

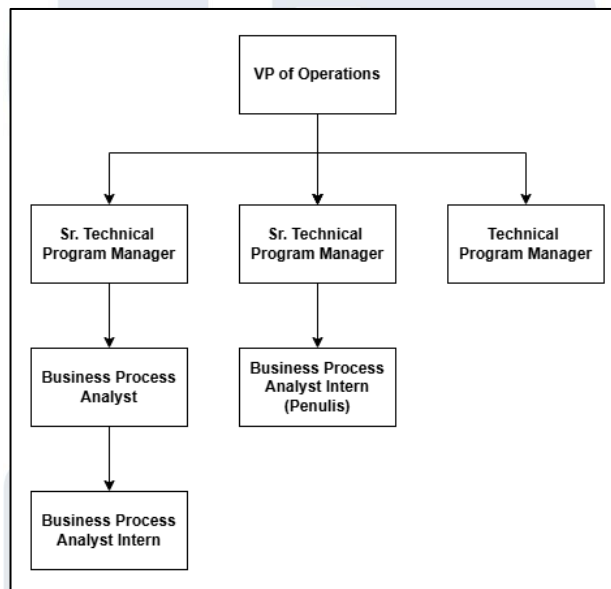
BAB III

PELAKSANAAN KERJA

3.1 Kedudukan dan Koordinasi

Bagian ini menjelaskan posisi atau kedudukan penulis selama melaksanakan kegiatan magang di PT Vidio Dot Com serta menjabarkan alur koordinasi kerja yang dilakukan antara penulis, pembimbing lapangan, dan pihak-pihak terkait lainnya dalam pelaksanaan proyek pengembangan sistem internal perusahaan.

3.1.1 Kedudukan



Gambar 3. 1 Kedudukan Business Process Analyst Internship

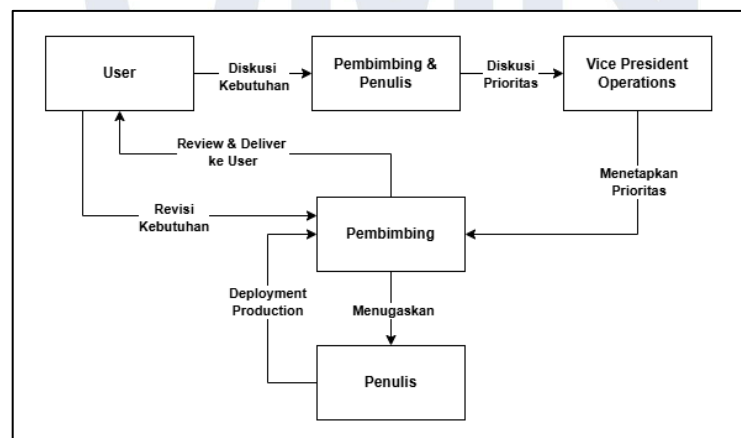
Berdasarkan struktur organisasi yang ditunjukkan pada Gambar 3.1, penulis menempati posisi sebagai Business Process Analyst Intern di bawah Divisi Operations PT Vidio Dot Com. Divisi ini berada di bawah tanggung jawab Vice President of Operations (VP of Operations) yang memiliki peran strategis dalam memastikan kelancaran kegiatan operasional perusahaan, termasuk pengelolaan dan optimalisasi sistem internal yang mendukung berbagai tim lintas fungsi.

Dalam struktur organisasi, posisi Business Process Analyst Intern berada di bawah supervisi Sr. Technical Program Manager, yang berperan sebagai pembimbing lapangan selama program magang berlangsung. Sr. Technical Program Manager bertanggung jawab dalam mengoordinasikan berbagai proyek pengembangan sistem internal dan memastikan keselarasan antara kebutuhan operasional dengan solusi teknis yang dikembangkan.

Sebagai Business Process Analyst Intern, penulis terlibat dalam berbagai kegiatan pendukung operasional, seperti analisis kebutuhan bisnis dan teknis, pemodelan proses kerja, penyusunan dokumen spesifikasi sistem, serta pengujian dan evaluasi fungsionalitas sistem yang sedang dikembangkan. Selain itu, penulis juga turut membantu dalam proses revisi dan deployment setelah sistem melewati tahap validasi dari tim terkait.

Melalui kedudukan ini, penulis memperoleh pengalaman langsung mengenai bagaimana koordinasi lintas divisi, penentuan prioritas, serta proses pengambilan keputusan dilakukan di lingkungan perusahaan teknologi media berskala besar seperti PT Vidio Dot Com. Kedudukan ini juga memberikan wawasan praktis mengenai penerapan metode analisis bisnis dan implementasi sistem yang dilakukan secara kolaboratif antara tim teknis, analis, dan pihak manajemen operasional.

3.1.2 Koordinasi



Gambar 3. 2 Alur Koordinasi Pengerjaan Proyek

Selama menjalankan program magang, penulis menjalani proses koordinasi yang sistematis dengan beberapa pihak internal perusahaan, seperti User, Pembimbing Lapangan yaitu Sr. Technical Program Manager), Penulis yaitu Business Process Analyst Intern, serta Vice President of Operations. Proses ini digambarkan pada Gambar 3.2 yang menunjukkan alur komunikasi dan mekanisme kerja dalam pelaksanaan proyek pengembangan sistem internal Perusahaan.

Tahapan koordinasi dimulai dari User yang memerlukan pengembangan, perbaikan, atau pembaruan terhadap sistem kerja yang digunakan. User akan melakukan diskusi kebutuhan bersama pembimbing dan penulis untuk menjelaskan kendala, kebutuhan fungsional, maupun pengembangan proses kerja yang diinginkan.

Setelah proses identifikasi kebutuhan selesai, pembimbing dan penulis melakukan diskusi prioritas dengan Vice President of Operations untuk menentukan urgensi serta skala prioritas berdasarkan dampak terhadap operasional dan ketersediaan sumber daya. Keputusan yang diambil oleh VP Operations menjadi dasar dalam penentuan proyek atau inisiatif yang akan dilanjutkan pada tahap pengembangan.

Selanjutnya, pembimbing menugaskan penulis untuk melaksanakan aktivitas analisis kebutuhan, penyusunan flow proses, serta membantu dalam pengembangan sistem menggunakan Google Spreadsheet, AppSheet, dan Google Apps Script sebagai teknologi utama. Setelah tahap pengembangan selesai, hasil pekerjaan penulis akan melalui proses review dan uji coba bersama User untuk memastikan kesesuaian dengan kebutuhan bisnis dan standar perusahaan.

Apabila terdapat masukan atau permintaan perubahan, User akan menyampaikan revisi kepada pembimbing yang kemudian diteruskan kepada penulis untuk dilakukan penyesuaian. Proses ini bersifat iteratif dan

kolaboratif, sehingga setiap pengembangan yang dilakukan selalu mengacu pada umpan balik pengguna dan keputusan manajemen operasional.

Koordinasi yang dijalankan selama masa magang menekankan pada prinsip komunikasi terbuka, kolaboratif, dan berjenjang. Setiap keputusan terkait pengembangan atau perbaikan sistem selalu melalui proses diskusi lintas peran antara User, Pembimbing, dan VP Operations untuk memastikan keselarasan antara kebutuhan teknis dengan tujuan operasional perusahaan.

Melalui mekanisme koordinasi tersebut, penulis tidak hanya memahami aspek teknis dalam pengembangan sistem, tetapi juga memperoleh pemahaman mendalam mengenai dinamika koordinasi profesional di lingkungan perusahaan.

3.2 Tugas yang Dilakukan

Selama melaksanakan program magang di PT Vidio Dot Com, penulis berperan dalam proses pengembangan sistem internal berbasis data, khususnya pada proyek Sport VOD Calendar dan proyek Improvment PT XYZ Digital Systems. Tugas yang dilakukan meliputi analisis kebutuhan pengguna, perancangan alur sistem, penyusunan struktur database, implementasi aplikasi menggunakan AppSheet, hingga pembuatan automasi proses menggunakan Apps Script.

Seluruh aktivitas dilakukan untuk mendukung efisiensi operasional tim internal serta memastikan alur pencatatan dan pelaporan konten berjalan lebih terstruktur. Pelaksanaan tugas dilakukan secara bertahap, dimulai dari orientasi kerja hingga implementasi aplikasi ke tahap produksi. Penulis berinteraksi dengan beberapa tim seperti Sports, Editor, Conten Manager dan team sales dan marketing PT XYZ Digital untuk mengidentifikasi kebutuhan fitur dan memastikan sistem yang dikembangkan sesuai dengan proses bisnis yang berjalan.

Tabel 3. 1 Detail Pekerjaan yang Dilakukan

No	Kegiatan	Tanggal Mulai	Tanggal Selesai
Proyek 1 - VOD Sport Calender			
1	Diskusi dengan tim terkait untuk mengidentifikasi kebutuhan proyek.	10 April 2025	24 April 2025
2	Menganalisis dan menyusun alur kerja aplikasi Sport VOD Calendar	14 April 2025	30 April 2025
3	Menyusun struktur database pencatatan konten berbasis Google Spreadsheet/AppSheet.	5 Mei 2025	8 Juli 2025
4	Mengembangkan tampilan aplikasi, form input, dan workflow otomatis	8 Mei 2025	3 Juni 2025
5	Pengujian internal, perbaikan bug, dan penyesuaian fitur berdasarkan masukan pengguna	1 Mei 2025	2 Juni 2025
6	Perbaikan bug, penyesuaian fitur berdasarkan feedback, pengoptimalan alur input dan automations	2 Juni 2025	4 Juli 2025
7	Melakukan peluncuran aplikasi serta memberikan panduan penggunaan kepada tim terkait	2 Juni 2025	4 Juni 2025
8	Monitoring penggunaan aplikasi dan memastikan alur pencatatan berjalan tanpa kendala	2 Juni 2025	11 Juli 2025
Proyek 2 – Imporvement PT XYZ Digital Systems			
9	Pemahaman aplikasi PT XYZ (Pre-Sales) dan analisis kebutuhan improvement	10 Juli 2025	17 Juli 2025
10	Improvement pengembangan Apps Script untuk database user	28 Juli 2025	8 Agustus 2025
11	Wawancara kebutuhan Brand dan Agency dengan User	11 Agustus 2025	22 Agustus 2025
12	Pengembangan database Brand & Agency	28 Agustus 2025	19 September 2025
13	Pengembangan Brand & Agency ke Production	8 September 2025	26 September 2025
14	Testing sistem dan proses rollout aplikasi	29 September 2025	30 September 2025

Seluruh rangkaian kegiatan magang sebagaimana dirangkum pada **Tabel 3.1** dilaksanakan secara bertahap, dimulai dari proses orientasi hingga keterlibatan langsung dalam pengembangan sistem internal di PT Vidio Dot Com. Kegiatan tersebut mencakup proses pengumpulan kebutuhan pengguna, analisis dan penyusunan alur kerja aplikasi, perancangan struktur database, pengembangan fitur sistem, hingga tahap pengujian dan implementasi aplikasi pada lingkungan operasional perusahaan.

Dalam pelaksanaannya, setiap tahapan pekerjaan dilakukan secara kolaboratif melalui diskusi rutin dengan tim Sports, Editor, Content Manager, serta tim Sales dan Marketing PT XYZ Digital. Proses pengembangan sistem dilakukan secara iteratif, dimulai dari perancangan awal, pengembangan fitur, pengujian internal, pencatatan umpan balik pengguna, hingga penyempurnaan sistem agar sesuai dengan kebutuhan bisnis yang berjalan. Pendekatan ini bertujuan untuk memastikan bahwa sistem yang dikembangkan tidak hanya stabil secara teknis, tetapi juga mampu meningkatkan efisiensi dan keteraturan alur kerja operasional.

Selain pengembangan aplikasi utama Sport VOD Calendar, penulis juga terlibat dalam proses pemahaman alur kerja PT XYZ Pre-Sales serta pengembangan database dan automasi untuk kebutuhan Brand dan Agency. Aktivitas tersebut meliputi analisis dokumen bisnis, pemetaan alur data, serta penyusunan struktur database yang mendukung proses input, monitoring, dan pelaporan secara lebih terstruktur. Untuk mendukung seluruh rangkaian kegiatan tersebut, digunakan beberapa tools dan platform pendukung yang berperan penting dalam proses perancangan, pengembangan, serta automasi sistem. Penjelasan mengenai tools yang digunakan selama pelaksanaan kerja magang, beserta fungsi dan perannya dalam mendukung pengembangan sistem di PT Vidio Dot Com, akan diuraikan pada subbab berikut.

1) Appsheet



Gambar 3. 3 Logo Appsheet

AppSheet adalah platform no-code yang dikembangkan oleh Google untuk membangun aplikasi berbasis data tanpa memerlukan kemampuan pemrograman. Platform ini memungkinkan pengguna membuat aplikasi langsung dari sumber data seperti Google Sheets, Excel, SQL, atau berbagai layanan cloud lainnya [17], [18], [19]. Dengan AppSheet, pembuatan aplikasi dapat dilakukan secara cepat melalui antarmuka visual yang intuitif, di mana pengguna cukup menentukan struktur data, logika bisnis, serta tampilan aplikasi, dan AppSheet akan secara otomatis menghasilkan aplikasi yang siap digunakan [20].

AppSheet menyediakan berbagai fitur seperti form, tampilan tabel, kartu, kalender, grafik, hingga otomasi proses yang dapat merespons perubahan data atau menjalankan alur kerja tertentu [20]. Selain itu, AppSheet mendukung logika perhitungan dan kondisi melalui expressions, yang memungkinkan pengguna membangun proses kompleks seperti validasi input, filtering data, pengiriman notifikasi, atau pembaruan otomatis antar tabel [20].

Keunggulan AppSheet terletak pada fleksibilitas, integrasi yang kuat dengan ekosistem Google, serta kemampuannya untuk berjalan di berbagai perangkat (web, Android, dan iOS) tanpa perlu melakukan pengembangan terpisah [19]. Hal ini menjadikannya alat yang sangat efektif untuk membuat aplikasi operasional, manajemen data, dan alur kerja internal secara cepat, efisien, dan mudah diadaptasikan pada kebutuhan bisnis.

2) Apps Script



Gambar 3. 4 Logo Apps Script

Google Apps Script merupakan platform pemrograman berbasis JavaScript yang disediakan oleh Google untuk mengotomatisasi berbagai proses serta memperluas fungsi layanan dalam Google Workspace, seperti Google Sheets, Docs, Forms, dan Gmail [21]. Melalui Apps Script, pengguna dapat membuat automasi untuk mengelola data, misalnya menarik respons dari Google Form ke Google Sheets, melakukan pembersihan data, hingga menghitung statistik secara otomatis [21]. Selain itu, Apps Script juga memungkinkan penjadwalan tugas tertentu, seperti mengirim email pengingat atau menghasilkan laporan berkala dari data yang tersedia.

Keunggulan utama Apps Script terletak pada kemudahan penggunaannya karena dapat dijalankan langsung melalui browser tanpa instalasi tambahan, serta integrasinya yang kuat dengan seluruh ekosistem Google [22]. Kombinasi ini menjadikannya alat yang sangat efektif untuk meningkatkan efisiensi dalam sebuah proyek, terutama pada tahap pengumpulan, pengolahan, dan pembuatan laporan data secara otomatis serta terstruktur [21].

Secara teknis, Apps Script bekerja dengan konsep bound script dan standalone script, yang memungkinkan skrip melekat pada file Google tertentu (misalnya Google Sheets) atau berdiri sendiri. Platform ini mendukung penggunaan trigger seperti onEdit, onChange, dan time-driven triggers untuk menjalankan proses otomatis berdasarkan peristiwa tertentu [21]. Selain itu, Apps Script menyediakan akses penuh ke Google Workspace APIs, sehingga developer dapat

membuat fungsi kustom, memanipulasi spreadsheet secara dinamis, berinteraksi dengan layanan eksternal menggunakan URL Fetch, hingga membangun workflow otomatis yang kompleks dalam skala yang mudah dikelola [23].

3) Spreadsheet



Gambar 3. 5 Logo Google Sheets

Spreadsheet adalah sebuah alat pengolah data berbentuk tabel yang terdiri dari baris dan kolom, digunakan untuk menyimpan, mengorganisir, dan menganalisis informasi secara terstruktur [24]. Setiap perpotongan antara baris dan kolom disebut sel, dan setiap sel dapat berisi angka, teks, rumus, maupun fungsi otomatis. Spreadsheet memudahkan pengguna untuk melakukan perhitungan, membuat rangkuman data, hingga menampilkan hasil analisis dalam bentuk grafik [25].

Dalam konteks pengembangan sistem atau proyek kerja, spreadsheet sering digunakan sebagai basis data ringan karena fleksibel, mudah diakses, dan cepat dimodifikasi. Melalui formula dan fungsi bawaan seperti SUM, IF, VLOOKUP, atau ARRAYFORMULA, spreadsheet dapat melakukan pemrosesan data secara otomatis tanpa memerlukan program tambahan [20]. Selain itu, spreadsheet modern seperti Google Sheets memungkinkan kolaborasi real-time, sehingga beberapa pengguna dapat mengedit, memeriksa, atau memvalidasi data secara bersamaan.

Kemudahan integrasi dengan alat lain misalnya AppSheet, Google Apps Script, maupun API eksternal membuat spreadsheet sangat ideal untuk digunakan sebagai fondasi awal dalam pembuatan prototipe sistem, automasi proses, atau pengelolaan database berbasis cloud [20].

4) Draw.io



Gambar 3. 6 Logo Draw Io

Draw.io adalah sebuah aplikasi berbasis web yang digunakan untuk membuat berbagai jenis diagram, seperti flowchart, ERD, UML, wireframe, dan diagram proses bisnis [26]. Platform ini sangat populer karena gratis, mudah digunakan, dan dapat diakses langsung melalui browser tanpa instalasi tambahan [26]. Pengguna tinggal menarik (drag) dan meletakkan (drop) elemen diagram untuk membangun visualisasi yang dibutuhkan.

Salah satu keunggulan draw.io adalah kemampuannya untuk terintegrasi dengan layanan penyimpanan seperti Google Drive, OneDrive, dan GitHub. Hal ini memudahkan pengguna menyimpan, berbagi, serta mengedit diagram secara kolaboratif [27]. Selain itu, draw.io menyediakan ribuan bentuk (shapes) dan template bawaan yang mempercepat proses pembuatan diagram, mulai dari diagram teknis sistem hingga rancangan alur kerja organisasi [26].

Karena fleksibilitas dan fitur-fiturnya yang lengkap, draw.io sering digunakan dalam pengembangan sistem, dokumentasi proyek, perancangan arsitektur aplikasi, hingga kebutuhan akademik seperti pembuatan ERD atau flowchart.

