanto	31/07/2023 14.16.16	data_images/0b8cdfba.Foto.071407.jpg	0.000000, 0.000000
f9ad78e7	HAS200281	Santi Yulia	31/07/2023 14.13.59	data_images/f9ad78e7.Foto.071446.jpg	-7.911605, 112.652796
e17c03b5	HAS200280	Wulandari	31/07/2023 14.09.24	data_images/e17c03b5.Foto.070954.jpg	-6.916390, 107.610006
28a3e40e	HAS200276	Supriyanto	31/07/2023 13.48.49	data_images/28a3e40e.Foto.064641.jpg	0.000000, 0.000000
4cebecd0	HAS200280	Wulandari	31/07/2023 13.09.54	data_images/4cebecd0.Foto.061012.jpg	-6.908255, 107.614377
3d1280f1	HAS200276	Supriyanto	31/07/2023 13.02.34	data_images/3d1280f1.Foto.060022.jpg	0.000000, 0.000000
f0a3b789	HAS200315	Budyanto Setyo	31/07/2023 12.07.45	data_images/f0a3b789.Foto.050822.jpg	-7.706169, 110.603332
52103cc9	HAS200280	Wulandari	31/07/2023 12.06.02	data_images/52103cc9.Foto.050621.jpg	-6.894401, 107.615415
3552b107	HAS200281	Santi Yulia	31/07/2023 11.56.59	data_images/3552b107.Foto.045717.jpg	-7.948191, 112.657745
5ec131b0	HAS200281	Santi Yulia	31/07/2023 11.43.10	data_images/5ec131b0.Foto.044348.jpg	-7.973310, 112.637854
ee7fd917	HAS200315	Budyanto Setyo	31/07/2023 11.41.00	data_images/ee7fd917.Foto.044137.jpg	-7.702424, 110.575101
24a6ca85	HAS200315	Budyanto Setyo	31/07/2023 11.33.38	data_images/24a6ca85.Foto.043642.jpg	-7.702422, 110.575096
67033862	HAS200276	Supriyanto	31/07/2023 11.29.10	data_images/67033862.Foto.042701.jpg	0.000000, 0.000000
52a68107	HAS200276	Supriyanto	31/07/2023 10.31.43	data_images/52a68107.Foto.032934.jpg	0.000000, 0.000000
79859299	HAS200281	Santi Yulia	31/07/2023 10.15.46	data_images/79859299.Foto.031613.jpg	-7.976891, 112.627923
eafecfb0	HAS200280	Wulandari	31/07/2023 09.19.42	data_images/eafecfb0.Foto.022000.jpg	-6.912663, 107.649118
2f3b9920	HAS200276	Supriyanto	31/07/2023 09.02.30	data_images/2f3b9920.Foto.020042.jpg	0.000000, 0.000000
f725f6e1	HAS200312	Ade Kurnia	31/07/2023 08.20.32	data_images/f725f6e1.Foto.012112.jpg	-6.993864, 110.411640
418a14e3	HAS200276	Supriyanto	31/07/2023 07.56.19	data_images/418a14e3.Foto.005405.jpg	0.000000, 0.000000
7f23ec8f	HAS200280	Wulandari	29/07/2023 14.21.19	data_images/7f23ec8f.Foto.072136.jpg	-6.922669, 107.651308
a5887fca	HAS200312	Ade Kurnia	29/07/2023 14.10.29	data_images/a5887fca.Foto.071052.jpg	-6.983725, 110.437542
47a5ab6f	HAS200312	Ade Kurnia	29/07/2023 13.45.18	data_images/47a5ab6f.Foto.064600.jpg	-7.008657, 110.435913
ad3a07af	HAS200281	Santi Yulia	29/07/2023 13.31.26	data_images/ad3a07af.Foto.063147.jpg	-7.911594, 112.652801
378f55e9	HAS200312	Ade Kurnia	29/07/2023 13.22.06	data_images/378f55e9.Foto.062244.jpg	-6.987864, 110.436066
036d747c	HAS200276	Supriyanto	29/07/2023 13.04.44	data_images/036d747c.Foto.060222.jpg	0.000000, 0.000000

Gambar 1. Rancangan database di Google Sheets

2. Rancangan Form Aplikasi

Dari database yang telah dibuat, selanjutnya ditarik melalui AppSheet. Selanjutnya setiap kolom pada tabel, disesuaikan tipe data, aturan, maupun formulanya jika dibutuhkan. Seperti pada gambar 2 di bawah ini.

