

## BAB III

### PELAKSANAAN KERJA MAGANG

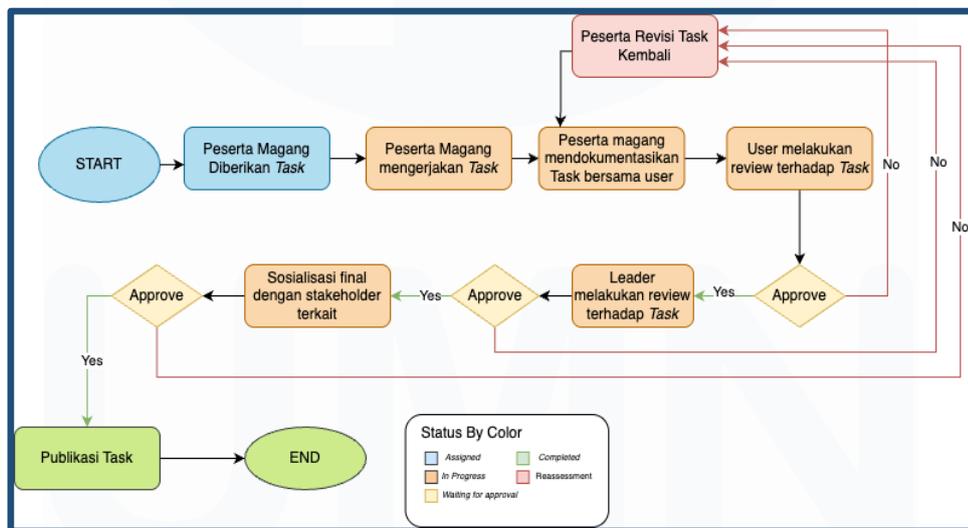
#### 3.1 Kedudukan dan Koordinasi

Posisi di PT Baller Indonesia Sentosa adalah sebagai *Data Engineer*, yang berada di bawah bimbingan seorang mentor dan supervisor dalam unit Executive Department yaitu Ibu Yuana Sari yang Secara struktural, posisi ini berada langsung di bawah president director yaitu sebagai Executive Assistant for President Director, yang berperan sebagai penghubung antara kebutuhan operasional CEO dan tim terkait. Dalam konteks ini, tugas utama *Data Engineer* mencakup mendukung pengembangan sistem berbasis *JavaScript* serta membangun *dashboard* data operasional sesuai dengan kebutuhan strategis perusahaan. Mentor bertanggung jawab untuk menerjemahkan dan menyampaikan berbagai kebutuhan CEO, termasuk dalam aspek pengolahan dan analisis data Untuk mendukung pengambilan keputusan. Oleh karena itu, peran ini berfokus pada pengembangan solusi teknologi yang dapat meningkatkan efisiensi operasional, seperti membangun sistem berbasis *Google Apps Script*, mengembangkan *dashboard* menggunakan alat visualisasi data, serta mengoptimalkan pemrosesan data agar selaras dengan kebutuhan bisnis perusahaan. Dengan demikian, posisi *Data Engineer* tidak hanya berfokus pada aspek teknis, tetapi juga memiliki peran strategis dalam mendukung eksekusi keputusan di tingkat manajemen melalui solusi berbasis data.

Bekerja dalam tim yang terdiri dari 4 anggota, dengan hanya 1 peserta *intern*, Meskipun bekerja dalam tim, terkait pengembangan form dan *JavaScript* untuk UI sepenuhnya dibuat oleh Mentor magang, serta seluruh coding dan pengembangan sistem dilakukan sepenuhnya oleh *Data Engineer Intern*. Selain itu, *dashboard* operasional perusahaan yang digunakan untuk analisis data dan pengambilan keputusan juga sepenuhnya dikembangkan oleh posisi *Data Engineer Intern*. Dengan demikian, posisi *Data Engineer* yang dijalani tidak hanya berfokus pada aspek teknis, tetapi juga memiliki peran strategis dalam mendukung eksekusi

keputusan di tingkat manajemen melalui solusi berbasis data yang dikembangkan secara langsung

Koordinasi dalam unit *Executive Department* diawali dengan *Morning Coffee*, sebuah pertemuan bulanan yang tidak hanya membahas nilai-nilai yang diterapkan dalam perusahaan, tetapi juga menjadi wadah untuk berbagi perkembangan pekerjaan antar tim. Selain diskusi terkait budaya kerja, sesi ini juga digunakan untuk update pekerjaan, di mana setiap anggota tim dapat berbagi progres tugas yang sedang dikerjakan. Selain itu, terdapat sesi review status pekerjaan untuk menilai apakah suatu tugas berada dalam kategori *done*, *in progress*, *in review*, atau *placeholder*, sehingga transparansi dan efektivitas kerja dapat terus terjaga. Melalui mekanisme ini, koordinasi dalam tim menjadi lebih terstruktur, memastikan bahwa setiap anggota memiliki pemahaman yang jelas mengenai tanggung jawabnya serta mendukung sinergi dalam mencapai tujuan perusahaan.



Gambar 3. 1 Alur Kerja Pelaksanaan Magang

Berdasarkan Gambar 3.1, alur kerja pelaksanaan magang dimulai dengan pemberian task, yang kemudian dikerjakan dan didokumentasikan bersama user. Setelah itu, user melakukan review terhadap task yang telah dikerjakan. Jika disetujui, task akan diteruskan ke leader untuk dilakukan review dan approval. Selanjutnya, task akan disosialisasikan kepada stakeholder terkait Untuk memastikan apakah terdapat tambahan user requirements. Jika ada, maka task akan

direvisi sesuai kebutuhan. Setelah mendapatkan persetujuan akhir, task akan dipublikasikan dan dianggap selesai. Namun, jika pada tahap manapun task tidak mendapatkan persetujuan, sehingga melakukan revisi hingga memenuhi standar yang ditetapkan.

PT Baller Indonesia Sentosa menggunakan *Slack* sebagai platform komunikasi utama, dengan berbagai channel yang disesuaikan berdasarkan kebutuhan masing-masing tim. Jika terjadi *issue* atau masalah yang memerlukan penanganan segera, tim dapat langsung mengangkatnya di channel yang relevan, sehingga seluruh anggota tim yang terkait dapat segera berkoordinasi untuk menemukan solusi dengan cepat. Untuk keperluan *meeting* atau diskusi formal, perusahaan memanfaatkan Google Meet, dengan jadwal yang terorganisir melalui Google Calendar Untuk menghindari bentrok antar anggota tim. Sementara itu, komunikasi yang bersifat lebih informal serta pengajuan perizinan biasanya dilakukan melalui WhatsApp.

### **3.2 Tugas dan Uraian Kerja Magang**

Tugas sebagai *Data Engineer* selama periode magang di PT Baller Indonesia Sentosa mencakup berbagai aktivitas yang mendukung pengolahan dan analisis data untuk kebutuhan operasional perusahaan. Tugas utama meliputi pengembangan sistem berbasis *JavaScript*, membangun dan mengoptimalkan *dashboard* data operasional, serta mengelola data *pipeline* Untuk memastikan integrasi data yang efisien. Selain itu, *data engineer intern* bertanggung jawab untuk menganalisis dan membersihkan data, mengembangkan automasi proses bisnis menggunakan *Google Apps Script*, serta mendukung kebutuhan tim dalam penyajian data yang akurat untuk pengambilan keputusan. Selama periode magang, peserta juga berkolaborasi dengan berbagai tim untuk memahami kebutuhan bisnis, melakukan debugging dan troubleshooting sistem, serta memastikan bahwa solusi data yang dikembangkan sesuai dengan standar perusahaan. Rincian aktivitas *Data Engineer Intern* dapat dilihat pada Tabel 3.1

Tabel 3. 1 Tugas Dan Uraian Kerja Magang

No	Pekerjaan yang dilakukan	Tim	Start Date	End Date	Hasil
<b>On Boarding dan Sesi Training Pengenalan Lingkungan Kantor</b>					
3.2.1	<i>Introduction Work Environment dan mempelajari Data Engineer Division SOPs</i>	<i>System And Data</i>	05/02/2025	05/02/2025	Lebih Mengetahui Nilai lingkungan kerja dan SOP
<b>HR &amp; Sales Operational System Development with JavaScript and Google Apps Script</b>					
3.2.2	Mengembangkan <i>Budget Usage Form</i> dengan <i>Submit, Clear, Load, Approval, dan Print</i> Fungsi dengan <i>Java Programming Language</i>	<i>System And Data</i>	05/02/2025	14/02/2025	Budget Usage Form yang dapat digunakan untuk <i>HR</i> jika dinilai terdapat Permintaan untuk penggunaan Budget
3.2.3	Mengembangkan <i>KOL Collaboration Agreement Form</i> dengan <i>Submit, Clear, Load, Approval, dan Print Functions</i> menggunakan <i>Java Programming Language</i>	<i>System And Data</i>	17/02/2025	21/02/2025	<i>KOL Collaboration Agreement Form</i> yang dapat digunakan untuk <i>sales team</i> terkait kebutuhan laporan Kolaborasi dengan KOL
3.2.4	Mengembangkan <i>Submit and Clear Buttons for Master Product/Stock Product Form</i> menggunakan <i>Java dan Google Apps Script Connected to Google Spreadsheet</i>	<i>System And Data</i>	24/02/2025	28/02/2025	<i>Master Product/Stock Product Form</i> yang dapat digunakan untuk <i>Sales Team</i>
3.2.5	Mengembangkan <i>Submit, Clear, Load, Approval, dan Print Buttons</i> untuk <i>Business Trip Forms</i> menggunakan <i>Java Script</i>	<i>System And Data</i>	07/03/2025	10/03/2025	<i>Business Trip Forms</i> yang dapat digunakan untuk kebutuhan HR team
3.2.6	Mengembangkan <i>Submit, Clear, Load, Approval, dan Print Buttons</i> untuk <i>Realization Business</i>	<i>System And Data</i>	10/03/2025	12/03/2024	<i>Realization Business Trip Forms</i> yang dapat digunakan untuk

No	Pekerjaan yang dilakukan	Tim	Start Date	End Date	Hasil
	<i>Trip Forms menggunakan Java Script</i>				kebutuhan HR team
3.2.7	<i>Mengembangkan Submit, Clear, Load, Approval, dan Submit Payment Buttons untuk Invoice Professional organizing forms dengan JavaScript</i>	<i>System And Data</i>	12/03/2025	17/03/2025	<i>Professional organizing form digunakan untuk kebutuhan team Sales</i>
3.2.8	<i>Mengembangkan Email Finance Accounting Buttons database Customer POGL forms menggunakan JavaScript</i>	<i>System And Data</i>	12/03/2025	18/03/2025	<i>database Customer POGL forms untuk kebutuhan team Accounting</i>
3.2.9	<i>Mengembangkan Submit, Clear Buttons Database Client B2B forms dengan JavaScript</i>	<i>System And Data</i>	12/03/2025	19/03/2025	<i>Database Client B2B forms digunakan untuk kebutuhan team Sales</i>
3.2.10	<i>Mengembangkan of Petty Cash Form untuk Submit, Clear, Load, Approval, dan Print Functions dengan Java Programming Language</i>	<i>System And Data</i>	13/03/2025	20/03/2025	<i>Petty Cash Form digunakan untuk team Sales</i>
3.2.11	<i>Mengembangkan of Log Book Stock Form dengan Submit, dan Clear Functions menggunakan Java Programming Language</i>	<i>System And Data</i>	14/03/2025	21/03/2025	<i>Log Book Stock Form digunakan untuk Team Sales</i>
3.2.12	<i>Mengembangkan of Log Book Coffe Bar Stock Form dengan Submit, dan Clear Functions menggunakan Java Programming Language</i>	<i>System And Data</i>	14/03/2025	24/03/2025	<i>Log Book CoffeBar Form digunakan untuk Team Sales</i>
3.2.13	<i>Mengembangkan Product Usage Report POGL form dengan Submit Functions menggunakan Java</i>	<i>System And Data</i>	14/03/2025	25/03/2025	<i>Product Usage Report POGL form digunakan Team Warehouse</i>

No	Pekerjaan yang dilakukan	Tim	Start Date	End Date	Hasil
	<i>Programming Language</i>				
3.2.14	Mengembangkan Internal Transfer Request form dengan Submit Functions menggunakan Java Programming Language	System And Data	26/03/2025	19/04/2025	Internal Transfer Request form digunakan Team Warehouse
3.2.15	Development Leave Request Form with Submit clear load and approval Functions Using Java Programming Language	System And Data	06/05/2025	17/05/2025	Leave Request Form Digunakan Hr divisi
<b>Development of HR &amp; Sales Operational Dashboard Using Google Looker Studio for Data Visualization and Performance Monitoring</b>					
3.2.16	Mengembangkan Customer Purchase & Warranty Dashboard dengan Sales Trends & SKU-Based Analysis	System And Data	25/02/2025	27/02/2025	Customer Purchase & Warranty Dashboard yang digunakan untuk team Sales
3.2.17	Mengembangkan 2025 Marketing Department KPI Dashboard untuk PT Baller Indonesia Sentosa	System And Data	03/03/2025	04/03/2025	Marketing Department KPI Dashboard berupa fitur - fitur nya berupa Sales Target Nett Yearly, Sales Target Gross Yearly, Gross Profit (%), ROI Marketing, ROI Digital, CAC dan Brand Awareness Growth.
3.2.18	Mengembangkan 2025 Sales Department KPI Dashboard untuk PT Baller Indonesia Sentosa	System And Data	05/03/2025	06/03/2025	Sales Department KPI Dashboard fitur fitur nya berupa B2B Sales, Sales Target, Customer & Service Center Satisfaction rate, Sales Target on Store, Customer Service online sales Satisfaction Rate, Live

No	Pekerjaan yang dilakukan	Tim	Start Date	End Date	Hasil
					<i>Streaming AVG revenue per sessions,</i>
3.2.19	Mengembangkan 2025 Warehouse & Operation Department KPI Dashboard for PT Baller Indonesia Sentosa	System And Data	06/03/2025	07/03/2025	<i>Warehouse &amp; Operation Department KPI Dashboard fiturnya berupa Productivity Rate, Inventory Accuracy / Shrinkage, Report on WH Accuracy, On time Order Processing, Order Accuracy, Service Warranty Satisfaction Rate, Layouting</i>
3.2.20	Mengembangkan E-Commerce Performance Dashboard for Traffic, Purchase, and Conversion Analysis	System And Data	11/03/2025	14/03/2025	<i>E-Commerce Performance Dashboard dengan fitur berupa Traffic, Purchase, and Conversion Analysis</i>
3.2.21	Mengembangkan Inventory Movement & Stock Status Dashboard PT Baller Indonesia	System And Data	21/04/2025	06/05/2025	<i>Inventory Movement &amp; Stock Status Dashboard dengan fitur berupa tren aktivitas barang masuk dan keluar</i>

Berdasarkan Tabel 3.1 tugas selama magang sebagai *Data Engineer* di PT Baller Indonesia Sentosa. Pekerjaan mencakup onboarding, pelatihan SOP lingkungan kerja, serta pengembangan sistem menggunakan *JavaScript* dan *Google Apps Script*. Proyek utama meliputi pembuatan berbagai formulir operasional (*Budget Usage, KOL Collaboration, Business Trip, Petty Cash, dan Stock Log*) dengan fungsi submit, approval, dan cetak. Selain itu, magang juga mengembangkan *dashboard* HR dan Sales di *Google Looker Studio* untuk pemantauan kinerja, termasuk *dashboard* tren penjualan, KPI pemasaran, operasional gudang, dan analisis kinerja E-commerce.

### 3.2.1 Pengenalan Lingkungan Kerja dan Pembelajaran SOP Divisi Data Engineer

Pada minggu pertama magang, dilakukan pengenalan lingkungan kerja, anggota tim, serta struktur organisasi PT Baller Indonesia Sentosa. Pemahaman mengenai bisnis perusahaan, sejarah, rantai bisnis, dan produk yang dihasilkan juga diperkenalkan. Selain itu, dilakukan rapat awal untuk membahas proyek yang akan dikerjakan, yaitu pengembangan sistem operasional berbasis *JavaScript* serta *dashboard* perusahaan. *Intern* diberikan tugas sesuai dengan keahlian yang dimiliki, terutama dalam mengotomatisasi form bisnis operasional menggunakan *JavaScript*, serta melakukan visualisasi data menggunakan Power BI dan proses *business intelligence*. Pada minggu pertama, *intern* juga melakukan setup perangkat kerja seperti laptop, Slack, dan Gmail untuk kebutuhan komunikasi dan koordinasi.



Gambar 3. 2 Training Dengan HR

Berdasarkan Gambar 3.2, sesi *training* dengan tim HR dilakukan untuk memahami sejarah perusahaan, produk, mitra B2B, serta nilai-nilai yang berpedoman pada visi dan misi perusahaan. Selain itu, HR juga menjelaskan aturan kerja (*do and don'ts*) yang harus dipatuhi selama bekerja di PT Baller Indonesia Sentosa. Hasil psikotes juga dibacakan untuk memberikan gambaran mengenai potensi dan karakteristik *intern* dalam bekerja Sebagai

bagian dari onboarding, *intern* juga mengikuti sesi foto untuk LinkedIn Untuk mendukung profil profesional selama menjalani program magang.

*Intern* juga mengikuti meeting bersama user untuk memahami alur kerja serta sistem yang sudah tersedia di perusahaan. Dalam sesi ini, diberikan penjelasan lebih rinci mengenai job description dan tanggung jawab yang harus dijalankan selama periode magang, Untuk memastikan keselarasan dengan kebutuhan perusahaan. Selain itu, *intern* diberikan akses ke dokumentasi sistem yang sudah ada untuk mempelajari proses kerja secara mandiri serta memahami alur bisnis yang berjalan. Diskusi dengan tim juga dilakukan untuk mengklarifikasi tugas, ekspektasi, serta target yang harus dicapai selama magang berlangsung

### 3.2.2 *Developmet Budget Usage Form dengan Function Java Programming Language*

*Budget Usage Form* adalah sebuah formulir digital yang digunakan untuk mencatat, mengajukan, dan menyetujui penggunaan anggaran dalam perusahaan. Periode pengerjaan formulir ini berlangsung selama dua minggu. Tujuannya adalah untuk mempermudah proses administrasi keuangan, memastikan transparansi dalam penggunaan dana, serta mempercepat proses persetujuan anggaran oleh pihak terkait.