3.3 Uraian Pelaksanaan Kerja

3.3.1 Proyek 1 - VOD Sport Calender

Pada bagian ini, penulis menjelaskan pekerjaan yang dilakukan dalam proyek Sport VOD Calendar, yaitu membuat sistem yang membantu mengatur dan mengotomasi alur kerja Video On Demand (VOD) untuk konten olahraga. Pengerjaan proyek ini meliputi memahami masalah di proses lama, membuat rancangan solusi, membangun aplikasi menggunakan

Appsheets dan Appscript serta mendampingi proses implementasinya. Seluruh proses dikerjakan dari tahap perancangan awal hingga aplikasi siap dipakai oleh tim Sport, Video Editor, dan Content Management.

3.3.1.1 Latar Belakang Proyek 1

Proyek Sport VOD Calendar berawal dari inisiatif VP Operation untuk memperbaiki alur kerja pembuatan konten Video On Demand (VOD). Sebelum sistem ini dikembangkan, seluruh proses mulai dari permintaan hingga penayangan VOD masih dilakukan secara manual melalui spreadsheet dan komunikasi di Google Chat. Cara kerja tersebut membuat koordinasi antar tim menjadi kurang efisien karena data sering tidak konsisten, informasi mudah berubah, dan tidak ada pencatatan yang benar-benar terstruktur.

Di sisi lain, proses Sport Live Stream sudah lebih tertata karena lebih dulu menggunakan Appsheets. Perbedaan inilah yang mendorong kebutuhan untuk menghadirkan sistem serupa pada pengelolaan VOD, agar seluruh alur kerja memiliki standar yang sama, lebih modern, dan mudah dipantau.

Pada alur permintaan VOD, tim Sport harus mengetik request secara manual, sementara tim Content Management menyalin data satu per satu ke CMS (Content Management System). Proses ini menimbulkan risiko seperti human error, keterlambatan penugasan, kesalahan input saat upload, serta duplikasi pekerjaan. Panjangnya rantai kerja dan tidak adanya pusat data yang jelas menjadi alasan utama perlunya sistem yang lebih tertata dan terintegrasi.

Berdasarkan kondisi tersebut, penulis bersama tim mengembangkan Sport VOD Calendar, sebuah aplikasi yang menggabungkan Appsheets, Appscript, dan Google Spreadsheet untuk mengotomasi seluruh tahapan dalam workflow. Mulai dari pengajuan request, penugasan editor, pengunggahan hasil edit, hingga

penyediaan data siap pakai untuk CMS (Content Management System) . Dengan adanya sistem ini, proses kerja menjadi lebih cepat, rapi, dan menyatu, sehingga mengurangi kesalahan serta membantu setiap tim bekerja lebih efektif.

3.3.1.2 Perancangan dan Implementasi Proses Proyek 1

Dalam tahap awal perancangan sistem Sport VOD Calendar, penulis melakukan wawancara dengan perwakilan Tim Sport, Editor Leads, Editor, serta tim Content Management untuk memahami proses kerja yang berlangsung. Hasil wawancara tersebut kemudian dituangkan ke dalam model SIPOC (Supplier - Input - Process - Output - Customer) sebagai alat untuk memetakan alur kerja dari hulu ke hilir.

Penyusunan SIPOC bertujuan untuk menggambarkan pihak-pihak yang terlibat, informasi yang digunakan, alur proses, keluaran yang dihasilkan, serta penerima keluaran. Dengan pemetaan ini, seluruh proses dari permintaan hingga penayangan VOD menjadi lebih jelas dan terstruktur, sehingga memudahkan identifikasi masalah dalam proses lama sekaligus kebutuhan perbaikan untuk sistem baru.

Process	Supplier (Responsible)	Input	Output (Deliverable)	Customer (Accountable)	Customer (Informed)	Artefak	Remaks
Request VOD	Team Sports	- Input Match - Input VOD Type - Source Materi	Informasi request VOD yang lengkap	Team Editor	Editor Leads	Spreadsheet Request VOD	Masih diketik manual di spreadsheet sehingga sering terjadi ketidaksesuaian data dan human error
Mention team editor leads	Team Sports	Google Chat: Informasikan request VOD	Notifikasi request diterima	Editor Leads	Team Editor	Google Chat message	Proses notifikasi dilakukan manual
Menugaskan editor	Editor Leads	Informasi request dari Team Sports	Assignment editor untuk pengerjaan	Editor	Team Editor	Spreadsheet Request VOD	Penugasan masih dilakukan manual
Edit Materi VOD	Team Editor	Materi mentah VOD	Video VOD yang sudah diedit	Team Content Manager	Editor Leads	File VOD edit	-
Upload Materi ke Share Drive	Team Editor	File VOD hasil edit	Link materi VOD	Team Content Manager	Team Editor	Shared Drive Folder	-
Notifikasi bahwa materi siap	Team Editor	Google Chat: Informasikan lokasi materi	Notifikasi bahwa materi siap	Team Content Manager	Team Editor	Google Chat message	Proses notifikasi dilakukan manual
Upload Materi VOD ke CMS	Team Content Manager	- Materi VOD - Informasi VOD	Materi berhasil tayang di Vidio.com	Team Sports	Team Editor	Masih dilakukan copy-paste dari spreadsheet ke CMS sehingga rawan salah input	Masih di lakukan Copy paste dari spreadsheet ke CMS

Gambar 3. 7 SIPOC Pre development VOD Matches

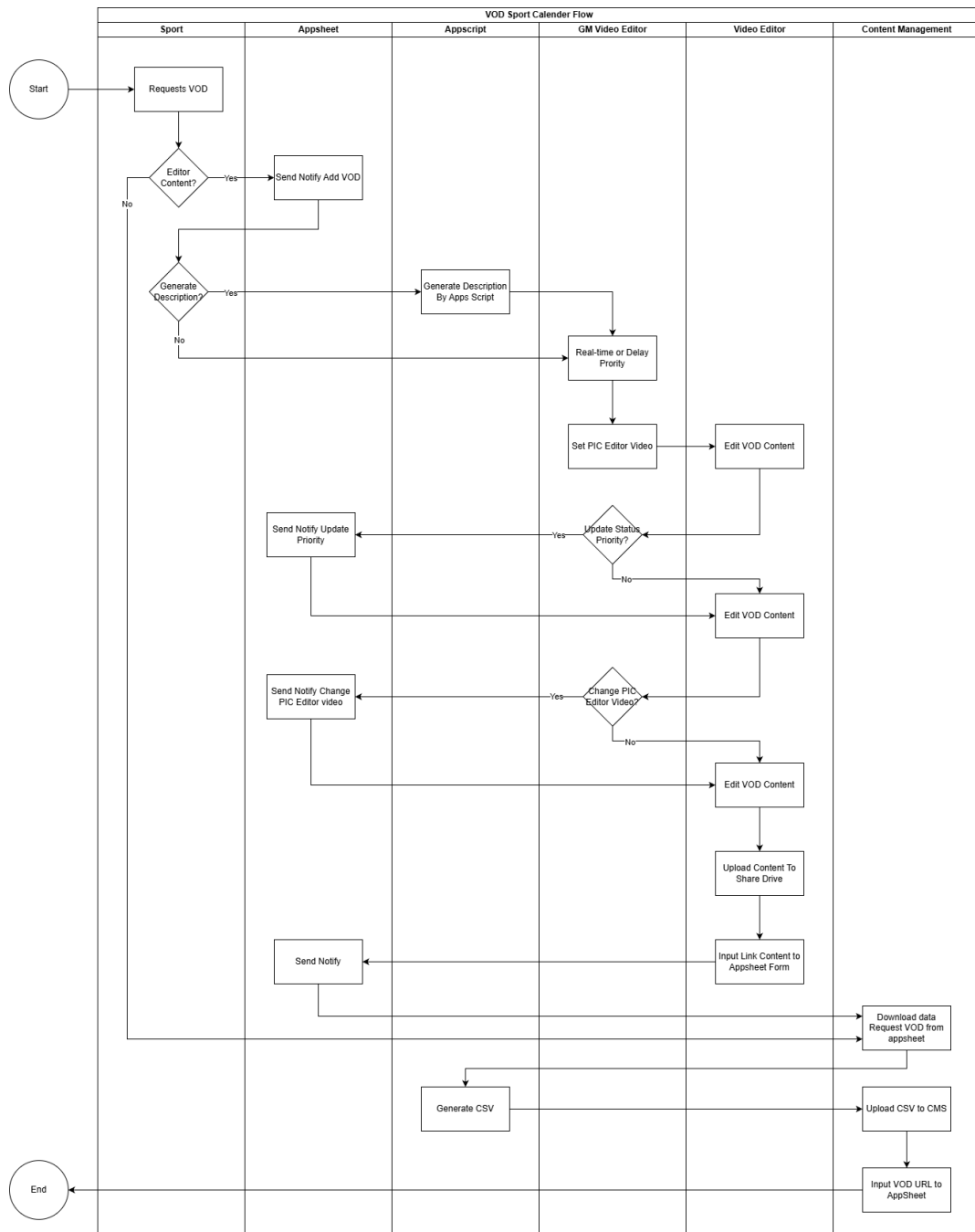
Dari hasil pemetaan pada gambar 3.7, terlihat bahwa alur kerja VOD sebelumnya melibatkan banyak titik manual, seperti pengetikan request di spreadsheet, penyampaian informasi melalui Google Chat, hingga penyalinan data secara copy-paste ke CMS (Content

Management System). Kondisi ini menunjukkan adanya risiko human error, ketidaksinkronan data, serta ketiadaan mekanisme otomatisasi dalam penugasan maupun notifikasi. SIPOC membantu memahami bahwa akar permasalahan utama terletak pada kurangnya integrasi antar tim dan ketiadaan sistem terpusat yang mengelola seluruh proses VOD secara end-to-end.

Selain itu, Tim Sport menyampaikan bahwa proses permintaan VOD sangat memakan waktu karena harus mengisi spreadsheet secara manual dan mengulang format yang sama setiap hari. Sementara itu, Content Manager menjelaskan bahwa proses upload ke CMS membutuhkan ketelitian tinggi, namun sering terjadi ketidaksinkronan data sehingga harus melakukan pengecekan ulang, yang memperlambat penayangan konten. Masukan dari masing-masing tim ini menjadikan SIPOC sebagai fondasi penting untuk merancang solusi yang benar-benar sesuai kebutuhan operasional.

Berdasarkan model SIPOC, kemudian dirancang alur kerja baru yang lebih ringkas dan terotomasi. Setiap permasalahan mulai dari request, penugasan, pengeditan, unggahan, hingga publikasi diterjemahkan menjadi kebutuhan fitur sistem. Dengan demikian, Appsheet dan Appscript tidak hanya digunakan untuk membangun aplikasi, tetapi juga untuk merepresentasikan ulang proses bisnis yang lebih ideal dan terstandarisasi.

Selanjutnya, dibuat diagram swimlane yang menggambarkan alur kerja baru sebagai dasar pembangunan aplikasi di Appsheet, yang memperlihatkan interaksi antar tim, titik keputusan, serta otomatisasi yang berjalan dalam sistem.



Gambar 3. 8 Swimlane Diagram Development: Proses VOD

Diagram Swimlane yang ditunjukkan pada Gambar 3.8 menetapkan Appsheet dan Appscript sebagai sistem terpusat atau Single Source of Truth, yang memecah alur kerja menjadi empat jalur interaksi utama: Tim Sport, Appsheet/Appscript, Video Editor, dan Content Management. Proses dimulai ketika Tim Sport mengajukan

permintaan VOD melalui formulir di Appsheet. Sistem akan secara otomatis memvalidasi permintaan dan, melalui Appscript, segera mengirimkan notifikasi penugasan kepada tim terkait, menggantikan komunikasi manual melalui Google Chat yang rentan terhadap human error. Selain itu, Appscript dapat menjalankan fungsi otomatisasi pembuatan deskripsi VOD, yang secara signifikan meminimalkan waktu dan risiko kesalahan pengetikan manual.

Jalur Video Editor kini menerima penugasan secara otomatis dan melakukan editing berdasarkan data yang terpusat. Setelah proses editing selesai, Editor wajib menginput tautan konten hasil edit kembali ke dalam Appsheet. Langkah ini memastikan bahwa status pekerjaan diperbarui secara real-time dan semua data yang diperlukan untuk publikasi telah tervalidasi serta terpusat.

Pada tahapan akhir, Content Management mendapatkan manfaat terbesar dari integrasi ini. Mereka tidak lagi perlu mengumpulkan metadata (judul, deskripsi, tautan drive, dll.) dari berbagai sumber yang tidak konsisten. Sebaliknya, mereka hanya mengunduh data metadata yang telah terintegrasi dalam format CSV langsung dari Appsheet untuk diunggah ke CMS. Pendekatan ini secara drastis mengurangi risiko human error saat penyalinan data, memungkinkan tim Content Management berfokus pada proses unggah dan validasi akhir.

Dengan demikian, Diagram Swimlane ini menjadi panduan teknis utama dalam pengembangan sistem. Diagram ini berhasil mengubah proses kerja yang sebelumnya terpecah-pecah dan manual menjadi sebuah sistem yang terpadu, otomatis, dan memiliki standar yang jelas. Hal ini memastikan bahwa setiap langkah, mulai dari permintaan awal konten hingga konten tersebut ditayangkan sebagai VOD, dapat berjalan dengan cepat dan efisien.

Selanjutnya, penulis membangun database untuk seluruh proses VOD, yang kemudian divisualisasikan dalam bentuk Entity Relationship Diagram (ERD). ERD ini menampilkan entitas utama seperti Tournament, Matches, Teams, Leagues, VOD_Matches, dan Users, serta hubungan antar entitas tersebut. Entitas Tournament menyimpan informasi turnamen atau kompetisi olahraga, termasuk judul, tanggal, jenis olahraga, dan musim, yang menjadi acuan bagi Matches untuk setiap pertandingan live stream yang terjadi.



Gambar 3. 9 ERD VOD Matches

Matches berfungsi sebagai database untuk Live Stream, mencatat data pertandingan, termasuk nama pertandingan, tim yang bertanding, tanggal, dan status pertandingan. Matches terhubung dengan Teams melalui atribut team_home dan team_away, dan dengan Tournament

melalui match_tournament. Teams menyimpan informasi mengenai nama tim, cabang olahraga, liga, negara, tag, dan gender, yang terhubung dengan Leagues sebagai referensi liga tempat tim tersebut berkompetisi. Struktur ini memastikan data pertandingan live stream tersimpan secara konsisten dan terorganisir.

Sementara itu, VOD_Matches merupakan database utama untuk request VOD. Entitas ini mencatat seluruh data terkait VOD, termasuk judul, deskripsi, tipe konten, tautan, status, catatan, dan metadata lainnya, dengan relasi ke Matches melalui match_id agar tetap terkait dengan pertandingan yang relevan. VOD_Matches juga terhubung ke Users melalui match_vod_pic_editor untuk menandai editor yang bertanggung jawab.

Users menyimpan informasi seluruh pengguna sistem, mulai dari tim Sport, Video Editor, hingga Content Management, sehingga penugasan dan validasi konten VOD dapat dilakukan dengan jelas. Dengan adanya ERD ini, setiap data memiliki primary key dan foreign key yang membangun relasi antar tabel, menjamin integritas data, mendukung otomatisasi alur kerja, dan menjadi fondasi penting sebelum implementasi Appsheet dan Appscript.

Setelah struktur entitas dan relasinya ditetapkan melalui ERD, langkah berikutnya adalah mendalami setiap entitas pada level kolom, tipe data, dan ekspresi yang digunakan. Penjelasan ini penting sebagai dasar untuk membangun aplikasi di Appsheet. Dengan fokus pada kolom utama, tipe data yang sesuai, serta ekspresi untuk validasi dan pengolahan data, setiap entitas mulai dari Tournament, Matches, Teams, Leagues, VOD_Matches, hingga Users dapat diimplementasikan secara konsisten dan saling terhubung. Detail per kolom akan dijabarkan dalam bentuk tabel agar lebih terstruktur dan mudah dipahami, sekaligus menjadi referensi teknis untuk pengembangan aplikasi, sebagai berikut:

1) VOD_Matches Database

Tabel 3. 2 Type Database & Expression Databse VOD Matches

Name Columns	Type	Expression
match_vod_id (PK)	text	UNIQUEID()
match_vod_reference	enum	-
match_vod_reference_url	url	IFS([match_vod_reference] = "Catch-Up", [match_stream_admin], OR(AND(IN([match_vod_type], LIST("Mini Match", "Extended Mini Match", "Shorts")), [match_vod_reference] <> "Editor"), IN([match_vod_reference], LIST("Mediahub", "Partner"))), "TRUE", TRUE, ""))
match_vod_title	text	SWITCH([match_vod_type], "Moment", "[moment] " & IF(CONTAINS([match_name], " - "), LEFT([match_name], LEN([match_name]) - LEN(INDEX(SPLIT([match_name], " - "), COUNT(SPLIT([match_name], " - ")))))

		<pre> - LEN(" - ")), [match_name]) & " " & [match_tournament], "Customs", "", IF(CONTAINS([match_name], " - "), LEFT([match_name], LEN([match_name]) - LEN(INDEX(SPLIT([match_name], " - "), COUNT(SPLIT([match_name], " - ")))) - LEN(" - ")), [match_name]) & " - " & [match_vod_type] & " " & [match_tournament]) </pre>
match_vod_description	longtext	-
match_vod_type	enum	-
match_vod_models	enum	<pre> IF(ISBLANK([match_vod_models]), SWITCH([match_vod_type], "Highlight", "AVOD", "Moment", "AVOD", "Full Match", "SVOD", "Mini Match", "SVOD", "Extended Mini Match", "SVOD", "Shorts", "AVOD", </pre>

		"Sosmed", "AVOD", "") [match_vod_models])
match_vod_link	url	-
match_vod_link_backup	url	-
match_vod_notes	url	-
match_vod_pic (FK)	ref (UserID)	SELECT(UserID[Email Address], AND(IN([Department], LIST("Operations", "Content")), OR(CONTAINS(LOWER([Employee Title]), "content"), CONTAINS(LOWER([Employee Title]), "video"))))
match_vod_pic_editor (FK)	ref (UserID)	SELECT(UserID[Email Address], AND(IN([Department], LIST("Marketing", "Content", "Creative Studio")), OR(CONTAINS(LOWER([Employee Title]), "design"), CONTAINS(LOWER([Employee Title]), "creative"), CONTAINS(LOWER([Employee Title]), "video"))))

match_vod_status	enum	IF(ISNOTBLANK([match_vod_link]), "Published", "")
match_id (FK)	text	IF(CONTEXT("View") = "All Matches VOD_Form", LOOKUP([_THISROW].[match_id], "Matches", "match_id", "match_id"), "")
match_name	name	IF(CONTEXT("View") = "All Matches VOD_Form", LOOKUP([_THISROW].[match_id], "Matches", "match_id", "match_name"), "")
match_date_start	datetime	IF(CONTEXT("View") = "All Matches VOD_Form", LOOKUP([_THISROW].[match_id], "Matches", "match_id", "match_date_start"), "")
match_tournament	ref (Matches)	IF(CONTEXT("View") = "All Matches VOD_Form", LOOKUP([_THISROW].[match_id], "Matches", "match_id", "match_tournament"), "")
match_vod_main_genre	text	IF(CONTEXT("View") = "VOD Request_Form", "", LOOKUP([_THISROW].[match_id], "Matches", "match_id", "match_main_genre"))

match_vod_genre	text	LOOKUP([_THISROW].[match_tournament], "Tournaments", "tournament_title", "tournament_genre")
match_vod_date_created	datetime	IF(ISBLANK([match_vod_date_created]), NOW(), [match_vod_date_created])
match_vod_date_edited	datetime	NOW()
match_vod_pic_editor_priority	enum	"Realtime"
match_stream_id	text	IF(CONTEXT("View") = "All Matches VOD_Form", LOOKUP([_THISROW].[match_id], "Matches", "match_id", "match_stream_id"), "")

Tabel VOD_Matches adalah database utama yang mencatat seluruh request VOD. Setiap record di tabel ini mewakili satu konten VOD yang terkait dengan pertandingan tertentu, termasuk judul, deskripsi, tipe konten, tautan, status, catatan, serta editor yang bertanggung jawab. Tabel ini terhubung ke Matches melalui match_id untuk memastikan VOD terkait dengan pertandingan yang tepat, dan ke Users melalui match_vod_pic dan match_vod_pic_editor agar penugasan editor dapat dilakukan secara jelas. Dengan adanya tabel ini, proses permintaan VOD, penugasan, dan monitoring status konten dapat dilakukan secara terpusat, sehingga meminimalkan risiko human error.

