

**PENGEMBANGAN SISTEM INFORMASI MENGGUNAKAN
GOOGLE APPS SCRIPT
STUDI KASUS: TOKO ASIH**



Disusun Oleh:

Nama : Dwi Candra Yovan Rindho
NIM : 20523236

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2024

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN SISTEM INFORMASI MENGGUNAKAN
GOOGLE APPS SCRIPT
STUDI KASUS: TOKO ASIH**

TUGAS AKHIR



Disusun Oleh:

N a m a

: Dwi Candra Yovan Rindho

NIM

: 20523236

الجامعة الإسلامية
الاندونيسية

Yogyakarta, 30 Oktober 2024

Pembimbing,

(Dr Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN SISTEM INFORMASI MENGGUNAKAN
GOOGLE APPS SCRIPT
STUDI KASUS: TOKO ASIH

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 11 November 2024

Tim Penguji

Dr. Raden Teduh Dirgahayu, S.T., M.Sc.

Anggota 1

Arrie Kurniawardhani, S.Si., M.Kom.

Anggota 2

Kholid Haryono, S.T., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Dwi Candra Yovan Rindho

NIM : 20523236

Tugas akhir dengan judul:

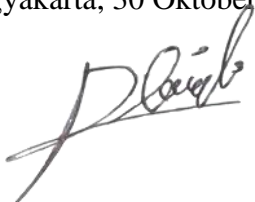
**PENGEMBANGAN SISTEM INFORMASI MENGGUNAKAN
GOOGLE APPS SCRIPT STUDI
KASUS: TOKO ASIH**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 30 Oktober 2024




(Dwi Candra Yovan Rindho)

HALAMAN PERSEMBAHAN

Dengan segala kerendahan hati dan rasa syukur, penulis mempersembahkan skripsi ini kepada orang tua dan kakak penulis. Doa, dukungan, dan pengorbanan yang tak terhitung dari mereka telah membantu penulis menjalani perjalanan ini dengan baik. Skripsi ini adalah ungkapan terima kasih penulis kepada orang tua dan kakak yang telah menjadi sumber semangat dalam proses penulisannya. Kelulusan ini adalah hadiah yang penulis berikan, meskipun tidak dapat terbayar hanya dengan tulisan di skripsi ini. Penulis berharap kelulusan ini menjadi kebanggaan tidak hanya bagi penulis, tetapi juga bagi mereka.

Penulis juga ingin mengucapkan terima kasih kepada diri sendiri yang telah berjuang dalam penulisan skripsi ini dengan mengorbankan banyak waktu dan tenaga. Terima kasih atas ketahanan yang telah ditunjukkan hingga sejauh ini. Penulis bersyukur karena selalu percaya kepada Allah SWT, yang senantiasa memberikan kekuatan dan petunjuk dalam setiap langkah. Meskipun masih banyak tantangan dan perjuangan yang akan dihadapi di masa depan, masa perkuliahan ini sangat berarti bagi penulis dan telah memberikan banyak makna serta pelajaran berharga yang akan diambil.

HALAMAN MOTO

“Cukuplah Allah (menjadi penolong) bagi kami dan Dia sebaik-baik pelindung.”

(Q.S Ali Imran : 173)

“Keberhasilan bukanlah milik orang yang pintar, keberhasilan adalah kepunyaan mereka yang senantiasa berusaha.”

(B.J Habibie)

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur kehadiran Allah SWT atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini yang berjudul “Pengembangan Sistem Informasi Menggunakan *Google Apps Script* Studi Kasus Toko Kelontong Asih.” Semoga shalawat dan salam selalu tercurah kepada Nabi Muhammad SAW, yang telah membimbing umatnya ke jalan kebenaran dan menjadi rahmat bagi seluruh alam.

Penyusunan skripsi ini dilakukan untuk memenuhi salah satu syarat guna memperoleh gelar Sarjana Komputer pada program studi Informatika di Fakultas Teknologi Industri, Universitas Islam Indonesia. Selain itu, penulisan ini juga bertujuan untuk memenuhi kebutuhan akademik serta sebagai bentuk pengabdian penulis dalam mengembangkan ilmu pengetahuan, khususnya di bidang informatika. Dalam penyelesaian skripsi ini, penulis mengucapkan terima kasih kepada Bapak Raden Teduh Dirgahayu, selaku dosen pembimbing, yang telah meluangkan waktu untuk memberikan bimbingan dan dukungan dalam penulisan ini. Penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan dan jauh dari kesempurnaan. Oleh karena itu, penulis sangat menghargai berbagai kritik dan saran yang membangun untuk penelitian ini. Penulis berharap skripsi ini dapat memberikan manfaat dalam pengembangan ilmu pengetahuan dan kontribusi positif bagi dunia akademik.

Penulis sepenuhnya menyadari bahwa penyusunan skripsi ini tidak akan berhasil tanpa pertolongan-Nya serta dukungan dari berbagai pihak. Oleh karena itu, dengan tulus, penulis menyampaikan ucapan terima kasih yang sebanyak-banyaknya kepada:

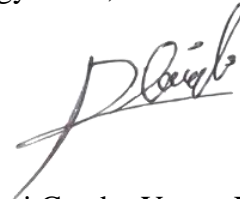
1. Allah SWT, yang telah memberikan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik dan lancar.
2. Orangtua penulis, Bapak Lasno dan Ibu Asih, yang telah memberikan doa, kasih sayang, pengorbanan, dan dukungan yang tiada henti selama proses penulisan skripsi ini. Tiada balasan yang sepadan selain tulisan ini yang merupakan ungkapan cinta kepada mereka. Semoga Allah SWT membalas semua kebaikan dan pengorbanan mereka dengan kebahagiaan.
3. Kakak penulis, Syafrudin Candra Prayikno, S.Kom., M.M., dan Thania Kencanasari, S.Pd serta seluruh keluarga besar, atas perhatian, dukungan, dan nasihat berharga yang menginspirasi penulis untuk terus berusaha.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., sebagai dosen pembimbing skripsi yang telah sabar memberikan arahan, bimbingan, dan masukan berharga dari awal hingga akhir penulisan skripsi ini. Terima kasih atas kesabaran dan waktu yang telah diberikan.
5. Bapak/Ibu Dosen Jurusan Informatika, yang telah memberikan ilmu dan wawasan berharga yang sangat bermanfaat dalam penulisan penelitian ini.
6. Teman-teman dan sahabat penulis yang selalu memberikan dukungan dan bersedia mendengarkan keluh kesah penulis selama proses penulisan skripsi demi meraih gelar yang diimpikan.
7. Kepada teman sekelompok bimbingan penulis yang telah memberikan dukungan dan semangat selama pengerjaan skripsi ini. Terima kasih telah saling mendukung dan mendorong dalam perjuangan ini.

8. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah memberikan dukungan, bantuan, dan motivasi selama proses penulisan skripsi ini. Setiap dukungan dan dorongan dari siapapun sangat berarti bagi penulis. Semoga Allah SWT membalas semua kebaikan dari berbagai pihak.

Akhir kata, penulis berharap skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan dan bagi almamater Universitas Islam Indonesia serta pihak-pihak terkait.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 30 Oktober 2024



(Dwi Candra Yovan Rindho)

SARI

Pengelolaan toko kelontong sering menghadapi berbagai tantangan, terutama dalam mencatat transaksi, mengelola stok, dan menangani layanan tambahan seperti BRILink, terutama jika dilakukan secara manual. Proses manual ini tidak hanya memakan waktu, tetapi juga berisiko tinggi terhadap kesalahan pencatatan yang dapat berdampak negatif pada kinerja operasional toko dan akan membuat kerugian. Penelitian ini bertujuan untuk mengembangkan sistem informasi pengelolaan toko kelontong dengan memanfaatkan *Google Apps Script* (GAS) sebagai *backend* dan *Spreadsheets* sebagai basis data. Sistem yang dirancang dapat mengotomatiskan berbagai proses penting, termasuk pencatatan transaksi penjualan, pengelolaan data barang, serta pengolahan transaksi BRILink, pembayaran PDAM dan listrik, serta pengelolaan hutang. Dengan otomatisasi ini, diharapkan dapat mengurangi beban kerja manual dan meningkatkan akurasi dalam pengelolaan data. Antarmuka pengguna dikembangkan dengan menggunakan HTML, CSS, dan *Bootstrap* untuk memastikan tampilan yang responsif dan mudah digunakan, sehingga pengguna dari berbagai latar belakang dapat mengakses sistem dengan nyaman. Pengujian dilakukan dengan metode black box testing, yang menunjukkan bahwa semua fitur utama berfungsi sesuai dengan spesifikasi yang ditetapkan, tanpa adanya kesalahan signifikan. Hasil penelitian ini menunjukkan bahwa sistem informasi yang dikembangkan mampu meningkatkan efisiensi operasional, transparansi, dan memudahkan pemantauan aktivitas toko secara real-time, sehingga dapat memenuhi kebutuhan pengelolaan toko kelontong secara efektif dan memberikan nilai tambah bagi pemilik usaha.

Kata Kunci : Pengelolaan toko kelontong, Google Apps Script, sistem informasi.