FORM PENGAJUAN PEMAKAIAN BUDGET					
1					
2					
3					
4	Tanggal		No. Surat		
5					
6	<b>KETERANGAN FORM</b>				
7	Nama Brand				
8	Pakai Budget Department		Sub-Department		
9	Status				
10	Nama Pembuat Form		Timestamp	E-Sign by	
11	Nama Approver 1		Timestamp	E-Sign by	
12	Nama Approver 2		Timestamp	E-Sign by	
13					
14	<b>DETAIL FORM</b>				
15	Tipe Pemakaian Budget		Kategori Keperluan / Barang		
16					
17	<b>DETAIL PEMBAYARAN / REIMBURSE</b>				
18	PT atau Pribadi		PPh	0.00%	PPN 0%
19	Nama PT				

Gambar 3. 3 UI form Pemakaian Budget

Gambar 3.3 menunjukkan *UI Form* Pemakaian Budget, yang dirancang untuk mempermudah pencatatan, pengajuan, dan persetujuan anggaran di perusahaan. Formulir ini memiliki berbagai kolom yang menyerupai format *spreadsheet*, memungkinkan pengguna untuk menginput dan mengelola data anggaran dengan lebih sistematis. Dalam pengembangannya, beberapa fungsi utama telah ditambahkan, *Submit*, *Clear*, *Load Submit for Approval 1 & Approval 2* untuk proses persetujuan bertahap, serta *Print* untuk mencetak laporan. Seluruh data yang diinput akan otomatis tersimpan dalam *database* perusahaan.

```

function submitData() {
    const url = Spreadsheet.getURL();
    const response = $.ajax({url: url});
    if (response == "success") {
        ui.alert("Pengiriman data dibatalkan.");
        return;
    }
    const ss = Spreadsheet.getActiveSpreadsheet();
    const spreadsheet = ss.getActiveSheet();
    const databaseId = "0K_F0h0u0d1L0k0bJ0d097A0P0u0B0m0d0z0u0g";
    const spreadsheetName = "DB-002";
    const userEmail = session.getActiveUser().getEmail();

    const cellMapping = [
        "B4", "C3", "E8", "H8", "I18", "J18", "K18", "L18", "M18", "N18", "O18", "P18", "Q18", "R18", "S18", "T18", "U18", "V18", "W18", "X18", "Y18", "Z18", "AA18", "AB18", "AC18", "AD18", "AE18", "AF18", "AG18", "AH18", "AI18", "AJ18", "AK18", "AL18", "AM18", "AN18", "AO18", "AP18", "AQ18", "AR18", "AS18", "AT18", "AU18", "AV18", "AW18", "AX18", "AY18", "AZ18", "BA18", "BB18", "BC18", "BD18", "BE18", "BF18", "BG18", "BH18", "BI18", "BJ18", "BK18", "BL18", "BM18", "BN18", "BO18", "BP18", "BQ18", "BR18", "BS18", "BT18", "BU18", "BV18", "BW18", "BX18", "BY18", "C4", "D4", "E4", "F4", "G4", "H4", "I4", "J4", "K4", "L4", "M4", "N4", "O4", "P4", "Q4", "R4", "S4", "T4", "U4", "V4", "W4", "X4", "Y4", "Z4", "AA4", "AB4", "AC4", "AD4", "AE4", "AF4", "AG4", "AH4", "AI4", "AJ4", "AK4", "AL4", "AM4", "AN4", "AO4", "AP4", "AQ4", "AR4", "AS4", "AT4", "AU4", "AV4", "AW4", "AX4", "AY4", "AZ4", "BA4", "BB4", "BC4", "BD4", "BE4", "BF4", "BG4", "BH4", "BI4", "BJ4", "BK4", "BL4", "BM4", "BN4", "BO4", "BP4", "BQ4", "BR4", "BS4", "BT4", "BU4", "BV4", "BW4", "BX4", "BY4", "C8", "D8", "E8", "F8", "G8", "H8", "I8", "J8", "K8", "L8", "M8", "N8", "O8", "P8", "Q8", "R8", "S8", "T8", "U8", "V8", "W8", "X8", "Y8", "Z8", "AA8", "AB8", "AC8", "AD8", "AE8", "AF8", "AG8", "AH8", "AI8", "AJ8", "AK8", "AL8", "AM8", "AN8", "AO8", "AP8", "AQ8", "AR8", "AS8", "AT8", "AU8", "AV8", "AW8", "AX8", "AY8", "AZ8", "BA8", "BB8", "BC8", "BD8", "BE8", "BF8", "BG8", "BH8", "BI8", "BJ8", "BK8", "BL8", "BM8", "BN8", "BO8", "BP8", "BQ8", "BR8", "BS8", "BT8", "BU8", "BV8", "BW8", "BX8", "BY8", "C12", "D12", "E12", "F12", "G12", "H12", "I12", "J12", "K12", "L12", "M12", "N12", "O12", "P12", "Q12", "R12", "S12", "T12", "U12", "V12", "W12", "X12", "Y12", "Z12", "AA12", "AB12", "AC12", "AD12", "AE12", "AF12", "AG12", "AH12", "AI12", "AJ12", "AK12", "AL12", "AM12", "AN12", "AO12", "AP12", "AQ12", "AR12", "AS12", "AT12", "AU12", "AV12", "AW12", "AX12", "AY12", "AZ12", "BA12", "BB12", "BC12", "BD12", "BE12", "BF12", "BG12", "BH12", "BI12", "BJ12", "BK12", "BL12", "BM12", "BN12", "BO12", "BP12", "BQ12", "BR12", "BS12", "BT12", "BU12", "BV12", "BW12", "BX12", "BY12", "C16", "D16", "E16", "F16", "G16", "H16", "I16", "J16", "K16", "L16", "M16", "N16", "O16", "P16", "Q16", "R16", "S16", "T16", "U16", "V16", "W16", "X16", "Y16", "Z16", "AA16", "AB16", "AC16", "AD16", "AE16", "AF16", "AG16", "AH16", "AI16", "AJ16", "AK16", "AL16", "AM16", "AN16", "AO16", "AP16", "AQ16", "AR16", "AS16", "AT16", "AU16", "AV16", "AW16", "AX16", "AY16", "AZ16", "BA16", "BB16", "BC16", "BD16", "BE16", "BF16", "BG16", "BH16", "BI16", "BJ16", "BK16", "BL16", "BM16", "BN16", "BO16", "BP16", "BQ16", "BR16", "BS16", "BT16", "BU16", "BV16", "BW16", "BX16", "BY16", "C20", "D20", "E20", "F20", "G20", "H20", "I20", "J20", "K20", "L20", "M20", "N20", "O20", "P20", "Q20", "R20", "S20", "T20", "U20", "V20", "W20", "X20", "Y20", "Z20", "AA20", "AB20", "AC20", "AD20", "AE20", "AF20", "AG20", "AH20", "AI20", "AJ20", "AK20", "AL20", "AM20", "AN20", "AO20", "AP20", "AQ20", "AR20", "AS20", "AT20", "AU20", "AV20", "AW20", "AX20", "AY20", "AZ20", "BA20", "BB20", "BC20", "BD20", "BE20", "BF20", "BG20", "BH20", "BI20", "BJ20", "BK20", "BL20", "BM20", "BN20", "BO20", "BP20", "BQ20", "BR20", "BS20", "BT20", "BU20", "BV20", "BW20", "BX20", "BY20", "C24", "D24", "E24", "F24", "G24", "H24", "I24", "J24", "K24", "L24", "M24", "N24", "O24", "P24", "Q24", "R24", "S24", "T24", "U24", "V24", "W24", "X24", "Y24", "Z24", "AA24", "AB24", "AC24", "AD24", "AE24", "AF24", "AG24", "AH24", "AI24", "AJ24", "AK24", "AL24", "AM24", "AN24", "AO24", "AP24", "AQ24", "AR24", "AS24", "AT24", "AU24", "AV24", "AW24", "AX24", "AY24", "AZ24", "BA24", "BB24", "BC24", "BD24", "BE24", "BF24", "BG24", "BH24", "BI24", "BJ24", "BK24", "BL24", "BM24", "BN24", "BO24", "BP24", "BQ24", "BR24", "BS24", "BT24", "BU24", "BV24", "BW24", "BX24", "BY24", "C28", "D28", "E28", "F28", "G28", "H28", "I28", "J28", "K28", "L28", "M28", "N28", "O28", "P28", "Q28", "R28", "S28", "T28", "U28", "V28", "W28", "X28", "Y28", "Z28", "AA28", "AB28", "AC28", "AD28", "AE28", "AF28", "AG28", "AH28", "AI28", "AJ28", "AK28", "AL28", "AM28", "AN28", "AO28", "AP28", "AQ28", "AR28", "AS28", "AT28", "AU28", "AV28", "AW28", "AX28", "AY28", "AZ28", "BA28", "BB28", "BC28", "BD28", "BE28", "BF28", "BG28", "BH28", "BI28", "BJ28", "BK28", "BL28", "BM28", "BN28", "BO28", "BP28", "BQ28", "BR28", "BS28", "BT28", "BU28", "BV28", "BW28", "BX28", "BY28", "C32", "D32", "E32", "F32", "G32", "H32", "I32", "J32", "K32", "L32", "M32", "N32", "O32", "P32", "Q32", "R32", "S32", "T32", "U32", "V32", "W32", "X32", "Y32", "Z32", "AA32", "AB32", "AC32", "AD32", "AE32", "AF32", "AG32", "AH32", "AI32", "AJ32", "AK32", "AL32", "AM32", "AN32", "AO32", "AP32", "AQ32", "AR32", "AS32", "AT32", "AU32", "AV32", "AW32", "AX32", "AY32", "AZ32", "BA32", "BB32", "BC32", "BD32", "BE32", "BF32", "BG32", "BH32", "BI32", "BJ32", "BK32", "BL32", "BM32", "BN32", "BO32", "BP32", "BQ32", "BR32", "BS32", "BT32", "BU32", "BV32", "BW32", "BX32", "BY32", "C36", "D36", "E36", "F36", "G36", "H36", "I36", "J36", "K36", "L36", "M36", "N36", "O36", "P36", "Q36", "R36", "S36", "T36", "U36", "V36", "W36", "X36", "Y36", "Z36", "AA36", "AB36", "AC36", "AD36", "AE36", "AF36", "AG36", "AH36", "AI36", "AJ36", "AK36", "AL36", "AM36", "AN36", "AO36", "AP36", "AQ36", "AR36", "AS36", "AT36", "AU36", "AV36", "AW36", "AX36", "AY36", "AZ36", "BA36", "BB36", "BC36", "BD36", "BE36", "BF36", "BG36", "BH36", "BI36", "BJ36", "BK36", "BL36", "BM36", "BN36", "BO36", "BP36", "BQ36", "BR36", "BS36", "BT36", "BU36", "BV36", "BW36", "BX36", "BY36", "C40", "D40", "E40", "F40", "G40", "H40", "I40", "J40", "K40", "L40", "M40", "N40", "O40", "P40", "Q40", "R40", "S40", "T40", "U40", "V40", "W40", "X40", "Y40", "Z40", "AA40", "AB40", "AC40", "AD40", "AE40", "AF40", "AG40", "AH40", "AI40", "AJ40", "AK40", "AL40", "AM40", "AN40", "AO40", "AP40", "AQ40", "AR40", "AS40", "AT40", "AU40", "AV40", "AW40", "AX40", "AY40", "AZ40", "BA40", "BB40", "BC40", "BD40", "BE40", "BF40", "BG40", "BH40", "BI40", "BJ40", "BK40", "BL40", "BM40", "BN40", "BO40", "BP40", "BQ40", "BR40", "BS40", "BT40", "BU40", "BV40", "BW40", "BX40", "BY40", "C44", "D44", "E44", "F44", "G44", "H44", "I44", "J44", "K44", "L44", "M44", "N44", "O44", "P44", "Q44", "R44", "S44", "T44", "U44", "V44", "W44", "X44", "Y44", "Z44", "AA44", "AB44", "AC44", "AD44", "AE44", "AF44", "AG44", "AH44", "AI44", "AJ44", "AK44", "AL44", "AM44", "AN44", "AO44", "AP44", "AQ44", "AR44", "AS44", "AT44", "AU44", "AV44", "AW44", "AX44", "AY44", "AZ44", "BA44", "BB44", "BC44", "BD44", "BE44", "BF44", "BG44", "BH44", "BI44", "BJ44", "BK44", "BL44", "BM44", "BN44", "BO44", "BP44", "BQ44", "BR44", "BS44", "BT44", "BU44", "BV44", "BW44", "BX44", "BY44", "C48", "D48", "E48", "F48", "G48", "H48", "I48", "J48", "K48", "L48", "M48", "N48", "O48", "P48", "Q48", "R48", "S48", "T48", "U48", "V48", "W48", "X48", "Y48", "Z48", "AA48", "AB48", "AC48", "AD48", "AE48", "AF48", "AG48", "AH48", "AI48", "AJ48", "AK48", "AL48", "AM48", "AN48", "AO48", "AP48", "AQ48", "AR48", "AS48", "AT48", "AU48", "AV48", "AW48", "AX48", "AY48", "AZ48", "BA48", "BB48", "BC48", "BD48", "BE48", "BF48", "BG48", "BH48", "BI48", "BJ48", "BK48", "BL48", "BM48", "BN48", "BO48", "BP48", "BQ48", "BR48", "BS48", "BT48", "BU48", "BV48", "BW48", "BX48", "BY48", "C52", "D52", "E52", "F52", "G52", "H52", "I52", "J52", "K52", "L52", "M52", "N52", "O52", "P52", "Q52", "R52", "S52", "T52", "U52", "V52", "W52", "X52", "Y52", "Z52", "AA52", "AB52", "AC52", "AD52", "AE52", "AF52", "AG52", "AH52", "AI52", "AJ52", "AK52", "AL52", "AM52", "AN52", "AO52", "AP52", "AQ52", "AR52", "AS52", "AT52", "AU52", "AV52", "AW52", "AX52", "AY52", "AZ52", "BA52", "BB52", "BC52", "BD52", "BE52", "BF52", "BG52", "BH52", "BI52", "BJ52", "BK52", "BL52", "BM52", "BN52", "BO52", "BP52", "BQ52", "BR52", "BS52", "BT52", "BU52", "BV52", "BW52", "BX52", "BY52", "C56", "D56", "E56", "F56", "G56", "H56", "I56", "J56", "K56", "L56", "M56", "N56", "O56", "P56", "Q56", "R56", "S56", "T56", "U56", "V56", "W56", "X56", "Y56", "Z56", "AA56", "AB56", "AC56", "AD56", "AE56", "AF56", "AG56", "AH56", "AI56", "AJ56", "AK56", "AL56", "AM56", "AN56", "AO56", "AP56", "AQ56", "AR56", "AS56", "AT56", "AU56", "AV56", "AW56", "AX56", "AY56", "AZ56", "BA56", "BB56", "BC56", "BD56", "BE56", "BF56", "BG56", "BH56", "BI56", "BJ56", "BK56", "BL56", "BM56", "BN56", "BO56", "BP56", "BQ56", "BR56", "BS56", "BT56", "BU56", "BV56", "BW56", "BX56", "BY56", "C60", "D60", "E60", "F60", "G60", "H60", "I60", "J60", "K60", "L60", "M60", "N60", "O60", "P60", "Q60", "R60", "S60", "T60", "U60", "V60", "W60", "X60", "Y60", "Z60", "AA60", "AB60", "AC60", "AD60", "AE60", "AF60", "AG60", "AH60", "AI60", "AJ60", "AK60", "AL60", "AM60", "AN60", "AO60", "AP60", "AQ60", "AR60", "AS60", "AT60", "AU60", "AV60", "AW60", "AX60", "AY60", "AZ60", "BA60", "BB60", "BC60", "BD60", "BE60", "BF60", "BG60", "BH60", "BI60", "BJ60", "BK60", "BL60", "BM60", "BN60", "BO60", "BP60", "BQ60", "BR60", "BS60", "BT60", "BU60", "BV60", "BW60", "BX60", "BY60", "C64", "D64", "E64", "F64", "G64", "H64", "I64", "J64", "K64", "L64", "M64", "N64", "O64", "P64", "Q64", "R64", "S64", "T64", "U64", "V64", "W64", "X64", "Y64", "Z64", "AA64", "AB64", "AC64", "AD64", "AE64", "AF64", "AG64", "AH64", "AI64", "AJ64", "AK64", "AL64", "AM64", "AN64", "AO64", "AP64", "AQ64", "AR64", "AS64", "AT64", "AU64", "AV64", "AW64", "AX64", "AY64", "AZ64", "BA64", "BB64", "BC64", "BD64", "BE64", "BF64", "BG64", "BH64", "BI64", "BJ64", "BK64", "BL64", "BM64", "BN64", "BO64", "BP64", "BQ64", "BR64", "BS64", "BT64", "BU64", "BV64", "BW64", "BX64", "BY64", "C68", "D68", "E68", "F68", "G68", "H68", "I68", "J68", "K68", "L68", "M68", "N68", "O68", "P68", "Q68", "R68", "S68", "T68", "U68", "V68", "W68", "X68", "Y68", "Z68", "AA68", "AB68", "AC68", "AD68", "AE68", "AF68", "AG68", "AH68", "AI68", "AJ68", "AK68", "AL68", "AM68", "AN68", "AO68", "AP68", "AQ68", "AR68", "AS68", "AT68", "AU68", "AV68", "AW68", "AX68", "AY68", "AZ68", "BA68", "BB68", "BC68", "BD68", "BE68", "BF68", "BG68", "BH68", "BI68", "BJ68", "BK68", "BL68", "BM68", "BN68", "BO68", "BP68", "BQ68", "BR68", "BS68", "BT68", "BU68", "BV68", "BW68", "BX68", "BY68", "C72", "D72", "E72", "F72", "G72", "H72", "I72", "J72", "K72", "L72", "M72", "N72", "O72", "P72", "Q72", "R72", "S72", "T72", "U72", "V72", "W72", "X72", "Y72", "Z72", "AA72", "AB72", "AC72", "AD72", "AE72", "AF72", "AG72", "AH72", "AI72", "AJ72", "AK72", "AL72", "AM72", "AN72", "AO72", "AP72", "AQ72", "AR72", "AS72", "AT72", "AU72", "AV72", "AW72", "AX72", "AY72", "AZ72", "BA72", "BB72", "BC72", "BD72", "BE72", "BF72", "BG72", "BH72", "BI72", "BJ72", "BK72", "BL72", "BM72", "BN72", "BO72", "BP72", "BQ72", "BR72", "BS72", "BT72", "BU72", "BV72", "BW72", "BX72", "BY72", "C76", "D76", "E76", "F76", "G76", "H76", "I76", "J76", "K76", "L76", "M76", "N76", "O76", "P76", "Q76", "R76", "S76", "T76", "U76", "V76", "W76", "X76", "Y76", "Z76", "AA76", "AB76", "AC76", "AD76", "AE76", "AF76", "AG76", "AH76", "AI76", "AJ76", "AK76", "AL76", "AM76", "AN76", "AO76", "AP76", "AQ76", "AR76", "AS76", "AT76", "AU76", "AV76", "AW76", "AX76", "AY76", "AZ76", "BA76", "BB76", "BC76", "BD76", "BE76", "BF76", "BG76", "BH76", "BI76", "BJ76", "BK76", "BL76", "BM76", "BN76", "BO76", "BP76", "BQ76", "BR76", "BS76", "BT76", "BU76", "BV76", "BW76", "BX76", "BY76", "C80", "D80", "E80", "F80", "G80", "H80", "I80", "J80", "K80", "L80", "M80", "N80", "O80", "P80", "Q80", "R80", "S80", "T80", "U80", "V80", "W80", "X80", "Y80", "Z80", "AA80", "AB80", "AC80", "AD80", "AE80", "AF80", "AG80", "AH80", "AI80", "AJ80", "AK80", "AL80", "AM80", "AN80", "AO80", "AP80", "AQ80", "AR80", "AS80", "AT80", "AU80", "AV80", "AW80", "AX80", "AY80", "AZ80", "BA80", "BB80", "BC80", "BD80", "BE80", "BF80", "BG80", "BH80", "BI80", "BJ80", "BK80", "BL80", "BM80", "BN80", "BO80", "BP80", "BQ80", "BR80", "BS80", "BT80", "BU80", "BV80", "BW80", "BX80", "BY80", "C84", "D84", "E84", "F84", "G84", "H84", "I84", "J84", "K84", "L84", "M84", "N84", "O84", "P84", "Q84", "R84", "S84", "T84", "U84", "V84", "W84", "X84", "Y84", "Z84", "AA84", "AB84", "AC84", "AD84", "AE84", "AF84", "AG84", "AH84", "AI84", "AJ84", "AK84", "AL84", "AM84", "AN84", "AO84", "AP84", "AQ84", "AR84", "AS84", "AT84", "AU84", "AV84", "AW84", "AX84", "AY84", "AZ84", "BA84", "BB84", "BC84", "BD84", "BE84", "BF84", "BG84", "BH84", "BI84", "BJ84", "BK84", "BL84", "BM84", "BN84", "BO84", "BP84", "BQ84", "BR84", "BS84", "BT84", "BU84", "BV84", "BW84", "BX84", "BY84", "C88", "D88", "E88", "F88", "G88", "H88", "I88", "J88", "K88", "L88", "M88", "N88", "O88", "P88", "Q88", "R88", "S88", "T88", "U88", "V88", "W88", "X88", "Y88", "Z88", "AA88", "AB88", "AC88", "AD88", "AE88", "AF88", "AG88", "AH88", "AI88", "AJ88", "AK88", "AL88", "AM88", "AN88", "AO88", "AP88", "AQ88", "AR88", "AS88", "AT88", "AU88", "AV88", "AW88", "AX88", "AY88", "AZ88", "BA88", "BB88", "BC88", "BD88", "BE88", "BF88", "BG88", "BH88", "BI88", "BJ88", "BK88", "BL88", "BM88", "BN88", "BO88", "BP88", "BQ88", "BR88", "BS88", "BT88", "BU88", "BV88", "BW88", "BX88", "BY88", "C92", "D92", "E92", "F92", "G92", "H92", "I92", "J92", "K92", "L92", "M92", "N92", "O92", "P92", "Q92", "R92", "S92", "T92", "U92", "V92", "W92", "X92", "Y92", "Z92", "AA92", "AB92", "AC92", "AD92", "AE92", "AF92", "AG92", "AH92", "AI92", "AJ92", "AK92", "AL92", "AM92", "AN92", "AO92", "AP92", "AQ92", "AR92", "AS92", "AT92", "AU92", "AV92", "AW92", "AX92", "AY92", "AZ92", "BA92", "BB92", "BC92", "BD92", "BE92", "BF92", "BG92", "BH92", "BI92", "BJ92", "BK92", "BL92", "BM92", "BN92", "BO92", "BP92", "BQ92", "BR92", "BS92", "BT92", "BU92", "BV92", "BW92", "BX92", "BY92", "C96", "D96", "E96", "F96", "G96", "H96", "I96", "J96", "K96", "L96", "M96", "N96", "O96", "P96", "Q96", "R96", "S96", "T96", "U96", "V96", "W96", "X96", "Y96", "Z96", "AA96", "AB96", "AC96", "AD96", "AE96", "AF96", "AG96", "AH96", "AI96", "AJ96", "AK96", "AL96", "AM96", "AN96", "AO96", "AP96", "AQ96", "AR96", "AS96", "AT96", "AU96", "AV96", "AW96", "AX96", "AY96", "AZ96", "BA96", "BB96", "BC96", "BD96", "BE96", "BF96", "BG96", "BH96", "BI96", "BJ96", "BK96", "BL96", "BM96", "BN96", "BO96", "BP96", "BQ96", "BR96", "BS96", "BT96", "BU96", "BV96", "BW96", "BX96", "BY96", "C100", "D100", "E100", "F100", "G100", "H100", "I100", "J100", "K100", "L100", "M100", "N100", "O100", "P100", "Q100", "R100", "S100", "T100", "U100", "V100", "W100", "X100", "Y100", "Z100", "AA100", "AB100", "AC100", "AD100", "AE100", "AF100", "AG100", "AH100", "AI100", "AJ100", "AK100", "AL100", "AM100", "AN100", "AO100", "AP100", "AQ100", "AR100", "AS100", "AT100", "AU100", "AV100", "AW100", "AX100", "AY100", "AZ100", "BA100", "BB100", "BC100", "BD100", "BE100", "BF100", "BG100", "BH100", "BI100", "BJ100", "BK100", "BL100", "BM100", "BN100", "BO100", "BP100", "BQ100", "BR100", "BS100", "BT100", "BU100", "BV100", "BW100", "BX100", "BY100", "C104", "D104", "E104", "F104", "G104", "H104", "I104", "J104", "K104", "L104", "M104", "N104", "O104", "P104", "Q104", "R104", "S104", "T104", "U104", "V104", "W104", "X104", "Y104", "Z104", "AA104", "AB104", "AC104", "AD104", "AE104", "AF104", "AG104", "AH104", "AI104", "AJ104", "AK104", "AL104", "AM104", "AN104", "AO104", "AP104", "AQ104", "AR104", "AS104", "AT104", "AU104", "AV104", "AW104", "AX104", "AY104", "AZ104", "BA104", "BB104", "BC104", "BD104", "BE104", "BF104", "BG104", "BH104", "BI104", "BJ104", "BK104", "BL104", "BM104", "BN104", "BO104", "BP104", "BQ104", "BR104", "BS104", "BT104", "BU104", "BV104", "BW104", "BX104", "BY104", "C108", "D108", "E108", "F108", "G108", "H108", "I108", "J108", "K108", "L108", "M108", "N108", "O108", "P108", "Q108", "R108", "S108", "T108", "U108", "V108", "W108", "X108", "Y108", "Z108", "AA108", "AB108", "AC108", "AD108", "AE108", "AF108", "AG108", "AH108", "AI108", "AJ108", "AK108", "AL108", "AM108", "AN108", "AO108", "AP108", "AQ108", "AR108", "AS108", "AT108", "AU108", "AV108", "AW108", "AX108", "AY108", "AZ108", "BA108", "BB108", "BC108", "BD108", "BE108", "BF108", "BG108", "BH108", "BI108", "BJ108", "BK108", "BL108", "BM108", "BN108", "BO108", "BP108", "BQ108", "BR108", "BS108", "BT108", "BU108", "BV108", "BW108", "BX108", "BY108", "C112", "D112", "E112", "F112", "G112", "H112", "I112", "J112", "K112", "L112", "M112", "N112", "O112", "P112", "Q112", "R112", "S112", "T112", "U112", "V112", "W112", "X112", "Y112", "Z112", "AA112", "AB112", "AC112", "AD112", "AE112", "AF112", "AG112", "AH112", "AI112", "AJ112", "AK112", "AL112", "AM112", "AN112", "AO112", "AP112", "AQ112", "AR112", "AS112", "AT112", "AU112", "AV112", "AW112", "AX112", "AY112", "AZ112", "BA112", "BB112", "BC112", "BD112", "BE112", "BF112", "BG112", "BH112", "BI112", "BJ112", "BK112", "BL112", "BM112", "BN112", "BO112", "BP112", "BQ112", "BR112", "BS112", "BT112", "BU112", "BV112", "BW112", "BX112", "BY112", "C116", "D116", "E116", "F116", "G116", "H116", "I116", "J116", "K116", "L116", "M116", "N116", "O116", "P116", "Q116", "R116", "S116", "T116", "U116", "V116", "W116", "X116", "Y116", "Z116", "AA116", "AB116", "AC116", "AD116", "AE116", "AF116", "AG116", "AH116", "AI116", "AJ116", "AK116", "AL116", "AM116", "AN116", "AO116", "AP116", "AQ116", "AR116", "AS116", "AT116", "AU116", "AV116", "AW116", "AX116", "AY116", "AZ116", "BA116", "BB116", "BC116", "BD116", "BE116", "BF116", "BG116", "BH116", "BI116", "BJ116", "BK116", "BL116", "BM116", "BN116", "BO116", "BP116", "BQ116", "BR116", "BS116", "BT116", "BU116", "BV116", "BW116", "BX116", "BY116", "C120", "D120", "E120", "F120", "G120", "H120", "I120", "J120", "K120", "L120", "M120", "N120", "O120", "P120", "Q120", "R120", "S120", "T120", "U120", "V120", "W120", "X120", "Y120", "Z120", "AA120", "AB120", "AC120", "AD120", "AE120", "AF120", "AG120", "AH120", "AI120", "AJ120", "AK120", "AL120", "AM120", "AN120", "AO120", "AP120", "AQ120", "AR120", "AS120", "AT120", "AU120", "AV120", "AW120", "AX120", "AY120", "AZ120", "BA120", "BB120", "BC120", "BD120", "BE120", "BF120", "BG120", "BH120", "BI120", "BJ120", "BK120", "BL120", "BM120", "BN120", "BO120", "BP120", "BQ120", "BR120", "BS120", "BT120", "BU120", "BV120", "BW120", "BX120", "BY120", "C124", "D124", "E1
```