Beberapa kolom menggunakan expression untuk mendukung otomatisasi dan validasi. Misalnya, match_vod_id menggunakan UNIQUEID() untuk membuat ID unik secara otomatis, match_vod_status menggunakan IF(ISNOTBLANK([match_vod_link]) , "Published", " ") agar status VOD berubah otomatis saat link tersedia, dan match_vod_reference_url menggunakan IFS() untuk menyesuaikan URL referensi berdasarkan

tipe konten. Ekspresi-ekspresi ini mempermudah input data, mengurangi pekerjaan manual, serta memastikan data yang tersimpan konsisten dan terintegrasi dengan tabel lain.

2) Matches Database

Tabel 3. 3 Type Database & Expression Database Matches

Name Columns	Type	Expression
match_id (PK)	text	UNIQUEID()
match_name	text	-
match_date_start	datetime	-
match_date_end	datetime	-
match_tournament (FK)	ref (Tournament)	SORT(SELECT(Tournaments[tournament_title], AND([tournament_sport] = [match_main_genre], [status] <> "Concluded")))
match_channel	Enum	-
match_stream_id	text	-
team_home (FK)	ref (Teams)	SORT(SELECT(Teams[team_name], TRUE))
team_away (FK)	ref (Teams)	SORT(SELECT(Teams[team_name], TRUE)) & [team_home] <> [team_away]
match_main_genre	text	SORT(UNIQUE(SELECT(Tournaments[tournament_sport], [status] <> "Concluded")))
match_svod_to_avid	enum	-
match_status	text	IF(ISNOTBLANK([match_stream_id]), "Published", "")

Tabel Matches menyimpan data pertandingan atau live stream, termasuk nama pertandingan, tanggal mulai dan berakhir, tim yang bertanding, saluran, dan status pertandingan. Tabel ini terhubung ke Tournament melalui match_tournament dan ke Teams melalui team_home serta team_away, sehingga setiap pertandingan selalu

konsisten dengan turnamen dan tim yang bersangkutan. Tabel ini menjadi dasar untuk membuat VOD terkait pertandingan, memastikan semua konten memiliki referensi pertandingan yang valid.

Beberapa kolom menggunakan expression untuk otomatisasi dan validasi. Kolom `match_status` menggunakan `IF(ISNOTBLANK ([match_stream_id]), "Published", " ")` agar status pertandingan otomatis diperbarui saat live stream tersedia, sedangkan `match_tournament` menggunakan `SORT(SELECT (...))` untuk menampilkan daftar turnamen aktif yang sesuai dengan genre pertandingan. Dengan ekspresi ini, pemilihan turnamen dan status pertandingan menjadi lebih cepat, akurat, dan terstandarisasi.

3) Tournament Database

Tabel 3. 4 Type Database & Expression Databse Tournament

Name Columns	Type	Expression
<code>tournament_title (PK)</code>	text	-
<code>tournament_start_date</code>	date	<code>NOW()</code>
<code>tournament_end_date</code>	date	<code>NOW()</code>
<code>status</code>	enum	-
<code>tournament_sport</code>	text	<code>SORT(SELECT(Tournaments[tournament_sport], TRUE, TRUE))</code>
<code>tournament_name</code>	text	-
<code>tournament_season</code>	text	-
<code>tournament_coverage</code>	enumlist	-
<code>tournament_gender</code>	text	-
<code>tournament_main_genre</code>	text	<code>[tournament_sport]</code>

Tabel Tournament menyimpan informasi turnamen atau kompetisi olahraga, termasuk judul, tanggal mulai dan berakhir, jenis olahraga, musim, genre, dan status turnamen. Tabel ini menjadi acuan utama bagi tabel Matches dan VOD_Matches untuk memastikan setiap pertandingan dan konten VOD dikategorikan sesuai olahraga, musim, dan klasifikasi turnamen.

Kolom seperti tournament_sport menggunakan ekspresi SORT(SELECT(...)) agar daftar olahraga muncul secara terurut saat membuat pertandingan atau VOD baru. Hal ini mempermudah pengguna dalam memilih kategori olahraga, memastikan data turnamen konsisten, dan meminimalkan kesalahan input yang dapat terjadi jika dipilih secara manual.

4) Teams Database

Tabel 3. 5 Type Database & Expression Databse Teams

Name Columns	Type	Expression
team_name (PK)	text	-
team_sport	enum	SELECT(Tournaments[tournament_sport], TRUE, TRUE)
team_country	enum	SORT(SELECT(TEAMS[team_country],TRUE,TRUE))
team_league (FK)	ref (Leagues)	SELECT(Leagues[league_id], AND([league_sport] = [_THISROW].[team_sport], [league_country] = [_THISROW].[team_country], [league_gender] = [_THISROW].[team_gender]))
team_tag	text	-
team_display	text	-
team_gender	text	-

Tabel Teams berisi informasi tim olahraga, termasuk nama tim, cabang olahraga, negara, liga, gender, dan tag. Tabel ini terhubung ke Leagues melalui team_league untuk memastikan setiap tim berada pada liga yang sesuai. Data dari tabel ini digunakan di Matches untuk mencatat tim yang bertanding, serta di VOD_Matches untuk menautkan konten VOD ke tim terkait.

Beberapa kolom menggunakan ekspresi, misalnya team_league menggunakan SELECT() untuk menampilkan hanya liga yang sesuai

dengan olahraga, negara, dan gender tim. Dengan cara ini, pemilihan liga menjadi otomatis dan akurat, sehingga meminimalkan kesalahan saat membuat data tim atau pertandingan baru dan menjaga konsistensi database.

5) Leagues Database

Tabel 3. 6 Type Database & Expression Databse Leagues

Name Columns	Type	Expression
league_id (PK)	text	UNIQUEID()
league_name (FK)	text	-
league_sport	enum	-
league_country	enum	SORT(SELECT(Leagues[league_country],TRUE,TRUE))

Tabel Leagues menyimpan informasi liga atau kompetisi tempat tim berkompetisi, seperti nama liga, jenis olahraga, dan negara. Tabel ini menjadi referensi penting bagi Teams untuk memastikan tim terhubung dengan liga yang tepat, serta membantu validasi saat membuat pertandingan.

Kolom league_id menggunakan UNIQUEID() untuk membuat ID unik, sedangkan league_country menggunakan SORT(SELECT(...)) agar daftar negara muncul secara terurut saat dipilih. Dengan ekspresi ini, proses input data liga menjadi lebih cepat, rapi, dan minim kesalahan, mendukung integritas database secara keseluruhan.

6) UserID Database

Tabel 3. 7 Type Database & Expression Databse UserID

Name Columns	Type	Expression
Userid (PK)	text	UNIQUEID()
Email Address	email	-
Fullname	text	-
Employee Title	text	-
Department	text	-

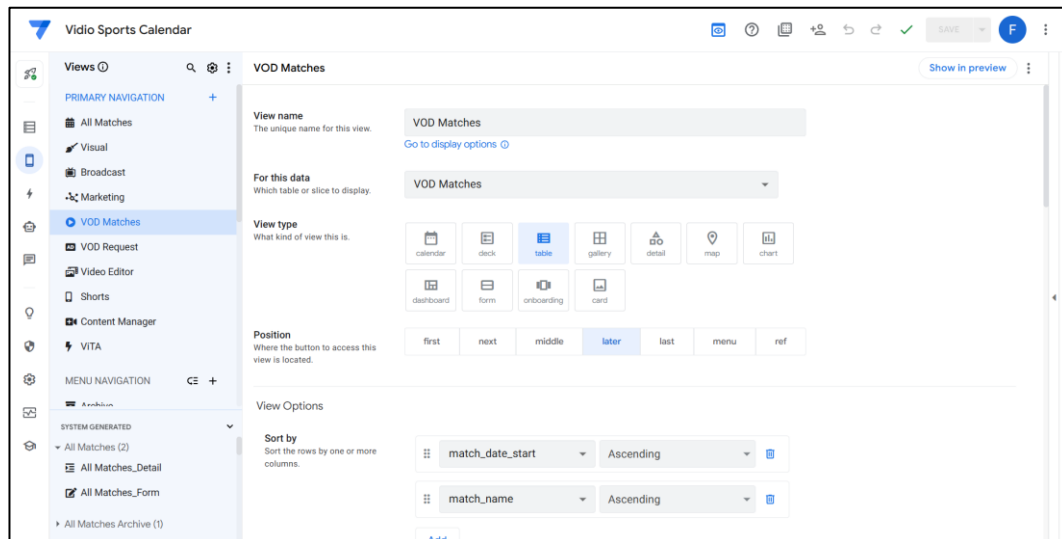
Tabel Users menyimpan data seluruh pengguna sistem, mulai dari Tim Sport, Video Editor, hingga Content Management. Kolom yang tersimpan antara lain email, nama lengkap, jabatan, dan departemen. Tabel ini menjadi acuan untuk penugasan VOD dan validasi konten, sehingga setiap aktivitas di VOD dapat dilacak dengan jelas.

Kolom Userid menggunakan UNIQUEID() untuk membuat ID unik bagi setiap pengguna, memastikan setiap user dapat dikenali secara unik dalam sistem. Dengan struktur ini, penugasan editor dapat dilakukan secara tepat, integritas data terjaga, dan sistem dapat mendukung otomatisasi workflow VOD dari permintaan hingga publikasi.

Setelah ERD dan expression selesai dibuat, langkah berikutnya adalah mengatur tampilan atau front-end di Appsheet agar pengguna dapat mengakses dan mengelola data dengan mudah. View disesuaikan dengan peran masing-masing, seperti tim Sport untuk membuat request, Video Editor untuk memperbarui progress, dan Content Management untuk mengunduh metadata. Penggunaan expression mendukung otomatisasi dan validasi data sehingga workflow VOD berjalan lebih efisien dan minim kesalahan, sebagai berikut:

UNIVERSITAS
MULTIMEDIA
NUSANTARA

7) VOD Matches View



Gambar 3. 10 Back End View VOD Matches

Penjelasan View VOD Matches merupakan pintu masuk utama bagi pengguna untuk mengakses daftar pertandingan yang sudah selesai dan siap diubah menjadi konten VOD. View ini menggunakan VOD Matches Slice sebagai sumber data dan ditampilkan dalam format tabel, yang diurutkan berdasarkan `match_date_start` secara ascending dan `match_name` secara ascending. Slice ini memiliki beberapa filter logika yang penting, yaitu:

```

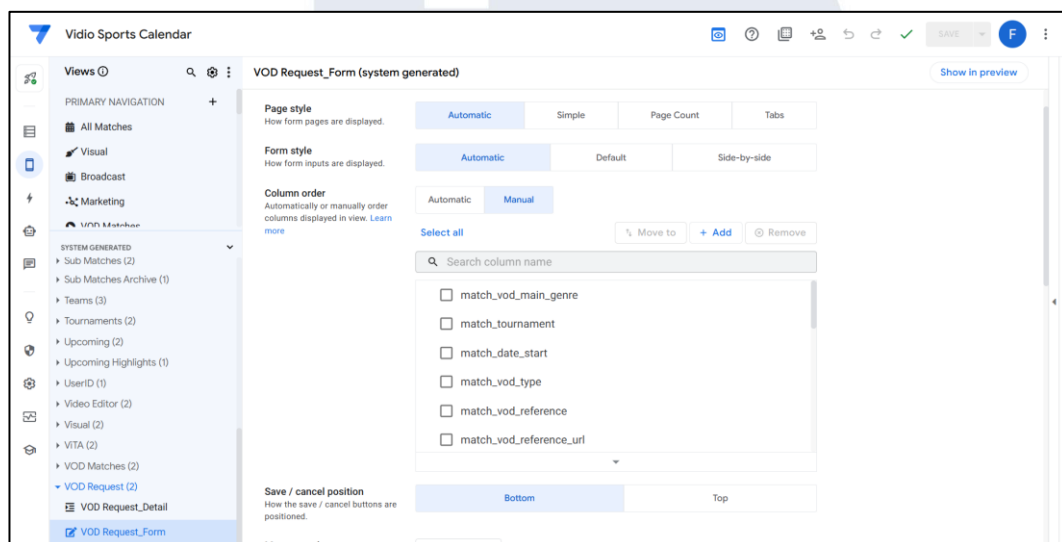
1  AND(
2    [match_tournament].[tournament_publish] = "Publish",
3    ISNOTBLANK([match_premier_status]),
4    ISNOTBLANK([match_date_start]),
5    [match_date_start] >= IF(WEEKDAY(TODAY()) = 1, TODAY() - 6,
6      TODAY() - (WEEKDAY(TODAY()) - 2)),
7    [match_parent] <> TRUE
8  )

```

Gambar 3. 11 Row Filter Slice VOD Matches

Dari expression di atas yaitu memfilter hanya pertandingan dari turnamen yang telah berstatus Publish (`[match_tournament].[tournament_publish] = "Publish"`), memastikan pertandingan sudah memiliki status akhir atau sudah tayang (`ISNOTBLANK([match_premier_status])`), memeriksa

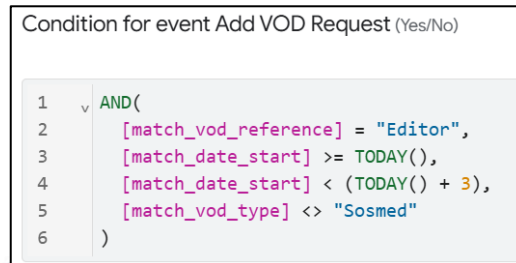
apakah tanggal pertandingan terisi (ISNOTBLANK([match_date_start])), menampilkan hanya pertandingan dalam satu minggu kalender berjalan ([match_date_start] >= IF(WEEKDAY(TODAY()) = 1, TODAY() - 6, TODAY() - (WEEKDAY(TODAY()) - 2))), dan mengecualikan entri parent ([match_parent] <> TRUE). Pengaturan ini memastikan hanya data yang relevan dan siap diproses yang ditampilkan, sehingga workflow menjadi lebih efisien dan risiko kesalahan input dapat diminimalkan.



Gambar 3. 12 Back End Form VOD Matches

Dari View ini, pengguna dapat melakukan dua aksi utama, yaitu Add (Request VOD) dan Edit. Tombol Add memanfaatkan relasi AppSheet untuk membuat entri baru di tabel VOD_Matches yang terkait dengan baris pertandingan saat ini. Saat diklik, tombol ini membuka VOD Request_Form, di mana pengguna dapat mengisi informasi seperti genre utama VOD, turnamen, tipe konten, dan kolom metadata lainnya. Sedangkan tombol Edit memungkinkan pengguna untuk memperbarui data VOD yang sudah ada, sehingga setiap perubahan pada konten dapat dilakukan secara real-time dan terintegrasi dengan data pertandingan. Dengan mekanisme ini, proses

pembuatan dan pembaruan VOD menjadi lebih cepat, rapi, dan terstandarisasi.



Gambar 3. 13 Condition Automation Add VOD Request

Setelah data VOD diisi dan disimpan, sistem secara otomatis memeriksa kondisi tertentu untuk mengirimkan notifikasi. Logika yang digunakan adalah kombinasi dari beberapa syarat yang harus terpenuhi secara bersamaan (menggunakan operator AND). Notifikasi hanya dikirim jika pertandingan ditugaskan kepada Editor, tanggal mulai pertandingan berada dari hari ini hingga tiga hari ke depan, dan jenis VOD yang bersangkutan bukan tipe “Sosmed”. Dengan mekanisme ini, sistem memastikan bahwa hanya data relevan yang membutuhkan perhatian editor yang memicu notifikasi, sehingga alur kerja menjadi lebih efisien dan terfokus.