GLOSARIUM

BRILink	layanan perbankan yang disediakan oleh Bank Rakyat Indonesia (BRI) yang memungkinkan masyarakat untuk melakukan transaksi keuangan secara langsung di lokasi yang lebih dekat dan mudah dijangkau, seperti di toko kelontong, warung, atau agen BRILink.
<i>In-sourcing</i>	praktik di mana perusahaan memilih untuk menggunakan sumber daya internal untuk melakukan fungsi tertentu atau menyediakan layanan.
<i>Outsourcing</i>	praktik di mana perusahaan mengalihkan sebagian fungsi bisnisnya kepada pihak ketiga atau vendor luar.
<i>Cosourcing</i>	pendekatan di mana perusahaan bekerja sama dengan pihak ketiga untuk melakukan fungsi tertentu.
<i>GAS (Google Apps Script)</i>	platform pemrograman berbasis <i>cloud</i> yang memungkinkan pengguna untuk mengotomatisasi, mengintegrasikan, dan memperluas fungsi aplikasi Google seperti <i>Google Sheets</i> , <i>Google Docs</i> , <i>Google Drive</i> , <i>Gmail</i> , dan banyak lagi.
<i>Google Workspace</i>	kumpulan alat produktivitas dan kolaborasi berbasis <i>cloud</i> yang ditawarkan oleh <i>Google</i> .
Struk	Bukti fisik dari transaksi yang menunjukkan rincian pembelian atau pembayaran yang dilakukan.
IoT (Internet of Things)	Jaringan perangkat fisik yang terhubung ke internet, memungkinkan mereka untuk mengumpulkan dan bertukar data.
<i>Use Case</i>	alat dalam pengembangan perangkat lunak yang digunakan untuk mendefinisikan interaksi antara pengguna (aktor) dan sistem.
<i>Unified Modeling Language (UML)</i>	Bahasa standar yang digunakan untuk mendeskripsikan, merancang, dan mendokumentasikan sistem perangkat lunak serta sistem lainnya.
URL (Uniform Resource Locator)	alamat yang digunakan untuk mengidentifikasi dan mengakses sumber daya di internet.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penelitian	4
BAB II KAJIAN PUSTAKA	5
2.1 Landasan Teori.....	5
2.1.1 Google Apps Script.....	5
2.2 Tinjauan Pustaka	6
2.2.1 <i>Google Apps Script</i> untuk <i>Smart City</i>	6
2.2.2 <i>Google Apps Script</i> untuk Pendidikan	7
2.2.3 <i>Google Apps Script</i> untuk Bot Telegram dan <i>Push Notification</i>	8
2.2.4 <i>Google Apps Script</i> untuk Monitoring Sistem Berbasis IoT.....	9
2.2.5 <i>Google Apps Script</i> untuk Integrasi Layanan Eksternal	9
2.2.6 <i>Google Apps Script</i> untuk <i>Cloud Computing</i>	10
2.2.7 <i>Google Apps Script</i> untuk Perpustakaan.....	10
2.2.8 Kesimpulan Tinjauan Pustaka.....	11
BAB III METODOLOGI PENELITIAN	13
3.1 Analisis Kebutuhan	13
3.1.1 Analisis Proses Bisnis	13
3.1.2 <i>Use Case Diagram</i>	14
3.1.3 Deskripsi <i>Use Case</i>	15
3.2 Perancangan Perilaku	18
3.2.1 Insert Transaksi Pengadaan Barang	18
3.2.2 Insert Transaksi Penjualan Barang	19
3.2.3 Melihat Harga Jual dan Jumlah Barang	20
3.2.4 Insert Transaksi BRILink	20
3.2.5 Insert Hutang PDAM dan Listrik.....	21
3.2.6 Insert Hutang Pembeli.....	21
3.3 Analisis Tampilan	22
3.4 Perancangan Antarmuka	22
3.4.1 Antarmuka <i>Use Case</i> Insert Transaksi Pengadaan Barang	23

3.4.2	Antarmuka <i>Use Case</i> Insert Transaksi Penjualan Barang	23
3.4.3	Antarmuka <i>Use Case</i> Lihat Harga Jual dan Jumlah Barang	24
3.4.4	Antarmuka <i>Use Case</i> Transaksi BRILink	24
3.4.5	Antarmuka <i>Use Case</i> Insert Data PDAM	25
3.4.6	Antarmuka <i>Use Case</i> Insert Data Listrik	26
3.4.7	Antarmuka <i>Use Case</i> Insert Hutang Pembeli	27
3.5	Perancangan Basis Data	28
3.6	Perancangan Sistem dengan Google Apps Script	29
BAB IV HASIL DAN PEMBAHASAN.....		31
4.1	Hasil	31
4.1.1	Implementasi Basis Data.....	31
4.1.2	Implementasi Sistem	38
4.2	Pembahasan.....	63
4.3	Pengujian Sistem.....	64
4.3.1	Pengujian <i>Blackbox</i>	65
4.3.2	Pengujian <i>System Usability Scale (SUS)</i>	69
BAB V KESIMPULAN DAN SARAN		71
5.1	KESIMPULAN.....	71
5.2	SARAN	71
DAFTAR PUSTAKA.....		73
LAMPIRAN		75

DAFTAR TABEL

Tabel 3.1 Deskripsi <i>Use Case</i> Pemilik	15
Tabel 3.2 Deskripsi <i>Use Case</i> Admin.....	16
Tabel 3.3 Deskripsi <i>Use Case</i> Kasir	17
Tabel 4.1 Pengujian Aktor Pemilik.....	65
Tabel 4.2 Pengujian Aktor Admin	66
Tabel 4.3 Pengujian Aktor Kasir	67
Tabel 4.4 Tabel Hasil Pengujian <i>System Usability Scale</i> (SUS)	69

DAFTAR GAMBAR

Gambar 3.1 <i>Use Case Diagram</i>	15
Gambar 3.2 Diagram aktivitas Insert Transaksi Pengadaan Barang.....	19
Gambar 3.3 Diagram aktivitas Insert Transaksi Penjualan Barang	19
Gambar 3.4 Diagram aktivitas Melihat Harga Jual dan Jumlah Barang.....	20
Gambar 3.5 Diagram aktivitas Transaksi BRILink	20
Gambar 3.6 Diagram aktivitas Insert Hutang PDAM dan Listrik	21
Gambar 3.7 Diagram aktivitas Hutang Pembeli	22
Gambar 3.8 Transaksi Pengadaan Barang	23
Gambar 3.9 Transaksi Penjualan Barang	23
Gambar 3.10 Melihat Harga dan Stock.....	24
Gambar 3.11 Transaksi BRILink	24
Gambar 3.12 Insert Data PDAM	25
Gambar 3.13 Data PDAM.....	25
Gambar 3.14 Insert Data Listrik	26
Gambar 3.15 Data Listrik	27
Gambar 3.16 Hutang Pembeli.....	27
Gambar 3.17 Relasi Tabel.....	28
Gambar 4.1 <i>Sheet users</i>	31
Gambar 4.2 <i>Sheet Data_Barang</i>	32
Gambar 4.3 <i>Sheet Data_Penjualan</i>	33
Gambar 4.4 <i>Sheet BRILink</i>	34
Gambar 4.5 <i>Sheet PDAM</i>	35
Gambar 4.6 <i>Sheet Listrik</i>	36
Gambar 4.7 <i>Sheet Hutang</i>	37
Gambar 4.8 <i>Sheet Data_Pelanggan</i>	38
Gambar 4.9 <i>Form Login User</i>	39
Gambar 4.10 Script Login.....	40
Gambar 4.11 Tampilan Pemberitahuan Kesalahan.....	40
Gambar 4.12 Halaman Pengelolaan PDAM	41
Gambar 4.13 Script Mengambil Data PDAM.....	42
Gambar 4.14 Script Menampilkan Data PDAM.....	42
Gambar 4.15 Form <i>Add Data PDAM</i>	42

Gambar 4.16 <i>Script Add Data PDAM</i>	43
Gambar 4.17 Halaman Pengelolaan Listrik	44
Gambar 4.18 Script Mengambil Data Listrik	44
Gambar 4.19 Script Menampilkan Data Listrik.....	45
Gambar 4.20 <i>Form Add Data Listrik</i>	45
Gambar 4.21 Script <i>Add Data Listrik</i>	46
Gambar 4.22 Laporan Transaksi.....	47
Gambar 4.23 Script Mengambil Laporan Transaksi.....	47
Gambar 4.24 Halaman Data Barang	48
Gambar 4.25 Script Mengambil Data Barang.....	49
Gambar 4.26 Script Menampilkan Data Barang.....	49
Gambar 4.27 <i>Form Add Data Barang</i>	49
Gambar 4.28 Script <i>Add Data Barang</i>	50
Gambar 4.29 Halaman Melihat Harga Jual dan Stock Barang	51
Gambar 4.30 Script Menampilkan Harga Jual dan Stock Barang	51
Gambar 4.31 Penjualan Barang	52
Gambar 4.32 Script <i>Add Data Penjualan Barang</i>	53
Gambar 4.33 Script Menampilkan Data Penjualan Barang	53
Gambar 4.34 Transaksi BRILink.....	54
Gambar 4.35 Script Mengambil Data BRILink.....	55
Gambar 4.36 Script Menampilkan Data BRILink	55
Gambar 4.37 <i>Form Add Data BRILink</i>	55
Gambar 4.38 Script <i>Add Data BRILink</i>	56
Gambar 4.39 Transaksi Data PDAM	57
Gambar 4.40 Script Transaksi PDAM	57
Gambar 4.41 Tampilan <i>Action PDAM</i>	58
Gambar 4.42 <i>Script Action PDAM</i>	58
Gambar 4.43 Transaksi Data Listrik.....	59
Gambar 4.44 Script Transaksi Listrik.....	59
Gambar 4.45 Tampilan <i>Action Listrik</i>	60
Gambar 4.46 Script <i>Action Listrik</i>	60
Gambar 4.47 Halaman Data Hutang.....	61
Gambar 4.48 Script Mengambil Data Hutang	62
Gambar 4.49 Script Menampilkan Data Hutang.....	62

Gambar 4.50 <i>Form Add Data Hutang</i>	62
Gambar 4.51 Script Add Data Hutang	63

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penggunaan teknologi informasi telah menjadi suatu keharusan untuk meningkatkan efisiensi dan produktivitas dalam berbagai jenis bisnis, termasuk toko kelontong (Simandjuntak, 2009). Toko kelontong seringkali berlokasi di kawasan perumahan atau pemukiman penduduk, menjadi pilihan utama masyarakat sekitar karena dianggap lebih dekat dan dapat dicapai dengan berjalan kaki. Toko kelontong merupakan usaha kecil yang menyediakan berbagai macam kebutuhan sehari-hari, seperti beras, bumbu dapur, peralatan mandi, sabun mencuci pakaian, pembersih rumah, dan makanan.

Pada umumnya, toko kelontong masih menggunakan pendekatan manual dalam mencatat pengelolaan toko, termasuk jumlah barang, harga barang, dan transaksi jual beli. Kendati demikian, pengelolaan yang masih bersifat manual seringkali menyulitkan pemilik toko saat melakukan pengecekan terhadap data toko. Dengan pertumbuhan jumlah barang yang terus berubah setiap harinya, diperlukan pencatatan yang lebih efisien. Penggunaan buku atau kertas sebagai alat pencatatan, yang seringkali tulisannya kurang jelas, dapat menyebabkan kebingungan pemilik toko terkait jumlah barang (atulrofida, 2021).

Terkadang juga saat membeli barang dari supplier atau toko grosiran yang notanya terkadang menggunakan sobekan kertas yang sering terselip dan juga hilang. Nota kertas yang menumpuk dan juga harga yang berbeda beda setiap pembelian barang yang sama membuat salah saat memberi harga jual. Harga jual yang pasti akan berganti saat ada kenaikan harga beli membuat pemilik harus teliti saat mengecek nota pembelian yang baru. Permasalahan tersebut juga terjadi di salah satu Toko Asih.

Toko Asih berdiri sejak tahun 2018 yang bertepatan di Desa Sejomulyo RT 006/001 ,Juwana ,Pati Jawa Tengah. Toko Asih sekaligus menjadi agen BRILink di daerah tersebut. Terkadang toko ini masih suka bingung membedakan transaksi jual beli barang dan transaksi BRILink. Selain itu pengelolaan data pada transaksi BRILink juga masih manual dengan menumpuk banyak struk. Transaksi juga tidak semuanya terbayar adapun hutang dalam penjualan dalam toko kelontong. Toko ini juga biasanya menalangi pembayaran PDAM dan juga listrik melalui BRILink hal tersebut membuat penumpukan struk yang banyak dan membuat Toko Asih kebingungan saat ada yang mau membayar hutangnya. Bukan hanya

hutang dalam transaksi BRILink melainkan ada hutang dari penjualan barang yang membuat pemilik toko kadang lupa untuk hal tersebut.