NAME	TYPE	KEY?	LABEL?	FORMULA
_RowNumber	Number	<input type="checkbox"/>	<input type="checkbox"/>	=
FormID	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>	=
HASID	Ref	<input type="checkbox"/>	<input checked="" type="checkbox"/>	=
Nama	Text	<input type="checkbox"/>	<input type="checkbox"/>	= [H/
Tanggal	Date	<input type="checkbox"/>	<input type="checkbox"/>	=
Area	Text	<input type="checkbox"/>	<input type="checkbox"/>	= [H/
NOO	Number	<input type="checkbox"/>	<input type="checkbox"/>	=
Call	Number	<input type="checkbox"/>	<input type="checkbox"/>	=
Effective Call	Number	<input type="checkbox"/>	<input type="checkbox"/>	=

Gambar 2. Rancangan Setingan Kolom-kolom

Dari setingan tampilan-tampilan kolom ini, nanti secara otomatis akan berubah menjadi *field-field form* pada aplikasi. Ditunjukkan pada gambar 3 di bawah ini.

FORMULA	SHOW?	EDITABLE?	REQUIRE?	INITIAL VALUE
=	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	=
=	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	= UNIQUEID()
=	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	=
= [HASID] . [Nama]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	=
=	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	= TODAY()
= [HASID] . [Area]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	=
=	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	=
=	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	=
=	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	=

Gambar 3. Setingan Kolom-kolom

Pada gambar 3 ada beberapa kolom yang dimasukkan formula dan *initial value*. Kolom *Form ID* diisi initial value fungsi `uniqueid()`, yang membuat isi dari *Form ID* secara otomatis terisi dengan nilai yang unik. Kolom *Nama* dan *Area* merupakan referensi dari tabel *Master Data*. Sedangkan tabel tanggal secara otomatis mengambil data tanggal hari ini.

data : Lokasi
 type: LatLong Done

You can also provide a 'Show_If' expression to decide.

Type
 Column data type: LatLong

Type Details

Optional Url for KML File

Data Validity

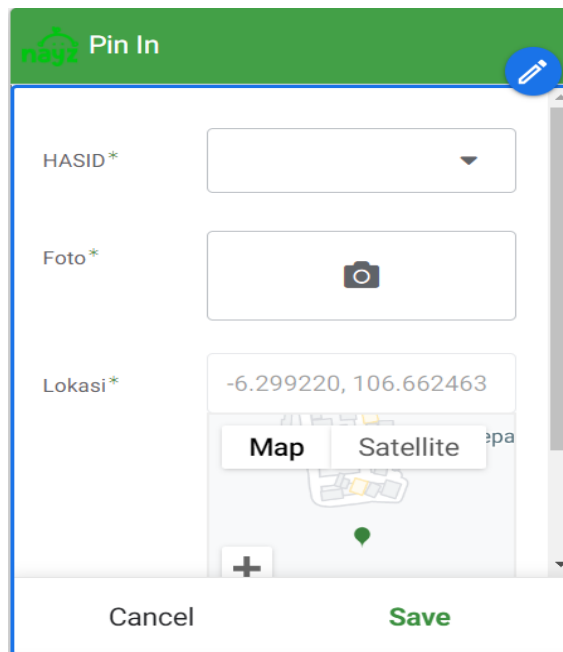
Auto Compute

App formula
 Compute the value for this column instead of allowing user input. =

Initial value
 The default starting value for this column. = HERE()