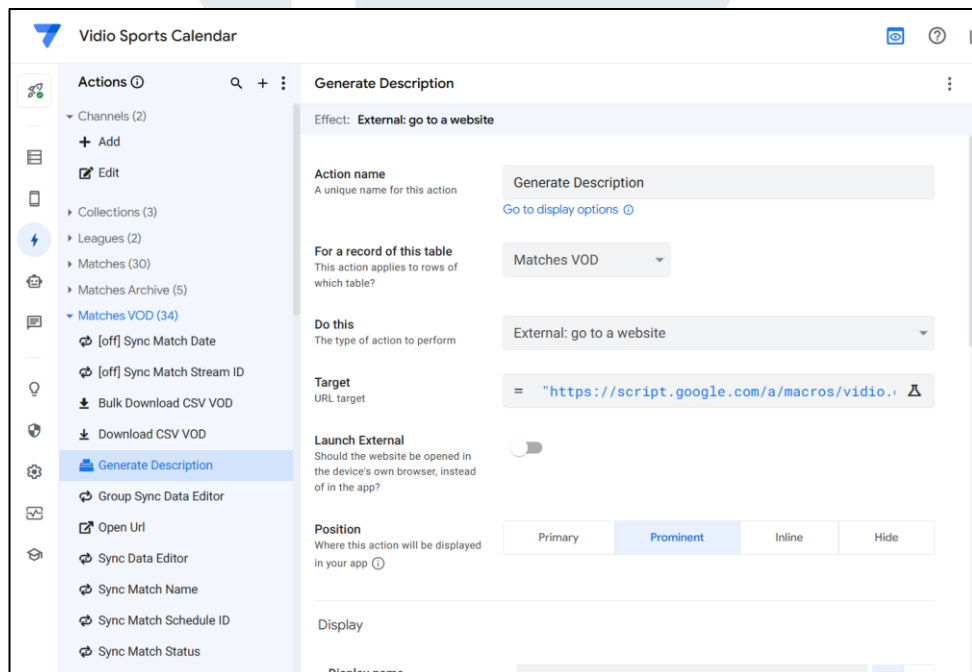


Gambar 3. 14 Chat Masssage: Expression Add VOD Request

Setiap kali VOD baru ditambahkan, sistem secara otomatis membuat notifikasi yang menampilkan judul VOD (menggunakan [match_vod_title] jika tipe VOD adalah *Moment*, atau [match_name] untuk tipe lainnya), nama turnamen ([match_tournament]), tanggal dan waktu mulai pertandingan ([match_date_start]), serta tipe VOD

([match_vod_type]), disertai informasi siapa yang menambahkan ([_INPUT].[PIC]) dan daftar pengguna terkait yang menerima notifikasi, yang terdiri dari ID pengguna default serta PIC turnamen yang diperoleh dari tabel Tournaments, diolah melalui pengecekan email dan pembersihan format sehingga setiap pihak yang relevan langsung mendapatkan pemberitahuan tanpa harus memeriksa database secara manual.

Selain itu, terdapat aksi khusus yaitu Generate Description yang mengintegrasikan AI Gemini untuk otomatisasi pembuatan deskripsi konten. Saat tombol ini diklik, AppSheet memicu Apps Script dengan function generateVideoDescriptions(), yang mengambil data dari sheet VOD Matches, menyusun prompt sesuai aturan SEO, dan memanggil API Gemini untuk menghasilkan deskripsi dan tagar.



Gambar 3. 15 Back End Action Generate Description

Seperti pada gambar, trigger aksi "Generate Description" menggunakan Action di AppSheet dengan tipe External: go to a website, yang dikonfigurasi untuk satu baris data pada tabel Matches

VOD. Aksi ini memiliki Target URL yang mengarah ke Web App Apps Script dan posisi tombol ditetapkan sebagai Prominent, sehingga mudah terlihat oleh pengguna. Karena tipe aksinya adalah "External: go to a website", setiap kali tombol diklik, AppSheet akan memicu eksekusi Apps Script secara otomatis. Dengan pengaturan ini, skrip dapat berjalan di latar belakang tanpa memerlukan input manual dari pengguna, sehingga proses pembuatan deskripsi VOD menjadi terotomasi dan konsisten.

```

1 function generateVideoDescriptions() {
2   const apiKey = [GEMINI_API_KEY_DISINI];
3   const ss = SpreadsheetApp.getActiveSpreadsheet();
4   const sheet = ss.getSheetByName('Develop');
5
6   const outputColumn = 5;
7
8   for (let i = 2; i <= sheet.getLastRow(); i++) {
9     const genre = sheet.getRange(i, 28).getValue();
10    const tournament = sheet.getRange(i, 22).getValue();
11    const title = sheet.getRange(i, 4).getValue();
12    const type = sheet.getRange(i, 9).getValue();
13    const dateStart = sheet.getRange(i, 21).getValue();
14    const matchName = sheet.getRange(i, 28).getValue();
15
16    const formattedDate = Utilities.formatDate(new Date(dateStart), Session.getScriptTimeZone(), 'N/d/yyyy HH:mm');
17
18    const prompt = `Buatkan deskripsi untuk olahraga ${genre} dalam turnamen ${tournament} pada tanggal ${formattedDate} WIB dengan judul video ${title} dan untuk type videonya ${type}.
19    Syarat deskripsinya:
20    1. Pada pertandingan ${matchName}
21    2. Deskripsi mengikuti prinsip SEO terbaru.
22    3. Panjang deskripsi minimal 70 kata dan maksimal 100 kata.
23    4. Harus terdapat keyword yang sesuai dengan pertandingan.
24    5. Kalimat pertama harus diawali dengan keyword sesuai pertandingan.
25    6. Setelah paragraf, beri tag maksimal 5 sesuai dengan pertandingan, dengan format (#keyword1 #keyword2 #keyword3 #keyword4 #keyword5) dengan jarak satu enter dari paragraf.
26    7. Jangan gunakan kalimat pembuka atau penutup tambahan.
27    8. Teks atau kalimat tidak ada bold.
28    9. Tag tidak ditulis otomatis, pastikan muncul di bawah paragrafnya.
29    10. Pastikan penonton tahu bahwa deskripsi ini adalah tayangan ulang, bukan pertandingan yang akan datang.
30    11. Tambahkan emoticon secukupnya agar deskripsi terasa lebih hidup.`;
31
32    const response = callGeminIAPI(prompt, apiKey);
33
34    if (response && response.candidates && response.candidates.length > 0 && response.candidates[0].content && response.candidates[0].content.parts && response.candidates[0].content.parts.length > 0) {
35      const generatedText = response.candidates[0].content.parts[0].text;
36
37      const parts = generatedText.split('\n');
38      const description = parts[0].trim();
39      const tagLine = parts.length > 1 ? parts[1].trim() : '';
40
41      if (description.split(' ').length >= 10 && description.split(' ').length <= 100) {
42        const output = `${description}\n\n${tagLine}`;
43        sheet.getRange(i, outputColumn).setValue(output);
44      } else {
45        sheet.getRange(i, outputColumn).setValue('Gagal: Panjang deskripsi tidak sesuai.');

```

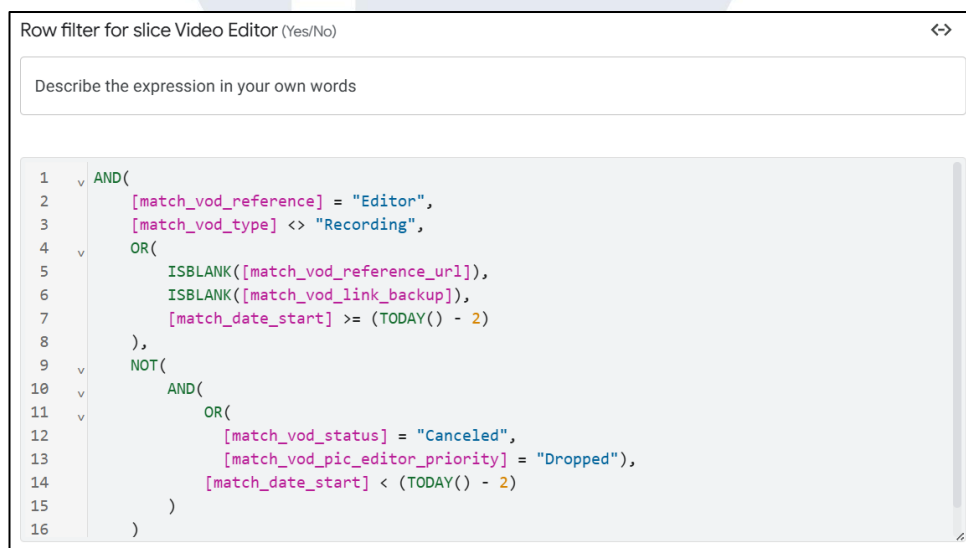
Gambar 3. 16 Apps Script Generate Description

Kode Apps Script untuk Generate Deskripsi VOD terdiri dari dua fungsi utama, yaitu `generateVideoDescriptions()` yang mengelola alur kerja data dari Google Sheet hingga output deskripsi, dan `callGeminIAPI()` yang menangani komunikasi dengan layanan Google Gemini AI. Fungsi `generateVideoDescriptions()` memulai dengan mengambil referensi Spreadsheet dan Sheet "Develop", menetapkan kolom output (Kolom 5), lalu melakukan iterasi dari baris ke-2 hingga akhir untuk membaca detail pertandingan seperti genre, turnamen, judul, tipe, tanggal mulai, dan nama pertandingan. Data tersebut kemudian dirangkai menjadi prompt yang terstruktur dengan persyaratan agar hasil AI SEO-ready, terformat, dan memuat tagar.

Hasil dari Gemini dipisahkan menjadi deskripsi dan tagar menggunakan pemisah baris kosong, divalidasi panjangnya (10–100 kata), dan ditulis kembali ke kolom output.

Fungsi `callGeminiAPI()` membuat HTTP POST ke endpoint Gemini menggunakan model *gemini-2.0-flash* dengan pengaturan temperature 0.5 dan `maxOutputTokens` 150, serta dilengkapi *try-catch* untuk menangani kesalahan jaringan atau API. Seluruh mekanisme ini memungkinkan setiap klik tombol "Generate Description" di AppSheet memicu eksekusi skrip secara otomatis, memproses semua baris data di Sheet, dan menghasilkan deskripsi VOD yang konsisten, terstruktur, dan siap dipublikasikan. Kode lengkap dari skrip ini dapat dilihat pada Gambar 3.10.

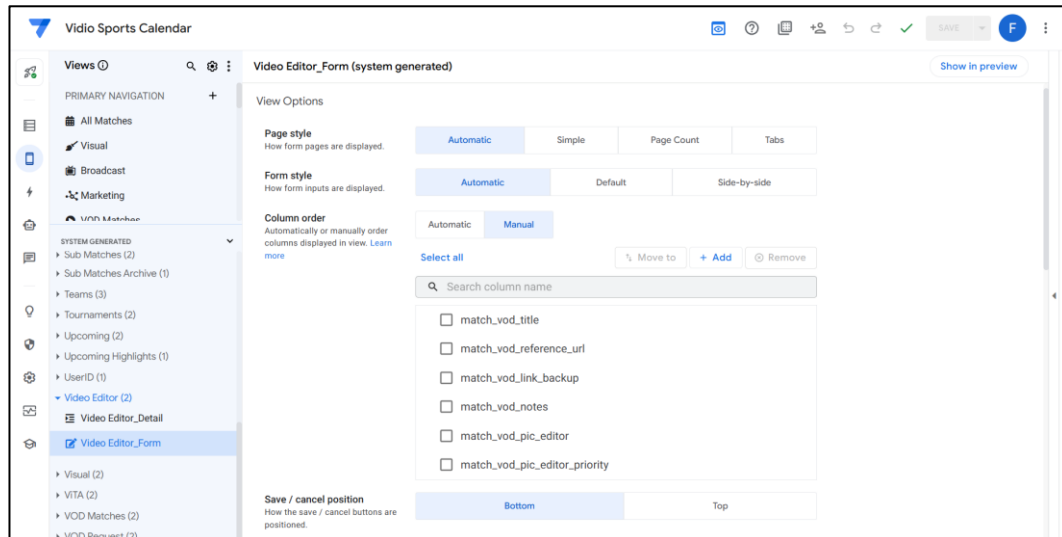
8) Video Editor View



Gambar 3. 17 Row Filter Slice Video Editor

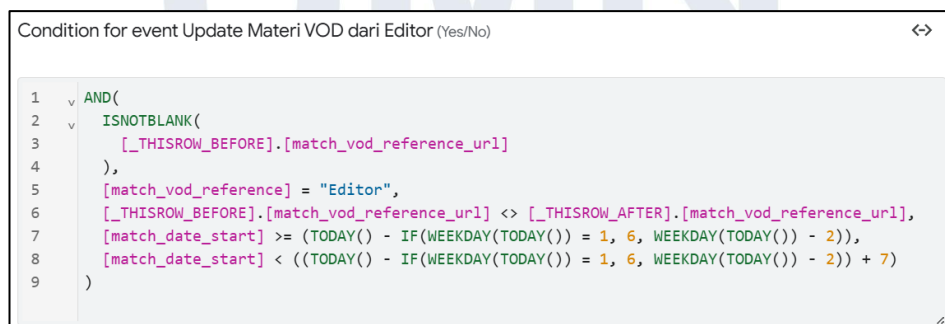
View Video Editor adalah antarmuka utama bagi Editor untuk melihat, mengedit, dan memproses VOD yang ditugaskan kepada mereka. Data dibatasi oleh Slice Video Editor, sehingga hanya menampilkan VOD yang relevan, bukan tipe "Recording", baru atau belum lengkap URL nya, dan tidak termasuk VOD lama yang

dibatalkan atau prioritasnya dihapus, sehingga Editor dapat fokus pada konten yang perlu dikerjakan.



Gambar 3. 18 Back End View Video Editor

Editor melakukan input atau pembaruan melalui Form Video Editor_Form, yang memiliki Page style dan Form style otomatis, dengan urutan kolom manual agar sesuai alur kerja. Kolom yang tersedia untuk diisi meliputi: match_vod_title, match_vod_reference_url (URL utama), match_vod_link_backup (URL cadangan), match_vod_notes, match_vod_pic_editor (Editor bertugas), dan match_vod_pic_editor_priority (Prioritas tugas).



Gambar 3. 19 Condition Automation Update Materi VOD

Setiap kali Editor menyimpan perubahan URL VOD, sistem memicu notifikasi otomatis. Kondisi pemicu memastikan notifikasi hanya dikirim jika VOD ditugaskan ke Editor, URL sebelum dan

sesudah penyimpanan berbeda
 ([_THISROW_BEFORE].[match_vod_reference_url] <>
 [_THISROW_AFTER].[match_vod_reference_url]), dan tanggal
 pertandingan berada dalam pekan kalender berjalan.

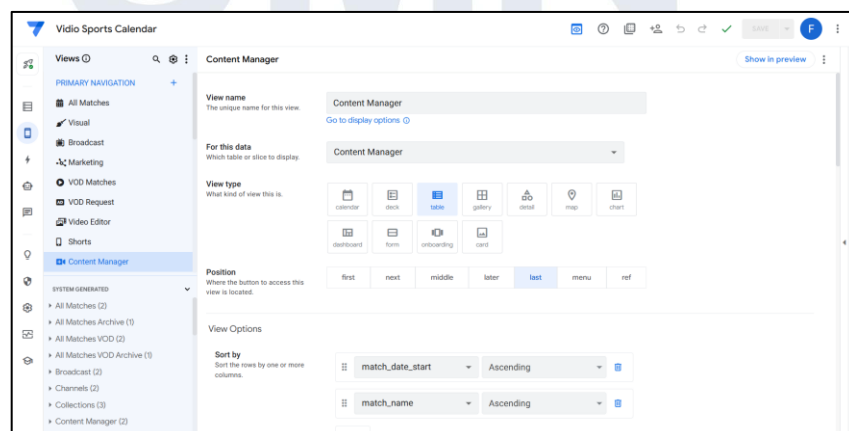
```
Webhook Body (Text)

1 {
2   v "text": "📺 ⚡ Materi VOD Baru Tersedia ⚡ 📺 \n\n🏆 <<[match_name]>>\n🏆
   <<[match_tournament]>>\n📅 <<TEXT([match_date_start], "dd MMM YYYY HH:mm")>>\n✅ Materi
   menjadi 📺 <<[match_vod_reference_url]>> \n\n👤 Diperbarui oleh: <<[_INPUT].[PIC]>>\n\n📢
   Fyi: <<IF(
3     v AND(
4       ISNOTBLANK([match_vod_reference_url]),
5       ISNOTBLANK(SUBSTITUTE(CONCATENATE(SELECT(OnCall[GoogleChat], [Group] =
   "vidio_content_management")), ", ", " "))
6     ),
```

Gambar 3. 20 Chat Massage: Expression Update Materi VOD

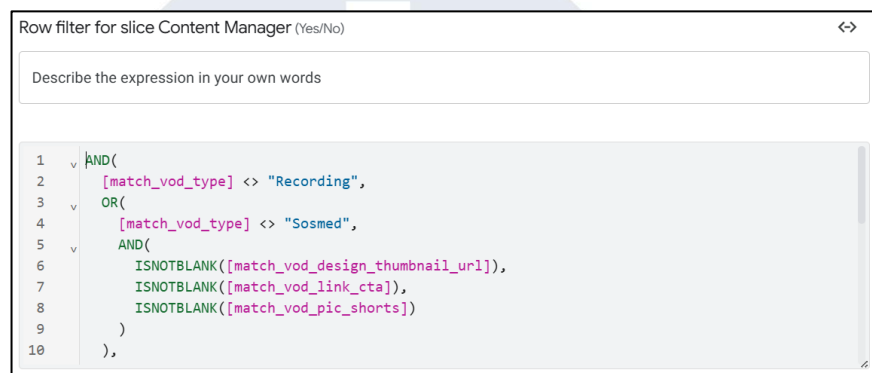
Notifikasi menampilkan judul pertandingan, turnamen, waktu mulai, URL VOD baru, serta PIC yang melakukan update, diformat dalam JSON untuk Webhook. Penerima notifikasi difilter agar hanya dikirim ke grup Google Chat yang valid (vidio_content_management) dan jika URL VOD sudah tersedia. Dengan pengaturan ini, setiap update Editor tersampaikan secara real-time ke tim terkait, memastikan workflow VOD tetap terkontrol, tepat waktu, dan konsisten

9) Content Manager View



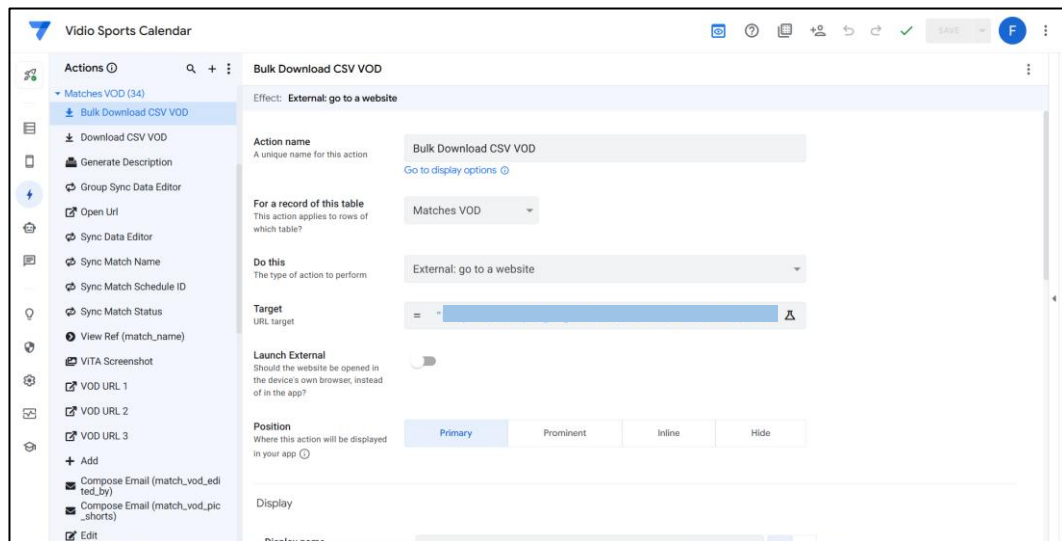
Gambar 3. 21 Back End View Content Manager

View Content Manager adalah pusat kerja bagi tim konten yang bertanggung jawab atas tahap final VOD, termasuk pengisian metadata lengkap, deskripsi, dan persiapan untuk publikasi. View ini menggunakan Content Manager Slice sebagai sumber data dan ditampilkan dalam format tabel, diurutkan berdasarkan match_date_start secara ascending dan match_name secara ascending, sehingga VOD diproses sesuai urutan tayang paling awal.