Dalam pengembangan sistem informasi, Sistem informasi merupakan suatu kebutuhan yang tidak dapat dipisahkan oleh manusia di era digital saat ini. Sistem informasi terus berkembang seiring dengan perkembangan teknologi informasi yang sangat cepat dan keberadaan sistem informasi sangat dibutuhkan untuk mendukung kinerja dalam meningkatkan efisiensi, efektivitas, dan produktivitas berbagai organisasi. Sistem informasi biasanya dikembangkan dengan beberapa metode, seperti *in-sourcing*, *outsourcing*, dan *co-sourcing*. Setiap metode memiliki kelebihan dan kekurangan tersendiri (Sari, 2019). Sehingga, toko kelontong mungkin akan menghadapi kesulitan dalam mengembangkan sistem informasi sendiri karena kurangnya tenaga ahli, waktu yang dibutuhkan, dan biaya yang harus dikeluarkan. Selain itu, perubahan teknologi informasi yang cepat juga dapat menjadi kendala dalam pengembangan sistem informasi. Oleh karena itu, pengembangan Sistem informasi dapat dilakukan menggunakan *Google Apps Script* yang akses penggunaannya dapat dilakukan secara gratis.

Penggunaan *Google Apps Script* dapat digunakan untuk mengotomatisasi tugas-tugas yang sering dilakukan di *Google Workspace*. *Google Apps Script* satu satunya platform pengembang aplikasi cepat berbasis *cloud*. Selain itu, *Google Apps Script* mudah digunakan bahkan oleh pemula sekalipun dan menyediakan banyak fungsi yang siap digunakan untuk mengoptimalkan pekerjaan (Nur Aidah, 2023). Salah satu alasan memilih *Google Apps Script* adalah biayanya yang sangat terjangkau, bahkan gratis, yang memungkinkan toko kelontong dengan anggaran terbatas untuk mengakses teknologi efisien (Google, 2024). Selain itu, *Google Apps Script* mendukung pengembangan aplikasi yang cepat dan mudah tanpa memerlukan keterampilan teknis mendalam, serta dapat mengotomatisasi pengelolaan data seperti pencatatan transaksi, perhitungan stok, dan pengelolaan hutang, sehingga mengurangi kesalahan dan meningkatkan efisiensi operasional toko. Dengan adanya pengembangan sistem informasi menggunakan *Apps Script*, diharapkan dapat meminimalisir waktu saat pengelolaan data dan juga mengurangi biaya serta memudahkan pengguna dalam mengakses dan mengelola data.

Dari latar belakang masalah yang telah diuraikan, hal ini membuat peneliti untuk mengajukan skripsi dengan judul “ Pengembangan Sistem Informasi menggunakan *Google Apps Script* untuk Toko Kelontong “. Sistem ini berisi beberapa fitur yang akan memudahkan

pengelolaan dalam transaksi toko kelontong dan juga biaya penggunaan sistem yang tidak terlalu banyak.

1.2 Rumusan Masalah

Berdasarkan Latar Belakang masalah yang telah diuraikan, maka peneliti merumuskan masalah yang akan dibahas yaitu “Bagaimana mengembangkan Sistem Informasi penjualan dan pengolahan data barang pada Toko Asih dengan memanfaatkan *Google Apps Script*”.

1.3 Batasan Masalah

Dalam penelitian ini agar pembuatan sistem informasi untuk Toko Asih dapat terselesaikan sesuai tujuan dan sasaran yang diharapkan, maka dibuat batasan masalah sebagai berikut:

- A. Sistem Informasi Toko Asih berbasis web ini dibuat menggunakan layanan Google yaitu Apps Script.
- B. Sistem ini dibuat untuk pengelolaan data pada Toko Asih, termasuk pengelolaan data penjualan, data barang, data pengadaan, laporan transaks, dan transaksi layanan BRILink.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu untuk mengembangkan sistem informasi yang dapat mengelola penjualan dan pengolahan data barang pada Toko Asih dengan menggunakan *Google Apps Script*.

1.5 Manfaat Penelitian

Dengan melakukan penelitian ini diharapkan dapat memberikan sejumlah manfaat yang signifikan, sebagai berikut:

1. Pemilik toko mendapat kemudahan untuk pengelolaan data barang lebih efisien.
2. Pemilik toko dapat meminimalisir waktu dan biaya untuk pengelolaan data yang lebih efektif.
3. Karyawan toko mendapat kemudahan untuk mencatat transaksi pada toko.
4. Memberikan informasi kepada pembeli terkait hutang piutang.

1.6 Metodologi Penelitian

Metode penelitian yang digunakan untuk membuat sistem informasi toko kelontong menggunakan *Google Apps Script* meliputi:

- A. Analisis Kebutuhan, yaitu melakukan analisis kebutuhan dengan mengidentifikasi masalah dan kebutuhan untuk perancangan sistem seperti *Use Case Diagram* yang menggambarkan interaksi pengguna dengan sistem, serta mengidentifikasi kebutuhan *input*, *output*, proses, dan antarmuka.
- B. Perancangan, yaitu melakukan sebuah perancangan sistem berdasarkan analisis kebutuhan seperti membuat *Activity Diagram*, Skema Basis Data, dan *User Interface*.
- C. Implementasi, yaitu dilakukan setelah melakukan perancangan sistem yang selanjutnya akan dilakukan pembuatan sistem ke dalam bahasa pemrograman
- D. Pengujian, yaitu digunakan untuk menguji sistem apakah berjalan sesuai harapan atau apakah masih ada yang error.

1.7 Sistematika Penelitian

Sistematika penulisan pada tugas akhir ini bertujuan untuk memudahkan dalam memahami isi dari laporan ini. Pada laporan ini terdiri dari berbagai pokok pembahasan, sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi pembahasan mengenai latar belakang dibuatnya sistem informasi pengelolaan data Toko Asih yang berbasis web, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi pembahasan mengenai teori teori yang berhubungan dengan penelitian yaitu Google Apps Script dan penjelasan mengenai tinjauan pustaka.

BAB III METODOLOGI PENELITIAN

Bab ini berisi pembahasan tentang analisis dan perancangan sistem, seperti use case diagram, activity diagram, perancangan perilaku, dan perancangan basis data.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi pembahasan mengenai hasil pembuatan sistem yang telah dilakukan. Pembahasan yang dilakukan seperti implementasi sistem dan pengujian sistem, serta perancangan antarmuka untuk sistem.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil dan analisis membangun sistem yang telah dilakukan, serta saran untuk perbaikan dan pengembangan pada sistem informasi pengelolaan Toko Asih ini untuk kedepannya.

BAB II

KAJIAN PUSTAKA

2.1 Landasan Teori

Menjelaskan tentang teori-teori dasar yang berkaitan dengan pembuatan sistem sebagai berikut:

2.1.1 Google Apps Script

Google Sheet menciptakan sebuah bahasa pemrograman yang disebut *Google Apps Script* (GAS). *Apps Script* merupakan cara mudah untuk menyatukan Google dan layanan web lainnya untuk membentuk satu aplikasi web yang kuat dan interaktif. *Google Apps Script* adalah platform pengkodean dan pengembangan aplikasi yang dibangun ke dalam *Google Apps*, bisa juga untuk menambahkan fungsionalitas ke *spreadsheet*, *Gmail*, *Sites*, dan layanan lain dari *Google*.

Google Apps Script dapat dibuat sebagai file mandiri di Drive, di dalam dokumen atau spreadsheet ini dikenal sebagai penampung, atau di *Google Sites*. Pada *Apps Script* menggunakan bahasa JavaScript versi 1.6 tetapi ada beberapa versi 1.7 dan 1.8 serta *Google Apps Script* memberikan akses mudah sehingga pengguna dapat mengotomatiskan tugas di seluruh produk *Google* dan layanan pihak ketiga (Ganapathy, 2016).

Kegunaan *Google Apps Script*:

1. Membuat fungsi pengguna khusus untuk *Google Sheet*.
2. Pengembangan Aplikasi ringan berbasis *Google Spreadsheet*.
3. Membuat aplikasi dalam bentuk makro sederhana.
4. Integrasi dengan berbagai produk dan layanan *Google* lainnya.
5. Berkomunikasi dengan database relasional berbasis *cloud* melalui layanan *Google JDBC* (Ferreira, 2014).

Google Apps Script ditujukan untuk Aplikasi Google, dan beroperasi dengan cara yang sama seperti Visual untuk aplikasi (VBA) pada aplikasi Microsoft Excel, untuk membantu mengotomatiskan aplikasi dan layanan *Google* lainnya (Ganapathy, 2016).

Pada *Google I/O* tahun 2013, *Google Apps Script* menerima pembaruan dalam menyajikan UI. Bukan cuma menggunakan bahasa *JavaScript* tetapi bisa menggunakan

HTML standar dan juga *Cascading Style Site* (CSS) untuk menampilkan halaman. Ada beberapa cara untuk membuat antarmuka pengguna di *Google Apps Script* dengan Layanan Ui

App yang lebih lama, sebagai gadget untuk *Google Sites*, dan menggunakan Layanan *HTML* (Ferreira, 2014)

Google Apps Script berjalan di server *Google*, sehingga tidak dapat berjalan terus menerus selama lebih dari enam menit mungkin akan berubah jika ada perubahan mengenai sistem *Apps Script* di masa mendatang (Ganapathy, 2016). *Google Apps Script* dapat terhubung ke produk atau layanan google lainnya, misalnya:

1. *Google Drive*, Sebuah aplikasi penyimpanan file untuk menyimpan dan menyinkronkan file di server *Google*. *Google Drive* dapat digunakan untuk menyimpan data yang diperlukan dalam pengembangan aplikasi, seperti file *HTML*, *CSS*, dan *JavaScript* (Ganapathy, 2016).
2. *Google Sites*, suatu produk pada *Google* yang digunakan untuk menampilkan UI pada *Google apps script* dan dapat menulis dan mengelola kode dari dalam konsol pengolahan site (Ferreira, 2014).
3. *Google Calendar*, *Google Apps Script* menyediakan akses untuk layanan kalender yang berfungsi untuk mengakses dan memodifikasi menggunakan *Class Calendar App*.
4. *Google Docs*, di *Google Apps Script* dapat digunakan untuk menambahkan atau membuat dokumen, memformat konten menggunakan *source code* yang ada dalam *Apps Script* (Ganapathy, 2016).