merah yang sebelumnya ditandai. Sebelum data dihapus, sistem juga meminta konfirmasi pengguna untuk memastikan tindakan ini tidak dilakukan secara tidak sengaja.

```

function loadData(showConfirmation = true) {
  // Jika showConfirmation = true, tampilkan pop-up konfirmasi
  if (showConfirmation) {
    const ui = SpreadsheetApp.getUi();
    const response = ui.alert(
      "Konfirmasi",
      "Apakah Anda yakin ingin me-LOAD data dari database need approval?",
      ui.ButtonSet.YES_NO
    );
  }

  if (response == ui.Button.NO) {
    ui.alert("Pemuatan data dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form");

  // Daftar sel yang perlu dikosongkan
  const clearCells = [
    "B4", "C2", "C8", "H8", "C10", "G10", "I10", "C11", "G11", "I11", "C12", "G12", "I12",
    "C15", "H15", "C18", "G18", "I18", "C19", "G19", "C22", "H22", "C25", "G26", "C27", "G9", "J9"
  ];
  clearCells.forEach(cell => formSheet.getRange(cell).setValue(""));

  // Hapus isi tabel barang mulai dari baris 18 ke bawah
  const lastRow = formSheet.getLastRow();
  const startRow = 35;
  if (lastRow >= startRow) {
    formSheet.getRange("A5[startRow]:F5[lastRow]").clearContent();
    formSheet.getRange("H5[startRow]:H5[lastRow]").clearContent();
  }

  function updateApprover1() {
    // Tampilkan pop-up konfirmasi
    const ui = SpreadsheetApp.getUi();
    const response = ui.alert(
      "Konfirmasi",
      "Apakah Anda yakin ingin melakukan APPROVE IT?",
      ui.ButtonSet.YES_NO
    );
  }

  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses APPROVE IT dibatalkan.");
    return;
  }

  const formSheetName = "Form"; // Nama sheet form tempat tombol berada
  const databaseIndex = "0K-Pu8m2dZLahdYp80W8AP8Bc4880t0wng"; // Ganti dengan ID file Google Spreadsheet tempat "DB" berada
  const databaseSheetName = "DB-DB"; // Nama sheet database sementara di file lain
  const nomorSurat = "G4"; // Lokasi No Surat di sheet Form

  const timestamp01 = 3; // Kolom untuk Timestamp Approver1 di Database Sementara (kolom C = 3)
  const email01 = 4; // Kolom untuk email Approver1 di Database Sementara (kolom D = 4)
  const nomorSurat01 = 7; // Kolom untuk No Surat di Database Sementara (kolom G = 7)

  const approverEmail = Session.getActiveUser().getEmail(); // Mendapatkan email pengguna yang menjalankan skrip
  const timestamp = new Date();

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName(formSheetName);

  if (!formSheet) {
    throw new Error("Sheet Form tidak ditemukan");
  }

  const nomorSurat = formSheet.getRange(nomorSuratCell).getValue();
  
```

Gambar 3. 5 Query Java Untuk Button Load dan Approval serta Approval 2

Berdasarkan Gambar 3.5 Kode *JavaScript* untuk *Google Apps Script* dalam *Google Spreadsheet* ini berisi fungsi untuk memuat data berdasarkan nomor surat yang dimasukkan serta melakukan proses persetujuan bertahap. Fungsi pertama memungkinkan pengguna untuk mengambil data dari *database* berdasarkan nomor surat yang dimasukkan di sel G4 pada sheet "*Form*", mengosongkan sel tertentu, menghapus isi tabel barang dari baris 35 ke bawah, lalu mengisi kembali informasi yang ditemukan, seperti status, anggaran, dan detail barang, serta memberikan notifikasi keberhasilan. Sementara itu, fungsi persetujuan terdiri dari dua tahap, yaitu *updateApprover1()* yang mencatat waktu dan email approver pertama saat memperbarui data di *database* sementara, serta *updateApprover2()* yang mencatat informasi approver kedua, memindahkan data ke *database* final, dan mengirimkan notifikasi email kepada penerima yang tertera di sel I10 pada sheet "*Form*". Jika nomor surat tidak ditemukan dalam proses persetujuan, pengguna akan menerima peringatan melalui pop-up.

```

function submitFA() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin melakukan SUBMIT FA?",
    ui.ButtonSet.YES_NO
  );

  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses SUBMIT FA dibatalkan.");
    return;
  }

  // 1. Baca Sheet Form
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form");
  if (!formSheet) {
    ui.alert("Sheet 'Form' tidak ditemukan.");
    return;
  }

  // 2. Tentukan ID dan Nama Sheet dari Database
  const databases = [
    { id: "1U5SD_1JUGx00m9h4qR15945Q_18Scu0YphxzmmPc", sheetName: "DB-BUD" }, // Database Final
    { id: "1OM_Dm0nvx1x1eLx8mjoh89XFAP8PbcN4G0z4Vuog", sheetName: "DB-BUD" } // Database Tambahan
  ];

  // 3. Ambil Data dari Form (C25, C26, C27)
  const newData = [
    formSheet.getRange("C25").getValue(), // Data untuk AH
    formSheet.getRange("C26").getValue(), // Data untuk AI
    formSheet.getRange("C27").getValue(), // Data untuk AJ
  ];
}

function printAndSendPDF() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin PRINT dokumen ini?",
    ui.ButtonSet.YES_NO
  );

  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses PRINT dibatalkan.");
    return;
  }

  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = ss.getSheetByName("Print");
  var totalRows = 0; // Jumlah total baris yang ada di sheet
  var reservedRows = 4; // Jumlah baris paling bawah yang tidak akan disembunyikan
  var range = sheet.getRange(1, 1, totalRows - reservedRows); // Ambil data di kolom B, kecuali 2 baris terakhir
  var columnData = range.getValues(); // Ambil nilai di kolom C
  var lastRowWithDataC = 0;

  // Sembunyi semua baris
  sheet.showRows(1, totalRows);

  // Log untuk debugging
  console.log("Data di kolom C:");
  columnData.forEach(function(row, index) {
    console.log("Baris " + (index + 1) + ": " + row[0]);
  });

  // Cari baris terakhir yang memiliki data di kolom B
  for (var i = columnData.length - 1; i >= 0; i--) {
    if (columnData[i][0] != "") { // Periksa apakah sel di kolom C tidak kosong
      lastRowWithDataC = i + 1;
    }
  }
}

```

Gambar 3.6 Query Java Untuk Button Submit FA dan PrintandSendpdf

Berdasarkan Gambar 3.6. Kode tersebut berisi dua fungsi utama dalam *Google Apps Script*, yaitu *submitFA()* dan *printAndSendPDF()*. Fungsi *submitFA()* digunakan untuk memperbarui data dari sheet "Form" ke dua *database* *Google Sheets* berdasarkan nomor surat, kemudian mengirim email notifikasi beserta bukti transfer. Prosesnya meliputi konfirmasi pengguna, validasi data, pencarian data di *database*, pembaruan data, dan pengiriman email. Sementara itu, fungsi *printAndSendPDF()* digunakan untuk mencetak dokumen dari sheet "Print" dengan menyembunyikan baris kosong, kemudian mengonversinya menjadi PDF, dan mengirimkannya melalui email berdasarkan alamat yang tertera di sheet. Kedua fungsi ini melibatkan interaksi dengan *Google Sheets*, validasi data, serta integrasi dengan *Google Mail* untuk otomatisasi pengiriman email.

### 3.2.3 Developmet KOL Form dengan Function Java Programming Language

*Form of the Work Agreement Letter KOL* adalah sebuah formulir digital yang dikembangkan menggunakan bahasa pemrograman Java untuk memfasilitasi proses pembuatan surat perjanjian kerja antara perusahaan dan Key Opinion Leader (KOL). Formulir ini dirancang untuk memastikan bahwa semua persyaratan kerja, ketentuan pembayaran, serta hak dan kewajiban

kedua belah pihak terdokumentasi dengan jelas dan sistematis. Periode pengerjaan formulir ini berlangsung selama satu minggu, dengan tujuan utama untuk meningkatkan efisiensi administrasi, mengurangi kesalahan input data, serta memastikan transparansi dalam kesepakatan kerja.

*Gambar 3. 7* UI form Surat Perjanjian Kerja Sama KOL

Gambar 3.7 UI Form Surat Perjanjian Kerja Sama KOL menampilkan formulir digital untuk mengelola kerja sama dengan Key Opinion Leader (KOL). Form ini mencakup input field seperti Nama Brand, Department, Status, serta Nama Pembuat Form, Approver 1, dan Approver 2 untuk persetujuan. Detail influencer yang terdokumentasi meliputi No Database Influencer, Pribadi/Perusahaan, Nama Influencer/Nama Brand, Nama Account/Nama Perusahaan, PIC (jika ada), Alamat, No Telp, dan Jenis Kerja Sama. Tersedia tombol fungsional Submit, Clear, Load, Approve 1 & 2, serta Print GL & BL. Dengan UI intuitif, form ini mempermudah administrasi, transparansi, dan efisiensi pengelolaan perjanjian kerja sama KOL. Selain itu, tampilan form ini dirancang user-friendly agar memudahkan semua pihak yang terlibat dalam proses administrasi. Fitur-fitur yang tersedia juga meminimalkan potensi human error dalam pencatatan data kerja sama.

```

function submitData() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin meng-SUBMIT data ke database need approval?",
    ui.ButtonSet.YES_NO
  );

  if (response == ui.Button.NO) {
    ui.alert("Pengiriman data dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("FormPKM");
  const databaseId = "1e3jM80q0b_Hj8E0wRtCt-bw8VEV7zq0kuTE0pJfs";
  const databaseSheetName = "DD-PKM";

  const userEmail = Session.getActiveUser().getEmail();

  const cellMapping = [
    "B4", "C7", "C8", "H8", "C18", "C11", "C12",
    "C15", "C16", "C17", "C18", "C19", "C28", "C21", "C22", "C38", "C34", "C35", "D35", "C36", "C39", "C40", "C41"
  ];

  const data = cellMapping.map(cell => formSheet.getRange(cell).getValue());

  // Menambahkan pemeriksaan kolom yang kosong
  const missingDataColumns = [];

  cellMapping.forEach(function(cell, index) {
    function clearForm(showConfirmation = true) {
      const ui = SpreadsheetApp.getUi();

      // Tampilkan pop-up konfirmasi hanya jika showConfirmation bernilai true
      if (showConfirmation) {
        const response = ui.alert(
          "Konfirmasi",
          "Apakah Anda yakin ingin meng-CLEAR form?",
          ui.ButtonSet.YES_NO
        );

        // Jika pengguna memilih "No", hentikan eksekusi script
        if (response == ui.Button.NO) {
          ui.alert("Pembersihan form dibatalkan.");
          return;
        }
      }

      // 1. Baca Sheet Form
      const ss = SpreadsheetApp.getActiveSpreadsheet();
      const formSheet = ss.getSheetByName("FormPKM"); // Ganti nama sheet sesuai kebutuhan

      // 2. Definisikan Lokasi Sel untuk Data Utama
      const cellMapping = [
        "B4", "B18", "D11", "I11", "I12", "I13", "I5", "I4", "B4", "C7", "C8", "H8", "C18", "C11", "C12",
        "C15", "C16", "C17", "C18", "C19", "C28", "C21", "C22", "C38", "C34", "C35", "D35", "C36", "C39", "C40", "C41"
      ];

      // 3. Kosongkan Sel untuk Data Utama dan ubah border menjadi putih
      cellMapping.forEach(cell => {
        function loadData() {
          const ui = SpreadsheetApp.getUi();
          const response = ui.alert(
            "Konfirmasi",
            "Apakah Anda yakin ingin memuat data berdasarkan Nomor Surat?",
            ui.ButtonSet.YES_NO
          );
        };

        if (response == ui.Button.NO) {
          ui.alert("Proses dibatalkan.");
          return;
        }
      }
    }
  }
}

```

Gambar 3. 8 Query Java Untuk Button Submit, Clear Dan Load

Berdasarkan Gambar 3.8 Fungsi dalam *Google Apps Script* ini bertujuan untuk mengelola data formulir Surat Perjanjian Kerja Sama KOL dalam Google Sheets melalui tiga proses utama: *submit*, *clear*, dan *load*. Fungsi *submitData()* mengirimkan data formulir ke *database* setelah validasi ketat, termasuk pengecekan kolom wajib, konten, dan produk, serta memberikan nomor unik secara otomatis sebelum menyimpan data dan menampilkan notifikasi sukses. Fungsi *clearForm(showConfirmation = true)* menghapus isi formulir pada sheet FormPKM, termasuk data utama, konten, dan barang dengan pengecualian kolom tertentu, serta menampilkan konfirmasi jika diperlukan sebelum memberikan border putih pada sel yang dibersihkan. Fungsi *loadData()* mengambil data dari *database* berdasarkan Nomor Surat, memeriksa keberadaannya, dan mengisikan kembali formulir setelah membersihkan data lama menggunakan *clearForm(false)*, kemudian menampilkan notifikasi keberhasilan pemuatan data. Fungsi-fungsi ini saling terintegrasi untuk menjaga akurasi, kelengkapan, dan konsistensi data dalam pengelolaan administrasi kerja sama KOL. Dengan penerapan otomatisasi ini, proses administrasi menjadi lebih efisien dan meminimalkan risiko kesalahan input manual.

```

function updateApprover1() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin melakukan APPROVE 1?",
    ui.ButtonSet.YES_NO
  );
  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses APPROVE 1 dibatalkan.");
    return;
  }
  const formSheetName = "FormPKM"; // Nama sheet form tempat tombol berada
  const databaseId = "1oYMNqRb_JG6DNvXcT-3MVEVYzqgkUEdZpJfa"; // ID file database sementara
  const databaseSheetName = "DD-PKM"; // Nama sheet database sementara
  const finalDatabaseId = "110d8E6thgEzCvE_VYFhazd6e-weo4d0V43M"; // Ganti dengan ID file database final
  const finalDatabaseSheetName = "DB-PKM"; // Nama sheet di database final
  const noSuratCol = "G4"; // Lokasi No Surat di sheet Form
  const notifikasiEmailCol = "I10"; // Lokasi email notifikasi di Form

  const timestampCol = 3; // Kolom untuk timestamp Approver1 di Database Sementara (kolom C = 3)
  const emailCol = 4; // Kolom untuk email Approver1 di Database Sementara (kolom D = 4)
  const noSuratCol = 7; // Kolom untuk No Surat di Database Sementara (kolom G = 7)

  const approverEmail = Session.getActiveUser().getEmail(); // Mendapatkan email pengguna yang menjalankan skrip
  const timestamp = new Date();

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName(formSheetName);

  function printGL() {
    // Tampilkan pop-up konfirmasi
    const ui = SpreadsheetApp.getUi();
    const response = ui.alert(
      "Konfirmasi",
      "Apakah Anda yakin ingin PRINT GL dokumen ini?",
      ui.ButtonSet.YES_NO
    );
    // Jika pengguna memilih "No", hentikan eksekusi script
    if (response == ui.Button.NO) {
      ui.alert("Proses PRINT dibatalkan.");
      return;
    }
  }

  function printBL() {
    // Tampilkan pop-up konfirmasi
    const ui = SpreadsheetApp.getUi();
    const response = ui.alert(
      "Konfirmasi",
      "Apakah Anda yakin ingin PRINT BL dokumen ini?",
      ui.ButtonSet.YES_NO
    );
    // Jika pengguna memilih "No", hentikan eksekusi script
    if (response == ui.Button.NO) {
      ui.alert("Proses PRINT dibatalkan.");
      return;
    }
  }
}

```

Gambar 3. 9 Query Java Function Approval1 Approval2 Print GL dan Print BL

Berdasarkan Gambar 3.9 Fungsi *updateApprover1()* digunakan untuk melakukan persetujuan pertama terhadap data yang terdapat di "Database Need Approval" dengan menampilkan pop-up konfirmasi kepada pengguna. Jika pengguna memilih "No", proses akan dibatalkan. Skrip kemudian mengambil No Surat dari sel G4 di sheet FormPKM, membuka file *database* sementara (DD-PKM), mencari baris yang sesuai dengan No Surat, lalu memperbarui timestamp dan email *Approver 1* pada kolom yang sesuai. Setelah itu, sistem akan menampilkan notifikasi apakah pembaruan berhasil atau tidak. Sementara itu, *updateApprover2()* bertujuan untuk melakukan persetujuan kedua dan memindahkan data yang telah disetujui ke *database* final (DB-PKM). Setelah menampilkan pop-up konfirmasi, jika pengguna memilih untuk melanjutkan, skrip akan mengambil No Surat dari sel G4, serta email notifikasi dari sel I10.