Gambar 3. 22 Row Filter Slice Content Manager

Slice memfilter data dengan logika kompleks: menampilkan VOD bukan tipe "Recording", menampilkan VOD bukan "Sosmed" atau jika tipenya "Sosmed" hanya jika Thumbnail, Link CTA, dan PIC Shorts sudah tersedia, memilih VOD baru (2 hari terakhir) atau yang link finalnya kosong, serta termasuk VOD dengan prioritas "Delayed" untuk tipe "Full Match" atau "Highlight" yang belum direview ulang, sambil mengecualikan VOD lama yang dibatalkan atau prioritasnya dihapus lebih dari 2 hari lalu. Dengan mekanisme ini, Content Manager hanya melihat VOD yang perlu difinalisasi, sehingga alur kerja lebih efisien dan focus.



Gambar 3. 23 Back End Action Bulk Download CSV VOD

Editor atau Content Manager dapat melakukan aksi khusus melalui AppSheet Action, yaitu Bulk Download CSV VOD, yang dikonfigurasi sebagai External: go to a website. Action ini memicu eksekusi Web App Google Apps Script melalui URL, baik untuk satu baris maupun seluruh daftar VOD, sehingga memungkinkan pengguna mengunduh data VOD secara massal. Dengan Action ini, tim dapat mengekspor data yang menjadi tanggung jawab mereka ke format CSV secara otomatis, termasuk metadata, judul, tipe, tanggal pertandingan, dan kolom penting lainnya, tanpa perlu memproses manual satu per satu. Pengaturan Action ini memastikan alur kerja finalisasi konten lebih cepat, rapi, dan terstandarisasi.

```

25 function downloadCSV(e) {
26   var rowId = e.parameter.id;
27   if (!rowId) {
28     return ContentService.createTextOutput("Error: No ID provided.");
29     .setMimeType(ContentService.MimeType.TEXT);
30   }
31
32   var fileId = "1";
33   var file = SpreadsheetApp.openById(fileId);
34   var sheet = file.getSheetByName("Matches VOD");
35
36   var range = sheet.getDataRange();
37   var values = range.getValues();
38
39   var headers = getCSVHeaders();
40   var columns = getColumnMapping();
41
42   var rowData = null;
43   for (var i = 1; i < values.length; i++) {
44     var currentRowId = values[i][0];
45     if (currentRowId.toString() === rowId) {
46       rowData = values[i];
47       break;
48     }
49   }
50
51   if (!rowData) {
52     return ContentService.createTextOutput("Error: ID not found.");
53     .setMimeType(ContentService.MimeType.TEXT);
54   }
55
56   var csvContent = headers.join(",") + "\n";
57
58   var processedRow = processRowData(rowData, columns);
59   var escapedRow = processedRow.map(escapeForCsv);
60   csvContent += escapedRow.join(",") + "\n";
61
62   var matchName = rowData[3];
63   var matchDate = rowData[20];
64
65   var matchDateTimeStr = "";
66   if (matchDate instanceof Date) {
67     matchDateTimeStr = Utilities.formatDate(matchDate, Session.getScriptTimeZone(), "d MMM yyyy HHmm");
68   } else {
69     var dateValue = new Date(matchDate);
70     if (!isNaN(dateValue)) {
71       matchDateTimeStr = Utilities.formatDate(dateValue, Session.getScriptTimeZone(), "d MMM yyyy HHmm");
72     } else {
73       matchDateTimeStr = "";
74     }
75   }
76 }

```

Gambar 3. 24 Apps Script Download CSV VOD

Skrip pada gambar di atas menangani proses download massal data VOD dari sheet "Matches VOD" ke CSV. Fungsi doGet(e) sebagai entry point memeriksa parameter action dari URL untuk mengeksekusi fungsi yang sesuai, misalnya downloadBulkCSV(e) untuk download massal. Fungsi ini memvalidasi parameter userEmail, mengambil semua data sheet, memfilter baris berdasarkan email Content Manager dan tanggal pertandingan (hari ini atau setelahnya), lalu memproses setiap baris melalui processRowData() dan escapeForCsv() sesuai mapping kolom dari getColumnMapping(). Skrip juga menangani konversi tipe data (boolean, datetime), logika khusus kolom (misal Schedule ID untuk tipe Highlight), serta membersihkan nama file melalui sanitizeFileName(). Hasil akhir adalah file CSV dengan nama dinamis VODs_[cleanEmail]_[yyyyMMdd].csv yang siap diunduh, rapi, valid, dan siap digunakan untuk publikasi atau upload massal VOD ke CMS.

3.3.1.3 Hasil Pengembangan Proyek 1

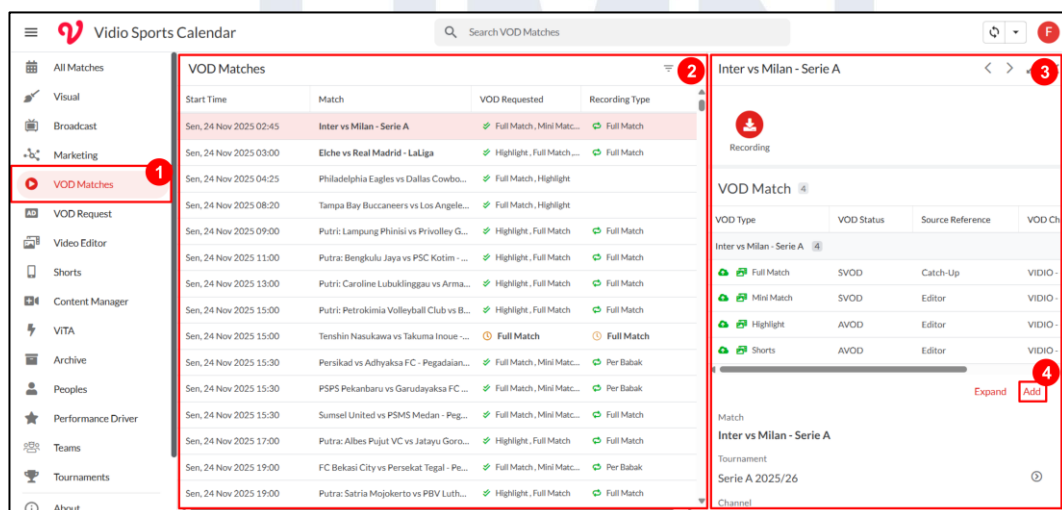
Setelah tahap perancangan proses, penyusunan SIPOC, pembuatan diagram swimlane, serta penyusunan struktur database

melalui ERD, tahap berikutnya adalah melakukan implementasi sistem Sport VOD Calendar menggunakan Appsheet, AppScript, dan Google Spreadsheet sebagai fondasi utama. Hasil pengembangan ini menghasilkan sebuah sistem end-to-end yang mampu mengotomasi seluruh proses pembuatan konten VOD, mulai dari pengajuan permintaan, penugasan editor, proses editing, hingga publikasi final oleh Content Management.

Berikut merupakan hasil pengembangan sistem berupa tampilan (view) utama yang digunakan oleh setiap tim dalam alur kerja Sport VOD Calendar:

1) View VOD Matches

VOD Matches View merupakan halaman utama yang digunakan untuk mengelola seluruh proses pembuatan konten VOD dalam sistem Sport VOD Calendar. Tampilan ini ditunjukkan pada Gambar 3.1, yang berfungsi sebagai pusat manajemen seluruh pertandingan yang akan atau telah diproses menjadi konten VOD. Pada tampilan ini terdapat empat area utama, masing-masing diberi nomor pada gambar untuk mempermudah identifikasi dan penjelasan.



Gambar 3. 25 View VOD Matches

Pada Gambar 3.25 di Nomor 1, ditampilkan panel VOD Matches yang berada di bagian navigasi kiri. Panel ini merupakan modul utama yang sedang aktif, memberikan akses bagi operator untuk masuk ke daftar pertandingan yang akan dikelola. Panel ini memastikan bahwa pengguna tetap berada pada konteks manajemen VOD dan menjadi titik awal saat mengoperasikan system.

Berlanjut ke Gambar 3.25 di Nomor 2, yaitu panel Daftar Pertandingan VOD yang terletak di bagian tengah. Panel ini menampilkan seluruh pertandingan yang tersedia untuk diproses menjadi konten VOD. Informasi yang ditampilkan meliputi *Start Time*, nama pertandingan (*Match*), jenis VOD yang diminta (*VOD Requested*), dan *Recording Type*. Baris yang disorot menandakan pertandingan yang sedang dipilih, di mana detail lengkapnya akan muncul di panel kanan. Panel ini berfungsi sebagai ringkasan komprehensif bagi operator untuk memantau progres VOD secara keseluruhan.

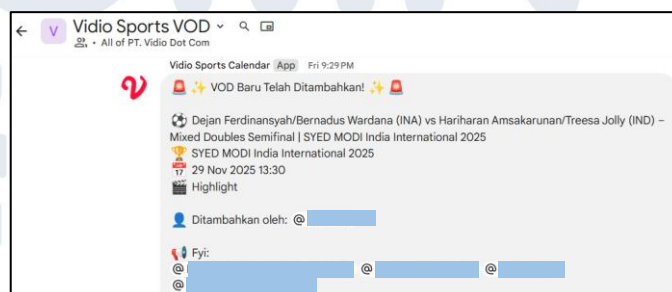
Selanjutnya, Gambar 3.25 di Nomor 3 menunjukkan panel Detail Pertandingan yang Dipilih, yang berada pada bagian kanan atas. Panel ini menampilkan informasi detail dari pertandingan yang dipilih, seperti nama pertandingan (contoh: Inter vs Milan – Serie A), status rekaman asli (ikon merah yang menandakan rekaman belum tersedia), serta tombol navigasi kiri dan kanan untuk berpindah antar pertandingan dalam daftar. Panel ini membantu operator memastikan ketersediaan dan status rekaman sebelum membuat konten VOD.

Kemudian pada Gambar 3.25 di Nomor 4, terdapat panel Tambah/Kelola Jenis VOD di bagian kanan bawah. Panel ini menjadi area utama bagi operator untuk mengatur berbagai jenis konten VOD berdasarkan satu pertandingan, seperti Full Match, Mini Match, Highlight, atau Shorts. Tabel informasi menampilkan VOD Type, VOD Status, tipe monetisasi (SVOD atau AVOD), serta Source

Reference. Tombol Add pada panel ini digunakan untuk membuka formulir pembuatan VOD baru.

Gambar 3. 26 Form Add VOD Request

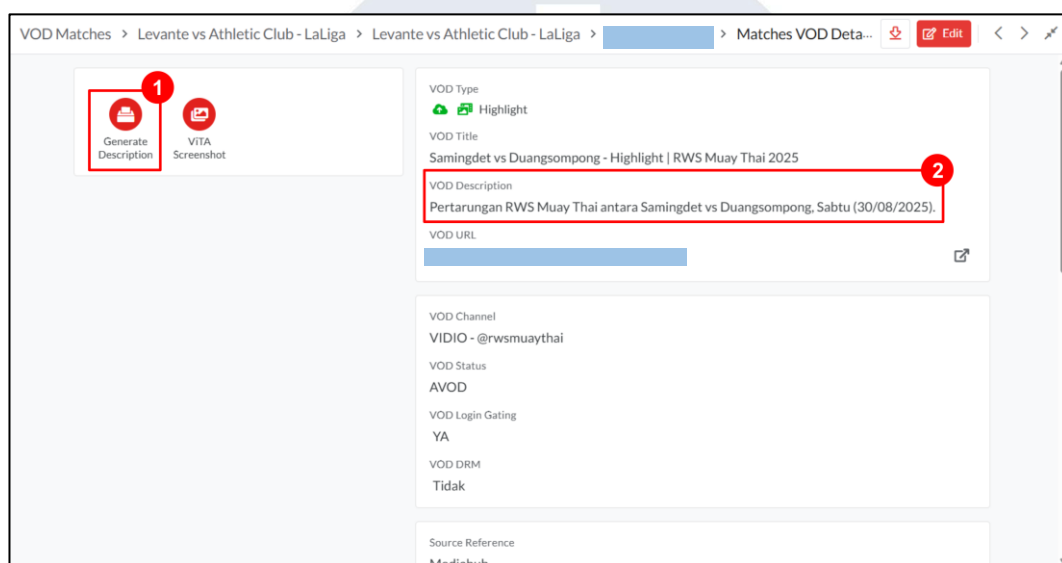
Ketika tombol Add ditekan, sistem akan menampilkan Formulir VOD Matches seperti yang terlihat pada Gambar 3.26. Formulir ini digunakan untuk memasukkan detail spesifik dari satu jenis VOD. Field-field yang tersedia meliputi informasi pertandingan, turnamen, channel penayangan, VOD Type. Source Reference. Operator dapat menyimpan data melalui tombol Save atau membatalkan melalui Cancel.



Gambar 3. 27 Notification Google Chat: VOD Request

Setelah operator menyimpan formulir tersebut, sistem secara otomatis mengirimkan notifikasi ke Google Chat seperti pada Gambar

3.27. Notifikasi “VOD Baru Telah Ditambahkan” berfungsi sebagai konfirmasi bahwa proses penambahan VOD berhasil dan sebagai pemberitahuan bagi tim terkait. Notifikasi ini menampilkan informasi lengkap seperti nama pertandingan, kategori yang ditambahkan, turnamen, jadwal pertandingan, pengguna yang menambahkan, serta daftar anggota tim yang di mention untuk memastikan koordinasi yang cepat dan transparan.



Gambar 3. 28 View Button Generate Description

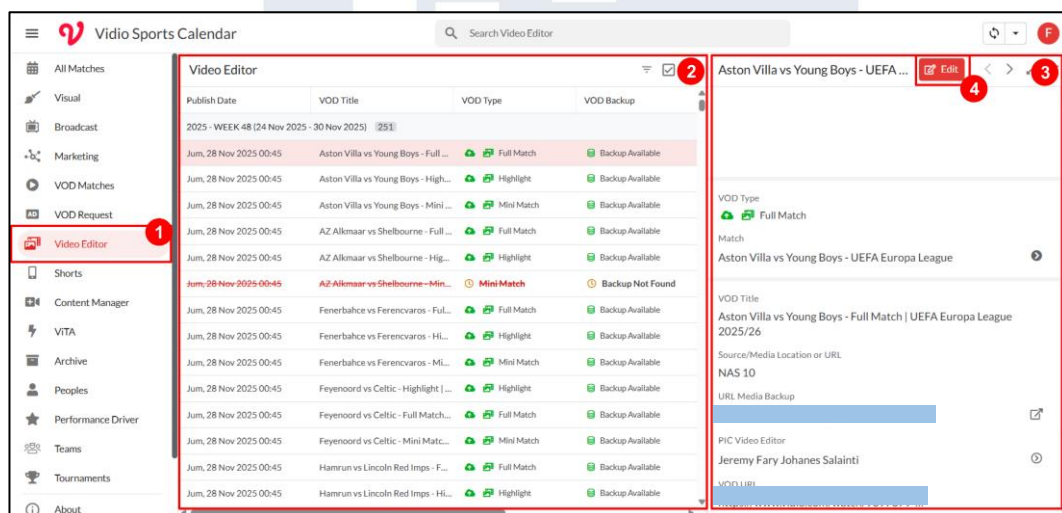
Selain itu, Gambar 3.28. menampilkan tampilan detail VOD yang menyoroti proses otomatisasi dalam pembuatan deskripsi konten. Pada Nomor 1, terdapat tombol Generate Description yang berfungsi untuk memicu script otomatis melalui App Script yang mengambil metadata seperti nama pertandingan, turnamen, tanggal, dan jenis VOD. Data tersebut kemudian digunakan untuk membentuk deskripsi secara otomatis menggunakan template. Fitur ini dirancang untuk mempercepat pekerjaan operator sekaligus menjaga konsistensi format deskripsi di setiap konten VOD.

Pada Nomor 2, ditampilkan field VOD Description yang akan terisi otomatis setelah proses generate selesai dijalankan. Hasil deskripsi langsung muncul pada field ini tanpa memerlukan input

manual, sehingga operator hanya perlu melakukan pengecekan atau penyesuaian kecil bila diperlukan. Contoh deskripsi yang dihasilkan pada gambar adalah: “Pertarungan RWS Muay Thai antara Samingdet vs Duangsompong, Sabtu (30/08/2025).”

2) View Video Editor

Video Editor View pada Gambar 3.5 merupakan tampilan yang digunakan oleh tim editor untuk memantau, mengelola, dan memperbarui materi VOD yang telah diproduksi. Tampilan ini terdiri dari beberapa komponen utama yang saling terhubung untuk memastikan proses editing dan publikasi materi VOD berjalan terstruktur.



Gambar 3. 29 View Video Editor

Pada panel navigasi kiri pada Gambar 3.29 - nomor 1, pengguna memilih modul Video Editor, yang merupakan pusat kontrol bagi editor dalam menangani daftar VOD yang sudah tersedia. Setelah modul ini dipilih, panel tengah pada Gambar 3.29 - nomor 2 menampilkan seluruh daftar materi VOD lengkap dengan informasi penting seperti Publish Date, VOD Title, VOD Type, serta status VOD Backup. Panel ini membantu editor melihat prioritas pekerjaan, menandai materi yang sudah memiliki file cadangan, dan mengidentifikasi konten yang masih memerlukan perhatian.