Gambar 4. Setingan Kolom Lokasi

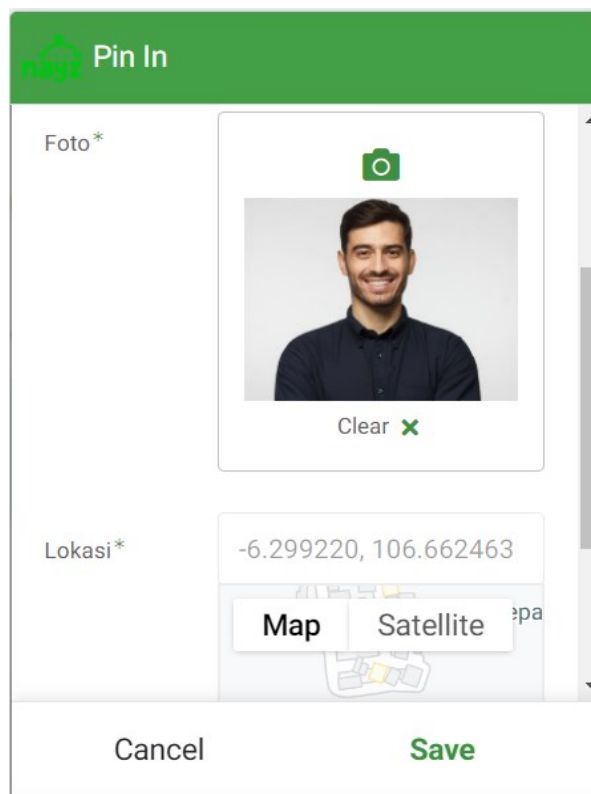
Untuk kolom Lokasi, tipe kolomnya diset menjadi LatLong dan initial value diisi fungsi HERE(). Hal ini agar aplikasi langsung bisa membaca titik lokasi pada saat aplikasi dijalankan. Tampilan formnya ditunjukkan pada gambar 5 di bawah ini.



Gambar 5. Rancangan *Form* Aplikasi

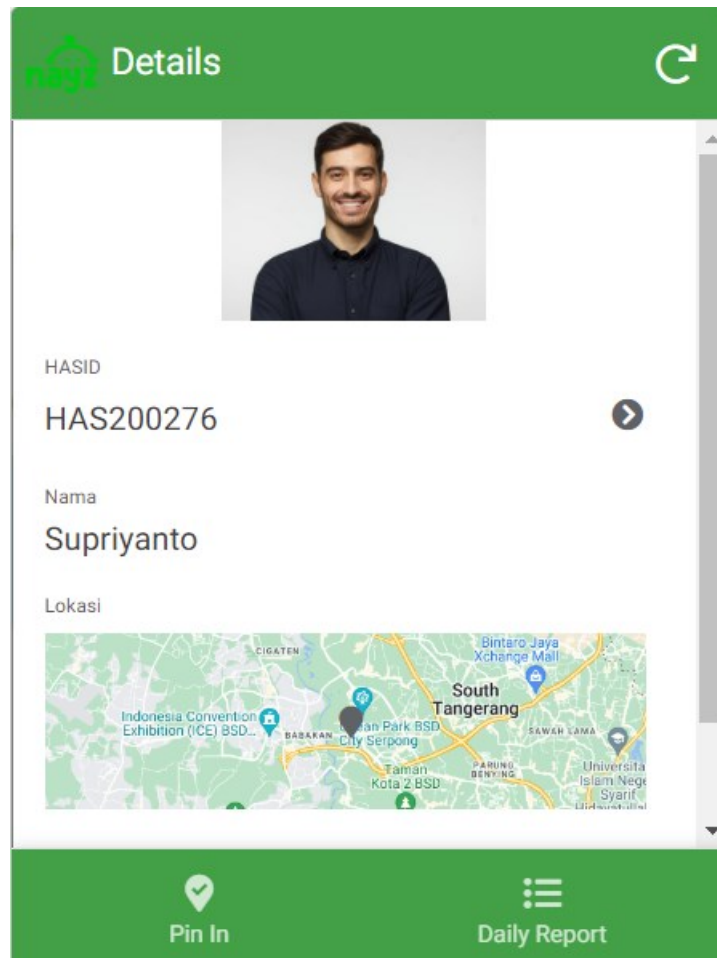
3.3. Implementasi

Aplikasi ini sudah selesai dibuat dan digunakan oleh sales PT. Nutrisi Juara Asia. Tampilan penggunaan bisa dilihat pada gambar 6 di bawah ini.