Sistem membuka file *database* sementara (DD-PKM) dan *database* final (DB-PKM), mencari baris yang sesuai dengan No Surat, memperbarui timestamp serta email *Approver 2* di *database* sementara, lalu menyalin seluruh data ke *database* final. Setelah proses selesai, email notifikasi dikirim

ke alamat yang terdaftar di I10, dan sistem menampilkan notifikasi keberhasilan atau kegagalan proses. Selain itu, dalam proses pencetakan dan pengiriman dokumen BL dalam format PDF, skrip akan menampilkan pop-up konfirmasi sebelum melanjutkan. Jika pengguna menyetujui, skrip memeriksa keberadaan sheet "PRI-PKMBL", serta memvalidasi data seperti email di I10 dan nomor surat di I11 agar tidak kosong. Proses konversi sheet ke PDF menggunakan URL ekspor Google Sheets dengan parameter seperti ukuran A4, mode portrait, serta tanpa gridlines, dan menggunakan OAuth Token untuk autentikasi. Jika konversi berhasil, dokumen PDF akan dikirim ke email yang terdaftar menggunakan MailApp.sendEmail(), dengan subjek yang mencantumkan No Surat, kemudian sistem akan memberikan notifikasi kepada pengguna mengenai status pengiriman dokumen.

#### ***3.2.4 Developmet Master Product/Stock SKU Form dengan Function Java Programming Language***

Product/Stock SKU Form adalah formulir yang dirancang untuk mendokumentasikan dan mengelola informasi terkait produk atau stok berdasarkan Stock Keeping Unit (SKU). Formulir ini berfungsi sebagai referensi utama dalam sistem manajemen inventaris, memungkinkan perusahaan untuk melacak ketersediaan, kategori, harga, dan spesifikasi produk dengan lebih akurat. Dalam pengembangannya, formulir ini dibuat menggunakan Java Programming Language, yang memungkinkan integrasi dengan *database* serta otomatisasi proses validasi data Untuk meningkatkan efisiensi operasional. Periode pengerjaan formulir ini berlangsung selama satu minggu, dengan tujuan utama untuk memastikan struktur data yang terorganisir, meningkatkan kecepatan pencarian informasi produk, serta mengurangi risiko kesalahan pencatatan dalam manajemen stok. Tampilan form ini disusun secara sistematis agar mempermudah pengguna dalam memahami alur pengisian data.

FORM - Master SKU Sales - Product				
		Rincian SKU Sales		
Brand :		No	SKU - Nama Product	Name Product
		1		
		2		
		3		
		4		
		5		
		6		
		7		
		12		
		13		
		14		
		15		
		16		
		17		
		18		
		19		
		20		
		21		
		22		
		23		
		24		
		25		
		26		
		27		

Gambar 3. 10 UI Form Master SKU Sales Product

Berdasarkan Gambar 3.10 UI Form Master SKU Sales - Product dirancang sebagai antarmuka pengguna (UI) yang memudahkan pengelolaan dan pencatatan data produk dalam sistem penjualan. Formulir ini terdiri dari beberapa bidang input utama, seperti Brand, yang mencatat merek produk; Tipe Produk, untuk menentukan jenis barang yang dijual SKU Varian Marketplace, yang berisi kode unik SKU sesuai dengan platform marketplace serta Nama Produk Marketplace, yang mencantumkan nama resmi produk yang ditampilkan di marketplace. Selain itu, terdapat Category dan Sub-Category untuk mengelompokkan produk berdasarkan jenisnya, sementara Color, Size, dan Material digunakan untuk mendetailkan atribut fisik produk. UI ini juga dilengkapi dengan dua tombol utama, yaitu Button Submit, yang berfungsi untuk menyimpan dan mengirim data produk ke *database* sistem, serta Button Clear, yang memungkinkan pengguna menghapus semua input yang telah dimasukkan untuk mengisi ulang formulir dengan data baru. Tampilan form ini dibuat sederhana dan terstruktur untuk memudahkan proses input data oleh pengguna dari berbagai divisi. Penggunaan form ini juga mendukung keteraturan database produk agar mempermudah proses monitoring dan evaluasi penjualan. Dengan form ini, perusahaan dapat memastikan seluruh informasi produk tercatat secara akurat, rapi, dan terintegrasi.

```

function ClearDataSKUsales() {
  var ui = SpreadsheetApp.getUi();
  var sheetForm = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("FORM - Master SKU Sales");
  var response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin menghapus data dari form?", ui.ButtonSet.YES_NO);

  if (response == ui.Button.YES) {
    // List of cells to be cleared
    var cellsToClear = [
      "06", "08", "014", "016", "018", "020", "022", "018", "012",
      "07:052", "J7:J92" // Vertical input fields
    ];

    // Clear the cells
    cellsToClear.forEach(function(cell) {
      sheetForm.getRange(cell).setValue(""); // Empty the form fields
    });

    // Reset border colors back to grey for the cleared cells
    cellsToClear.forEach(function(cell) {
      var range = sheetForm.getRange(cell);
      range.setBorder(true, true, true, true, true, "white", SpreadsheetApp.BorderStyle.SOLID); // Reset borders to grey
    });

    // Reset border colors for SKU Nama Produk column (G7:G)
    var maxRows = sheetForm.getLastRow(); // Determine the last row with data
    sheetForm.getRange("G7:G" + maxRows).setBorder(true, true, true, true, true, "grey", SpreadsheetApp.BorderStyle.SOLID);

    // Reset border colors for SKU Varian Marketplace column (J7:J)
    sheetForm.getRange("J7:J" + maxRows).setBorder(true, true, true, true, true, "grey", SpreadsheetApp.BorderStyle.SOLID);

    ui.alert("Sukses!", "Formulir telah berhasil dihapus.", ui.ButtonSet.OK);
  } else {

```

```

function SubmitDataSKUsales() {
  var ui = SpreadsheetApp.getUi();
  var sheetForm = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("FORM - Master SKU Sales");
  var sheetDatabase = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Master SKU Sales - Product");

  var data = [
    sheetForm.getRange("06").getValue(),
    sheetForm.getRange("08").getValue(),
    sheetForm.getRange("014").getValue(),
    sheetForm.getRange("016").getValue(),
    sheetForm.getRange("018").getValue(),
    sheetForm.getRange("020").getValue(),
    sheetForm.getRange("022").getValue(),
    sheetForm.getRange("018").getValue(),
    sheetForm.getRange("012").getValue()
  ];

  var missingDataColumns = [];
  var dataRanges = ["06", "08", "014", "016", "018", "020", "018", "012"];
  dataRanges.forEach(function(range, index) {
    if (sheetForm.getRange(range).getValue() === "") {
      missingDataColumns.push(range);
    }
  });

  if (missingDataColumns.length > 0) {
    missingDataColumns.forEach(function(range) {
      var cell = sheetForm.getRange(range);
      cell.setBorder(true, true, true, true, true, "red", SpreadsheetApp.BorderStyle.SOLID);
    });
    ui.alert("Peringatan", "Harap lengkapi semua data sebelum menginput.", ui.ButtonSet.OK);
    return;
  }
}

```

Gambar 3. 11 Code Java Function Submit Dan Clear

Kode *JavaScript* tersebut berfungsi untuk mengelola data pada formulir Master SKU Sales - Product di Google Sheets dengan dua fungsi utama: *SubmitDataSKUsales()* dan *ClearDataSKUsales()*. Fungsi *SubmitDataSKUsales()* digunakan untuk menginput data dari sheet formulir ke *database*, dimulai dengan validasi kelengkapan data, menampilkan pop-up konfirmasi, mencari baris kosong berikutnya dalam *database*, serta menyimpan data SKU yang telah diisi. Jika ada kolom yang kosong, sistem akan menyorotnya dengan border merah sebagai peringatan kepada pengguna. Setelah data berhasil ditambahkan, formulir akan dikosongkan secara otomatis. Sementara itu, fungsi *ClearDataSKUsales()* memungkinkan pengguna untuk menghapus semua data dalam formulir setelah mendapatkan konfirmasi melalui pop-up. Fungsi ini juga mengembalikan warna border ke tampilan awal untuk menjaga keteraturan tampilan formulir.

### 3.2.5 Developmet Business Trip Form dengan Function Java Programming Language

Business trip form adalah formulir yang digunakan untuk mengelola dan mencatat perjalanan dinas karyawan dalam PT Baller Indonesia Sentosa. Formulir ini berfungsi sebagai alat untuk mengajukan, menyetujui, dan mendokumentasikan perjalanan bisnis, termasuk informasi seperti tujuan perjalanan, durasi, biaya yang dikeluarkan, serta persetujuan dari pihak terkait. Dalam pengembangannya, formulir ini dibuat menggunakan Java Programming Language, yang memungkinkan integrasi dengan *database* serta

otomatisasi dalam proses persetujuan dan pelaporan. Periode pengerjaan formulir ini berlangsung selama dua minggu, dengan tujuan utama untuk meningkatkan efisiensi pengelolaan perjalanan dinas, memastikan transparansi pengeluaran, serta mempercepat proses persetujuan dan pelaporan kepada manajemen.

Gambar 3. 12 UI Form Business Trip

Berdasarkan Gambar 3.12 UI Form Business Trip dirancang untuk memudahkan pengajuan dan persetujuan perjalanan dinas dalam perusahaan. Form ini terdiri dari beberapa bagian utama, dimulai dengan informasi identitas, termasuk Nama Brand, Department, dan Status, yang menunjukkan posisi serta status pengajuan perjalanan. Selanjutnya, terdapat Nama Pembuat Form, Nama Approver 1, dan Nama Approver 2, yang berfungsi untuk mencatat pihak yang mengajukan dan menyetujui perjalanan dinas. Pada bagian inti, terdapat Tipe Perjalanan, yang memungkinkan pengguna memilih jenis perjalanan yang akan dilakukan. Form ini juga mencakup kolom Dari Tanggal (Jam) dan Sampai Tanggal (Jam) untuk mencatat durasi perjalanan secara akurat. Selain itu, kolom Keperluan disediakan untuk menjelaskan alasan perjalanan dinas. Terakhir, terdapat kolom Berangkat Dari dan Menuju Ke, yang digunakan untuk mencatat lokasi keberangkatan dan tujuan perjalanan. Dalam Form Terdapat Button Submit, Clear, Load, Approval 1 dan Approval 2.



jika berhasil, nomor surat akan disimpan di sel I5, dan pengguna akan menerima notifikasi sukses.

```

function updateApprover1() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin melakukan APPROVE 1?",
    ui.ButtonSet.YES_NO
  );

  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses APPROVE 1 dibatalkan.");
    return;
  }

  const formSheetName = "FormDIN"; // Nama sheet form tempat tombol berada
  const databaseFileId = "1iz2nEeygrRULS0z1QDxaATM4nSqX8PLHfncKcTFlok"; // Ganti dengan ID file Google
  const databaseSheetName = "DB-DIN"; // Nama sheet database sementara di file lain
  const noSuratCell = "G4"; // Lokasi No Surat di sheet Form

  const timestampCol = 3; // Kolom untuk timestamp Approver1 di Database Sementara (kolom C = 3)
  const emailCol = 4; // Kolom untuk email Approver1 di Database Sementara (kolom D = 4)
  const noSuratCol = 7; // Kolom untuk No Surat di Database Sementara (kolom G = 7)

  function printGL() {
    const ui = SpreadsheetApp.getUi();
    const response = ui.alert(
      "Konfirmasi",
      "Apakah Anda yakin ingin mengirimkan notifikasi pengajuan perjalanan dinas ini ke HR?",
      ui.ButtonSet.YES_NO
    );

    if (response == ui.Button.NO) {
      ui.alert("Proses pengiriman notifikasi dibatalkan.");
      return;
    }

    var ss = SpreadsheetApp.getActiveSpreadsheet();
    var formSheet = ss.getSheetByName("FormDIN");
    if (formSheet) {
      SpreadsheetApp.getUi().alert("Sheet 'FormDIN' tidak ditemukan.");
    }
  }
}

function updateApprover2() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin melakukan APPROVE 2?",
    ui.ButtonSet.YES_NO
  );

  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses APPROVE 2 dibatalkan.");
    return;
  }

  const formSheetName = "FormDIN"; // Nama sheet form tempat tombol berada
  const databaseFileId = "1iz2nEeygrRULS0z1QDxaATM4nSqX8PLHfncKcTFlok"; // ID file database sementara
  const databaseSheetName = "DB-DIN"; // Nama sheet database sementara
  const finalDatabaseFileId = "1bP29NPSakSQBntvTVmCA1fPVE2XScVrWw4EjMCKFOAs"; // Ganti dengan ID file database final
  const finalDatabaseSheetName = "DB-DIN"; // Nama sheet di database final
  const noSuratCell = "G4"; // Lokasi No Surat di sheet Form
  const notificationEmailCell = "I10"; // Lokasi email notifikasi di Form
}

```

Gambar 3. 14 Function Approve 1, Approve 2, Print Java Script

Berdasarkan Gambar 3.14 Script ini berfungsi untuk mengelola proses persetujuan (approval) dalam pengajuan perjalanan dinas menggunakan *Google Apps Script* pada *Google Spreadsheet*. Fungsi *updateApprover1* menampilkan konfirmasi sebelum memperbarui data persetujuan pertama (Approver 1) pada *database* sementara berdasarkan No Surat yang diinput di sheet FormDIN, mencatat email pengguna sebagai *Approver 1*, dan mengirimkan notifikasi jika proses berhasil. Setelah itu, fungsi *printGL* akan meminta konfirmasi sebelum mengirimkan email pemberitahuan kepada HR terkait pengajuan yang telah disetujui. Selanjutnya, fungsi *updateApprover2* menangani persetujuan kedua (Approver 2) dengan mekanisme serupa, tetapi setelah persetujuan diberikan, data dari *database* sementara akan dipindahkan ke *database* final, memastikan bahwa pengajuan yang telah mendapat persetujuan penuh disimpan dalam sistem utama. Setiap proses dilengkapi dengan validasi, konfirmasi, serta pemberitahuan kepada pihak terkait untuk memastikan alur kerja yang transparan dan terdokumentasi dengan baik.

### 3.2.6 Developmet Realization Business Trip Form dengan Function Java Programming Language

Realisasi Business Trip adalah formulir yang digunakan untuk mencatat dan mengelola pengajuan serta pelaksanaan perjalanan dinas. Formulir ini berfungsi sebagai alat verifikasi untuk memastikan bahwa perjalanan dinas telah direalisasikan sesuai dengan ketentuan yang berlaku. Jika business trip dinilai telah terealisasi dan berjalan sesuai rencana, maka statusnya akan diubah menjadi "terealisasi". Dalam pengembangannya, formulir ini dibuat menggunakan Java Programming Language, yang memungkinkan integrasi dengan *database* serta otomatisasi dalam proses persetujuan dan pelaporan. Dengan adanya otomatisasi ini, proses administrasi menjadi lebih efisien, meminimalkan kesalahan manual, dan mempercepat alur kerja. Periode pengerjaan formulir ini berlangsung selama satu minggu, dengan tujuan utama untuk meningkatkan efisiensi dalam pengelolaan perjalanan dinas.

FORM REALISASI					
Tanggal			No. Surat		
<b>KETERANGAN FORM</b>					
Nama Brand			Department	Sub-Department	
Status					
Nama Pembuat Form		Timestamp	E-Sign by		
Nama Approver 1		Timestamp	E-Sign by		
Nama Approver 2	Marlina	Timestamp	E-Sign by		
<b>DETAIL REALISASI</b>					
Status		Peserta Perjalanan			1
Tipe Perjalanan					2
Dari Tanggal (Jam)	00:00:00				1
Sampai Tanggal (Jam)	00:00:00				2
Keperluan					1
					2

Gambar 3. 15 UI Realisasi Business Trip

Berdasarkan Gambar 3.15 UI Realisasi Business Trip menampilkan antarmuka untuk mencatat status perjalanan dinas yang telah terealisasi. Dalam sistem ini, status perjalanan akan ditandai sebagai "Terealisasi" setelah perjalanan benar-benar dilakukan. Terdapat beberapa informasi utama yang dicatat, seperti tipe perjalanan, tanggal dan waktu keberangkatan serta

keperluan, keperluan perjalanan, lokasi keberangkatan, dan tujuan akhir. Dengan adanya UI ini, perusahaan dapat dengan mudah memantau dan mengelola data perjalanan dinas secara lebih sistematis, memastikan transparansi, serta mendukung proses pelaporan yang lebih efisien.

```

function loadDataRL() {
  const ui = SpreadsheetApp.getUi();
  const formSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("FormRDI");

  // 1. Menampilkan Konfirmasi Sebelum Memuat Data
  const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin memuat data? Terkirim Request", ui.ButtonSet.YES_NO);
  if (response == ui.Button.NO) {
    ui.alert("Pemuatan data dibatalkan.");
    return;
  }

  // 2. Menghapus Spreadsheet dan Memasukkan Data Lama
  const clearCells = ["B4", "C7", "C8", "H8", "C10", "C11", "C15", "C16", "C17", "G10", "G11", "G12", "I10", "I11", "I12", "C18", "C19", "C20", "C21", "C9", "G4", "I5", "C66", "F66", "C67", "F67", "C68", "F68", "G4"];
  clearCells.forEach(cell => formSheet.getRange(cell).setValue(""));

  formSheet.getRange("D17").setValue("00:00:00");
  formSheet.getRange("D18").setValue("00:00:00");

  // Menusai sel tabel
  formSheet.getRange("H15:H24").clearContent();
  formSheet.getRange("A28:I37").clearContent();
  formSheet.getRange("A41:G58").clearContent();
  formSheet.getRange("I41:I58").clearContent();

  // 3. Mengambil Data dari Database Google Spreadsheet
  const databaseId = "1aWpKkCnZtV8dQpRt3kqWkqf3t37";
  const databaseFile = SpreadsheetApp.openById(databaseId);
  const databaseSheet = databaseFile.getSheetByName("DB");

  // Cek apakah nomor surat di G4 ada
  const searchKey = formSheet.getRange("G4").getValue();
  if (!searchKey) {
    ui.alert("Mohon masukkan no surat di sel G4.");
    return;
  }

  function submitDataRL() {
    const ui = SpreadsheetApp.getUi();
    const response = ui.alert("Konfirmasi", "Apakah anda yakin ingin meng-submit data ke database need approval?", ui.ButtonSet.YES_NO);
    if (response == ui.Button.NO) {
      ui.alert("Pengisian data dibatalkan.");
      return;
    }

    const ss = SpreadsheetApp.getActiveSpreadsheet();
    const formSheet = ss.getSheetByName("FormRDI");

    const cellMapping = ["B4", "C7", "C8", "H8", "C10", "C11", "C15", "C16", "C17", "G10", "G11", "G12", "I10", "I11", "I12", "C18", "C19", "C20", "C21", "C9", "G4", "I5", "C66", "F66", "C67", "F67", "C68", "F68", "G4"];
    let data = cellMapping.map(cell => {
      let value = formSheet.getRange(cell).getValue();
      if (value) {
        complete = false;
        range.push({range: true, true, true, true, true, "RDI", SpreadsheetApp.BordersStyle.DL3D});
      }
    });
    return value;
  }

  if (!complete) {
    ui.alert("Mohon mengisi seluruh data sebelum mengirimkan formulir.");
    return;
  }
}

```

```

function clearFormRL(showConfirmation = true) {
  const ui = SpreadsheetApp.getUi();

  // Jika showConfirmation true, tampilkan konfirmasi sebelum menghapus data
  if (showConfirmation) {
    const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin Menghapus semua data dalam form?", ui.ButtonSet.YES_NO);
    if (response == ui.Button.NO) {
      ui.alert("Penghapusan data dibatalkan.");
      return;
    }
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("FormRDI");

  const cellMapping = ["B4", "C7", "C8", "H8", "C10", "C11", "C15", "C16", "C17", "G10", "G11", "G12", "I10", "I11", "I12", "C18", "C19", "C20", "C21", "C9", "G4", "I5", "C66", "F66", "C67", "F67", "C68", "F68", "G4"];
  const pesertaRange = "H15:H24";
  const akomodasiRange = "A28:I37";
  const realisasiBiaya = "A41:G58";
  const addressBiaya = "I41:I58";

  // Menghapus nilai dari setiap sel dalam cellMapping, kecuali C12
  cellMapping.forEach(cell => {
    formSheet.getRange(cell).setValue("");
  });

  // Mengatur nilai menjadi "00:00:00"
  formSheet.getRange("D17").setValue("00:00:00");
  formSheet.getRange("D18").setValue("00:00:00");

  // Menghapus data peserta
  formSheet.getRange(pesertaRange).clearContent();

  // Menghapus data akomodasi
  formSheet.getRange(akomodasiRange).clearContent();
}

```

Gambar 3. 16 Function Load Data, Submit Data dan Clear

Berdasarkan Gambar 3.16 Fungsi *loadDataRL()*, *submitDataRL()*, dan *clearFormRL()* dalam *Google Apps Script* digunakan untuk mengelola data realisasi perjalanan dinas pada *spreadsheet "FormRDI"* dengan memastikan efisiensi, validasi, dan keakuratan data. *loadDataRL()* bertugas memuat data dari *database* berdasarkan nomor surat yang diinputkan di sel G4, menghapus data lama untuk memastikan formulir bersih sebelum pengisian ulang, dan menampilkan notifikasi jika nomor surat tidak ditemukan. Setelah menemukan data yang sesuai, skrip akan mengisi status perjalanan serta data peserta secara dinamis ke dalam form. Sementara itu, *submitDataRL()* berfungsi untuk

mengirim data dari formulir ke *database* dengan validasi ketat, termasuk pengecekan kelengkapan sel yang ditandai dengan batas merah jika kosong, memastikan minimal satu entri peserta dan biaya tersedia sebelum pengiriman, serta melakukan verifikasi tambahan jika perjalanan merupakan shooting. Setelah data dikompilasi dan disimpan di baris kosong pertama yang tersedia dalam *database*, skrip menghapus batas merah dan menampilkan notifikasi konfirmasi jika pengiriman berhasil. Terakhir, *clearFormRL()* digunakan untuk menghapus data formulir setelah meminta konfirmasi pengguna (jika diaktifkan), dengan mengosongkan sel yang telah ditentukan, membersihkan daftar peserta, akomodasi, dan realisasi biaya, serta mengatur ulang border menjadi putih. Jika pengguna membatalkan konfirmasi, proses penghapusan tidak dilakukan, memastikan bahwa data tidak terhapus secara tidak sengaja.