2.2 Tinjauan Pustaka

Dalam penyusunan skripsi ini, peneliti melakukan eksplorasi informasi dari studi-studi sebelumnya sebagai dasar perbandingan, mencakup metode-metode yang diterapkan, keunggulan, kelemahan, analisis penelitian dan Hasil dari penelitian.

2.2.1 *Google Apps Script* untuk *Smart City*

Studi yang dilakukan oleh Petrović et al. (2020) membahas pengembangan aplikasi *smart city* menggunakan *AppSheet* dan *Apps Script*. Penelitian ini bertujuan untuk menyajikan pendekatan terhadap perkembangan pesat aplikasi berbasis data untuk smart city, dengan fokus pada solusi teknologi inovatif seperti *Internet of Things* (IoT), kecerdasan buatan, dan teknologi blockchain yang membuat kota-kota di seluruh dunia semakin cerdas.

Tujuan utama penelitian ini adalah mengurangi waktu pengembangan aplikasi kota pintar dan memungkinkan penyampaian aplikasi seluler dengan cepat dan efisien. Manfaat utama dari penelitian ini adalah memberikan pendekatan yang dapat mengurangi waktu pengembangan, memungkinkan pakar domain membuat aplikasi seluler tanpa pengetahuan

pemrograman tingkat lanjut. Studi ini juga menyoroti penggunaan *Google Apps Script* dan *AppSheet* sebagai alat pengembangan, serta model prediksi konsumsi energi.

Metode penelitian mencakup teknik analisis data dan pemodelan, dengan pihak yang terlibat termasuk peneliti, layanan *Google Sheet*, dan *Apps Script*. Hasil evaluasi dari dua studi kasus, yaitu pengembangan aplikasi untuk memprediksi konsumsi energi dan pengelolaan limbah, menunjukkan keakuratan model prediksi dan peningkatan efisiensi melalui pendekatan yang diusulkan. Studi ini juga membahas potensi pendekatan untuk skenario kota pintar lainnya, seperti manajemen infrastruktur energi, cloud, dan komputasi cloud hybrid.

Hasil penelitian menunjukkan bahwa *Google Apps Script* dan *AppSheet* dapat digunakan bersama untuk membangun aplikasi mobile berbasis data dengan cepat dan mudah. Hal ini memungkinkan integrasi dengan berbagai layanan G Suite tanpa pengetahuan pemrograman tambahan, dan menciptakan aplikasi mobile canggih dengan antarmuka pengguna yang unggul (Petrovic et al., 2020).

2.2.2 *Google Apps Script* untuk Pendidikan

Studi yang dilakukan oleh (Khariono et al., 2021) membahas pembangunan sistem ujian berbasis file menggunakan bot Telegram dengan tujuan mempermudah pelaksanaan ujian tanpa biaya berlebih dibandingkan sistem ujian konvensional. Penelitian ini dilatarbelakangi oleh permasalahan dalam ujian konvensional yang memakan waktu lama untuk koreksi jawaban dan memerlukan biaya yang cukup besar. Metode penelitian melibatkan analisis masalah, analisis prosedur sistem, pembuatan *Google Spreadsheet*, pembuatan bot Telegram, dan pengkodean, dengan pendekatan eksperimental. Subjek penelitian melibatkan penulis dan mahasiswa dalam uji coba. Hasil penelitian menunjukkan bahwa sistem ini efektif, efisien, dan tidak memerlukan biaya berlebih.

Sementara itu, penelitian yang dilakukan oleh (Sopingi et al., 2023) membahas pengembangan integrasi sistem pembelajaran dengan *Google Classroom* melalui *Google Apps Script* menggunakan pendekatan Rational Unified Process (RUP) dengan konsep object-oriented. Tujuan penelitian ini adalah mengembangkan integrasi sistem pembelajaran dengan *Google Classroom* melalui *Google Apps Script* untuk mendukung implementasi *blended learning*. Metode penelitian ini praktis dengan pendekatan RUP dan fokus pada arsitektur. Hasil penelitian menunjukkan bahwa *Google Apps Script* efektif mengintegrasikan sistem pembelajaran dengan *Google Classroom*, meningkatkan efisiensi dan efektivitas *blended learning*.

(Rahman, 2020) melakukan penelitian tentang pengembangan sistem informasi penjadwalan menggunakan *Google Apps Script* di Politeknik Kampar. Penelitian ini dilatarbelakangi oleh permasalahan dalam proses penjadwalan di Politeknik Kampar yang masih menggunakan sistem manual. Tujuan penelitian ini adalah mengembangkan sistem informasi penjadwalan yang efektif dan efisien. Metode yang digunakan adalah metode Waterfall dengan subjek penelitian berupa penulis dan dukungan dari beberapa pihak. Hasil penelitian menunjukkan bahwa *Google Apps Script* efektif digunakan untuk membangun sistem penjadwalan yang membantu menyinkronkan informasi dan meminimalisir bentrokan pada jadwal mengajar.

Penelitian yang dilakukan oleh (Xu et al., 2021) membahas implementasi *Controlled Digital Lending* (CDL) sebagai solusi selama pandemi COVID-19 di NYU Shanghai Library menggunakan *Google Sheets* dan *Google Drive*. Tujuan penelitian ini adalah menjelaskan alasan dan cara NYU Shanghai Library menggunakan *Google Sheets* dan *Google Drive* untuk menerapkan CDL. Metode penelitian menggunakan pendekatan interpretatif tanpa data numerik atau analisis kuantitatif. Hasil penelitian menunjukkan bahwa *Google Drive* dan Spreadsheet menjadi dasar penerapan CDL dengan pengembangan *Google Apps Script* khusus, menunjukkan potensi *Google Apps Script* sebagai alat canggih untuk implementasi solusi pinjaman digital.

2.2.3 *Google Apps Script* untuk Bot Telegram dan *Push Notification*

(Prabowo et al., 2023), dalam penelitian mereka, fokus pada pembuatan bot Telegram untuk teknisi asuransi di PT.Telkom Akses Rajawali Kota Bandung dengan menggunakan metode Push Notification. Latar belakang penelitian muncul dari penurunan kualitas penanganan layanan gangguan jaringan wifi di wilayah Bandung Barat, khususnya pada pelanggan Indihome. Tujuan utama adalah mempermudah dan mempercepat komunikasi antara helpdesk, tim leader, dan teknisi asuransi dalam menangani tiket SQM.

Manfaatnya terletak pada efisiensi dan kecepatan komunikasi di antara pihak terkait, memungkinkan teknisi assurance melaporkan aktivitas pekerjaannya melalui bot Telegram yang terhubung dengan *Google Spreadsheet*. Metode yang digunakan melibatkan push notification yang menghubungkan web dengan bot Telegram, memungkinkan teknisi assurance mengirim tiket SQM pada web untuk otomatis diposting pada grup Telegram teknisi assurance. Beberapa pihak yang terlibat dalam penelitian ini mencakup helpdesk, tim leader, dan teknisi assurance, serta melibatkan layanan *Google Apps Script* dan aplikasi Telegram.

Analisis penelitian membahas bagaimana bot Telegram dapat memfasilitasi komunikasi di antara pihak terkait, sementara hasil penelitian menunjukkan bahwa *Google Apps Script* memiliki relevansi yang besar dalam pembuatan bot Telegram tersebut, berfungsi untuk menjalankan berbagai fungsi yang mendukung fitur pada bot Telegram, termasuk penyimpanan data laporan aktivitas pekerjaan pada *Google Spreadsheet*.

2.2.4 *Google Apps Script* untuk Monitoring Sistem Berbasis IoT

(Vallisree Sivathanu., 2023) membahas pengembangan sistem pemantauan penyimpanan biji-bijian makanan cerdas berbasis IoT untuk silo dengan menggunakan *Google Apps Script*. Latar belakang penelitian menyoroti kebutuhan untuk memantau kondisi penyimpanan biji-bijian makanan dalam silo secara real-time guna mencegah kerusakan dan kerugian. Tujuan penelitian adalah mengembangkan sistem pemantauan cerdas berbasis IoT untuk silo yang dapat memonitor suhu, kelembaban, dan tingkat karbon-dioksida dalam silo secara real-time, dilengkapi dengan prosedur darurat otomatis jika ada parameter yang melebihi nilai ambang batas maksimum.

Manfaatnya adalah membantu petani dan pengguna silo dalam memantau kondisi penyimpanan biji-bijian makanan secara efektif, mencegah kerusakan, dan mengurangi kerugian. Metode penelitian melibatkan pengembangan sistem berbasis IoT dengan pengumpulan data real-time menggunakan sensor di dalam silo. Subjek penelitian mencakup tim peneliti dari Departemen Teknik Elektronika dan Komunikasi, Geethanjali College of Engineering and Technology, Hyderabad, India, serta PriMera Scientific Engineering sebagai perusahaan yang terlibat dalam pengembangan sistem tersebut.

Analisis menyoroti inovasi sistem yang memberikan peringatan real-time melalui Gmail jika ada masalah dengan kondisi penyimpanan biji-bijian, meningkatkan efisiensi dan keamanan penyimpanan serta mengurangi kerugian. *Google Apps Script* memainkan peran kunci dalam pengembangan sistem ini sebagai alat untuk mengumpulkan data dan memberi peringatan kepada pengguna melalui email (Vallisree Sivathanu., 2023).

2.2.5 *Google Apps Script* untuk Integrasi Layanan Eksternal

(Díaz et al., 2015) melakukan penelitian tentang integrasi layanan eksternal untuk penggunaan internal dalam organisasi menggunakan *Google Apps Script*. Penelitian ini dilakukan setelah proyek penelitian sebelumnya terkait manajemen sumber daya menggunakan *Google Apps Script* untuk konsumsi internal bisnis. Tujuan penelitian adalah mengeksplorasi potensi integrasi layanan eksternal dengan *Google Apps Script* dan memahami dampaknya terhadap efisiensi bisnis dan lingkungan. Manfaat penelitian ini melibatkan peningkatan

efisiensi bisnis, penghapusan sumber daya yang diduplikasi, dan kemudahan akses data secara real-time untuk memantau peluang bisnis, pembaruan data ekonomi, dan pelacakan tugas harian.

Metode penelitian mencakup pendekatan eksploratori dan studi kasus, dengan penulis tunggal, Karina Diaz Arrunategui, memainkan peran utama. Beberapa pihak terlibat dalam pengembangan solusi perangkat lunak yang terintegrasi dengan layanan pihak ketiga seperti *Google Apps Script*, *Google Sheets*, *Google Sites*, *JotForms*, *Cyfe*, dan *Insightly*. Penelitian menekankan pentingnya menggunakan perangkat lunak gratis dan mudah digunakan untuk mempermudah penggunaan dan pemeliharaan. Hasil penelitian menunjukkan relevansi *Google Apps Script* sebagai bahasa pemrograman yang memungkinkan integrasi layanan pihak ketiga dalam pengembangan solusi perangkat lunak (Díaz et al., 2015).