X Aston Villa vs Young Boys - UEFA Europa League

Cancel Save

VOD Title
Aston Villa vs Young Boys - Full Match | UEFA Europa League 2025/26

Source/Media Location or URL
NAS 10

URL Media Backup

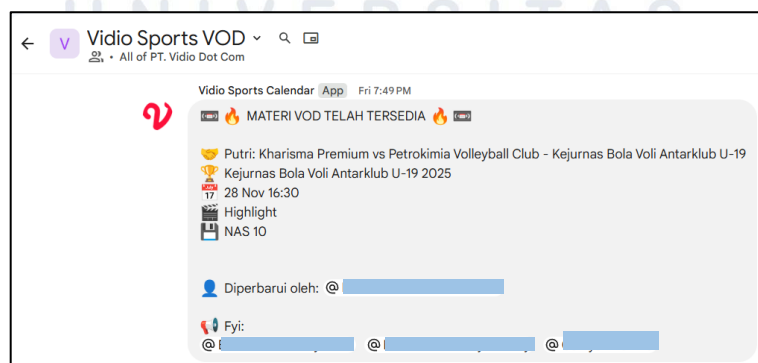
VOD Notes

PIC Video Editor

Video Editor Priority
Realtime Delayed Dropped

Gambar 3. 30 Form Video Editor

Ketika salah satu VOD dipilih, panel detail di sisi kanan pada Gambar 3.29 - nomor 3 menampilkan informasi lengkap terkait VOD tersebut, mulai dari jenis VOD, judul, lokasi media, hingga PIC editor yang bertanggung jawab. Tombol Edit pada Gambar 3.29 - nomor 4 digunakan untuk membuka formulir pengeditan VOD seperti ditunjukkan pada Gambar 3.30. Pada formulir ini, editor dapat memperbarui metadata seperti VOD Title, Source/Media Location, URL Media Backup, menambahkan VOD Notes, memilih PIC Video Editor, serta menentukan prioritas pekerjaan (Realtime, Delayed, atau Dropped). Setelah perubahan disimpan melalui tombol Save, sistem akan melakukan pembaruan data secara otomatis.



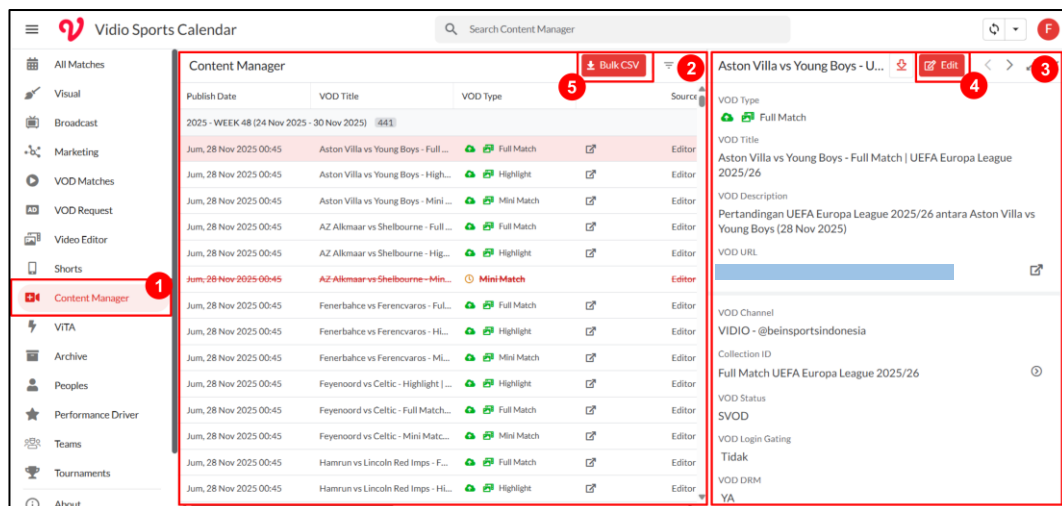
Gambar 3. 31 Notification Google Chat: Update Materi VOD

Sebagai bagian dari alur kerja terintegrasi, setiap kali editor memperbarui data VOD dan menyimpannya, sistem secara otomatis mengirimkan notifikasi ke Google Chat seperti pada Gambar 3.31. Notifikasi ini berupa pesan berjudul “MATERI VOD TELAH TERSEDIA”, yang berfungsi memberi tahu tim terkait bahwa materi VOD sudah siap atau telah diperbarui.

Notifikasi mencakup informasi lengkap seperti nama pertandingan, nama turnamen, tanggal dan waktu, jenis VOD, lokasi file, serta identitas editor yang melakukan update. Pada bagian akhir, sistem juga menampilkan daftar pengguna yang ikut ditandai sebagai bentuk koordinasi lintas tim untuk memastikan tidak ada hambatan dalam proses distribusi dan publikasi konten.

3) View Content Manager

Content Manager View pada Gambar 3.32 merupakan tampilan utama yang digunakan operator untuk mengelola seluruh VOD yang sudah siap tayang setelah melewati proses editing.

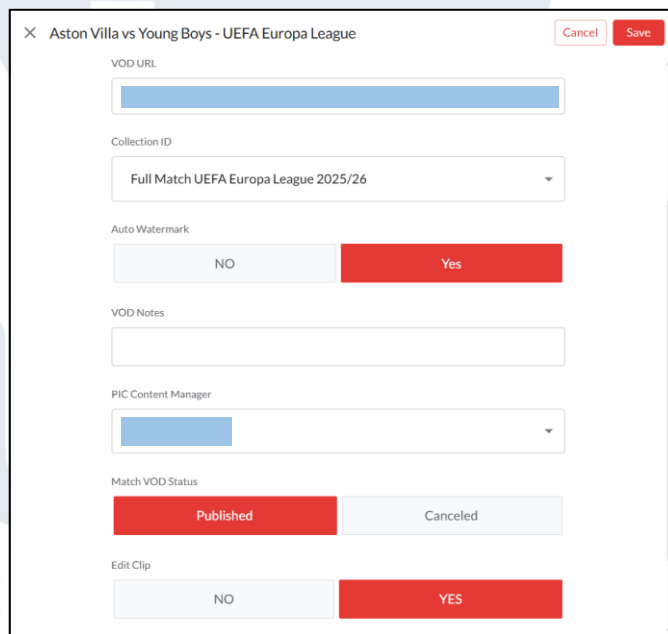


Gambar 3. 32 View Content Manager

Pada panel navigasi kiri pada nomor 1, operator memilih modul Content Manager sebagai langkah final sebelum konten dipublikasikan. Panel tengah pada nomor 2 menampilkan daftar

seluruh VOD yang telah diatur pada modul VOD Matches, dilengkapi informasi seperti Publish Date, VOD Title, VOD Type, Source, serta filter minggu yang membantu operator fokus pada batch VOD tertentu.

Ketika salah satu VOD dipilih, panel kanan pada nomor 3 menampilkan metadata lengkap seperti VOD Description, VOD URL, VOD Channel, Collection ID, VOD Status, Login Gating, hingga DRM, sehingga operator dapat meninjau informasi penayangan secara menyeluruh. Pada bagian atas panel kanan terdapat tombol Edit pada nomor 4 untuk membuka formulir pengelolaan metadata, serta tombol Bulk CSV pada nomor 5 untuk memproses unggahan massal data VOD melalui file CSV yang dijalankan dengan Apps Script.

The image shows a web-based form titled "Aston Villa vs Young Boys - UEFA Europa League". At the top right are "Cancel" and "Save" buttons. The form contains several input fields and buttons: "VOD URL" (a text box), "Collection ID" (a dropdown menu showing "Full Match UEFA Europa League 2025/26"), "Auto Watermark" (two buttons: "NO" and "Yes", with "Yes" being red), "VOD Notes" (a text box), "PIC Content Manager" (a dropdown menu), "Match VOD Status" (two buttons: "Published" and "Canceled", with "Published" being red), and "Edit Clip" (two buttons: "NO" and "YES", with "YES" being red).

Gambar 3. 33 Form Content Manager

Saat tombol Edit di klik, sistem membuka Formulir Content Manager seperti pada Gambar 3.33. Form ini digunakan untuk melakukan pengaturan teknis serta menentukan status penayangan akhir konten VOD. Field yang tersedia meliputi VOD URL, Collection ID, Auto Watermark, VOD Notes, PIC Content Manager,

Match VOD Status, dan Edit Clip, yang masing-masing berfungsi untuk mengontrol kualitas, struktur koleksi, penanggung jawab, serta status publikasi VOD.

Setelah data diperbarui, operator dapat menyimpannya melalui tombol Save untuk memperbarui status tayang di sistem atau memilih Cancel untuk membatalkan VOD. Dengan dua komponen utama ini tampilan utama Content Manager dan formulir pengeditan sistem memastikan bahwa setiap konten VOD dipublikasikan dengan metadata yang benar, lengkap, dan siap di upload ke CMS.

3.3.2 Proyek 2 - Improvement PT XYZ Digital Systems Pre-Sales

3.3.2.1 Latar Belakang Proyek 2

Proses Pre-Sales di PT XYZ Digital Systems sebelumnya berjalan tanpa struktur yang jelas, terutama dalam alur permintaan (request) campaign dari tim Sales kepada tim Marketing. Selama ini, proses tersebut masih dilakukan melalui email, sehingga tidak ada histori data yang tersentral dan sulit untuk ditelusuri kembali. Padahal, data histori campaign sangat penting sebagai dasar dalam penyusunan forecast, evaluasi performa, hingga perencanaan campaign di periode berikutnya.

Untuk menjawab kebutuhan tersebut, tim Business Process Analyst telah membangun sebuah sistem berbasis AppSheet dengan dukungan database Google Spreadsheet yang sudah tersentralisasi. Sistem ini membantu menyatukan data dan mengurangi ketergantungan pada proses manual. Namun, masih terdapat beberapa kekurangan yang perlu ditingkatkan. Salah satunya adalah pada Form Campaign Plan, di mana field Brand, Agency, dan Bill To masih menggunakan format free text. Cara ini menimbulkan risiko inkonsistensi data karena penulisan dapat berbeda-beda antar

pengguna, padahal daftar Brand, Agency, dan Bill To sebenarnya sudah tersedia dan tersimpan rapi di spreadsheet.

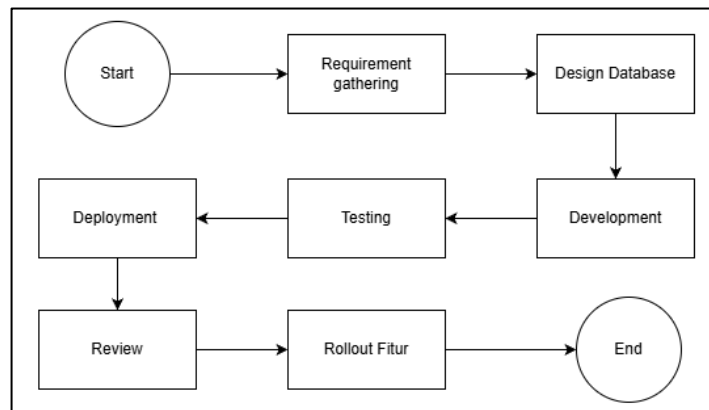
Selain itu, proses pendaftaran user baru (karyawan atau intern) ke appsheet juga masih bersifat manual. Data dari spreadsheet daftar karyawan harus dipindahkan satu per satu ke database AppSheet PT XYZ Digital Systems, sehingga membutuhkan waktu lebih lama dan rawan kesalahan input. Maka dari itu, diperlukan serangkaian perbaikan untuk meningkatkan efisiensi, akurasi, dan keterpaduan data dalam sistem Pre-Sales tersebut.

Untuk mengatasi permasalahan tersebut, penulis akan melakukan pengembangan lebih lanjut terhadap sistem yang sudah ada. Langkah yang dilakukan meliputi pembuatan database terstruktur untuk Brand, Agency, dan Bill To, kemudian melakukan mapping terhadap seluruh data yang telah ada sebelumnya agar selaras dengan standar database AppSheet yang baru. Selain itu, penulis juga akan mengembangkan Google Apps Script untuk mengotomatisasi proses sinkronisasi data user baru ke database utama, sehingga proses pendaftaran menjadi lebih cepat, efisien, dan minim kesalahan.

3.3.2.2 Perancangan dan Implementasi Proses Proyek 2

1) Improvement Struktur Brand, Agency, & Bill To

Setelah memahami permasalahan dan menetapkan ruang lingkup improvement yang perlu dilakukan, tahap berikutnya adalah merancang serta mengimplementasikan solusi secara terstruktur. Pada proyek ini, penulis menggunakan alur kerja pengembangan sistem yang bersifat iteratif dan berurutan untuk memastikan setiap kebutuhan dapat dianalisis, diuji, dan dipastikan berjalan dengan baik sebelum diluncurkan ke seluruh pengguna.



Gambar 3. 34 Proses Implementasi Fitur Brand & Agency

Alur proses tersebut digambarkan dalam bentuk diagram pada gambar di atas, dimulai dari pengumpulan kebutuhan melalui wawancara, perancangan database, pengembangan fitur, pengujian, hingga proses deployment dan rollout. Setiap tahapan memiliki tujuan dan peran penting dalam memastikan sistem Pre-Sales yang baru menjadi lebih akurat, konsisten, dan efisien. Berikut merupakan penjelasan dari setiap tahapan dalam alur proses tersebut.

a) Requirement Gathering (Wawancara User)

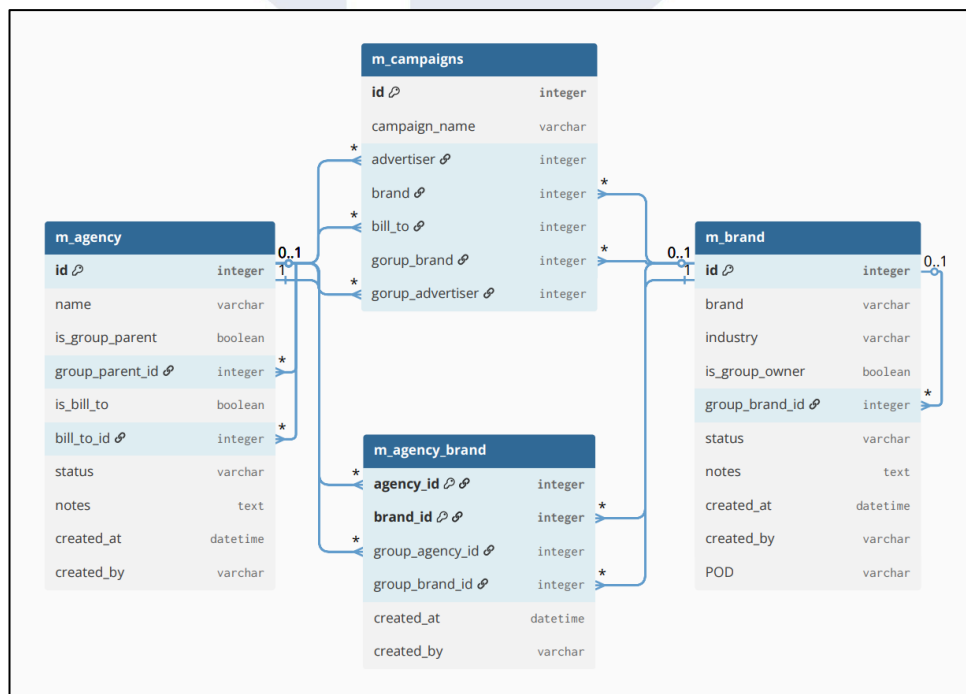
Tahap awal pengembangan dilakukan dengan melakukan sesi wawancara bersama VP Marketing serta Leaders dari tim Implementer PT XYZ Digital Systems. Tujuan wawancara ini adalah untuk memahami kebutuhan bisnis terkait perbaikan sistem Pre-Sales, khususnya pada proses pengelolaan Brand, Agency, dan Bill To. Dari hasil diskusi, ditemukan bahwa penambahan Brand dan Agency merupakan aktivitas yang cukup sering terjadi, dan masing-masing memiliki keterkaitan yang spesifik. Misalnya, suatu Brand biasanya dikelola oleh Agency tertentu, sementara Bill To dapat diarahkan ke Agency maupun langsung ke Brand, tergantung struktur kerjasamanya.

Selain itu, pihak user juga menjelaskan bahwa setiap Brand memiliki kategori industri yang berbeda, dan informasi ini penting karena dalam pengaturan campaign sering kali terdapat batasan

campaign dari industri yang sama atau pesaing tidak boleh tayang bersamaan. Temuan-temuan ini kemudian menjadi dasar dalam merancang database baru yang lebih terstruktur serta alur otomatisasi yang akan dikembangkan pada tahap implementasi berikutnya.

b) Design Database

Setelah kebutuhan bisnis diperoleh melalui wawancara dengan VP Marketing dan Leaders tim Implementer, tahap berikutnya adalah melakukan perancangan struktur database sebagai dasar dari seluruh improvement pada sistem Pre-Sales. Tahap ini dimulai dengan mengidentifikasi entitas inti yang diperlukan dalam proses bisnis, yaitu Agency, Brand, relasi Agency Brand, serta Campaigns yang memanfaatkan data tersebut. Perancangan ini dilakukan untuk mengatasi permasalahan inkonsistensi data akibat penggunaan free text, sekaligus memastikan bahwa hubungan antar data dapat dikelola secara menyeluruh dan terintegrasi.



Gambar 3. 35 ERD Dtabase Brand & Agency

Hasil identifikasi tersebut kemudian divisualisasikan dalam bentuk Entity Relationship Diagram (ERD). ERD ini berfungsi sebagai blueprint yang menggambarkan hubungan antar entitas, termasuk struktur grup Brand, hubungan banyak ke banyak antara Agency dan Brand melalui tabel m_agency_brand, serta keterkaitan masing-masing entitas terhadap Campaigns. Dengan adanya ERD, alur data menjadi lebih jelas dan mudah ditelusuri, sehingga meminimalkan risiko duplikasi maupun ketidaksesuaian data ketika sistem dijalankan di AppSheet maupun saat dilakukan.