The image contains three screenshots of JavaScript code. The top-left screenshot shows the `loadDataPPROVE(showConfirmation = true)` function, which handles data loading and confirmation. The top-right screenshot shows the `updateApprover1REAL()` function, which manages the first approval step. The bottom screenshot shows the `updateApprover2RL()` function, which manages the second approval step. All code includes comments in Indonesian and uses Google Apps Script APIs for spreadsheet and database operations.

Gambar 3. 17 Code JavaScript Approval 1, Approval 2, load Data

Berdasarkan Gambar 3.17 Fungsi *updateApprover1REAL()* dan *updateApprover2RL()* dalam *Google Apps Script* digunakan untuk mengelola

proses persetujuan (approval) pada dokumen di Google *Spreadsheet* yang berkaitan dengan sistem persetujuan RDI. *updateApprover1REAL()* menampilkan konfirmasi sebelum melanjutkan persetujuan pertama, memverifikasi keberadaan No Surat pada sheet form, serta memperbarui timestamp dan email approver pertama dalam *database* sementara. Jika No Surat ditemukan, proses approval dicatat dan diproses lebih lanjut. Sementara itu, *updateApprover2RL()* menangani persetujuan kedua dengan langkah serupa, tetapi juga memindahkan data yang telah disetujui dari *database* sementara ke *database* final serta mengirimkan email notifikasi ke pihak terkait. Selain itu, fungsi *loadDataAPPROVE* digunakan untuk mengambil dan mengisi data persetujuan dari Google *Spreadsheet* ke dalam lembar kerja "*FormRDI*" berdasarkan No Surat di sel G4. Jika ditemukan dalam *database* "*DD-RDI*", data seperti status, informasi peserta, rincian realisasi perjalanan dinas, dan detail biaya akan diisi ke dalam form, termasuk lampiran yang diperlukan. Jika No Surat tidak ditemukan, fungsi ini akan menampilkan pesan kesalahan. Secara keseluruhan, ketiga fungsi ini memastikan bahwa setiap tahap persetujuan terdokumentasi dengan baik, data tersinkronisasi dengan benar, dan pihak terkait menerima notifikasi yang sesuai.

### ***3.2.7 Developmet Invoice Professional organizing Form dengan Function Java Programming Language***

Invoice Professional Organizing Form adalah formulir yang digunakan untuk mengelola dan mengorganisir data terkait pembuatan serta persetujuan invoice secara profesional. Dalam pengembangannya, formulir ini dibuat menggunakan Java Programming Language, yang memungkinkan integrasi dengan *database* serta otomatisasi dalam proses persetujuan dan pelaporan. Dengan adanya otomatisasi ini, proses pencatatan, validasi, dan pelacakan invoice menjadi lebih efisien, dengan tujuan utama meningkatkan akurasi data, mempercepat alur kerja, serta meminimalkan kesalahan dalam pengelolaan keuangan.

Gambar 3. 18 UI Invoice Professional organizing Form

Berdasarkan Gambar 3.18 UI Invoice Professional Organizing Form menampilkan antarmuka pengguna dari formulir yang dirancang untuk mengelola proses invoice secara profesional. Formulir ini mencakup beberapa bidang input utama, seperti Nama Brand, Department, dan Status, yang digunakan untuk mengidentifikasi informasi perusahaan dan status transaksi. Selain itu, terdapat kolom Nama Pembuat Form, Nama Approver 1, dan Nama Approver 2, yang berfungsi untuk mencatat pihak yang bertanggung jawab dalam pembuatan dan persetujuan invoice. Formulir ini juga menyediakan kolom Jenis Transaksi, Invoice Date, dan Customer ID untuk mendokumentasikan jenis transaksi serta data pelanggan. Untuk aspek keuangan, terdapat Notes, Paid Value, Paid Date, dan Payment Notes, yang memungkinkan pencatatan detail pembayaran Untuk memastikan transparansi serta kemudahan dalam pelacakan dan verifikasi transaksi.

```
function submitData() {
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin meng-SUBMIT");
  if (response == ui.Button.NO) {
    ui.alert("Pengiriman data dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form-Invoice");
  const databaseId = "1ZH0ZWjsYtx91K5cIe5_lV6Pm9PSnAEVJU0J8ZffH256g";
  const databaseSheetName = "DD-Invoice";
  const userEmail = Session.getActiveUser().getEmail();

  const classFormSheetConfFormInvoice = true;
  const ui = SpreadsheetApp.getUi();

  // Jika sheetConfFormInvoice true, tampilan konfirmasi sebelum menginput data
  if (sheetConfFormInvoice) {
    const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin MEMASUKI semua data dalam formulir?", ui.Button.YES_NO);
    if (response == ui.Button.NO) {
      return;
    }
  }

  const ui = SpreadsheetApp.getActiveSpreadsheet();
  const spreadsheet = ui.getActiveSpreadsheet();
  const uiMapping = ["03B", "03T", "03D", "03M", "03Y", "04", "04", "04", "04", "04", "04", "04", "04", "04", "04", "04", "04", "04"];
  const uiMapping = ["03B", "03T", "03D", "03M", "03Y", "04", "04", "04", "04", "04", "04", "04", "04", "04", "04", "04"];

  // Menginput nilai ke sel yang dipecahkan
  uiMapping.forEach(uiMapping => {
    formSheet.getRange(uiMapping).setValue("");
  });

  const uiMapping = ["03B", "03T", "03D", "03M", "03Y", "04", "04", "04", "04", "04", "04", "04", "04", "04", "04"];
  uiMapping.forEach(uiMapping => {
    formSheet.getRange(uiMapping).setValue("");
  });

  // Menginput nilai berdasarkan pilihan di UI
  const uiMapping = formSheet.getRange("C3:C3").getValues();
}
```

```

function loadData(showConfirmation = true) {
  const ui = SpreadsheetApp.getUi();
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form-Invoice");

  // 1. Menampilkan Konfirmasi Sebelum Memuat Data (jika diperlukan)
  if (showConfirmation) {
    const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin memuat data?", ui.ButtonSet.YES_NO);
    if (response == ui.Button.NO) {
      ui.alert("Pemutaran data dibatalkan.");
      return;
    }
  }
}

```

Gambar 3. 19 *Function Submit, Clear, Load Data*

Berdasarkan Gambar 3.19 *Script Google Apps* ini berfungsi untuk mengelola data pada formulir invoice di *Google Spreadsheet* dengan tiga fungsi utama: mengirim data, menghapus data formulir, dan memuat data dari *database*. Fungsi pertama memungkinkan pengguna mengirim data dari formulir ke *spreadsheet database* setelah memastikan semua kolom wajib terisi, dengan validasi tambahan untuk berbagai jenis transaksi seperti *Commitment Fee, Down Payment, Settlement, Event Guest Speaker*, dan *Product Purchase after POGL*. Jika terdapat kolom kosong, sistem menandai sel dengan garis merah sebagai peringatan dan meminta pengguna untuk melengkapinya sebelum pengiriman. Setelah semua data terverifikasi, nomor unik otomatis diberikan, data disimpan di *database*, formulir dikosongkan, dan pengguna menerima notifikasi keberhasilan pengiriman. Fungsi kedua, *clearForm*, digunakan untuk menghapus data formulir dengan opsi konfirmasi sebelum eksekusi, mengosongkan sejumlah sel yang telah dipetakan, serta mengatur ulang format berdasarkan kategori transaksi yang dipilih, dengan beberapa sel diisi dengan tanda "-" atau dikosongkan sesuai kondisi tertentu.

Fungsi ketiga, *loadData*, memungkinkan pengguna mengambil data dari *database* dan mengisinya kembali ke dalam formulir berdasarkan nomor surat yang dimasukkan di sel G4, dengan konfirmasi sebelum pemuatan. Data yang diambil dari sheet "DD-Invoice" dalam *spreadsheet* lain akan dimasukkan ke dalam sel yang telah ditentukan, dan jika kategori transaksi tertentu dipilih, data tambahan yang sesuai juga akan dimuat. Setelah proses selesai, nomor draft surat disimpan di sel I5, dan pengguna diberi notifikasi bahwa data berhasil dimuat ke dalam formulir.

```

function submitPayment() {
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin meng-SUBMIT pembayaran ke database?", ui.ButtonSet.YES_NO);
  if (response == ui.Button.NO) {
    ui.alert("Pengiriman pembayaran dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form-Invoice");

  const databaseId1 = "1ZH0ZwjsYtx91KScIe5_lv6Ph9PSnAEVJ0J8ZffH256g"; // Database Utama
  const databaseSheetName1 = "DD-Invoice";

  const databaseId2 = "1is-W898My6XbE1qA3mhNoYpDG7AxvBQAAPyh390hcSU"; // Database Tambahan
  const databaseSheetName2 = "DB-Invoice";

  const paymentCells = ["C21", "C22", "C23"];
  let paymentData = paymentCells.map(cell => formSheet.getRange(cell).getValue());

  // ... (code continues) ...
}

function updateApprover1() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin melakukan APPROVE 1?",
    ui.ButtonSet.YES_NO
  );

  if (response == ui.Button.NO) {
    ui.alert("Proses APPROVE 1 dibatalkan.");
    return;
  }
}

```

Gambar 3. 20 Function Submit FA dan Function Approval

Berdasarkan Gambar 3.20 Fungsi `submitPayment` dalam *Google Apps Script* digunakan untuk mengirim data pembayaran dari formulir di *Google Spreadsheet* ke dua *database* setelah memastikan semua kolom wajib telah diisi. Sistem menampilkan kotak dialog konfirmasi sebelum memproses data dan memeriksa kelengkapan sel pembayaran (C21, C22, dan C23) serta nomor surat di G4. Jika ada data yang kosong, sistem memberikan peringatan dan menandai sel yang bersangkutan dengan garis merah. Jika nomor surat ditemukan, fungsi `updateDatabase` memperbarui data pembayaran pada *database* utama ("*DD-Invoice*") dan *database* tambahan ("*DB-Invoice*"). Setelah pembaruan selesai, batas merah dihapus, notifikasi sukses ditampilkan, dan formulir dikosongkan menggunakan `clearForm`. Sementara itu, fungsi `updateApprover1` dan `updateApprover2` digunakan dalam proses persetujuan invoice. `updateApprover1` memungkinkan persetujuan pertama dengan mencatat waktu dan email pengguna jika nomor surat ditemukan dalam "*DD-Invoice*". `updateApprover2` melakukan persetujuan kedua dengan tambahan pemindahan data ke "*DB-Invoice*" serta pengiriman notifikasi email ke penerima di sel I10. Kedua fungsi ini memastikan persetujuan invoice dilakukan secara bertahap, terdokumentasi, dan divalidasi melalui email untuk transparansi.



```

function sendInvoiceRequest() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var customerId = sheet.getRange("P1").getValue(); // Mengambil ID customer dari sel P1

  if (!customerId) {
    Logger.log("ID Customer tidak ditemukan!");
    SpreadsheetApp.getUi().alert("ID Customer di P1 kosong. Silakan isi terlebih dahulu.");
    return;
  }

  var data = sheet.getDataRange().getValues(); // Mengambil seluruh data di sheet
  var commitmentFee = null;

  for (var i = 1; i < data.length; i++) { // Mulai dari indeks 1 untuk menghindari header
    if (data[i][13] == customerId) { // Kolom N (index 13) adalah 'Copy ID Customer'
      commitmentFee = data[i][7]; // Kolom H (index 7) adalah 'Nominal Commitment Fee'
      break;
    }
  }

  if (commitmentFee === null) {
    Logger.log("Commitment Fee tidak ditemukan untuk ID Customer " + customerId);
    SpreadsheetApp.getUi().alert("Commitment Fee tidak ditemukan untuk ID Customer " + customerId);
    return;
  }

  var formattedCommitmentFee = new Intl.NumberFormat('id-ID', { style: 'currency', currency: 'IDR' }).format(commitmentFee);

  var recipient = "accounting@grinnliving.com";
  var subject = "Permintaan Invoice untuk ID Customer #" + customerId;
  var body = "Halo,\n\nMohon dibuatkan invoice untuk Commitment Fee sebesar " + formattedCommitmentFee +
    "\n\n" + " untuk ID customer no " + customerId + ".\n\nTerima kasih.";

  MailApp.sendEmail({

```

Gambar 3. 22 Function Code Submit Finance

Berdasarkan Gambar 3.22 Fungsi *sendInvoiceRequest()* dalam *Google Apps Script* digunakan untuk mengirim permintaan pembuatan invoice ke divisi keuangan berdasarkan data pelanggan yang terdapat dalam *Google Spreadsheet*. Fungsi ini pertama-tama mengambil ID pelanggan dari sel P1, lalu memverifikasi apakah ID tersebut ada; jika tidak, sistem akan menampilkan peringatan. Selanjutnya, fungsi membaca seluruh data dalam *spreadsheet* dan mencari Commitment Fee berdasarkan kecocokan Copy ID Customer yang terdapat di kolom N (indeks 13). Jika Commitment Fee ditemukan, nilainya diformat ke dalam mata uang Rupiah (IDR) menggunakan *Intl.NumberFormat*. Setelah itu, email permintaan invoice dikirim ke *accounting@grinnliving.com* dengan subjek berisi ID pelanggan dan isi pesan yang mencantumkan nominal Commitment Fee. Setelah email dikirim, sistem mencatatnya di log dan menampilkan notifikasi keberhasilan kepada pengguna.

### 3.2.9 Developmet Client B2B Form dengan Function Java Programming Language

Form B2B Client merupakan sebuah formulir digital yang dirancang untuk mengumpulkan informasi terkait klien bisnis dalam skema Business-to-Business (B2B). Formulir ini digunakan untuk mencatat data perusahaan yang berinteraksi dengan perusahaan utama dalam berbagai kapasitas, seperti pelanggan, mitra bisnis, atau vendor. Dengan menggunakan formulir ini, perusahaan dapat mengelola data klien secara lebih terstruktur, menghindari duplikasi, serta meningkatkan efisiensi dalam pencatatan informasi klien. Data yang dikumpulkan dapat digunakan untuk berbagai tujuan, termasuk analisis pelanggan, segmentasi pasar, dan perencanaan strategi bisnis.

FORM DATABASE CLIENT B2B		SUBMIT	CLEAR
Tanggal	<input type="text"/>		
<b>SUMBER</b>			
Nama Brand	<input type="text"/>		
Nama Penginput	<input type="text"/>		
Sumber Data	<input type="text"/>		
Nama Event (jika ada)	<input type="text"/>		
<b>DETAIL PERUSAHAAN</b>			
Nama Perusahaan	<input type="text"/>		
Industri	<input type="text"/>		
Alamat Perusahaan	<input type="text"/>		
Kode Pos	<input type="text"/>		
Kota	<input type="text"/>		
Propinsi	<input type="text"/>		
Website Perusahaan	<input type="text"/>		
	<small>Kode Pos tidak terdaftar. Silakan cek kembali</small>		
	<small>Kode Pos tidak terdaftar. Silakan cek kembali</small>		

Gambar 3. 23 UI B2B Form

Berdasarkan Gambar 3.23 Pada antarmuka form B2B, pengguna akan menemukan beberapa field input yang harus diisi dengan informasi terkait klien untuk memastikan data yang lengkap dan akurat. Elemen-elemen utama dalam UI form ini mencakup Nama Brand, yang mencatat merek perusahaan klien yang sedang didaftarkan, serta Nama Penginput, yang mengidentifikasi pengguna yang mengisi formulir. Sumber Data menunjukkan asal informasi yang digunakan dalam pendaftaran klien, sedangkan Nama Event (jika ada) mencatat acara tertentu yang berhubungan dengan klien. Nama Perusahaan

diisi dengan identitas perusahaan klien yang ingin didaftarkan, sementara Industri menunjukkan bidang usaha tempat perusahaan tersebut beroperasi. Alamat Perusahaan, Kode Pos, Kota, dan Provinsi digunakan untuk mencatat lokasi fisik perusahaan, memberikan gambaran jelas mengenai letak geografisnya. Selain itu, Website Perusahaan memungkinkan pencantuman situs resmi perusahaan untuk referensi lebih lanjut. Bagian DETAIL CONTACT PERSON berisi informasi kontak dari perwakilan perusahaan, termasuk nama, nomor telepon, dan alamat email, yang berguna untuk komunikasi lebih lanjut antara perusahaan dan pihak terkait.

```
function submitData() {
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin meng-SUBMIT data ke database need approval?", ui.ButtonSet.YES_NO);
  if (response == ui.Button.NO) {
    ui.alert("Pengiriman data dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("F-Client");
  const databaseId = "1dJN7IPJEf61-VN866S_Jtta0opL3lg3lfg-01u1F_d4";
  const databaseSheetName = "DB-Client";
  const userEmail = Session.getActiveUser().getEmail();

  // Daftar sel yang akan diambil datanya
  const cellMapping = ["B4", "C7", "C8", "C9", "C18", "C13", "C14", "C15", "C16", "C17", "C18", "C19", "C22", "C23", "C24"];

  // C24 digabung dengan D24, C25 digabung dengan D25
  const additionalCells = [{"C25", "D25"}, {"C26", "D26"}];

  let isComplete = true;
  let data = cellMapping.map(cell => {
    let range = formSheet.getRange(cell);
    let value = range.getValue();
    if (!value) {
      isComplete = false;
      range.setBorder(true, true, true, true, true, true, "red", SpreadsheetApp.BorderStyle.SOLID);
    }
    return value;
  });

  // Proses penggabungan C24+D24 dan C25+D25
  additionalCells.forEach(pair => {
    let val1 = formSheet.getRange(pair[0]).getValue();

```

Gambar 3. 24 Code Function Submit And Clear

Berdasarkan Gambar 3.24 Kode di atas adalah skrip Google Apps Script untuk mengelola formulir di Google Spreadsheet. Fungsi utama, submitData(), mengirimkan data dari sheet "F-Client" ke "DB-Client" dalam spreadsheet lain yang diidentifikasi dengan databaseId. Skrip meminta konfirmasi sebelum mengirim data, mengambil informasi dari sel yang ditentukan dalam cellMapping, dan memberi peringatan dengan mengubah border sel menjadi merah jika ada yang kosong. Setelah validasi, data dikirim, nomor surat ditentukan berdasarkan jumlah baris, serta waktu pengiriman dan email pengguna dicatat. Setelah sukses, border merah dikembalikan ke putih. Fungsi

clearForm() menghapus isi formulir setelah dikirim, meminta konfirmasi jika showConfirmation bernilai true, mengosongkan sel dalam cellMapping, dan mengubah border menjadi putih, kecuali untuk sel C25 dan C26 yang hanya mengubah bordernya. Skrip ini memastikan data dikirim dengan benar dan meningkatkan pengalaman pengguna.

### 3.2.10 Developmet Petty Cash Form dengan Function Java Programming Language

Petty Cash Form merupakan sistem yang dirancang untuk menyimpan dan mengelola data pelanggan secara terstruktur Untuk mendukung operasional Pt Baller Indonesia, khususnya dalam pencatatan dan pengelolaan dana kas kecil. Proses pengembangan sistem ini memerlukan waktu pengerjaan selama satu minggu, dengan tujuan utama memastikan transparansi, efisiensi, dan akurasi dalam pencatatan pengeluaran kas kecil, sehingga mempermudah divisi keuangan dalam memonitor dan mengontrol penggunaan dana secara *real-time*.