2.2.6 Google Apps Script untuk Cloud Computing

Airinei & Homocianu (2017) mengeksplorasi penggunaan *Google Apps Script* dan *cloud computing* dalam pembuatan aplikasi web yang dapat diinstal dan dapat diskalakan. Latar belakang penelitian menyoroti bahwa penggunaan istilah "*cloud*" dalam teknologi informasi dan komunikasi (TIK) biasanya terfokus pada penyimpanan file dan akses jarak jauh, tetapi penelitian ini fokus pada kolokasi komputasi awan.

Tujuan penelitian adalah memberikan contoh dan pertimbangan dalam pengembangan aplikasi web kecil dengan menggunakan bahasa scripting cloud Google yang bekerja di belakang *Google Drive*. Manfaatnya melibatkan memberikan wawasan kepada pengembang aplikasi web tentang penggunaan *Google Apps Script* dan *cloud computing*, termasuk keuntungan dan kerugian. Metode penelitian menggunakan pendekatan kualitatif dengan praktik langsung menggunakan produk Google dan API pihak ketiga.

Hasil penelitian menunjukkan bahwa *Google Apps Script* dapat digunakan untuk mengembangkan aplikasi web berbasis *cloud computing*, memberikan contoh aplikasi yang dapat dijalankan secara otomatis dan diakses melalui *URL* unik. Studi ini dapat membantu pengembang aplikasi web mempertimbangkan penggunaan *Google Apps Script* dalam pengembangan aplikasi yang efektif dan efisien.

2.2.7 Google Apps Script untuk Perpustakaan

Penelitian yang dilakukan oleh Thomas, A., & KP, M. S. (2019) bertujuan untuk memahami efektivitas dan kepuasan pengguna layanan Kesadaran Terkini (*Current Awareness Services*) di Central University of Kerala. Latar belakang penelitian menyoroti investasi besar universitas dalam sumber daya perpustakaan dan tanggung jawab perpustakaan untuk

memaksimalkan pemanfaatan sumber daya tersebut oleh pengguna. Tujuan penelitian mencakup pemahaman mendalam tentang efektivitas layanan, manfaat yang diperoleh peneliti, sejauh mana layanan membantu dalam menemukan informasi, evaluasi terhadap layanan yang ada, dan eksplorasi sistem pengiriman email di perpustakaan. Metode penelitian melibatkan distribusi kuesioner dan wawancara pribadi kepada peneliti di universitas untuk menggali pandangan mereka terhadap layanan. Manfaat penelitian ini mencakup pemahaman yang lebih baik tentang kepuasan pengguna dan efektivitas layanan Kesadaran Terkini, memberikan landasan untuk perbaikan layanan perpustakaan.

Hasil studi menunjukkan bahwa penerapan *Google App Scripts* dalam email secara signifikan meningkatkan efisiensi dan efektivitas layanan, memungkinkan pengiriman email otomatis yang disesuaikan dengan minat penelitian, menciptakan layanan yang lebih personal dan tepat waktu. Selain itu, integrasi yang mudah dengan produk Google lainnya, seperti *Google Spreadsheet* dan *Google Formulir*, menyederhanakan proses penyediaan layanan Kesadaran Terkini (Ashok et al., 2019)

2.2.8 Kesimpulan Tinjauan Pustaka

Berdasarkan berbagai penelitian yang telah diulas, dapat disimpulkan bahwa *Google Apps Script* adalah alat yang sangat fleksibel dan efektif untuk mengembangkan sistem berbasis web yang sederhana, efisien, dan terintegrasi secara optimal dengan layanan Google lainnya. Dalam berbagai bidang seperti *smart city*, pendidikan, integrasi layanan eksternal, dan monitoring berbasis IoT, *Google Apps Script* terbukti mampu mendukung pengembangan aplikasi dengan cepat, ringan, dan terjangkau. Setiap studi yang dikaji menunjukkan bahwa *Google Apps Script* memungkinkan pengembangan sistem dengan lebih mudah, mengurangi kebutuhan akan keterampilan pemrograman tingkat lanjut, serta mendukung integrasi dengan layanan pihak ketiga dan penyimpanan berbasis cloud, sehingga dapat menciptakan solusi yang praktis dan relevan.

Lebih lanjut, penelitian-penelitian terdahulu juga mengindikasikan bahwa penggunaan *Google Apps Script* dapat secara signifikan meningkatkan efisiensi dan efektivitas dalam berbagai konteks aplikasi, seperti manajemen data, penjadwalan, otomatisasi notifikasi melalui bot, serta pemantauan data secara real-time. Hal ini sangat relevan dengan penelitian yang dilakukan, yang bertujuan untuk mengembangkan sistem informasi guna pengelolaan penjualan dan pengolahan data barang di Toko Asih. Dengan mengadopsi *Google Apps Script*, sistem informasi yang dirancang diharapkan mampu menghadirkan kemudahan pengelolaan data secara terpusat dan meningkatkan efisiensi operasional.

Selain itu, studi-studi sebelumnya yang telah berhasil mengintegrasikan *Google Apps Script* dengan layanan seperti *Google Sheets*, bot Telegram, dan perangkat IoT memberikan pedoman teknis yang berharga serta inspirasi untuk pengembangan sistem ini. Pemanfaatan integrasi ini memungkinkan desain pemodelan data dan analisis yang mendukung pengelolaan inventaris secara akurat dan layanan transaksi yang andal. Dengan mempertimbangkan keberhasilan penerapan *Google Apps Script* dalam studi-studi sebelumnya, kesimpulan ini mengukuhkan keputusan untuk menggunakan *Google Apps Script* sebagai platform utama dalam membangun sistem informasi yang efisien, mudah diakses, dan relevan untuk kebutuhan pengelolaan toko kelontong.

BAB III

METODOLOGI PENELITIAN

3.1 Analisis Kebutuhan

Dalam proses analisis kebutuhan sistem, akan diuraikan kebutuhan yang perlu diimplementasikan pada sistem yang sedang dikembangkan. Sistem ini dapat digunakan oleh pengguna yang sudah terdaftar pada sistem. Kategori pengguna melibatkan pemilik toko (owner), admin, dan kasir (karyawan) dengan memiliki hak akses. Pembagian hak akses ini dilakukan berdasarkan peran dan tanggung jawab untuk memastikan bahwa data dan informasi dalam sistem dikelola dengan baik dan tidak terjadi kesalahan atau penyalahgunaan akses.

3.1.1 Analisis Proses Bisnis

Toko Asih merupakan toko kelontong yang melayani berbagai kebutuhan barang sehari-hari bagi masyarakat, serta menyediakan layanan tambahan berupa transaksi keuangan melalui BRILink dan pembayaran talangan tagihan PDAM dan listrik. Alur bisnis utama dimulai dengan pelanggan yang datang ke toko untuk membeli barang. Pelanggan yang ingin membeli barang akan langsung memilih dari rak dan membawa barang tersebut ke meja kasir untuk dilakukan pembayaran. Pembayaran dapat dilakukan secara tunai atau dengan metode non-tunai sesuai kebijakan toko. Selain itu, bagi pelanggan yang ingin membeli barang dengan cara kredit atau hutang, mereka dapat menginformasikan kepada kasir, yang kemudian akan mencatat transaksi tersebut dalam sistem pencatatan manual atau buku hutang. Setiap pembayaran cicilan hutang juga akan dicatat dan dipantau untuk memastikan pembayaran tepat waktu.

Selain melayani pembelian barang, Toko Asih juga berfungsi sebagai agen BRILink yang memungkinkan pelanggan untuk melakukan transaksi setor dan tarik tunai melalui sistem yang terintegrasi dengan bank. Pelanggan yang ingin melakukan setor atau tarik tunai akan langsung mendatangi kasir, di mana kasir akan memverifikasi identitas dan memproses transaksi menggunakan sistem BRILink yang terhubung dengan server bank. Setelah transaksi selesai, pelanggan akan menerima bukti transaksi berupa struk. Layanan tambahan lainnya yang disediakan oleh Toko Asih adalah pembayaran talangan untuk tagihan PDAM dan listrik. Sebelum melakukan penalangan pembayaran pelanggan memberikan Nomor pelanggan untuk PDAM dan memberikan Nomor meter / ID pelanggan untuk listrik ke pemilik Toko Asih. Pelanggan yang ingin melakukan pembayaran tagihan tersebut dapat meminta untuk menalangi pembayaran dengan mencatat informasi tagihan yang perlu dibayar. Pembayaran talangan ini

dicatat dalam pembukuan toko dan pelanggan dapat melakukan pelunasan dalam jangka waktu satu bulan kedepan hingga waktu tagihan dibulan selanjutnya. Untuk pembayaran talangan tagihan PDAM dan listrik akan otomatis dibayarkan oleh pemilik setiap bulannya. Sehingga pelanggan masih bersedia untuk ditalangin hanya datang ke toko untuk malukan pembayaran yang ada dalam struk bukti transaksi.

Setelah pelanggan melakukan transaksi pembelian barang atau menggunakan layanan lainnya, proses pendataan di Toko Asih melibatkan peran aktif dari admin. Admin bertanggung jawab untuk mengelola dan memperbarui data transaksi barang. Setiap barang yang masuk dan keluar akan dicatat oleh admin untuk pengelolaan data barang. Barang didapat dari kulakan dari toko grosir, yang akan dijual kembali di Toko Asih. Untuk beberapa barang juga ada yang diambil dari supplier yang datang langsung ke toko dan akan langsung dicatat oleh admin.

Untuk meningkatkan efisiensi operasional dan meminimalkan kesalahan manual, Google Apps Script digunakan untuk mengotomatisasi proses utama dalam bisnis Toko Asih. Misalnya, Apps Script mencatat transaksi secara otomatis ke dalam Google Sheets, memperbarui stok barang, dan menghasilkan laporan bulanan. Teknologi ini juga memungkinkan pencatatan penalangan PDAM dan listrik secara real-time, sehingga memastikan bahwa semua data transaksi tercatat dengan baik.

3.1.2 Use Case Diagram

Sistem ini memberikan beberapa hak akses yang berbeda kepada pengguna. Pemilik toko memiliki hak akses sepenuhnya untuk mengelola seluruh sistem, sementara admin merupakan karyawan yang bertugas dalam pengelolaan data barang. Kasir, sebagai pengguna lain, bertanggung jawab atas transaksi penjualan baik barang maupun BRILink.

Terdapat beberapa hak akses sistem yang tidak semua pengguna dapat mengakses beberapa fitur dalam sistem. Untuk admin, hak aksesnya terbatas pada pengelolaan data barang, ketersediaan barang, kelola harga beli dan jual, serta memasukkan transaksi pengadaan barang dan penjualan barang melalui Google Sheet.