Setelah struktur relasi digambarkan melalui ERD, langkah berikutnya adalah mendefinisikan setiap entitas pada level yang lebih detail, meliputi kolom, tipe data, fungsi kolom, serta bagaimana kolom tersebut digunakan dalam proses bisnis. Pendalaman ini penting untuk memastikan bahwa setiap tabel dapat berfungsi sesuai kebutuhan bisnis dan berada dalam standar data yang sama. Adapun penjelasan setiap entitas adalah sebagai berikut:

Tabel 3. 8 Type Database & Expression Agency

Name Columns	Type	Expression
id	text	UNIQUEID()
name	text	-
is_group_parent	yes/no	-
grou_parent_id	enum	SELECT(m_agency[id], [is_group_parent] = TRUE)
is_bill_to	yes/no	-
bill_to_id	enumlist	SELECT(m_agency[id], [is_bill_to] = TRUE)
status	text	"active"
notes	longtext	-
created_at	datetime	NOW()
created_by	email	USEREMAIL()

Tabel `m_agency` berfungsi sebagai master data utama untuk seluruh agency yang terdaftar dalam sistem. Tabel ini menyimpan informasi penting seperti nama agency, status keaktifan, struktur hubungan antar agency, serta konfigurasi penagihan melalui kolom `bill_to_id`. Relasi ini memungkinkan setiap agency diarahkan kepada entitas penagihan tertentu, baik ke dirinya sendiri maupun ke agency lain yang berperan sebagai bill-to. Selain itu, atribut `is_group_parent` digunakan untuk menandai agency yang berperan sebagai induk dalam sebuah grup, sehingga struktur hirarki agency dapat dipetakan dengan baik dalam proses pengelolaan brand, agency, maupun campaign yang saling berkaitan.

Beberapa kolom dalam tabel ini juga didukung oleh mekanisme otomatisasi, seperti `created_at` yang mengisi waktu pembuatan secara otomatis menggunakan `NOW()` serta `created_by` yang mengambil email pengguna melalui `USEREMAIL()`. Kolom `is_bill_to` memberikan penanda apakah agency dapat digunakan sebagai entitas penagihan, sehingga validasi pemilihan bill-to dapat dilakukan lebih akurat. Dengan kombinasi struktur relasional dan atribut otomatis ini, tabel `m_agency` memastikan data agency tersimpan secara konsisten, terstandarisasi, dan siap terintegrasi dengan kebutuhan operasional maupun modul lain dalam system.

Tabel 3. 9 Type Database & Expression Brand

Name Columns	Type	Expression
id	text	
brand	text	-
industry	enum	-
is_group_owner	yes/no	-
group_brand_id	enum	SELECT(m_brand[id], [is_group_owner] = TRUE)
status	text	"active"
notes	longtext	-
created_at	datetime	NOW()
created_by	email	USEREMAIL()

Tabel m_brand merupakan master data yang menyimpan seluruh informasi mengenai brand yang digunakan dalam proses Pre-Sales. Kolom utamanya meliputi nama brand, industri, status, serta catatan tambahan. Tabel ini juga mendukung struktur hierarki melalui kolom group_brand_id, yaitu relasi ke brand induk jika brand tersebut berada dalam satu grup. Atribut is_group_owner digunakan untuk menandai apakah sebuah brand berperan sebagai pemilik grup, sehingga hubungan antar brand dapat dikelola dengan lebih rapi dan konsisten.

Selain sebagai daftar referensi brand, tabel m_brand juga berfungsi penting dalam logika bisnis, terutama dalam pengaturan campaign yang sensitif terhadap kategori industri. Misalnya, brand dari industri yang sama biasanya tidak diperbolehkan tayang bersamaan untuk menghindari konflik atau kompetisi langsung. Dengan adanya informasi industri, struktur grup brand, serta status masing-masing brand, tabel ini memastikan proses validasi, pemilihan, dan pengelolaan campaign dapat berjalan lebih terarah dan terstandarisasi.

Tabel 3. 10 Type Database & Expression Related Agency & Brand

Name Columns	Type	Expression
id	text	UNIQUEID()
agency_id	Ref	-
group_agency_id	text	LOOKUP([_thisrow].[agency_id], "m_agency", "id", "group_parent_id")
brand_id	ref	-
group_brand_id	ref	LOOKUP([_thisrow].[brand_id], "m_brand", "id", "group_brand_id")
created_at	datetime	NOW()
created_by	email	USEREMAIL()

Tabel m_agency_brand berfungsi sebagai tabel hubungan yang menghubungkan advertiser dan brand dalam relasi banyak-ke-banyak.

Setiap baris berisi pasangan `agency_id` dan `brand_id` yang menunjukkan bahwa suatu agency mengelola atau berhubungan dengan brand tertentu. Kedua kolom ini menjadi kunci gabungan, sehingga tidak ada duplikasi hubungan antara agency dan brand yang sama. Validasi menggunakan `LOOKUP_THROW` memastikan bahwa setiap ID agency maupun brand yang dimasukkan benar-benar ada pada tabel master terkait sehingga integritas data tetap terjaga.

Selain mencatat hubungan dasar, tabel ini juga mendukung struktur grup melalui kolom `group_agency_id` dan `group_brand_id`. Kedua kolom ini digunakan untuk menunjukkan hubungan pada tingkat grup, misalnya ketika sebuah brand berada dalam grup tertentu atau agency dikelola oleh entitas induk. Dengan adanya struktur ini, proses pemetaan data antara agency dan brand menjadi lebih terorganisir dan memungkinkan sistem untuk menampilkan relasi secara konsisten baik di level individu maupun di level grup.

Tabel 3. 11 Type Database & Expression Campaign Plan

Name Columns	Type	Expression
id	text	UNIQUEID()
campaign_name	text	-
gorup_brand	enum	SELECT(m_brand[id], ([id] = [brand].[group_brand_id]))
brand	enum	SELECT("m_brand", ([is_group_owner] = FALSE))
gorup_agency	enum	SELECT(m_agency[id], ([id] = [advertiser].[group_parent_id]))

agency	enum	SELECT(m_agency[id], IN([id], SELECT(m_agency_brand[advertiser_id], ([brand_id] = [_THISROW].[brand]))))
bill_to	enum	SELECT(m_agency[id], ([id] = [_THISROW].[advertiser].[bill_to_id]))

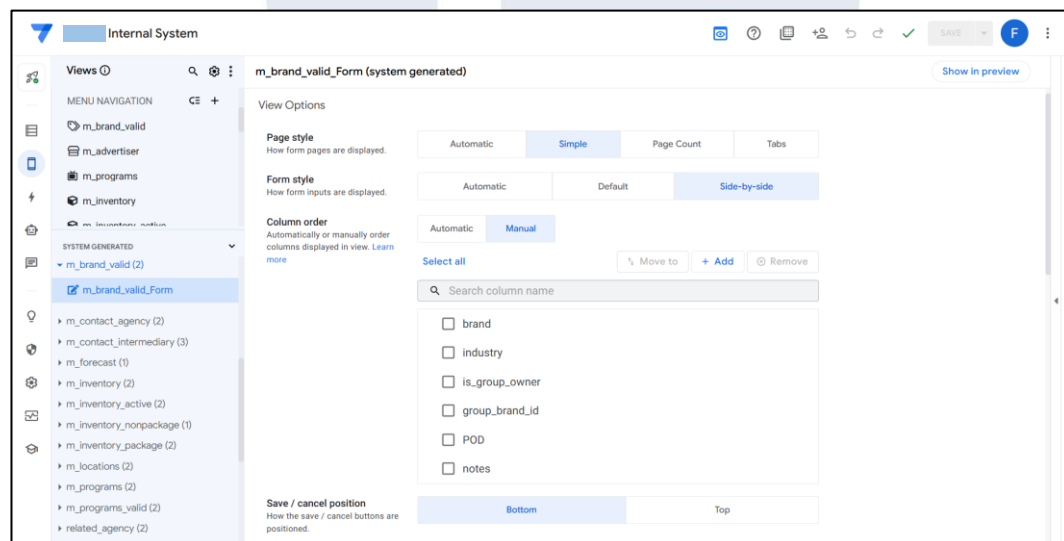
Tabel m_campaigns berfungsi sebagai pusat data untuk seluruh Campaign Plan, dan dirancang agar pemilihan agency, brand, maupun bill-to menjadi lebih konsisten dan akurat. Melalui kolom agency dan brand yang masing-masing mereferensikan tabel m_agency dan m_brand, setiap campaign yang dibuat di Form Campaign Plan akan selalu mengambil data dari master yang sudah distandardisasi, bukan lagi dari input free text. Dengan demikian, nama agency dan brand yang dipilih user akan selalu mengacu pada data resmi yang telah dimapping sebelumnya, sehingga menghilangkan variasi penulisan, duplikasi, dan inkonsistensi data.

Selain itu, kolom bill_to, group_brand, dan group_agency memastikan campaign dapat mengikuti struktur hubungan yang berlaku antara agency, brand, dan entitas penagihan. Misalnya, bill_to hanya bisa memilih agency yang memiliki atribut is_bill_to = TRUE, sehingga penagihan campaign tepat sasaran. Sementara itu, group_brand dan group_agency memungkinkan sistem secara otomatis memahami apakah brand atau agency berada dalam grup

tertentu, sehingga data di Form Campaign Plan menjadi lebih rapi, selaras, serta sesuai standar database baru yang telah dirancang.

c) Development

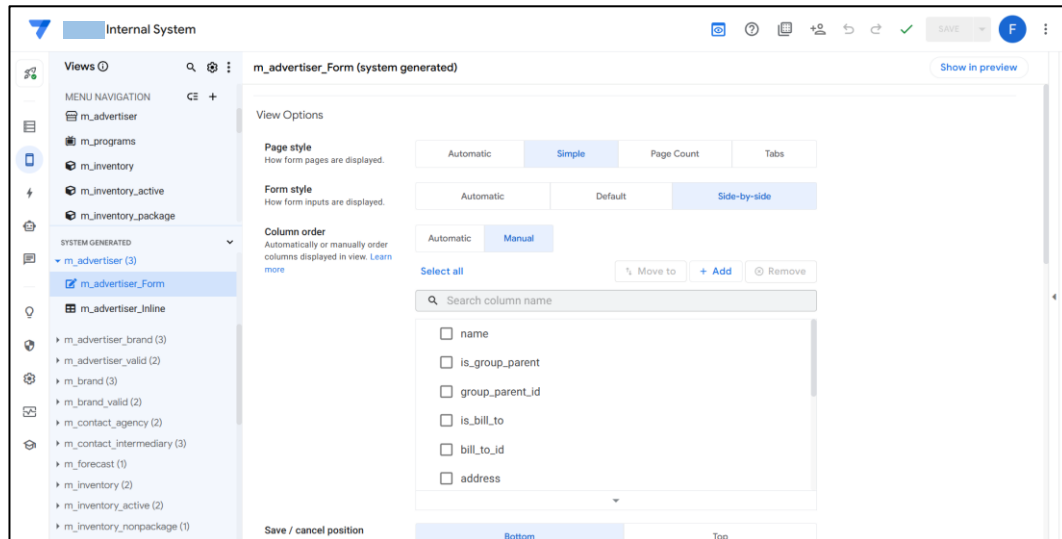
Tahap development dimulai setelah struktur database selesai dirancang dan ERD telah memastikan keterhubungan antar-entitas dapat berjalan secara konsisten. Proses implementasi dilakukan langsung di AppSheet dengan membangun setiap master data sesuai kebutuhan bisnis, dimulai dari entitas brand (m_brand), agency (m_agency), hingga integrasi keduanya melalui tabel m_agency_brand, dan akhirnya penerapan logika bisnis pada Campaigns (m_campaigns).



Gambar 3. 36 Back End Form Brand

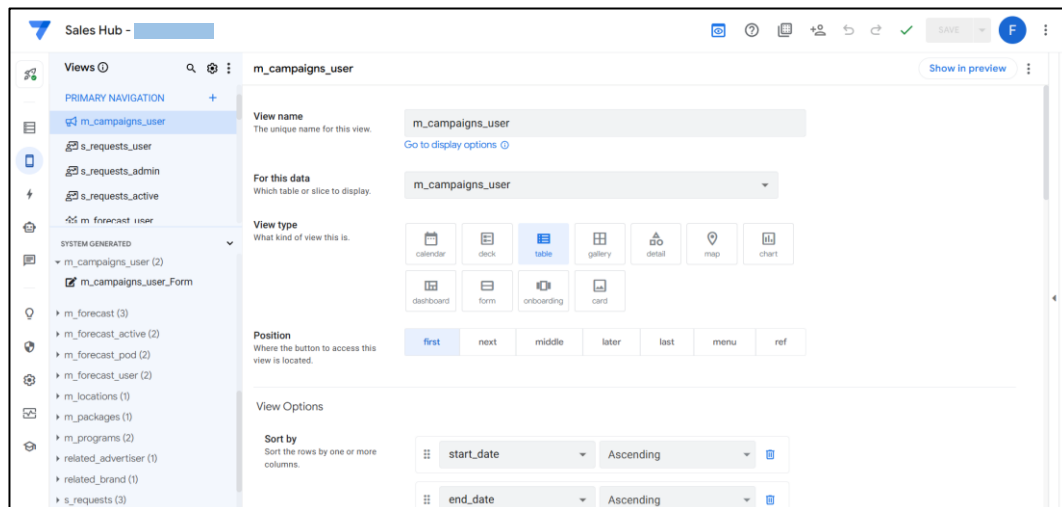
Implementasi pertama dilakukan pada entitas Brand (m_brand). Struktur backend tabel ini dibuat untuk mendukung pengelolaan merek secara terstruktur, baik sebagai brand individu maupun sebagai bagian dari grup brand. Kolom brand menyimpan nama merek, industry digunakan untuk mengidentifikasi kategori industri, sementara is_group_owner dan group_brand_id digunakan untuk mengatur relasi kepemilikan grup. Dengan struktur ini, sistem mampu

mengenali apakah suatu brand merupakan brand mandiri, brand induk, atau brand turunan.



Gambar 3. 37 Back End Form Agency

Selanjutnya, pengembangan dilakukan pada entitas agency. Tabel ini dirancang untuk mengelola struktur grup pengiklan sekaligus sistem penagihan. Kolom `is_group_parent` dan `group_parent_id` mengatur hirarki agency, sementara `is_bill_to` dan `bill_to_id` mengelola konfigurasi entitas penagihan. Dengan struktur ini, sistem dapat menentukan apakah sebuah agency dapat digunakan sebagai bill-to dan memastikan bahwa setiap campaign diarahkan ke entitas penagihan yang benar. Penambahan kolom seperti status dan notes membantu tim operasional memonitor kondisi dan catatan penting terkait masing-masing agency.



Gambar 3. 38 Back End View Campaign Plan

Tahap terakhir adalah pengembangan tabel Campaigns (m_campaigns) sebagai pusat dari seluruh perencanaan campaign. Tabel ini mengintegrasikan seluruh data dari m_agency, m_brand, dan struktur grup melalui kolom agency, brand, bill_to, group_brand, dan group_agency. Kolom agency dan brand mengambil data yang telah distandardisasi dari master, sedangkan bill_to hanya dapat memilih agency yang memiliki atribut is_bill_to = TRUE. Struktur ini memastikan bahwa setiap record campaign yang dibuat melalui Form Campaign Plan selalu konsisten, valid, serta mengikuti standar hubungan agency, brand, dan bill-to yang sudah ditetapkan. Dengan selesai-nya pembangunan tiga entitas inti ini, proses bisnis Pre-Sales menjadi lebih teratur, otomatis, dan minim kesalahan input dibandingkan proses sebelumnya yang masih menggunakan data bebas (free text).

d) Testing

Tahap testing dilakukan setelah seluruh struktur database dan logika utama di AppSheet selesai dikembangkan. Pengujian ini dilakukan oleh tim Business Process (BusPro) sebagai pengguna utama yang memahami alur kerja dan kebutuhan operasional. Proses testing mencakup pengecekan fungsi CRUD pada master data,

dimulai dari penambahan brand, agency, serta pengelolaan relasi agency dan brand untuk memastikan bahwa seluruh dependensi dan referensi berjalan sesuai desain.

Setelah master data dinyatakan valid, tim BusPro melanjutkan pengujian pada fitur inti yaitu Add Campaign melalui Form Campaign Plan. Pada tahap ini, mereka memastikan bahwa pemilihan agency dan brand muncul sesuai relasi yang telah diatur pada tabel `m_agency_brand`, termasuk verifikasi bahwa field seperti `bill_to`, `group_brand`, dan `group_agency` terisi otomatis dan konsisten. Selain itu, proses penyimpanan campaign baru diuji untuk memastikan tidak ada data yang tidak sinkron atau referensi yang gagal dikenali oleh system.

Hasil testing ini digunakan sebagai dasar untuk perbaikan minor dan penyesuaian logika ekspresi di AppSheet sebelum masuk ke tahap deployment. Dengan adanya pengujian end-to-end oleh BusPro, sistem dipastikan stabil dan siap digunakan sesuai kebutuhan operasional harian.

e) Deployment

Tahap deployment dilakukan setelah semua fitur diuji dan dipastikan berjalan dengan baik. Sebelum mulai melakukan perubahan di sistem, penulis terlebih dahulu memberikan pemberitahuan di grup PT XYZ Digital Systems Space bahwa aplikasi akan masuk masa maintenance selama 1 hari. Tujuannya agar seluruh user tahu bahwa selama proses ini berlangsung mereka tidak melakukan input data apa pun yang bisa mengganggu proses pembaruan.

Saat maintenance dimulai, penulis menerapkan semua perubahan yang sudah dirancang, seperti mengaktifkan struktur database baru, menyesuaikan data hasil mapping, dan memperbarui form di

AppSheet agar terhubung dengan tabel agency, brand, dan bill to yang terbaru. Semua pengaturan validasi dan otomatisasi juga disesuaikan agar pilihan agency, brand, dan bill to di Form Campaign Plan menjadi lebih akurat dan seragam.

Setelah semua konfigurasi selesai, penulis melakukan pengecekan ulang untuk memastikan aplikasi berjalan normal tanpa error. Ketika aplikasi dipastikan stabil, penulis membuka kembali akses penggunaan dan memberikan pemberitahuan bahwa maintenance sudah selesai serta aplikasi sudah bisa digunakan seperti biasa.

f) Review

Setelah proses deployment selesai dan sistem sudah berjalan normal, tahap berikutnya adalah melakukan sesi review untuk memastikan hasil pengembangan benar-benar sesuai dengan kebutuhan bisnis. Review ini dilakukan bersama VP Marketing, VP Sales, dan tiga sales staff yang paling aktif menggunakan sistem Pre-Sales.

Dalam sesi ini, penulis memperlihatkan alur baru mulai dari penambahan brand, penambahan agency, hingga pembuatan Campaign Plan menggunakan struktur database yang sudah distandardisasi. Para reviewer kemudian memberikan masukan terkait kemudahan penggunaan, konsistensi data, dan apakah pengaturan baru sudah membantu mengurangi kesalahan input yang sebelumnya sering terjadi.