Gambar 3. 25 UI Petty Cash Form

Berdasarkan Gambar 3.25 menunjukkan tampilan antarmuka (UI) Petty Cash Form yang digunakan untuk mencatat dan mengelola transaksi kas kecil dalam perusahaan. Formulir ini mencakup informasi utama seperti Nama Brand, Departemen, Status, serta identitas pembuat dan dua tingkat approver untuk memastikan validasi transaksi. Selain itu, terdapat bagian untuk

menentukan Tipe Pemakaian Budget dan Sumber Petty Cash, yang berfungsi untuk mencatat apakah transaksi berkaitan dengan penggunaan atau pengisian ulang dana kas kecil. Form ini juga mencatat detail transaksi, termasuk Tanggal Transaksi, kategori Uang Masuk atau Uang Keluar, Nama Barang atau Keperluan, Jumlah, Harga, Total, serta Keterangan tambahan. Dengan struktur ini, sistem Petty Cash Form membantu dalam pencatatan yang akurat dan transparan, mendukung akuntabilitas keuangan perusahaan.

```
function submitData() {
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin meng-SUBMIT data ke databa");
  if (response == ui.Button.NO) {
    ui.alert("Pengiriman data dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form");
  const databaseId = "1QR_Fa0wvK1a1LxMjoh09XFAKBPbcW4N0z4Vuog";
  const databaseSheetName = "DD-PET";
  const userEmail = Session.getActiveUser().getEmail();

  const cellMapping = ["B4", "C7", "C8", "H8", "C18", "C11", "C12", "C15", "C16"];
  const data = cellMapping.map(cell => formSheet.getRange(cell).getValue());

  const missingDataColumns = [];
  data.forEach((value, index) => {
    if (!value) missingDataColumns.push(cellMapping[index]);
  });

  const lastRow = formSheet.getLastRow();
  const itemsRange = formSheet.getRange("A18:"$J$[lastRow]).getValues();
  const filteredItems = itemsRange.filter(row => row[2] || row[3]);

  if (filteredItems.length === 0) {
    ui.alert("Mohon isi minimal 1 data petty cash.");
    return;
  }

  let incompleteRows = [];
  filteredItems.forEach((row, index) => {
    let missingColumns = [];
  });
}

function loadDate(showConfirmation = true) {
  // Tampilkan pop-up konfirmasi jika diperlukan
  if (showConfirmation) {
    const ui = SpreadsheetApp.getUi();
    const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin me-LOAD data dari database need approval?", ui.ButtonSet.YES_NO);
  }

  if (response == ui.Button.NO) {
    ui.alert("Pemuatan data dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form");

  // Definisikan sel yang perlu dikosongkan
  const clearCells = ["B4", "C7", "C8", "H8", "C9", "C18", "C11", "C12", "C15", "C16"];
  clearCells.forEach(cell => formSheet.getRange(cell).setValue(""));

  // Kosongkan tabel barang mulai dari baris 18 ke bawah
  const lastRow = formSheet.getLastRow();
  const startRow = 18;
  if (lastRow == startRow) {
    formSheet.getRange("A18:"$J$[lastRow]).clearContent();
    formSheet.getRange("H18:"$J$[lastRow]).clearContent();
  }

  const databaseId = "1QR_Fa0wvK1a1LxMjoh09XFAKBPbcW4N0z4Vuog";
  const databaseSheetName = "DD-PET";
  const databaseFile = SpreadsheetApp.openById(databaseId);
  const databaseSheet = databaseFile.getSheetByName(databaseSheetName);
}

function clearForm(showConfirmation = true) {
  if (showConfirmation) {
    const ui = SpreadsheetApp.getUi();
    const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin menghapus semua data di formulir?", ui.ButtonSet.YES_NO);
    if (response == ui.Button.NO) {
      ui.alert("Penghapusan data dibatalkan.");
      return;
    }
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form");

  const cellMapping = [
    "B4", "C7", "C8", "H8", "C18", "G10", "I10", "C11", "G11", "I11", "C12", "G12", "I12",
    "C15", "H15", "C18", "I18", "C19", "C22", "H22", "C25", "C26", "C27", "C9", "C16", "G4", "I5"
  ];
}

```

Gambar 3. 26 Function Submit, Clear, Load Data

Berdasarkan Gambar 3.26 Fungsi dalam *Google Apps Script* ini mengelola formulir Petty Cash di *Google Spreadsheet* dengan tiga fitur utama: pemuatan data (*loadData*), pengiriman data ke *database* (*submitData*), dan penghapusan data formulir (*clearForm*). *loadData()* mengambil data dari sheet *database* "DD-PET" berdasarkan nomor surat di sel G4, menghapus data lama di formulir, lalu mengisi kembali dengan data yang ditemukan. *submitData()* memastikan semua kolom wajib telah diisi, menandai sel kosong dengan border merah jika ada yang belum lengkap, lalu menyimpan transaksi ke

*database* dengan nomor unik sebelum mengosongkan formulir. `clearForm()` menghapus seluruh data di formulir setelah konfirmasi dari pengguna, termasuk daftar transaksi, serta menghapus border validasi. Setelah setiap proses, pengguna mendapat notifikasi sesuai dengan status eksekusi.

```
function updateApprover1() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin melakukan APPROVE 1?",
    ui.ButtonSet.YES_NO
  );
  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses APPROVE 1 dibatalkan.");
    return;
  }
}

function updateApprover2() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin melakukan APPROVE 2?",
    ui.ButtonSet.YES_NO
  );
  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses APPROVE 2 dibatalkan.");
    return;
  }
}

function printAndSendPDF() {
  // Tampilkan pop-up konfirmasi
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert(
    "Konfirmasi",
    "Apakah Anda yakin ingin PRINT dokumen ini?",
    ui.ButtonSet.YES_NO
  );
  // Jika pengguna memilih "No", hentikan eksekusi script
  if (response == ui.Button.NO) {
    ui.alert("Proses PRINT dibatalkan.");
    return;
  }
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = ss.getSheetByName("Print");
  var totalRows = 58; // Jumlah total baris yang ada di sheet
  var reservedRows = 2; // Jumlah baris paling bawah yang tidak akan disembunyikan
  var rangeC = sheet.getRange(1, 3, totalRows - reservedRows); // Ambil data di kolom B, kecuali 2 baris terakhir
  var columnCData = rangeC.getValues(); // Ambil nilai di kolom C
  var lastRowWithDataInC = 0;
  // Unhide semua baris
  sheet.showRows(1, totalRows);
  // Log untuk debugging
  console.log("Data di kolom C:");
  columnCData.forEach(function(row, index) {
    console.log("Baris " + (index + 1) + ": " + row[0]);
  });
  // Cari baris terakhir yang memiliki data di kolom B
  for (var i = columnCData.length - 1; i >= 0; i--) {
    if (columnCData[i][0] != "") { // Periksa apakah sel di kolom C tidak kosong
      lastRowWithDataInC = i + 1;
    }
  }
}
```

Gambar 3. 27 Function Approval dan Send PDF

Berdasarkan Gambar 3.27 Skrip ini mengelola proses persetujuan (approval) pada Google *Spreadsheet* dalam dua tahap, yaitu Approve 1 dan Approve 2. Pada tahap pertama, sistem menampilkan pop-up konfirmasi, lalu memperbarui kolom timestamp dan email approver pertama di sheet DD-PET dalam *database* sementara berdasarkan No Surat yang diinput pada sheet Form, dengan peringatan jika No Surat tidak ditemukan. Pada tahap kedua, sistem kembali meminta konfirmasi sebelum memperbarui timestamp dan email approver kedua, menyalin data dari DD-PET ke *database* final DB-PET, serta mengirimkan email notifikasi ke alamat yang tercantum di sel I10 pada Form. Sementara itu, fungsi `printAndSendPDF` menangani pencetakan dan

pengiriman lembar kerja sebagai PDF melalui email, diawali dengan pop-up konfirmasi untuk pengguna. Jika disetujui, skrip menampilkan semua baris pada sheet Print, menentukan baris terakhir dengan data di kolom C, lalu menyembunyikan baris yang tidak perlu dicetak. Setelah memastikan keberadaan sheet, email pada I10, dan nomor surat di G4, skrip mengonversi sheet menjadi file PDF melalui convertSheetToPDF, yang menggunakan URL ekspor Google Sheets dengan token otorisasi. Jika proses berhasil, PDF dikirim ke email tujuan dengan subjek dan isi pesan yang sesuai, sementara jika terjadi kesalahan dalam pembuatan PDF atau pengiriman, pengguna akan diberi notifikasi.

### ***3.2.11 Development of Log Book Stock Form dengan Function Java Programming Language***

Log Book Stock Form adalah sistem pencatatan stok yang digunakan untuk memantau pergerakan dan ketersediaan stok dalam suatu proses operasional. Pengembangannya menggunakan bahasa pemrograman Java, yang memungkinkan integrasi dengan berbagai sistem serta memastikan keandalan dan efisiensi dalam pencatatan data. Proses pengembangannya memerlukan waktu 1 hari untuk perancangan, implementasi, dan pengujian. Untuk memastikan bahwa sistem dapat berjalan dengan optimal sesuai kebutuhan bisnis. Tujuan utama dari pengembangan ini adalah untuk meningkatkan akurasi dalam pencatatan stok, meminimalkan kesalahan manusia, serta memberikan visibilitas yang lebih baik terhadap stok yang tersedia, sehingga pengambilan keputusan terkait persediaan dapat dilakukan secara lebih efektif. Dengan adanya sistem ini, tim operasional dapat melakukan pelacakan stok secara real-time dan lebih responsif terhadap kebutuhan pengadaan maupun distribusi barang.

Gambar 3. 28 Ui Form Log Book Stock Product

Gambar 3.28 UI Form Log Book Stock Product menampilkan antarmuka pencatatan stok produk, khususnya untuk penjualan koper, dengan beberapa elemen utama. Form ini mencakup kolom SKU sebagai kode unik identifikasi setiap produk, Nama Barang yang menampilkan nama koper yang dicatat, serta Display yang menunjukkan jumlah stok yang dipajang di setiap toko. Selain itu, terdapat kolom Barang Stock yang mencatat jumlah stok aktual yang tersedia di masing-masing toko, sehingga memungkinkan pemantauan ketersediaan barang secara *real-time*. Dengan tampilan ini, pengguna dapat dengan mudah mengontrol stok koper di berbagai lokasi penjualan, memastikan ketersediaan produk, dan mengelola restock secara lebih efisien.

```
function submitData() {
  const ui = SpreadsheetApp.getUi();

  const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin meng-SUBMIT data ke database need approval?");
  if (response == ui.Button.NO) {
    ui.alert("Pengiriman data dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form Log Book Stock");
  const databaseId = "1v-ZivCAEYIM1Jub0z0tugbx-zTAk9f6xUj0USts";
  const databaseSheetName = "DB_LB";
  const userEmail = Session.getActiveUser().getEmail();

  const cellMapping = ["B4", "C7", "C8", "M8", "C9", "C12", "H12", "C13", "H13"];
  let isComplete = true;

  let data = cellMapping.map((cell) => {
    let range = formSheet.getRange(cell);
    let value = range.getValue();
    if (!value) {
      isComplete = false;
      range.setBorder(true, true, true, true, true, "red", SpreadsheetApp.BorderStyle.SOLID);
    }
    return value;
  });

  if (!isComplete) {
    ui.alert("Mohon mengisi seluruh data sebelum mengirimkan formulir.");
    return;
  }

  // pengecekan apakah A17 kosong
}

function clearForm(showConfirmation = true) {
  const ui = SpreadsheetApp.getUi();

  // Jika showConfirmation true, tampilkan konfirmasi sebelum menghapus data
  if (showConfirmation) {
    const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin menghapus semua data dalam form?");
    if (response == ui.Button.NO) {
      ui.alert("Penghapusan data dibatalkan.");
      return;
    }
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form Log Book Stock");

  // Daftar sel yang akan dihapus datanya
  const cellMapping = ["B4", "C9", "C12", "H12", "C13", "H13"];
  const dataProdakRange = "A17:D16";

  // Menghapus data di sel yang ditentukan
  cellMapping.forEach((cell) => {
    formSheet.getRange(cell).setValue("");
  });

  // Menghapus data dalam range produk
  formSheet.getRange(dataProdakRange).clearContent();

  // Mengatur ulang border menjadi putih
  const allRanges = [...cellMapping, dataProdakRange];
  allRanges.forEach((range) => {
    formSheet.getRange(range).setBorder(true, true, true, true, true, "white", SpreadsheetApp.BorderStyle.SOLID);
  });
}
```

Gambar 3. 29 Function Submit Dan Clear

Berdasarkan Gambar 3.29 Fungsi `submitData()` dalam kode *JavaScript* untuk *Google Apps Script* ini digunakan untuk mengirimkan data dari Form Log Book Stock ke *database* need approval yang berada di *Google Spreadsheet* lain. Proses dimulai dengan menampilkan konfirmasi kepada pengguna sebelum data dikirim. Selanjutnya, fungsi ini mengambil nilai dari sel yang telah ditentukan dan memeriksa apakah semua data telah diisi dengan benar. Jika ada sel yang kosong, maka sel tersebut akan diberi border merah sebagai peringatan, dan proses pengiriman akan dibatalkan. Jika semua data lengkap, fungsi akan mengambil data produk dari range A17:J36, memfilter data yang valid, lalu menyimpannya ke dalam *database* dengan menambahkan nomor unik dan informasi pengguna. Setelah data berhasil dikirim, border merah pada sel yang telah diperiksa akan dikembalikan ke warna putih, dan pengguna akan menerima notifikasi keberhasilan. Fungsi `clearFormRL()` digunakan untuk menghapus semua data dalam formulir, dengan opsi konfirmasi untuk mencegah penghapusan yang tidak disengaja. Fungsi ini membersihkan data di sel yang ditentukan, menghapus isi tabel produk, serta mengatur ulang border menjadi putih untuk mengembalikan tampilan form ke kondisi semula.

### ***3.2.12 Development of Log Book Coffee Bar Stock Form dengan Function Java Programming Language***

Log Book Coffee Bar Stock Form adalah sebuah formulir pencatatan stok khusus untuk kebutuhan operasional di area Coffee Bar, yang dirancang untuk memonitor ketersediaan bahan baku seperti kopi, susu, sirup, dan perlengkapan lainnya. Proses pengembangannya memerlukan waktu 1 hari, dengan fokus utama pada efisiensi pencatatan dan akurasi data stok. Formulir ini dikembangkan menggunakan *Java Programming Language* untuk memastikan integrasi yang baik dengan sistem *database*, serta mempermudah pemrosesan dan analisis data stok. Tujuan utama dari pengembangan ini adalah untuk mengotomatiskan pencatatan stok, meminimalkan kesalahan input manual, serta memudahkan staf dalam melacak penggunaan dan kebutuhan bahan baku

secara *real-time*, sehingga operasional Coffee Bar dapat berjalan lebih efisien dan tanpa hambatan

The image shows a web form titled "FORM LOG BOOK COFFEE BAR". At the top right, there are two buttons: "SUBMIT" and "CLEAR". Below the title, there is a "Tanggal" field. The form is divided into three main sections: "KETERANGAN FORM", "DETAIL PERPINDAHAN", and "DETAIL PRODUK".

**KETERANGAN FORM**

Nama Brand	Baller		
Department	Sales	Sub-Department	Offline
Nama Pembuat Form	<input type="text"/>	Timestamp	E-Sign by

**DETAIL PERPINDAHAN**

Tanggal Perpindahan	<input type="text"/>		
Dari	<input type="text"/>	Ke	<input type="text"/>

**DETAIL PRODUK**

Nama Barang	Jumlah	Keterangan	
			1
			2
			3

Gambar 3. 30 Ui Log Book Coffee Bar Stock

Gambar 3.30 UI Log Book Coffee Bar Stock menampilkan antarmuka pencatatan stok bahan baku di Coffee Bar, dengan elemen utama seperti tanggal, jenis kopi, serta dari - ke yang menunjukkan pergerakan stok. Tanggal digunakan untuk mencatat waktu transaksi atau pergerakan stok, memastikan pencatatan yang terorganisir. Jenis kopi merujuk pada varian kopi yang dicatat dalam log, seperti arabika, robusta, atau campuran lainnya. Dari - ke menunjukkan sumber dan tujuan stok, misalnya dari gudang ke Coffee Bar atau dari satu cabang ke cabang lain, sehingga alur distribusi bahan baku dapat dipantau dengan jelas. Dengan UI ini, pencatatan stok menjadi lebih terstruktur, akurat, dan mudah ditelusuri untuk keperluan audit maupun pengelolaan inventaris. Sistem ini juga dirancang dengan antarmuka yang sederhana agar mudah dipahami dan digunakan oleh seluruh tim operasional. Selain itu, fitur validasi data ditambahkan guna memastikan bahwa seluruh input yang masuk sesuai dengan ketentuan yang berlaku. Dengan adanya sistem ini, perusahaan dapat memonitor pergerakan stok secara real-time sehingga mempermudah pengendalian dan perencanaan persediaan.

```

function submitDataLogBook() {
  const ui = SpreadsheetApp.getUi();
  const response = ui.alert("konfirmasi", "Apakah Anda yakin ingin meng-SUBMIT data ke database need approval?", ui.ButtonSet.YES_NO);
  if (response == ui.Button.NO) {
    ui.alert("Pengiriman data dibatalkan.");
    return;
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form Log Book Coffee Bar");
  const databaseSheet = "1v-2jvKdE7WJ_0m2D0tUde-zFW3FvkuJ002Bz";
  const databaseSheetName = "DB-LOG";
  const userEmail = Session.getActiveUser().getEmail();

  const cellMapping = [{"B4", "C4", "C8", "H8", "C9", "C12", "C13", "H13"}];
  let isComplete = true;

  let data = cellMapping.map(cell => {
    let range = formSheet.getRange(cell);
    let value = range.getValue();
    if (!value) {
      isComplete = false;
      range.setBorder(true, true, true, true, true, "red", SpreadsheetApp.BorderStyle.SOLID);
    }
  });

  return value;
}

if (!isComplete) {
  ui.alert("Mohon mengisi seluruh data sebelum mengirimkan formulir.");
  return;
}

// Pengisian apakah A17 kosong
function clearFormLogBook(showConfirmation = true) {
  const ui = SpreadsheetApp.getUi();
  // Jika showConfirmation true, tampilkan konfirmasi sebelum menghapus data
  if (showConfirmation) {
    const response = ui.alert("konfirmasi", "Apakah Anda yakin ingin menghapus semua data dalam form?", ui.ButtonSet.YES_NO);
    if (response == ui.Button.NO) {
      ui.alert("Penghapusan data dibatalkan.");
      return;
    }
  }

  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const formSheet = ss.getSheetByName("Form Log Book Coffee Bar");
  // Buffer sel yang akan dihapus datanya
  const cellMapping = [{"B4", "C4", "C8", "C12", "C13", "H13"}];
  const dataProduceRange = "A17:J18";

  // Menghapus data di sel yang ditentukan
  cellMapping.forEach(cell => {
    formSheet.getRange(cell).setValue("");
  });

  // Menghapus data dalam range produk
  formSheet.getRange(dataProduceRange).clearContent();

  // Mengatur ulang border menjadi putih
  const allRanges = [...cellMapping, dataProduceRange];
  allRanges.forEach(range => {
    formSheet.getRange(range).setBorder(true, true, true, true, true, "white", SpreadsheetApp.BorderStyle.SOLID);
  });
}

```

Gambar 3. 31 Function Submit and Clear Form

Berdasarkan Gambar 3.31 Fungsi `submitDataLogBook()` pada skrip di atas bertujuan untuk mengirim data dari Form Log Book Coffee Bar ke *database* *need approval* dalam Google *Spreadsheet* setelah melakukan validasi input. Skrip ini terlebih dahulu meminta konfirmasi dari pengguna sebelum melakukan pengiriman. Jika ada data yang belum diisi, sel yang kosong akan diberi border merah sebagai indikator. Selain itu, skrip memeriksa apakah setidaknya ada satu detail produk yang diisi sebelum melanjutkan proses. Jika semua validasi terpenuhi, data dari formulir akan dikirim ke *database* dengan menyertakan timestamp, email pengguna, dan nomor transaksi yang dihitung berdasarkan entri sebelumnya. Setelah pengiriman berhasil, skrip menghapus border merah pada sel yang telah diisi dan menampilkan notifikasi sukses. Sementara itu, fungsi `clearFormLogBook()` berfungsi untuk menghapus semua data dalam formulir, dengan opsi konfirmasi dari pengguna, serta mengatur ulang border menjadi putih untuk menjaga tampilan yang rapi. Kedua fungsi ini memastikan pencatatan log stok kopi yang lebih sistematis, mencegah kesalahan input, serta meningkatkan efisiensi dalam pengelolaan data inventaris.

### 3.2.13 Development of Product Usage Report POGL Form dengan Function Java Programming Language

Product Usage Report POGL (Professional Organizer Grinn Living) adalah sebuah formulir yang digunakan untuk mencatat dan melaporkan penggunaan produk dalam operasional layanan profesional organizer di Grinn



(deskripsi barang atau alat yang digunakan), serta kolom Received, Used, dan Return yang merekam jumlah barang yang diterima, digunakan, dan dikembalikan dalam proses organizing. Terakhir, kolom Keterangan Lokasi digunakan untuk mencatat lokasi spesifik di mana kegiatan organizing dilakukan, memastikan dokumentasi yang lebih rinci dan akurat. Formulir ini membantu dalam manajemen inventaris dan operasional organizing secara efisien.

```

118 function Submit() {
119   // Menampilkan konfirmasi pop-up
120   const ui = SpreadsheetApp.getUi();
121   const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin meng-SUBMIT data ke database need approval?", ui.ButtonSet.YES_NO);
122   if (response == ui.Button.NO) {
123     ui.alert("Pengiriman data dibatalkan.");
124     return;
125   }
126
127   // Mengakses Spreadsheet
128   const ss = SpreadsheetApp.getActiveSpreadsheet();
129   const formSheet = ss.getSheetByName("FORM-ProductGL REPORT");
130   const databaseId = "1dn5UsSZRY_0BK3bnK80zR3Vfs_EkjlwDoAPX6pJw0";
131   const databaseSheetName = "DB-ProductGL Report";
132   const userEmail = Session.getActiveUser().getEmail();
133
134   // Cell mapping (kolom yang harus diisi sebelum submit)
135   const cellMapping = ["I7", "F7", "L7", "B6", "B7"];
136   let isComplete = true;
137
138   // Validasi data utama
139   let formData = cellMapping.map((cell, index) => {
140     let range = formSheet.getRange(cell);
141     let value = range.getValue();
142
143     // Jika cell kosong, set flag tidak lengkap
144     if (!value) {
145       isComplete = false;
146     }
147
148     // Jika kolom F7 (index 1 dalam array) maka tambahkan " Hari"
149     if (index === 1 && value) {
150       value = value + " Hari";
151     }
152   });

```

Gambar 3. 33 Function Submit

Berdasarkan Gambar 3.33 Kode *JavaScript* di atas adalah skrip *Google Apps Script* yang digunakan untuk mengirimkan data dari *spreadsheet FORM-ProductGL REPORT* ke *spreadsheet database* bernama *DB-ProductGL Report* setelah melakukan validasi input. Skrip pertama-tama menampilkan pop-up konfirmasi kepada pengguna sebelum mengirimkan data. Jika pengguna mengonfirmasi, skrip kemudian memeriksa apakah semua kolom penting dalam *cellMapping* (seperti "I7", "F7", "L7", "B6", dan "B7") telah terisi. Jika ada yang kosong, pengiriman dibatalkan. Selanjutnya, skrip membuka *spreadsheet database* menggunakan *databaseId*, mencari baris kosong pertama untuk memasukkan data baru, dan mengambil data dari rentang "A12:J221" dalam formulir, tetapi hanya menyertakan baris yang

memiliki nilai pada kolom C (barang). Data yang dikumpulkan kemudian diformat dengan menambahkan timestamp dan email pengguna sebelum disimpan ke *database*. Jika ada data yang memenuhi syarat, skrip menuliskannya ke *database* dan menampilkan notifikasi sukses, sementara jika tidak ada data yang valid, pengguna akan diberi tahu bahwa tidak ada data yang bisa dikirim.