Kasir, sebagai pengguna dengan hak akses terbatas, dapat memasukkan transaksi pegadaian barang dan penjualan barang melalui sistem. Selain itu, kasir juga dapat melihat harga jual barang, jumlah barang, melayani transaksi BRILink, dan memasukkan hutang pembeli.



Gambar 3.1 Use Case Diagram

3.1.3 Deskripsi Use Case

Dalam Deskripsi *Use Case* berisi penjelasan mengenai bagaimana suatu sistem digunakan dalam situasi atau fungsi tertentu. Aktor pada sistem ini melibatkan pemilik, admin, dan juga kasir. Setiap aktor memiliki *use case* masing masing dan hak akses serta fungsi sendiri sendiri.

Pemilik

Tabel 3.1 Deskripsi Use Case Pemilik

Use Case (UC)	Keterangan	Layanan Google
Menambah, Edit, Hapus Data Barang	Pemilik mengelola dan memonitoring data barang.	Gsheet

Mengelola data pengguna	Pemilik mengelola data pengguna (admin, kasir)	Gsheet
Insert transaksi penjualan barang	Pemilik dapat mengelola total penjualan barang yang akan di rekapitulasi per harinya	<i>Apps Script</i>
Insert Hutang pembayaran PDAM dan Listrik	Pemilik mencatat penalangan PDAM dan listrik, berbeda dengan hutang pembeli yang jarang terjadi. Penalangan pembayaran PDAM dan listrik adalah tindakan menalangi atas nama orang tetap.	<i>Apps Script</i>
Melihat laporan Transaksi	Pemilik melihat laporan transaksi per hari dengan transaksi penjualan barang dan BRILink	Gsheet/ <i>Apps Script</i>

Dalam Tabel 3.1 use case kelola data pengguna, mengelola data barang, dan melihat laporan transaksi yang menggunakan layanan *Google Sheet (Gsheet)* sebagai basis datanya untuk menyimpan dan mengelola data tersebut. Sedangkan use case mengelola transaksi penjualan barang dan insert hutang pembayaran PDAM dan listrik melalui sistem yang menggunakan layanan *Google Apps Script* untuk mengelola dan menyimpan data tersebut. Hal ini dikarenakan Pemilik bertanggung jawab untuk memastikan data pengguna terkelola dengan baik dan laporan transaksi tersedia untuk dianalisis. Sementara itu, Pemilik diberikan hak akses untuk menambahkan data atau mengubahnya pada insert penalangan tagihan PDAM dan Listrik, guna menjaga integritas dan keamanan data.

Admin

Tabel 3.2 Deskripsi *Use Case* Admin

<i>Use Case (UC)</i>	Keterangan	Layana Google
Menambah, Edit, Hapus Data Barang	Admin mengelola data barang (ketersediaan barang, harga beli dan jual)	Gsheet/ <i>Apps Script</i>

Mencatat Transaksi pengadaan barang	Admin dapat mencatat data keluar masuk ketersediaan barang	Gsheet/ <i>Apps Script</i>
Insert Transaksi Penjualan barang	Admin dapat mencatat data penjualan barang yang akan di rekapitulasi per harinya	<i>Apps Script</i>
Melihat harga jual dan jumlah barang	Admin dapat melihat informasi untuk harga jual dan jumlah barang	<i>Apps Script</i>

Dalam Tabel 3.2 use case kelola data barang dan melihat harga jual atau jumlah barang yang menggunakan layanan *Google Sheet* (Gsheet) sebagai basis datanya untuk menyimpan dan mengelola data tersebut. Sedangkan use case mencatat transaksi pengadaan barang dan mengelola transaksi penjualan barang melalui sistem yang menggunakan layanan *Google Apps Script* untuk mengelola dan menyimpan data tersebut. Hal ini dikarenakan Admin bertanggung jawab untuk mengelola data barang, yaitu hak untuk menambah, mengubah, atau menghapus informasi terkait barang. Pemberian hak akses penuh juga memungkinkan Admin untuk mengawasi dan mengontrol semua aspek sistem yang berhubungan dengan pengelolaan barang dan transaksi.

Kasir

Tabel 3.3 Deskripsi *Use Case* Kasir

<i>Use Case</i> (UC)	Keterangan	Layanan Google
insert transaksi penjualan barang	Kasir dapat mengelola total penjualan barang yang akan di rekapitulasi per harinya	<i>Apps Script</i>
Melihat harga jual dan jumlah barang	Kasir hanya dapat melihat stok barang dan harga jual	<i>Apps Script</i>
Melayani Transaksi BRILink	Kasir dapat mencatat transaksi BRILink (bukan transaksi penjualan barang) termasuk mencatat pelanggan yang sudah membayar hutang penyalangan PDAM dan listrik	<i>Apps Script</i>

Mencatat Hutang Pembeli	Kasir mencatat hutang pembeli (transaksi penjualan barang)	<i>Apps Script</i>
-------------------------	---	--------------------

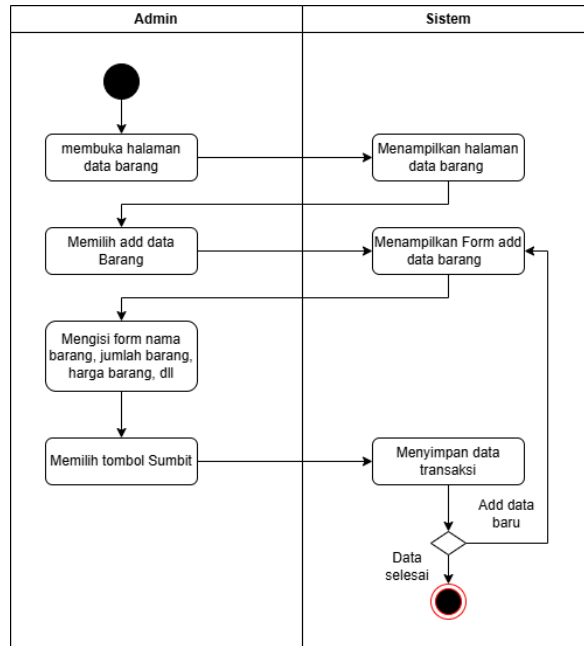
Dalam Tabel 3.3 keseluruhan *use case* menggunakan layanan *Google Apps Script* untuk mengelola dan menyimpan data dari use case insert transaksi penjualan barang, lihat harga jual dan jumlah barang, melayani transaksi BRILink, dan insert hutang pembeli. Hal ini dikarenakan kasir bertanggung jawab penuh dengan pada transaksi penjualan barang maupun transaksi BRILink. Pembatasan akses ini bertujuan untuk menjaga integritas data dan mencegah kesalahan atau penyalahgunaan, memastikan bahwa Kasir hanya dapat mengakses informasi yang relevan dengan tugas operasional mereka sehari-hari.

3.2 Perancangan Perilaku

Dalam perancangan perilaku, penulis menggunakan *Unified Modeling Language (UML)*. UML ini digunakan untuk gambaran awal sistem yang dapat menjelaskan interaksi antara objek dengan sistem yang akan dibangun. Diagram aktivitas memberikan gambaran mengenai alur kerja pada sistem dan menunjukkan urutan aktivitas antara sistem dan pengguna untuk deskripsi *use case* yang tidak menggunakan layanan *google sheet*.

3.2.1 Insert Transaksi Pengadaan Barang

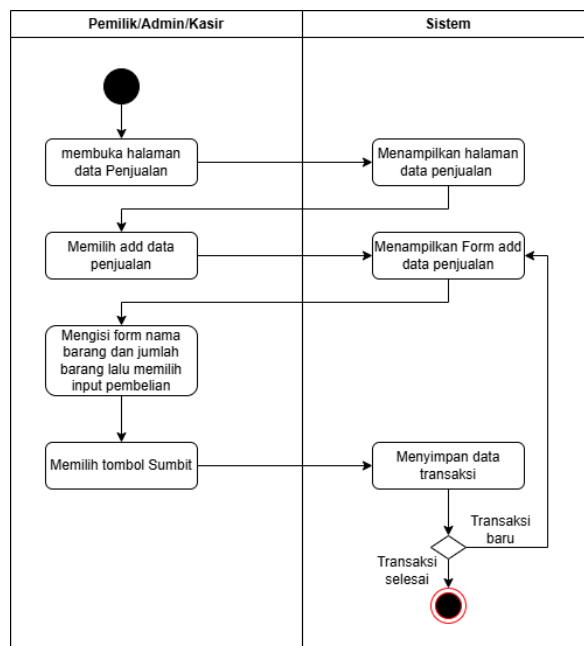
Pada *use case* ini, pengguna dapat melihat data yang sudah ada dan juga dapat menambahkan data baru. Untuk menambahkan data baru pengguna harus mengisi form yang tersedia dan melengkapi data baru. Supaya lebih jelas dapat melihat Diagram pada Gambar 3.2.



Gambar 3.2 Diagram aktivitas Insert Transaksi Pengadaan Barang

3.2.2 Insert Transaksi Penjualan Barang

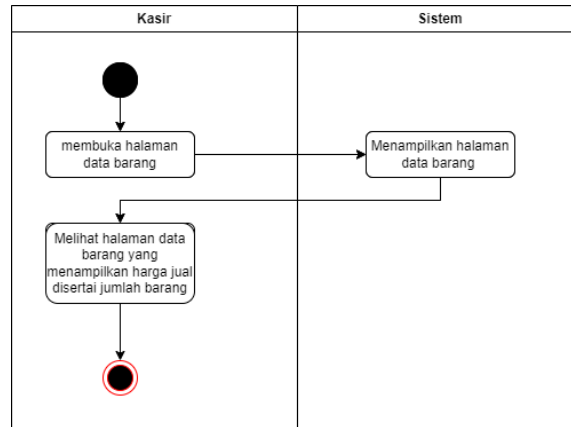
Pada *use case* ini, pengguna dapat melihat data yang sudah ada dan juga dapat menambahkan data baru. Untuk menambahkan data baru pengguna harus mengisi formulir yang tersedia dan melengkapi data baru. Supaya lebih jelas dapat melihat Diagram pada Gambar 3.3.