Gambar 6. Tampilan Aplikasi Pin-In saat diisi

Saat aplikasi dijalankan, sistem akan langsung mendeteksi lokasi sales saat itu juga dan akan menyimpan kordinat lokasi. Aplikasi ini juga meminta sales memasukkan foto selfie di tempat, sebagai bukti penguat.



Gambar 7. Tampilan Aplikasi saat Data disimpan

Saat form diisi dan disimpan, aplikasi langsung menunjukkan peta dengan ikon Pin yang sesuai titik dimana sales berada.

3.4. Pengujian

Sebelum aplikasi digunakan, dilakukan pengujian terlebih dahulu menggunakan beberapa perangkat oleh beberapa orang.

1. Pengujian Foto *Selfie*

Aplikasi coba menangkap foto selfie yang dilakukan oleh pengguna, dengan cara mengaktifkan kamera dan menyimpan foto langsung di *cloud*.

Tabel 1. Pengujian Foto *Selfie*

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
Foto melalui kamera	Kamera aktif dan foto langsung terupload di cloud	Kamera aktif dan foto ada di cloud	[x] Diterima [] Ditolak

2. Pengujian Pembacaan Lokasi

Aplikasi diharapkan langsung dapat membaca titik lokasi perangkat yang digunakan.

Tabel 3. Pengujian *pembacaan lokasi*

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
Aplikasi membaca kordinat perangkat	Kordinat terdetek dan tersimpan dalam bentuk <i>latitude</i> dan <i>longitude</i>	Kordinat tersimpan dan lokasi sesuai dengan posisi perangkat	[x] Diterima [] Ditolak

3. Pengujian *Input, Edit dan Delete* data Absensi

Berikut tabel pengujian *input, edit dan delete* data absensi.

Tabel 6. Pengujian *input* data absensi

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Input</i> data absensi dan detailnya	Data instruktur masuk ke dalam <i>database</i>	Data instruktur ada di <i>database</i>	[x] Diterima [] Ditolak

Tabel 7. Pengujian *edit* data absensi

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Edit</i> data absensi	Data absensi baru mengubah data lama di <i>database</i>	Data absensi baru ada di <i>database</i>	[x] Diterima [] Ditolak

Tabel 8. Pengujian *delete* data absensi

Data Masukan	Kejadian yang diharapkan	Pengamatan	Kesimpulan
<i>Delete</i> data absensi dan detailnya	Data absensi hilang dari <i>database</i>	Data absensi hilang dari <i>database</i>	[x] Diterima [] Ditolak

4. SIMPULAN

Kesimpulan yang dapat diambil setelah melakukan penelitian terhadap Perancangan Aplikasi Pin In Menggunakan AppSheet Dan Google Sheets adalah sebagai berikut:

1. Dengan menggunakan aplikasi khusus, maka pelaporan absensi menjadi lebih mudah dan efisien.
2. AppSheet membuat pembuatan aplikasi pelaporan absensi menjadi mudah karena tidak perlu mempelajari koding.
3. Dengan menggunakan aplikasi yang dirancang ini, terbukti pelaporan absensi menjadi mudah dan efisien.

DAFTAR PUSTAKA

- AppSheet. (2023). AppSheet: No-code application development platform. Diakses pada 13 November 2023. <https://www.appsheet.com/>
- Hamid, K., Widya, A., Ilham, K., & Dede, F. (2020). Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Penggajian Pada SMK Bina Karya Karawang. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 14(4), 58.
- Rosa, A.S., dan M. Shalahuddin. (2019). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek Edisi Revisi*. Bandung: Informatika
- Sholikhah, I., Sairan, M., & Syamsiah, N.O. (2017). Aplikasi Pembelian Dan Penjualan Barang Dagang Pada Cv Gemilang Muliatama Cikarang. *Jurnal Teknik Komputer Amik BSI*, III(1), 16–23.
- Suharna. (2021). Membuat Aplikasi Supervisi Tanpa Koding dengan AppSheet. *Deepublish*.
- Ushud, A.A.A. (2019). Perancangan Website Resepbayi.com Menggunakan Avada Untuk WordPress. *Jurnal Maklumatika*, 6(1), 10-19.
- Wicaksono , Yudhy., & Solusi Kantor. (2023). *Panduan Lengkap Excel dan Google Sheets untuk Pemula*. Elex Media Komputindo.