### 3.2.14 Development of Internal Transfer Request Form dengan Function Java Programming Language

*Internal Transfer Request barang* Form adalah sebuah formulir digital yang digunakan untuk mencatat dan memantau proses perpindahan barang penjualan antar lokasi atau divisi dalam perusahaan secara sistematis. Formulir ini dikembangkan menggunakan bahasa pemrograman *Java* dengan pendekatan berbasis fungsi, sehingga memungkinkan pengolahan data yang efisien dan modular. Proses pengembangannya memerlukan waktu sekitar 2 minggu, Tujuan utamanya adalah untuk memastikan transparansi dan akurasi dalam pelacakan pergerakan barang penjualan, mempercepat proses persetujuan internal, serta mendukung pengambilan keputusan yang berbasis data secara real-time.

Gambar 3. 34 UI Form Internal Transfer Request

Form Internal Transfer Request, seperti yang ditampilkan pada Gambar 3.34, merupakan formulir digital yang digunakan untuk mencatat dan mengelola proses perpindahan barang penjualan antar lokasi atau unit dalam perusahaan. Form ini mencakup informasi penting seperti tanggal permintaan, nomor surat, nama brand, departemen dan sub-departemen terkait, serta status permintaan. Selain itu, terdapat bagian identifikasi pihak yang terlibat dalam proses, termasuk nama pembuat form dan para approver, lengkap dengan timestamp dan tanda tangan elektronik (E-Sign) untuk memastikan keabsahan proses. Form juga memuat dua bagian utama, yaitu Detail Product Transfer Request yang mencatat informasi barang yang dipindahkan, seperti nama dan kode barang, jumlah, dan keterangan serta Detail Penerimaan Barang yang diisi oleh pihak FA, mencakup lokasi asal dan tujuan, alasan perpindahan, serta data penerimaan barang seperti tanggal diterima dan nama penerima. Dengan struktur yang lengkap dan sistematis ini, form berfungsi sebagai alat dokumentasi dan verifikasi yang efektif dalam mendukung transparansi serta akuntabilitas proses internal transfer barang di lingkungan perusahaan.

```

1 function submitData() {
2   const ui = SpreadsheetApp.getUi();
3   const response = ui.alert("Konfirmasi", "Apakah Anda yakin ingin meng-SUBMIT data ke database need approval?", ui.ButtonSet.YES_NO);
4   if (response == ui.Button.NO) {
5     ui.alert("Pengiriman data dibatalkan.");
6     return;
7   }
8
9   // PENYIMPANAN DI DATABASE
10  const ss = SpreadsheetApp.getActiveSpreadsheet();
11  const formSheet = ss.getSheetByName("Form");
12  const databaseId = "168XZ65b3MzK1E14E54ea1IPFVvpJ6LHGXeZ7o2JA";
13  const databaseSheetName = "DD-ITR";
14  const userEmail = Session.getActiveUser().getEmail();
15
16  // CELL MAPPING PENGISIAN DATA DI DATABASE
17  const cellMapping = ["B4", "C7", "C8", "H8", "C18", "C11", "C12", "C15", "C16", "C17", "C18", "C19"];
18  let isComplete = true;
19  let data = cellMapping.map(cell => {
20    let range = formSheet.getRange(cell);
21    let value = range.getValue();
22    if (!value) {
23      isComplete = false;
24      range.setBorder(true, true, true, true, true, true, "red", SpreadsheetApp.BorderStyle.SOLID);
25    }
26    return value;
27  });
28
29  if (!isComplete) {
30    ui.alert("Mohon mengisi seluruh data sebelum mengirimkan formulir.");
31    return;
32  }
33
34  // PENYIMPANAN DATA PRODUCT TRANSFER REQUEST

```

Gambar 3. 35 Code Dalam Form Internal Transfer Request

Berdasarkan Gambar 3.35 Kode tersebut dengan skrip Google Apps Script yang digunakan untuk mengelola pengisian dan persetujuan formulir permintaan transfer produk pada Google Sheets. Fungsi submitData()

digunakan untuk mengirim data dari sheet "Form" ke sheet database bernama "DD-ITR" setelah validasi isian dilakukan, serta memberikan nomor surat secara otomatis. Fungsi `clearForm()` menghapus seluruh isian dalam formulir, sedangkan `loadData()` memuat ulang data dari database berdasarkan nomor surat yang dimasukkan di sel G4. Fungsi `updateApprover1()` digunakan untuk mencatat persetujuan tahap pertama (Approve 1) ke dalam database setelah konfirmasi pengguna. Seluruh fungsi dilengkapi dengan notifikasi interaktif kepada pengguna, validasi input, serta pemetaan data antar sel agar pengelolaan permintaan lebih terdokumentasi secara sistematis.

### ***3.2.15 Development Leave Request Form with Submit clear load and approval Functions Using Java Programming Language***

Leave Request Form adalah sebuah formulir digital yang digunakan oleh karyawan untuk mengajukan cuti kepada atasan atau departemen terkait. Formulir ini dirancang menggunakan bahasa pemrograman Java dan dilengkapi dengan fungsi-fungsi seperti submit, clear, load, dan approval untuk mempermudah proses pengajuan dan persetujuan cuti. Waktu pengerjaan form ini diperkirakan selama 3 minggu. Tujuan dibuatnya Leave Request Form ini adalah untuk mempermudah, mempercepat, dan mengefisienkan proses pengajuan cuti karyawan, mengurangi penggunaan kertas (paperless), serta meningkatkan transparansi dan akurasi dalam pencatatan data cuti melalui sistem yang terkomputerisasi. Formulir ini juga dilengkapi dengan notifikasi otomatis kepada atasan terkait setiap kali ada pengajuan cuti baru. Sistem ini memudahkan monitoring status cuti karyawan, apakah sudah disetujui atau masih menunggu persetujuan. Selain itu, riwayat pengajuan cuti akan tersimpan dengan rapi dan dapat diakses kapan saja jika diperlukan. Dengan desain antarmuka yang sederhana dan ramah pengguna, form ini dapat digunakan oleh seluruh karyawan tanpa memerlukan pelatihan khusus. Kehadiran sistem ini diharapkan mampu meningkatkan efisiensi kerja HRD dalam pengelolaan data cuti.

FORM PENGAJUAN ANNUAL LEAVE			
1			
2			
3			
4	Tanggal		No. Surat
5			
6	<b>KETERANGAN FORM</b>		
7	Nama Brand		
8	Department		Sub-Department
9	Status		
10	Nama Pembuat Form		Timestamp
11	Nama Approver 1 (PIC)		E-Sign by
12	Nama Approver 2 (Direct Superior)		Timestamp
13	Nama Approver 3 (Human Resources)		E-Sign by
14			
15	<b>DETAIL FORM</b>		
16	Nama Employee	Rosa Aulia Wangi	
17	Employee ID	BLI 013	
18	Temporary Person in Charge		
19	Jenis Perizinan	Annual Leave	Sisa Jumlah Hari = 0

Gambar 3. 36 UI leave request form

Berdasarkan gambar 3.36, Form Pengajuan tersebut merupakan sebuah dokumen digital yang digunakan untuk proses permohonan cuti atau izin khusus oleh seorang karyawan dalam suatu organisasi atau perusahaan. Form ini terdiri dari dua bagian utama, yaitu Keterangan Form dan Detail Form.

Pada bagian Keterangan Form, terdapat informasi administratif seperti tanggal pengajuan, nomor surat, nama brand, departemen, sub-departemen, serta status pengajuan. Di bagian ini juga dicantumkan nama pembuat form beserta *timestamp* dan tanda tangan digital (*e-sign*), serta tiga level persetujuan: Approver 1 (PIC), Approver 2 (Atasan Langsung), dan Approver 3 (Human Resources), masing-masing dilengkapi dengan *timestamp* dan *e-sign* juga.

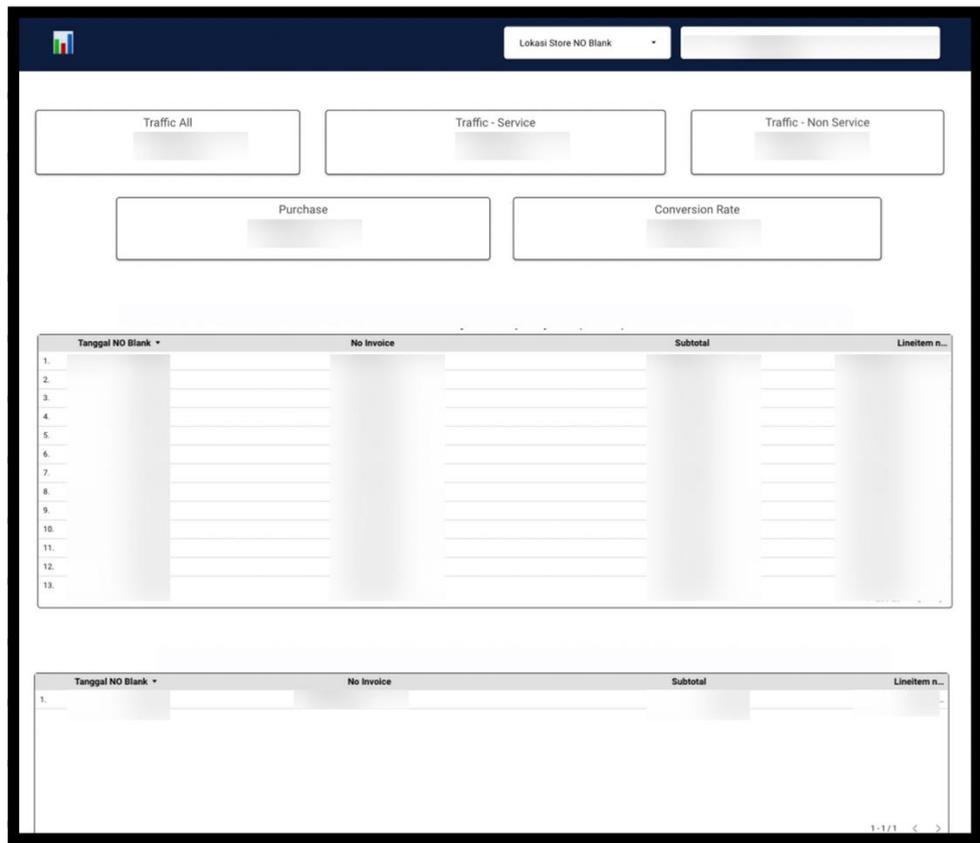
Bagian Detail Form berisi data personal karyawan yang mengajukan izin, seperti nama, Employee ID, dan siapa yang akan menggantikan sementara saat karyawan tersebut tidak masuk. Selanjutnya, terdapat pilihan jenis perizinan dan kolom khusus untuk menjelaskan jika pengajuan tersebut termasuk dalam kategori *special leave*. Kemudian dicantumkan tanggal cuti (Day Off), jumlah hari izin, serta rentang tanggal cuti dari tanggal berapa

sampai tanggal berapa lengkap dengan waktu (*timestamp* 00:00:00) yang menunjukkan kejelasan periode izin.

Setelah dilakukan proses User Acceptance Testing (UAT), dinilai perlu adanya penambahan fitur baru dalam pengelolaan database, yaitu penambahan warna sebagai penanda visual untuk memudahkan identifikasi jenis perizinan. Setiap kategori perizinan akan diberikan warna pastel yang berbeda agar lebih mudah dikenali: Annual Leave (C) dengan warna oranye pastel, Unpaid Leave (U) dengan warna merah pastel, Special Leave (H) dengan warna kuning pastel, Arrive Late (S) dengan warna biru pastel, Early Leave (S) dan Sakit (S) juga akan mengikuti skema warna yang serupa sesuai ketentuan yang disepakati. Penambahan warna ini bertujuan untuk meningkatkan keterbacaan data dan mempermudah proses monitoring perizinan di sistem.

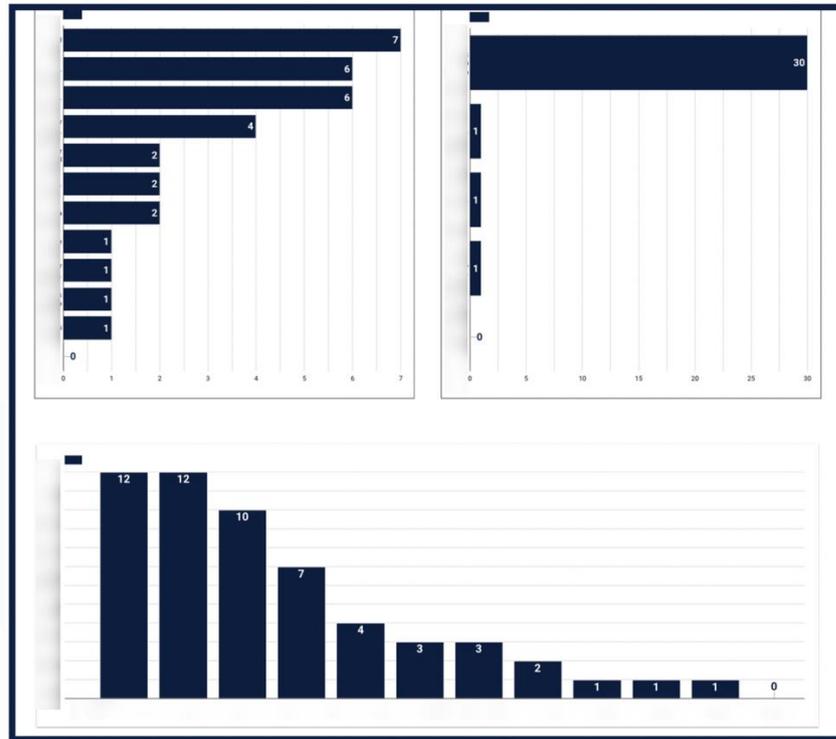
### ***3.2.16 Development of Customer Purchase & Warranty Dashboard dengan Sales Trends & SKU-Based Analysis***

*Customer Purchase & Warranty Dashboard* adalah sebuah *dashboard* yang berisi data pembelian pelanggan dan informasi garansi, yang disajikan dalam bentuk visualisasi interaktif. *Dashboard* ini mencakup analisis tren penjualan (Sales Trends) serta evaluasi berbasis SKU (SKU-Based Analysis) untuk memahami pola pembelian dan klaim garansi pelanggan. Pengembangannya menggunakan *Google Looker Studio*, dengan waktu pengerjaan selama 5 hari. Visualisasi ini bertujuan untuk memberikan wawasan mendalam mengenai performa produk, mengidentifikasi pola pembelian, serta mempermudah pemantauan klaim garansi, sehingga perusahaan dapat mengambil keputusan strategis berbasis data. *Dashboard* ini juga dirancang agar memudahkan manajemen dalam memantau performa penjualan dan layanan purna jual secara real-time.



Gambar 3. 37 Traffic Purchase *Dashboard*

Gambar 3.37 Traffic Purchase *Dashboard* menampilkan total traffic purchase dan conversion rate yang dianalisis melalui berbagai metrik. *Dashboard* ini mencakup tabel dengan informasi detail seperti tanggal, nomor invoice, kategori, line item, quantity, dan subtotal, memberikan gambaran lengkap tentang transaksi yang terjadi. Selain itu, analisis traffic purchase dilakukan dengan menghitung conversion rate menggunakan formula  $(\text{COUNT}(\text{Financial Status}) / \text{COUNT\_DISTINCT}(\text{No Invoice Traffic}) * 100)$  untuk mengukur efektivitas konversi dari transaksi. Visualisasi utama yang digunakan adalah dual line chart, yang membantu dalam memantau tren perubahan traffic purchase dan conversion rate secara bersamaan, sehingga memudahkan pengambilan keputusan berbasis data.



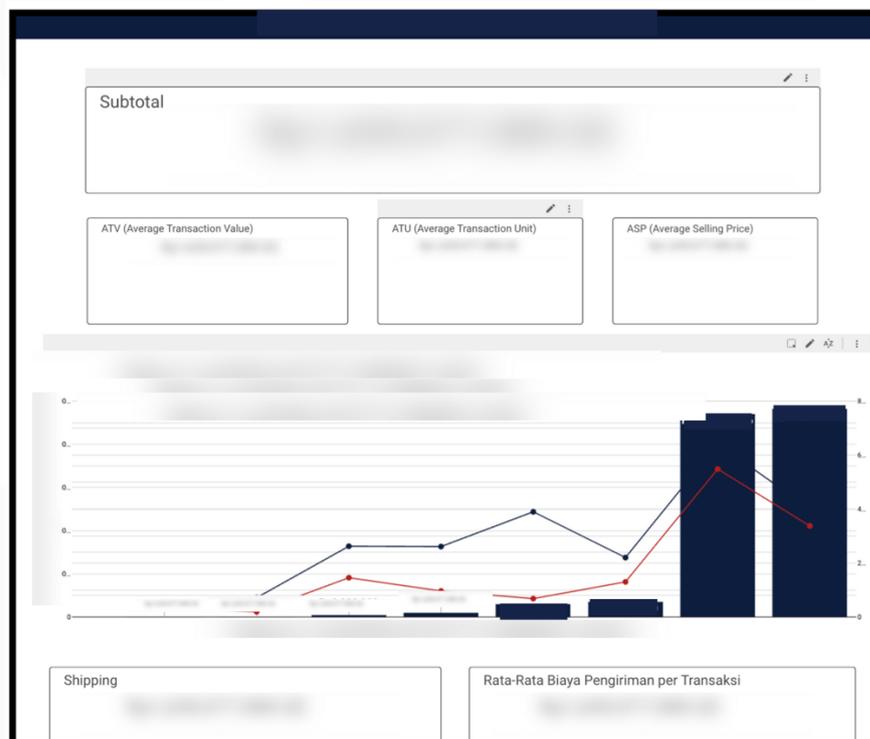
Gambar 3. 38 *Customer Insight Dashboard*

Berdasarkan Gambar 3.38 *Customer Insight Dashboard* menampilkan analisis mendalam tentang karakteristik pelanggan berdasarkan data survei penjualan koper. *Dashboard* ini mencakup demografi usia, domisili pelanggan, serta tipe customer berdasarkan pola kedatangan, yang memberikan wawasan mengenai kebiasaan belanja mereka. Selain itu, *dashboard* ini juga mengidentifikasi preferensi merek koper yang paling diminati serta kendala yang sering dihadapi pelanggan terkait koper, seperti daya tahan, ukuran, atau harga. Dengan visualisasi yang interaktif, *dashboard* ini membantu dalam memahami perilaku pelanggan dan menyusun strategi pemasaran yang lebih efektif untuk meningkatkan penjualan koper.



Gambar 3. 39 *Coffe Summary Dashboard*

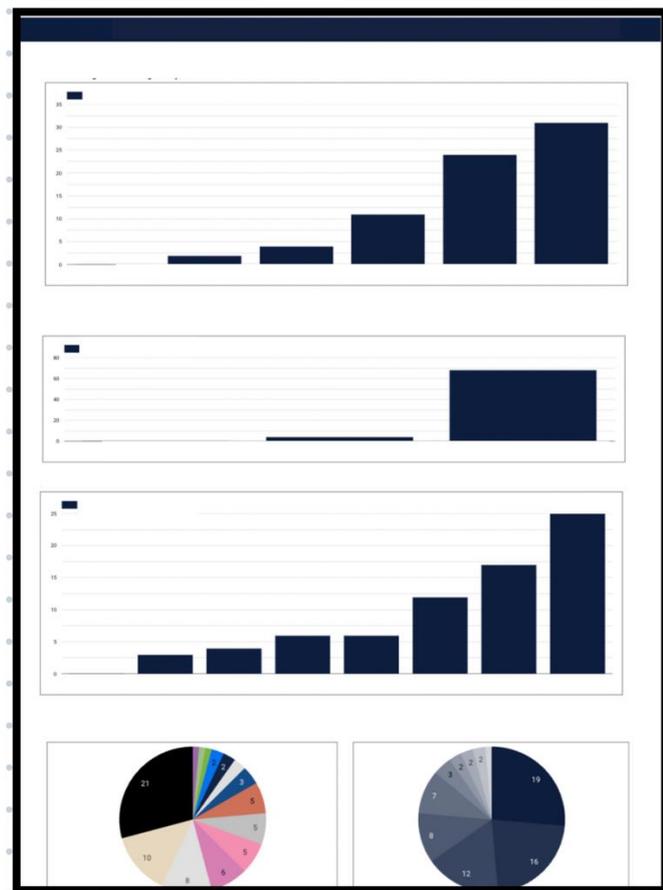
Berdasarkan Gambar 3.39 Coffee Summary Dashboard menampilkan ringkasan stok kopi dalam bentuk card summary, yang memberikan informasi terkini mengenai ketersediaan stok di Glaze Store. Dashboard ini membantu dalam memantau jumlah kopi yang masih tersedia, memastikan kelancaran operasional, serta menghindari kekurangan atau kelebihan stok. Dengan tampilan yang ringkas dan mudah dipahami, dashboard ini mendukung pengambilan keputusan yang lebih cepat dan efisien dalam manajemen persediaan kopi.