Gambar 3.3 Diagram aktivitas Insert Transaksi Penjualan Barang

3.2.3 Melihat Harga Jual dan Jumlah Barang

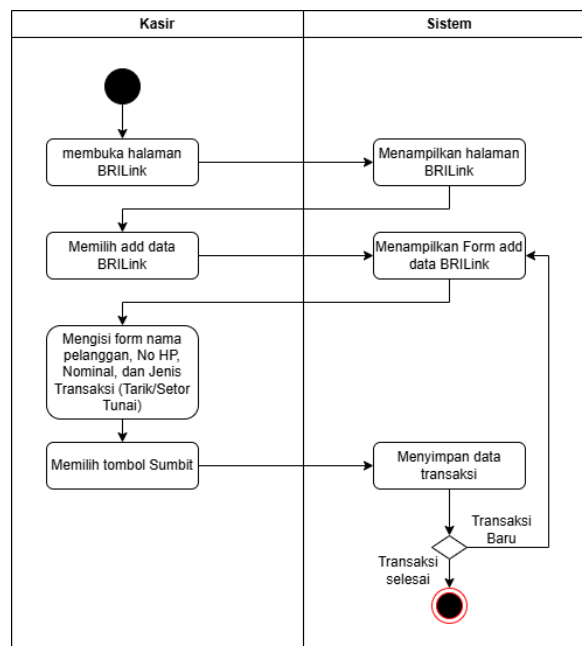
Pada *use case* ini, pengguna hanya dapat melihat data harga jual dan jumlah barang yang sudah tersedia. Supaya lebih jelas dapat melihat Diagram pada Gambar 3.4.



Gambar 3.4 Diagram aktivitas Melihat Harga Jual dan Jumlah Barang

3.2.4 Insert Transaksi BRILink

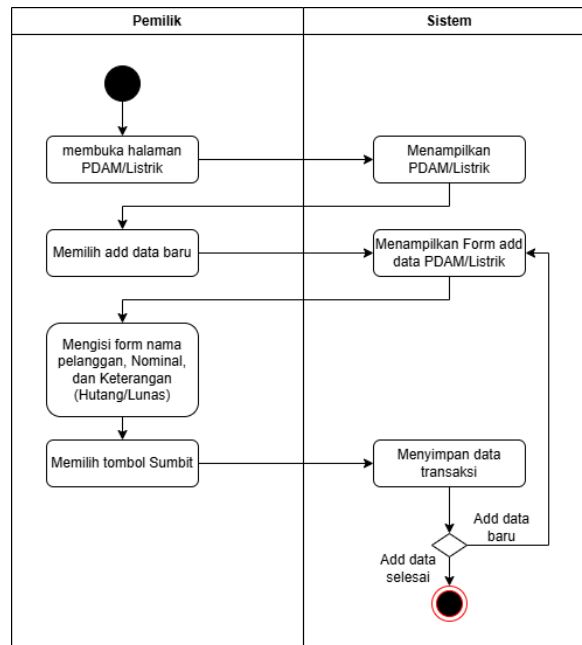
Pada *use case* ini, pengguna dapat melihat data yang sudah ada dan juga dapat menambahkan data baru. Pengguna dapat melakukan transaksi BRILink untuk setor tunai dan tarik tunai. Google Apps Script memastikan bahwa data ini tersimpan langsung dalam tabel di Google Sheets, yang kemudian dapat diakses oleh Pemilik untuk pelaporan. Supaya lebih jelas dapat melihat Diagram pada Gambar 3.5.



Gambar 3.5 Diagram aktivitas Transaksi BRILink

3.2.5 Insert Hutang PDAM dan Listrik

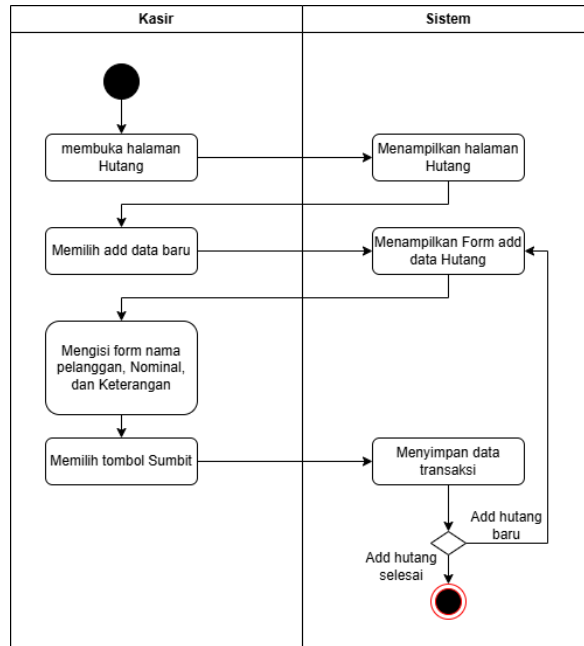
Pada *use case* ini, pengguna dapat melihat data yang sudah ada dan juga dapat menambahkan data baru. Pengguna dapat melakukan Insert Hutang PDAM dan Listrik untuk mencatat penalangan dan menyimpan datanya pada sistem. Supaya lebih jelas dapat melihat Diagram pada Gambar 3.6.



Gambar 3.6 Diagram aktivitas Insert Hutang PDAM dan Listrik

3.2.6 Insert Hutang Pembeli

Pada *use case* ini, pengguna melakukan pencatatan hutang pembeli. Pengguna juga dapat melihat detail hutang yang sudah di tambahkan. Supaya lebih jelas dapat melihat Diagram pada Gambar 3.7.



Gambar 3.7 Diagram aktivitas Hutang Pembeli

3.3 Analisis Tampilan

Penggunaan Google Apps Script dan Google Sheets dalam Toko Asih memberikan solusi yang efisien untuk mengelola proses bisnis yang berjalan. Google Apps Script memungkinkan otomatisasi pencatatan transaksi, pengelolaan stok barang, dan pencatatan pembayaran talangan tagihan PDAM serta listrik secara real-time, mengurangi kesalahan manual dan memastikan semua data tercatat dengan akurat dan tepat waktu. Pencatatan transaksi yang terintegrasi langsung dengan Google Sheets memungkinkan admin dan pemilik toko untuk memantau dan mengelola data transaksi dengan mudah tanpa perlu melakukan pencatatan manual. Perbedaan penyajian informasi di Halaman Web dan Google Sheets muncul karena setiap jenis data memiliki tujuan dan penggunaannya masing-masing. Informasi di Halaman Web biasanya digunakan langsung oleh pengguna aplikasi, seperti data transaksi atau informasi yang perlu diperbarui secara real-time. Sebaliknya, data di Google Sheets cenderung bersifat pendukung, seperti data pelanggan.

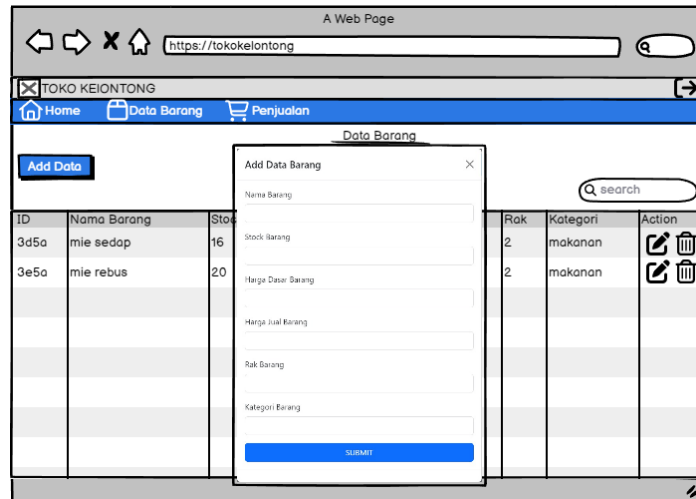
3.4 Perancangan Antarmuka

Rancangan antarmuka merupakan representasi visual dari desain tampilan pada sistem yang bertujuan untuk memudahkan proses pengembangan. Desain antarmuka memberikan gambaran visual mengenai penataan elemen-elemen antarmuka, seperti menu, tombol, dan area konten.

Rancangan ini dirancang untuk meningkatkan kemudahan penggunaan sistem serta memastikan keteraturan dan konsistensi dalam tampilan. Melalui gambar tersebut,

pengembang dapat lebih mudah memahami bagaimana setiap elemen interaktif berinteraksi dan diatur dalam konteks sistem yang sedang dikembangkan.

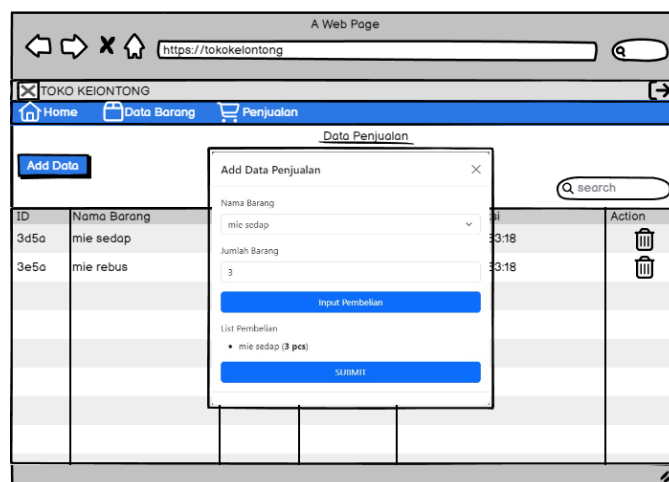
3.4.1 Antarmuka *Use Case* Insert Transaksi Pengadaan Barang



Gambar 3.8 Transaksi Pengadaan Barang

Gambar 3.8 merupakan rancangan antarmuka halaman “transaksi pengadaan barang” yang digunakan untuk admin pada sistem. Pada halaman ini terdapat form transaksi pengadaan barang yang berfungsi untuk menambah transaksi pengadaan barang baru oleh admin.

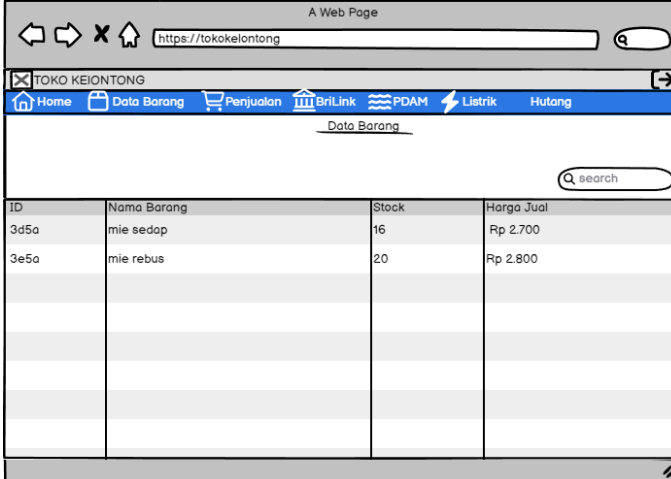
3.4.2 Antarmuka *Use Case* Insert Transaksi Penjualan Barang



Gambar 3.9 Transaksi Penjualan Barang

Gambar 3.9 merupakan rancangan antarmuka halaman “transaksi penjualan barang” yang digunakan untuk kasir pada sistem. Pada halaman ini terdapat form transaksi barang yang digunakan untuk menambah transaksi barang oleh kasir.