Masukan yang diberikan digunakan sebagai bahan evaluasi akhir untuk memastikan sistem sudah siap digunakan secara penuh dalam operasional harian. Dengan adanya review ini, penulis dapat memastikan bahwa fitur yang dikembangkan benar-benar menjawab

kebutuhan tim dan memberikan peningkatan nyata terhadap proses kerja.

g) Rollout



Gambar 3. 39 Dokumentasi Rollout Fitur dengan User

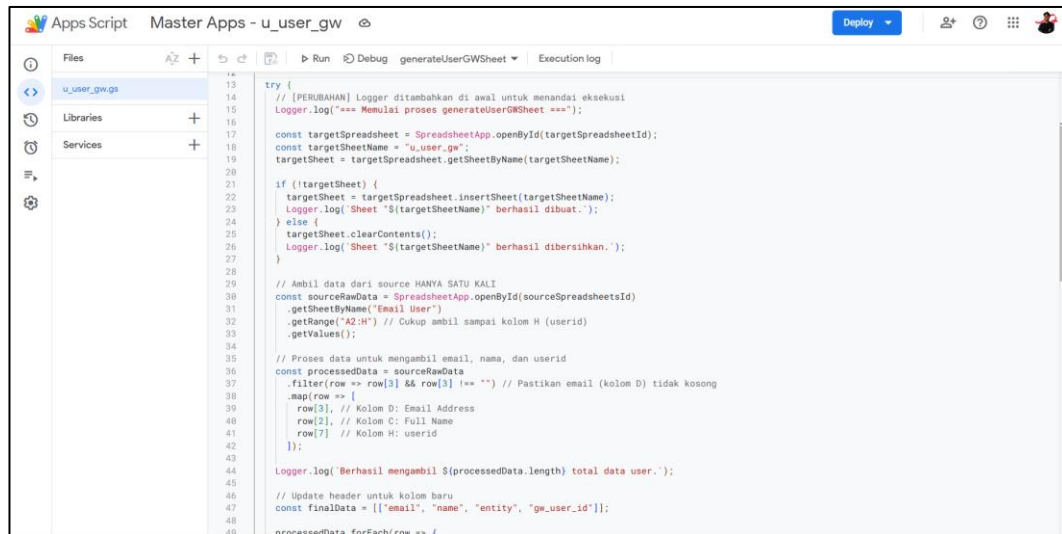
Tahap rollout dilakukan pada sesi Weekly Report Meeting yang dihadiri oleh tim Sales, tim Marketing, VP Marketing, dan VP Sales. Pada pertemuan ini, penulis memperkenalkan fitur baru terkait pengelolaan brand dan agency, termasuk cara penambahan data, hubungan antar-entitas, serta proses pembuatan Campaign Plan dengan struktur yang sudah diperbarui.

Penjelasan dilakukan secara langsung agar seluruh tim dapat memahami perbedaan antara sistem lama dan sistem baru, terutama bagaimana fitur ini membantu menjaga konsistensi data dan mempermudah proses input. Setelah sosialisasi selesai, fitur resmi dibuka untuk digunakan oleh seluruh tim Sales dalam aktivitas Pre-Sales harian.

2) Automation User Baru

Selain melakukan perbaikan pada struktur Brand dan Advertiser, penulis juga mengembangkan proses otomatisasi untuk memperbarui

daftar pengguna (user) dari database utama ke database AppSheet. Sebelumnya, pembaruan data user dilakukan secara manual dengan menyalin satu per satu dari spreadsheet HR ke dalam tabel user AppSheet, sehingga sering memakan waktu dan berpotensi terjadi kesalahan input.



```

13  try {
14    // [PERUBAHAN] Logger ditambahkan di awal untuk menandai eksekusi
15    Logger.log("=== Memulai proses generateUserGWSheet ===");
16
17    const targetSpreadsheet = SpreadsheetApp.openById(targetSpreadsheetId);
18    const targetSheetName = "u_user_gw";
19    targetSheet = targetSpreadsheet.getSheetByName(targetSheetName);
20
21    if (!targetSheet) {
22      targetSheet = targetSpreadsheet.insertSheet(targetSheetName);
23      Logger.log("Sheet '$(targetSheetName)' berhasil dibuat.");
24    } else {
25      targetSheet.clearContents();
26      Logger.log("Sheet '$(targetSheetName)' berhasil dibersihkan.");
27    }
28
29    // Ambil data dari source HANYA SATU KALI
30    const sourceRawData = SpreadsheetApp.openById(sourceSpreadsheetId)
31      .getSheetByName("Email User")
32      .getRange("A2:H") // Cukup ambil sampai kolom H (userid)
33      .getValues();
34
35    // Proses data untuk mengambil email, nama, dan userid
36    const processedData = sourceRawData
37      .filter(row => row[3] && row[3] !== "") // Pastikan email (kolom D) tidak kosong
38      .map(row => {
39        row[3], // Kolom D: Email Address
40        row[2], // Kolom C: Full Name
41        row[7] // Kolom H: userid
42      });
43
44    Logger.log('Berhasil mengambil $(processedData.length) total data user.');
```

Gambar 3. 40 Script Refresh User Baru

Untuk mengatasi masalah tersebut, penulis membuat sebuah Apps Script yang berfungsi menarik data user secara otomatis dari sheet Email User (database utama), lalu mengonversinya menjadi format yang sesuai dengan tabel u_user_gw di AppSheet. Script ini membaca email, nama, dan user ID, kemudian menambahkan kolom entity secara otomatis berdasarkan domain email (misalnya domain @PT XYZ.digital → PT XYZ Digital dan @vidio.com → Vidio). Seluruh proses tersebut kemudian dituangkan ke dalam sebuah fungsi automation. Berikut contoh tampilan kode Apps Script yang dikembangkan:

Sync User List

Effect: External: go to a website

Action name
A unique name for this action
Sync User List
[Go to display options](#)

For a record of this table
This action applies to rows of which table?
u_users

Do this
The type of action to perform
External: go to a website

Target
URL target
= "External: go to a website"

Launch External
Should the website be opened in the device's own browser, instead of in the app?
☐

Position
Where this action will be displayed in your app
Primary Prominent Inline Hide

Gambar 3. 41 Back End Action Sync User List

Setelah kode selesai dikembangkan dan diuji, tahap selanjutnya adalah membuat sebuah Action di AppSheet untuk menjalankan script tersebut. Action yang dibuat diberi nama “Sync User List” dengan tipe External: go to a website, kemudian URL Web App hasil deployment Apps Script dimasukkan ke dalam action tersebut. Dengan cara ini, tim cukup menekan satu tombol di AppSheet untuk memperbarui seluruh daftar user tanpa perlu melakukan input manual lagi.

3.3.2.3 Hasil pengembangan Proyek 2

1) Improvment Struktur Brand, Agency, & Bill To

Pada tahap ini, penulis telah mengimplementasikan perbaikan terhadap struktur data Brand, Agency, dan Bill To sesuai rancangan database yang telah disusun pada fase desain. Seluruh perubahan ini difokuskan pada peningkatan konsistensi data, validasi otomatis, serta integrasi penuh dengan master data yang ada dalam AppSheet. Hasil implementasi ini kemudian diterapkan langsung pada Form Brand dan Form Agency, seperti berikut:

Brand Form

Brand Name* Royco (Unilever)

Industry Others

Is Advertiser?* No Yes

Advertiser UNILEVER

Cancel Save

Gambar 3. 42 View Form Tambah Brand

Pada Form Brand pada Gambar 3.42, penulis melakukan penyesuaian agar seluruh data brand terhubung langsung dengan master yang sudah distandardisasi. Field Brand Name kini menggunakan referensi dari tabel m_brand, sehingga tidak lagi menerima input manual yang berpotensi menimbulkan duplikasi. Field Industry juga diperbarui menggunakan daftar kategori dari master industri agar klasifikasi brand lebih rapi. Selain itu, atribut Is Advertiser? (Group Brand) ditambahkan untuk menentukan apakah suatu brand memiliki Group Brand. Jika dipilih “Yes”, maka field Advertiser/Group Agency akan muncul dan mengambil data dari tabel m_advertiser. Dengan penyesuaian ini, data brand menjadi lebih konsisten dan siap digunakan dalam proses Campaign Plan tanpa membutuhkan perbaikan tambahan.

Agency Form

Overview Addresses Details

Agency* Mindshare Indonesia - Unilever

Is Group Agency?* No Yes

Group Agency GROUP M

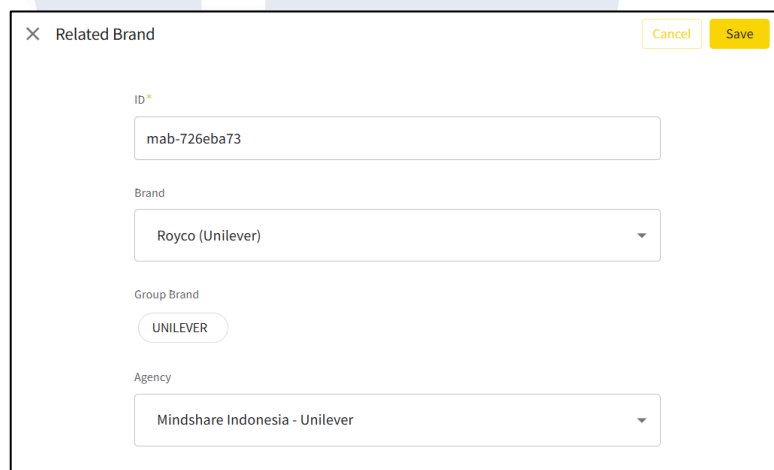
Is Bill To?* No Yes

Bill To PT. Citra Surya Indonesia Anymind Group Pte. Ltd.

Cancel Next >

Gambar 3. 43 View Form Tambah Agency

Penyesuaian serupa diterapkan pada Form Agency, seperti terlihat pada Gambar 3.43. Field Agency kini terhubung langsung dengan master m_advertiser sehingga seluruh data agency mengikuti struktur yang sama dengan master. Atribut Is Group Agency? digunakan untuk menandai apakah agency tersebut bagian dari sebuah grup, dan jika “Yes”, field Group Agency akan muncul untuk memilih grup yang tersedia. Selain itu, atribut Is Bill To? berfungsi untuk menentukan apakah agency berperan sebagai entitas penagihan, dan bila aktif, field Bill To akan menampilkan pilihan Bill To yang valid dari master. Dengan perubahan ini, struktur agency serta relasinya dengan group agency dan Bill To menjadi lebih jelas dan lebih mudah dikelola dalam proses Pre-Sales.



The image shows a web form titled "Related Brand" with a close button (X) and two buttons: "Cancel" and "Save". The form contains the following fields:

- ID ***: A text input field containing the value "mab-726eba73".
- Brand**: A dropdown menu showing "Royco (Unilever)".
- Group Brand**: A button labeled "UNILEVER".
- Agency**: A dropdown menu showing "Mindshare Indonesia - Unilever".

Gambar 3. 44 View Form Related Brand & Agency

Kemudian, di form Related Agency ke Brand pada gambar di atas digunakan untuk mencatat hubungan antara sebuah brand dan agensi yang menanganinya. Kolom ID berisi kode unik yang otomatis dibuat sistem. Kolom Brand menunjukkan brand yang ingin dikaitkan, sementara Group Brand menjelaskan grup atau perusahaan induknya. Kolom Agency mencantumkan agensi yang bekerja untuk brand tersebut, seperti untuk kebutuhan media, kreatif, atau digital. Form ini membantu memastikan setiap brand memiliki catatan agensi yang jelas dan mudah dilacak.

Gambar 3. 45 View Form Campaign Plan

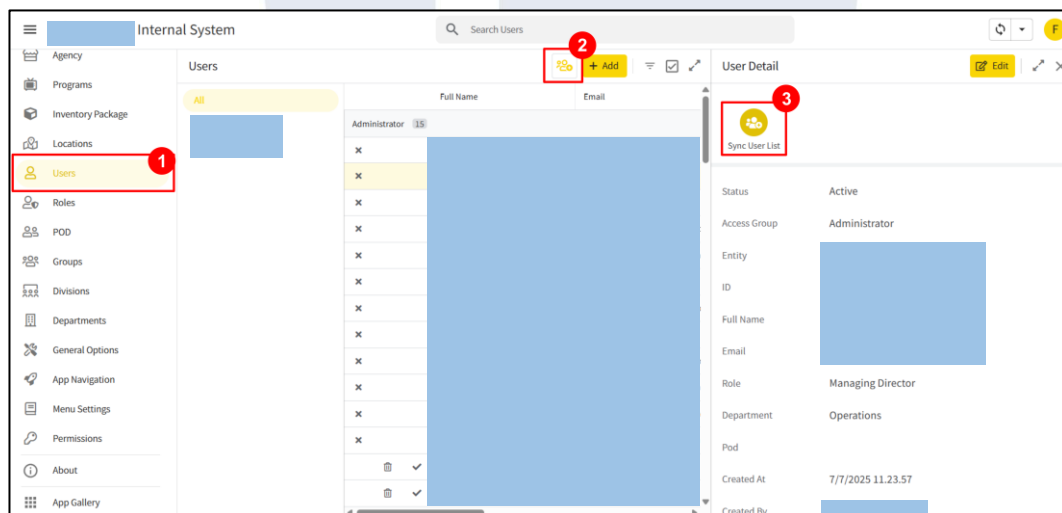
Pada tahap akhir, penyesuaian juga diterapkan pada Campaign Plan Form agar seluruh informasi kampanye mengambil data langsung dari master yang telah distandardisasi. Pada form di atas, user mengisi detail kampanye seperti Campaign Name, misalnya Campaign Testing, kemudian memilih Platform dan Billing Entity, yang dalam contoh terisi Vidio. Kolom Pitching menunjukkan apakah kampanye melibatkan proses pitching, sementara kolom Agency dan Group Agency menampilkan agensi serta grupnya melalui dropdown yang sudah terhubung ke master m_agency. Selanjutnya, kolom Brand seperti Royco (Unilever) dan Advertiser/Group Brand seperti UNILEVER memastikan pemilihan brand dan perusahaan induk selalu konsisten dengan relasi yang telah ditentukan pada database. Dengan struktur baru ini, seluruh data pada Campaign Plan menjadi jauh lebih rapi, akurat, dan otomatis terhubung dengan master data Brand, Agency, dan Bill to yang telah diperbaiki sebelumnya.

Secara keseluruhan, pengembangan pada struktur Brand, Agency, Bill To, hubungan Brand dan Agency, serta integrasi pada Campaign Plan Form berhasil meningkatkan kerapian dan konsistensi data dalam sistem Pre-Sales. Seluruh form kini terhubung dengan master data

terstruktur, sehingga proses input lebih efisien dan risiko duplikasi maupun inkonsistensi dapat ditekan. Fitur-fitur ini membuat alur kerja Pre-Sales menjadi lebih tertata, mudah ditelusuri, dan siap mendukung pengembangan lanjutan di tahap berikutnya.

3) Automation User Baru

Penyesuaian juga dilakukan pada fitur User Management untuk memastikan proses penambahan pengguna baru menjadi lebih efisien dan tetap selaras dengan master data karyawan. Pengembangan ini penting karena data user sering menjadi dasar akses dan otorisasi di seluruh modul sistem, sehingga konsistensi dan kemudahan pengelolaan menjadi prioritas.



Gambar 3. 46 View User & Button Sync User List

Pada halaman Users pada gambar di atas, admin dapat melihat daftar seluruh pengguna yang sudah terdaftar di sistem. Jika admin menambahkan user baru dan email yang dimasukkan belum tersedia, sistem menyediakan dua opsi untuk melakukan sinkronisasi. Tombol Sync User pada bagian header (nomor 2) dan tombol Sync User List pada panel detail (nomor 3) berfungsi untuk menarik data langsung dari database karyawan. Dengan fitur ini, informasi seperti nama

lengkap dan alamat email dapat terisi otomatis tanpa harus diketik manual.

Secara keseluruhan, fitur sinkronisasi user ini meningkatkan akurasi data, mempercepat proses penambahan pengguna, serta mengurangi potensi kesalahan input. Integrasi dengan database karyawan memastikan setiap user yang dibuat selalu sesuai dengan data internal perusahaan, sehingga pengelolaan akses dan peran pengguna menjadi lebih rapi dan terkontrol.

3.3.3 Kendala yang Ditemukan

Proses kerja magang sebagai bagian dari tim di PT Vidio Dot Com telah berjalan secara efektif. Namun demikian, sepanjang pelaksanaan kegiatan magang, terdapat beberapa kendala yang mempengaruhi kelancaran penyelesaian tugas dan proyek. Uraian kendala tersebut adalah sebagai berikut:

1) Akses Terbatas Terkait Generate API Gemini

Proses pembuatan API Gemini mengalami kendala karena akses yang diberikan terbatas. Hal ini menghambat eksekusi skrip secara mandiri dan memperlambat validasi data maupun pengujian fungsi API secara penuh.

2) Tidak terstandarisasi flow Brand & Advertiser

Alur kerja terkait Brand & Advertiser belum memiliki standarisasi yang jelas, sehingga proses koordinasi, pengolahan data, dan eksekusi tugas menjadi kurang efisien. Akibatnya, beberapa tahap validasi dan komunikasi membutuhkan waktu lebih lama karena prosedur yang belum seragam.

3) Batasan Platform AppSheet

Pengembangan aplikasi menggunakan AppSheet memiliki keterbatasan karena fitur-fitur tertentu tidak dapat dikustomisasi

secara bebas seperti pada website berbasis coding. Hal ini menyulitkan dalam membuat logika atau automasi yang lebih kompleks, sehingga beberapa kebutuhan teknis tidak dapat diimplementasikan secara langsung.

3.3.4 Solusi atas Kendala yang Ditemukan

Sebagai tindak lanjut atas kendala yang telah ditemukan, sejumlah langkah telah diambil untuk mengatasi permasalahan tersebut. Berikut adalah penjelasan solusi yang diterapkan:

1) Diskusi Bersama Tim Terkait Brand & Advertiser

Ketidakstandaran alur kerja diatasi dengan melakukan koordinasi intensif bersama tim terkait Brand & Advertiser. Diskusi ini membantu menyamakan pemahaman prosedur, memperjelas flow pekerjaan, dan mempercepat proses eksekusi tugas.

2) Bantuan Mentor untuk Akses API Gemini

Kendala akses API diselesaikan dengan meminta bantuan mentor untuk memperoleh hak akses yang diperlukan. Langkah ini memungkinkan eksekusi skrip dan pengujian API berjalan lebih lancar dan sesuai jadwal.

3) Konsultasi dengan Mentor, AppSheet Community, & AI

Untuk mengatasi keterbatasan AppSheet, dilakukan konsultasi dengan mentor, pencarian referensi dan solusi melalui AppSheet Community, serta memanfaatkan bantuan AI untuk menemukan alternatif logika dan pendekatan. Langkah ini membantu menemukan solusi yang paling memungkinkan dalam batasan platform AppSheet.