Gambar 3. 40 Sales Overview

Berdasarkan Gambar 3.40 Sales Overview menampilkan ringkasan penjualan yang mencakup subtotal penjualan, Average Transaction Value (ATV), dan Average Selling Price (ASP). Visualisasi utama berupa line chart with double chart, yang membandingkan ATV dan ASP berdasarkan kategori produk untuk mengidentifikasi tren penjualan. Selain itu, dashboard ini juga

menyajikan data terkait metode pengiriman, rata-rata biaya pengiriman, serta preferensi pelanggan dalam memilih metode pengiriman, sehingga dapat memberikan wawasan mendalam mengenai pola pembelian dan efisiensi logistik.



Gambar 3. 41 Category & Sub-Category Product Analysis

Berdasarkan Gambar 3.41 Category & Sub-Category Product Analysis menampilkan visualisasi analisis kategori produk berdasarkan penjualan terbaik, memberikan wawasan mengenai produk yang paling diminati. Selain itu, *dashboard* ini membandingkan penjualan produk bundling vs non-bundling, sehingga dapat mengidentifikasi efektivitas strategi bundling dalam meningkatkan penjualan. Analisis lebih lanjut mencakup sub-kategori produk, seperti warna, ukuran, dan material, untuk

memahami preferensi pelanggan terhadap karakteristik produk tertentu, yang dapat digunakan dalam perencanaan strategi pemasaran dan pengelolaan stok

### ***3.2.17 Development of 2025 Marketing Department KPI Dashboard Untuk PT Baller Indonesia Sentosa***

Marketing Department KPI *Dashboard* adalah sebuah *dashboard* yang digunakan untuk memantau kinerja utama departemen pemasaran PT Baller Indonesia Sentosa dalam mencapai target tahun 2025. *Dashboard* ini berisi berbagai metrik seperti jumlah leads, conversion rate, ROI marketing campaigns, customer engagement, dan sales performance, yang membantu dalam mengevaluasi efektivitas strategi pemasaran. *Dashboard* ini dikembangkan menggunakan *Google Looker Studio*, dengan waktu pengerjaan selama 5 hari, Untuk memastikan tampilan yang interaktif dan mudah dipahami. Visualisasi ini bertujuan untuk memberikan insight berbasis data dalam pengambilan keputusan strategis, sehingga perusahaan dapat mengoptimalkan kampanye pemasaran dan meningkatkan pertumbuhan bisnis secara efektif.

Selain itu, dashboard ini dilengkapi dengan fitur filter yang memungkinkan pengguna untuk menganalisis data berdasarkan periode waktu, channel pemasaran, dan segmentasi pasar tertentu. Hal ini memudahkan tim pemasaran dalam mengidentifikasi area mana yang memerlukan perbaikan maupun potensi yang dapat dimaksimalkan. Dengan tampilan visual yang menarik dan informatif, dashboard ini juga memudahkan manajemen dalam melakukan monitoring kinerja secara berkala serta membandingkan capaian dengan target yang telah ditetapkan. Kehadiran dashboard ini diharapkan mampu mendorong budaya kerja yang lebih data-driven dalam lingkungan pemasaran PT Baller Indonesia Sentosa.



Gambar 3. 42 Marketing Department KPI Dashboard

Berdasarkan Gambar 3.42 Marketing Department KPI Dashboard visualisasi menggunakan data fiktif yang memiliki berbagai fitur utama yang digunakan untuk memantau dan mengevaluasi kinerja pemasaran PT Baller Indonesia Sentosa. Fitur Sales Target Nett Yearly dan Sales Target Gross Yearly berfungsi untuk melacak pencapaian target penjualan bersih dan kotor dalam setahun, memberikan gambaran mengenai performa bisnis. Gross Profit (%) menunjukkan persentase keuntungan kotor yang diperoleh dari total penjualan, membantu dalam analisis profitabilitas. ROI Marketing dan ROI Digital mengukur efektivitas investasi pemasaran secara keseluruhan dan khusus dalam ranah digital, memastikan efisiensi strategi pemasaran yang diterapkan. Customer Acquisition Cost (CAC) memberikan wawasan terkait biaya yang dikeluarkan untuk mendapatkan pelanggan baru, sementara Brand Awareness Growth mencerminkan perkembangan kesadaran merek di pasar.

Dengan fitur-fitur ini, *dashboard* memberikan analisis mendalam yang mendukung pengambilan keputusan strategis berbasis data.

### ***3.2.18 Development of 2025 Sales Department KPI Dashboard Untuk PT Baller Indonesia Sentosa***

Sales Department KPI *Dashboard* adalah sebuah alat analitik yang dirancang untuk memantau dan mengukur kinerja penjualan PT Baller Indonesia Sentosa secara *real-time*. *Dashboard* ini berisi berbagai metrik seperti Total Sales Revenue, Sales Growth, Conversion Rate, Average Transaction Value (ATV), serta Customer Retention Rate, yang membantu dalam mengevaluasi efektivitas strategi pemasaran dan penjualan. Dikembangkan menggunakan *Google Looker Studio* dalam waktu 5 hari, *dashboard* ini dirancang agar interaktif dan mudah dipahami oleh tim penjualan. Visualisasi ini bertujuan untuk memberikan wawasan mendalam mengenai tren penjualan, membantu dalam pengambilan keputusan strategis, serta mengoptimalkan pencapaian target perusahaan di tahun 2025.

Selain itu, *dashboard* ini dilengkapi dengan fitur drill-down yang memungkinkan pengguna untuk melihat detail data penjualan berdasarkan produk, wilayah, atau periode tertentu. Dengan fitur ini, tim penjualan dapat lebih mudah mengidentifikasi area yang berkinerja baik maupun yang memerlukan perbaikan. Tampilan visual yang informatif dan terstruktur membantu mempermudah proses monitoring kinerja secara berkala. Kehadiran *dashboard* ini juga mendorong tim penjualan untuk lebih fokus dalam mencapai target yang telah ditetapkan dan meningkatkan efisiensi kerja secara keseluruhan. Dengan data yang akurat dan *real-time*, manajemen dapat mengambil keputusan yang lebih tepat dan cepat dalam merespons dinamika pasar.



Gambar 3. 43 Sales Department Dashboard

Berdasarkan Gambar 3.43 Sales Department KPI Dashboard dengan menggunakan data fiktif menampilkan berbagai fitur utama yang dirancang untuk memantau dan mengevaluasi kinerja penjualan secara menyeluruh. Fitur-fitur tersebut mencakup B2B Sales, yang mengukur performa penjualan dalam segmen bisnis ke bisnis; Sales Target, yang membandingkan target dengan pencapaian aktual; serta Customer & Service Center Satisfaction Rate, yang mengukur tingkat kepuasan pelanggan terhadap layanan yang diberikan. Selain itu, terdapat Sales Target on Store,

yang menunjukkan pencapaian target penjualan di toko fisik, serta Customer Service Online Sales Satisfaction Rate, yang mengukur kepuasan pelanggan terhadap layanan penjualan online. Fitur lainnya adalah **Live Streaming AVG Revenue per Session**, yang menganalisis rata-rata pendapatan yang dihasilkan dari sesi live streaming penjualan. *Dashboard* ini membantu tim sales dalam memahami performa mereka, mengidentifikasi area yang perlu ditingkatkan, serta mengoptimalkan strategi pemasaran dan penjualan secara lebih efektif.

### ***3.2.19 Development of 2025 Warehouse & Operation Department KPI Dashboard Untuk PT Baller Indonesia Sentosa***

Warehouse & Operation Department KPI *Dashboard* adalah sebuah *dashboard* yang dirancang untuk memantau dan mengevaluasi kinerja operasional dan manajemen gudang di PT Baller Indonesia Sentosa. *Dashboard* ini berisi berbagai metrik penting seperti Inventory Turnover, Order Fulfillment Rate, Warehouse Efficiency, dan Operational Cost Tracking, yang memberikan gambaran menyeluruh tentang efektivitas penyimpanan, distribusi, serta efisiensi operasional. Dikembangkan menggunakan *Google Looker Studio* dalam waktu 5 hari, *dashboard* ini dirancang agar interaktif dan mudah dipahami oleh tim operasional dan manajemen. Visualisasi ini bertujuan untuk meningkatkan efisiensi manajemen gudang, mengoptimalkan proses logistik, dan memastikan tingkat pemenuhan pesanan yang lebih baik, sehingga dapat mendukung kelancaran rantai pasok perusahaan.

Selain itu, dashboard ini dilengkapi dengan fitur monitoring real-time yang memungkinkan tim operasional untuk segera mengidentifikasi potensi masalah, seperti keterlambatan pengiriman atau ketidaksesuaian stok. Dengan adanya data yang terintegrasi dan mudah diakses, proses pengambilan keputusan menjadi lebih cepat dan akurat. Dashboard ini juga membantu perusahaan dalam menjaga kestabilan kinerja operasional serta meningkatkan kepuasan pelanggan melalui pelayanan yang lebih optimal.



Gambar 3. 44 Warehouse & Operations Department *Dashboard*

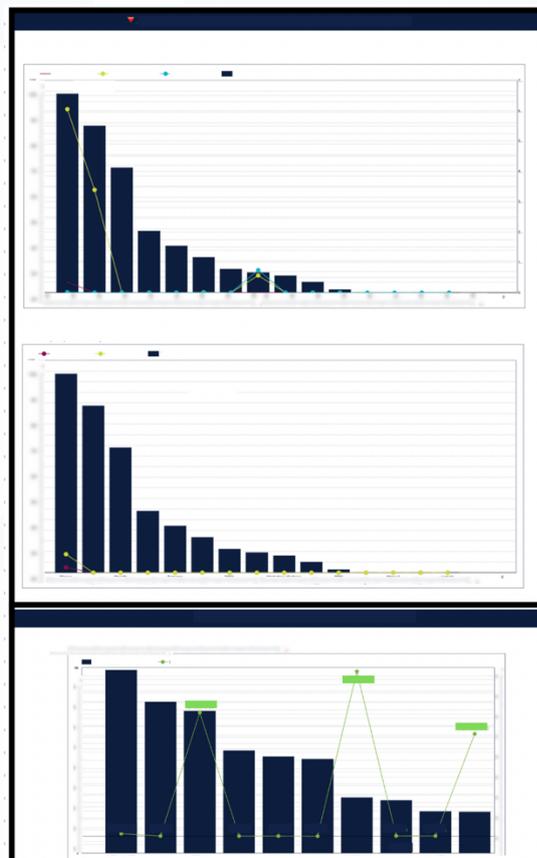
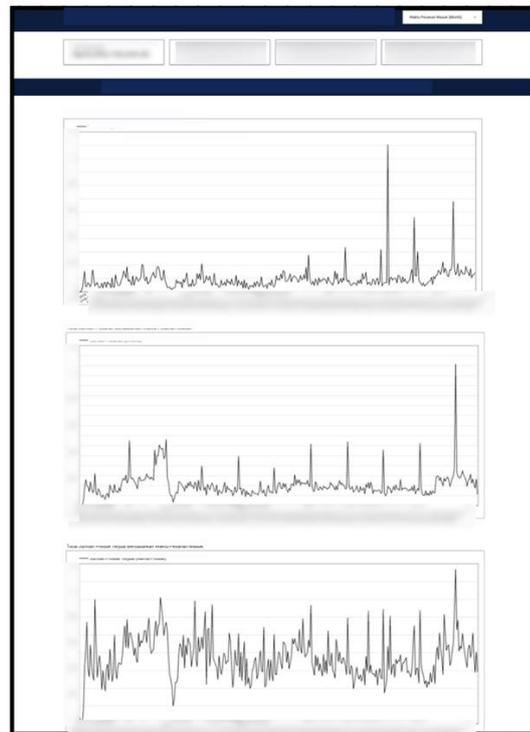
Berdasarkan Gambar 3.44 Warehouse & Operation Department KPI *Dashboard* yang menggunakan data fiktif memiliki berbagai fitur penting untuk memantau efisiensi dan kinerja operasional gudang di PT Baller Indonesia Sentosa. Fitur-fitur utama dalam *dashboard* ini mencakup Productivity Rate, yang mengukur produktivitas pekerja gudang; *Inventory Accuracy / Shrinkage*, yang membantu dalam mengontrol keakuratan stok

dan mengidentifikasi potensi kehilangan barang; serta *Report on WH Accuracy*, yang memberikan laporan terkait keakuratan data gudang. Selain itu, terdapat metrik On-time Order Processing untuk mengevaluasi ketepatan waktu pemrosesan pesanan, Order Accuracy untuk memastikan pesanan dikirim sesuai permintaan pelanggan, serta Service Warranty Satisfaction Rate untuk menilai kepuasan pelanggan terhadap layanan garansi. Fitur tambahan seperti Layouting juga disertakan Untuk mengoptimalkan tata letak gudang agar operasional lebih efisien. *Dashboard* ini memberikan insight yang komprehensif bagi tim operasional dalam meningkatkan efektivitas dan akurasi manajemen gudang.

### **3.2.20 Development of Ecommerce Performance Dashboard untuk PT Baller Indonesia Sentosa**

*Ecommerce Performance Dashboard* adalah sebuah dashboard yang dirancang untuk memantau dan menganalisis kinerja e-commerce di PT Baller Indonesia Sentosa. Dashboard ini menyajikan berbagai metrik penting seperti jumlah penjualan, traffic pengunjung, conversion rate, serta performa produk dan *channel* pemasaran. Dikembangkan menggunakan Google Looker Studio dalam waktu tiga hari, dashboard ini dirancang secara interaktif dan intuitif agar mudah dipahami dan digunakan oleh tim operasional maupun manajemen. Visualisasi yang disajikan bertujuan untuk memberikan insight yang cepat dan akurat, sehingga dapat mendukung pengambilan keputusan yang lebih efektif dan berbasis data.

Selain itu, dashboard ini dilengkapi dengan fitur filter dinamis berdasarkan periode waktu, kategori produk, dan saluran pemasaran, sehingga pengguna dapat melakukan analisis mendalam sesuai kebutuhan spesifik mereka. Dengan pendekatan ini, tim dapat dengan cepat mengidentifikasi tren penjualan, mengevaluasi efektivitas *Campaign* pemasaran, serta mengoptimalkan strategi produk yang kurang perform.



Gambar 3. 45 Dashboard Ecommerce Performance Dashboard

Berdasarkan Gambar 3.45, dashboard Ecommerce Performance menampilkan berbagai visualisasi penting yang menggambarkan kinerja penjualan secara keseluruhan. Terdapat grafik tren pendapatan, jumlah pesanan, serta produk terjual dari waktu ke waktu yang membantu memantau pertumbuhan bisnis. Selain itu, performa marketplace ditampilkan berdasarkan toko dan kategori channel, sehingga memudahkan dalam melihat kontribusi tiap channel penjualan. Visualisasi juga mencakup TOP 10 nama dan varian produk terlaris yang memberi gambaran produk mana yang paling diminati. Distribusi metode pembayaran disajikan secara jelas untuk melihat preferensi konsumen, disertai analisis diskon, subsidi, dan biaya marketplace berdasarkan channel yang berguna untuk mengukur efisiensi biaya promosi. Terakhir, status pesanan di marketplace juga ditampilkan untuk memberikan gambaran proses pemenuhan pesanan, dari yang telah selesai hingga yang masih dalam proses.

### **3.2.21 *Development of Inventory Movement & Stock Status Dashboard* untuk PT Baller Indonesia Sentosa**

*Inventory Movement & Stock Status* adalah dashboard yang dirancang untuk memantau pergerakan stok dan kondisi persediaan barang di PT Baller Indonesia Sentosa secara real-time. Dashboard ini menyajikan visualisasi yang mencakup informasi masuk dan keluarnya barang, jumlah stok tersedia di setiap gudang, serta status stok seperti overstock, understock, dan stockout. Dikembangkan menggunakan Google Looker Studio dalam waktu 7 hari, dashboard ini dibuat agar mudah digunakan oleh tim operasional dan logistik. Tujuannya adalah untuk membantu pengambilan keputusan terkait manajemen persediaan, mencegah kekurangan atau kelebihan stok, serta memastikan ketersediaan produk tetap optimal untuk mendukung kelancaran proses penjualan dan distribusi.



Gambar 3. 46 Inventory Movement & Stock Status Dashboard

Berdasarkan Gambar 3.46, Inventory Movement & Stock Status Dashboard menggunakan data fiktif menampilkan ringkasan pergerakan barang masuk dan keluar di PT Baller Indonesia Sentosa, khususnya pada Departemen Pengiriman Produk dan SUB-Departemennya. Visualisasi ini menyajikan tren aktivitas perpindahan barang secara detail, yang dikelompokkan berdasarkan nama produk, lokasi asal pengiriman, dan lokasi tujuan pengiriman. Melalui dashboard ini, tim operasional dapat memantau arus distribusi barang secara menyeluruh, mengidentifikasi frekuensi serta volume perpindahan stok antar lokasi, dan mengevaluasi efisiensi proses logistik. Data yang ditampilkan membantu perusahaan dalam mengontrol ketersediaan stok secara lebih akurat, meminimalisir keterlambatan pengiriman, serta mendukung pengambilan keputusan berbasis data dalam pengelolaan inventaris.

### 3.3 Kendala yang Ditemukan

Kendala yang dihadapi selama masa Internship Sebagai Data Engineer Intern pada PT Baller Indonesia Sentosa dapat dilihat pada point point dibawah antara lain:

- a) Kurangnya Familiaritas dengan Beberapa Istilah dalam Ranah Bisnis Penjualan Koper istilah teknis yang spesifik dan sering digunakan dalam operasional bisnis. Beberapa istilah dalam bahasa Inggris, seperti *SKU (Stock Keeping Unit)*, *inventory turnover*, dan *supply chain logistics*, masih belum terlalu familiar. Hal ini menyebabkan adanya tantangan dalam memahami konteks data serta proses bisnis yang berkaitan dengan analisis yang dilakukan.
- b) Penggunaan *spreadsheet* manual masih menjadi metode utama dalam beberapa proses di perusahaan. Hal ini membuat proses analisis data menjadi lebih lambat dan rentan terhadap human error.
- c) PT Baller Indonesia Sentosa sebagai startup memiliki ritme kerja yang cepat (*fast-paced environment*), di mana setiap tugas memiliki deadline yang ketat. Hal ini menjadi tantangan karena dituntut untuk menyelesaikan pekerjaan dalam waktu singkat, tetapi tetap harus mempertahankan kualitas hasil kerja. Oleh karena itu, kemampuan manajemen waktu menjadi keterampilan penting yang perlu ditingkatkan.

### 3.4 Solusi atas Kendala yang Ditemukan

Dari penjelasan berupa hambatan-hambatan intern Pada PT Baller Indonesia Seentosa pada poin sebelum ini, terdapat berbagai solusi yang diusung, sebagai berikut

- a) Mengatasi kurangnya familiaritas dengan istilah bisnis penjualan koper, yaitu dengan dapat memanfaatkan berbagai sumber referensi seperti artikel, jurnal, dan buku yang membahas konsep bisnis ritel dan manajemen rantai pasok dengan istilah seperti *SKU (Stock Keeping Unit)*, *inventory turnover*,

dan *supply chain logistics*. Selain itu, peserta intern juga bertanya langsung kepada rekan kerja atau mentor sehingga menjadi langkah efektif untuk memahami konteks penggunaan istilah tersebut dalam operasional bisnis perusahaan.

- b) Mengatasi keterbatasan akibat penggunaan spreadsheet manual, mulai mempelajari dan menerapkan tools analisis data yang lebih efisien seperti SQL untuk pengolahan data yang lebih cepat dan terstruktur. Selain itu, penggunaan tools *Business Intelligence (BI)*, *Google Data Studio* dapat membantu dalam visualisasi data yang lebih profesional dan interaktif. Mengusulkan penerapan automasi dalam pengolahan data dengan menggunakan *Java App Script* juga menjadi solusi efektif Untuk meningkatkan efisiensi serta mengurangi risiko human error dalam analisis data.
- c) Mengatasi tantangan bekerja di lingkungan startup yang bergerak cepat, dengan berupa meningkatkan keterampilan manajemen waktu dengan menerapkan metode Pomodoro Technique Untuk menyusun prioritas kerja yang lebih efektif. Memanfaatkan tools manajemen proyek seperti Notion dapat membantu dalam mengorganisir tugas dan menghindari penundaan pekerjaan.