3.4.3 Antarmuka Use Case Lihat Harga Jual dan Jumlah Barang

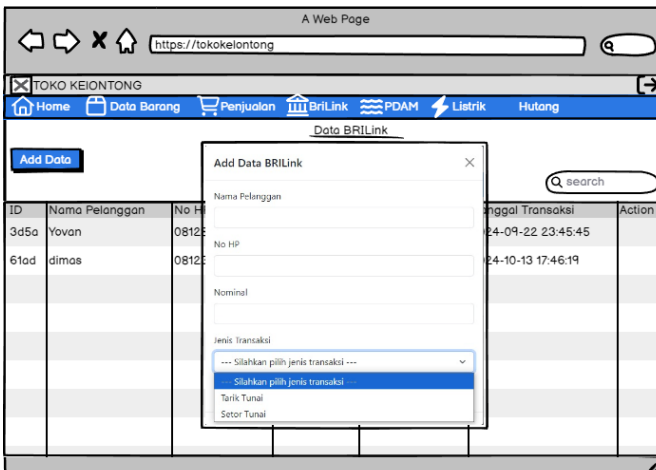


ID	Nama Barang	Stock	Harga Jual
3d5a	mie sedap	16	Rp 2.700
3e5a	mie rebus	20	Rp 2.800

Gambar 3.10 Melihat Harga dan Stock

Gambar 3.10 merupakan antarmuka halaman yang digunakan kasir untuk mengakses bagian harga barang. Pada bagian ini berfungsi untuk melihat harga dan stok barang yang tersedia di toko.

3.4.4 Antarmuka Use Case Transaksi BRILink

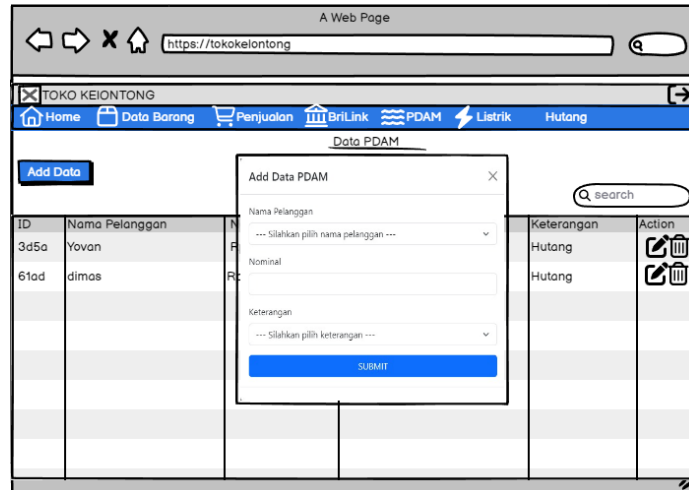


ID	Nama Pelanggan	No HP	Tanggal Transaksi	Action
3d5a	Yovan	08123	24-09-22 23:45:45	
61ad	dimas	08123	24-10-13 17:46:19	

Gambar 3.11 Transaksi BRILink

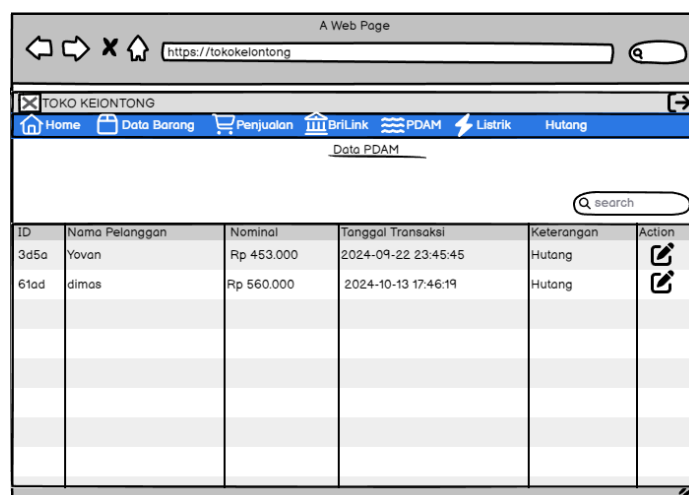
Gambar 3.11 merupakan rancangan antarmuka halaman “transaksi BRILink” yang digunakan untuk kasir pada sistem. Pada halaman ini terdapat form transaksi Setor/Tarik Tunai yang digunakan untuk menambah transaksi oleh kasir.

3.4.5 Antarmuka Use Case Insert Data PDAM



Gambar 3.12 Insert Data PDAM

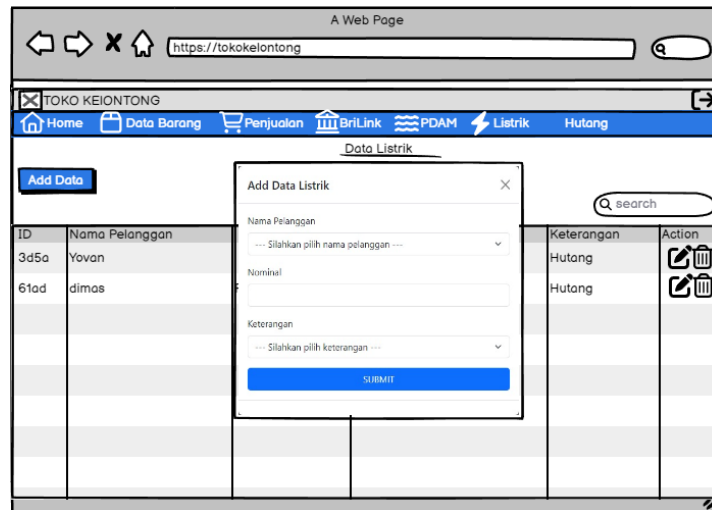
Gambar 3.12 merupakan rancangan antarmuka halaman “PDAM” yang digunakan untuk pemilik pada sistem. Pada halaman ini pemilik bisa memasukan data dengan mengisi formulir yang tersedia. Semua transaksi akan tercatat dan tersimpan secara otomatis dalam sistem ini.



Gambar 3.13 Data PDAM

Gambar 3.13 merupakan rancangan antarmuka halaman “PDAM” yang digunakan untuk kasir dalam sistem. Pada halaman ini, kasir hanya bisa melihat data dan mengedit untuk pelanggan yang sudah membayar hutangnya. Semua transaksi ini akan tercatat dan tersimpan secara otomatis dalam laporan ini.

3.4.6 Antarmuka *Use Case* Insert Data Listrik



Gambar 3.14 Insert Data Listrik

Gambar 3.14 merupakan rancangan antarmuka halaman “Listrik” yang digunakan untuk pemilik pada sistem. Pada halaman ini pemilik bisa memasukan data dengan mengisi formulir yang tersedia. Semua transaksi akan tercatat dan tersimpan secara otomatis dalam sistem ini.

ID	Nama Pelanggan	Nominal	Tanggal Transaksi	Keterangan	Action
3d5a	Yovan	Rp 453.000	2024-09-22 23:45:45	Hutang	
61ad	dimas	Rp 560.000	2024-10-13 17:46:19	Hutang	

Gambar 3.15 Data Listrik

Gambar 3.15 merupakan rancangan antarmuka halaman “Listrik” yang digunakan untuk kasir dalam sistem. Pada halaman ini, kasir hanya bisa melihat data dan mengedit untuk pelanggan yang sudah membayar hutangnya. Semua transaksi ini akan tercatat dan tersimpan secara otomatis dalam laporan ini.

3.4.7 Antarmuka Use Case Insert Hutang Pembeli

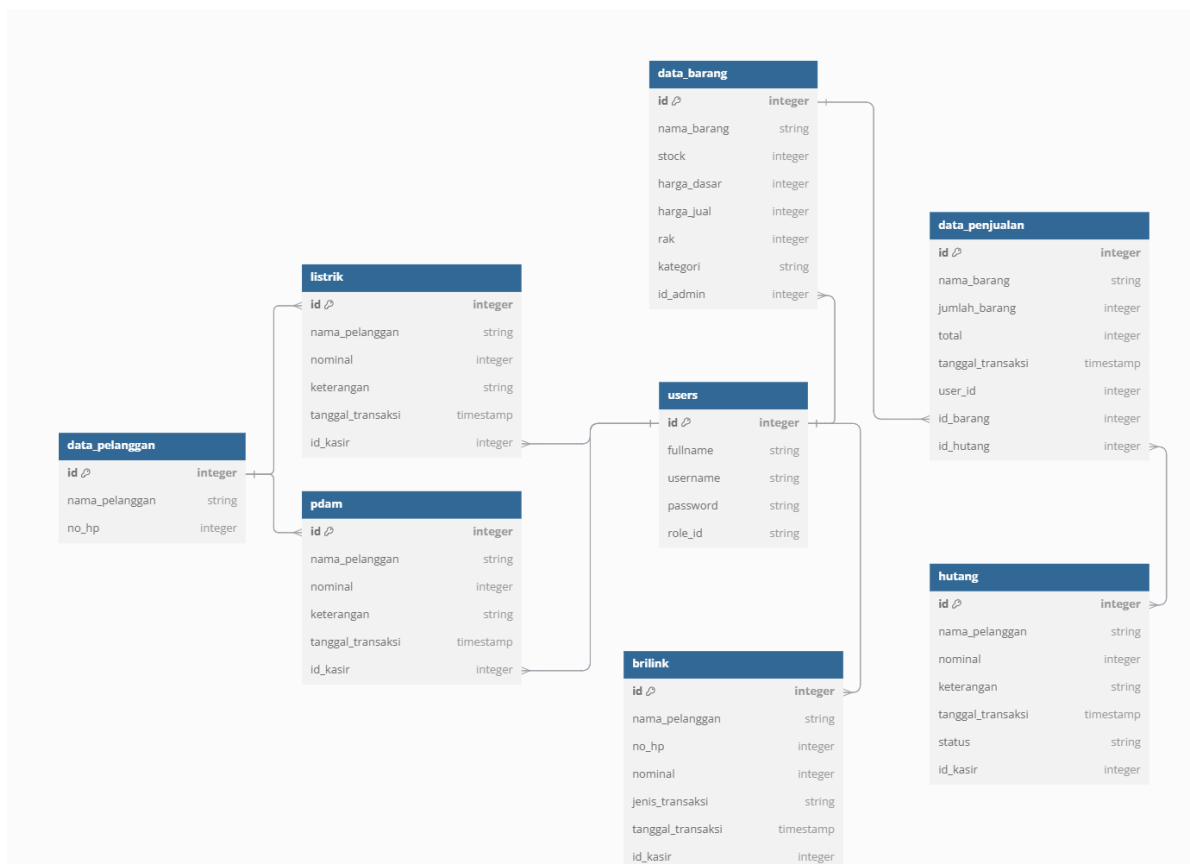
ID	Nama Pelanggan	Nominal	Tanggal Transaksi	action
3d5a	Yovan	Rp 453.000	4-09-22 23:45:45	
61ad	dimas	Rp 560.000	4-10-13 17:46:19	

Gambar 3.16 Hutang Pembeli

Gambar 3.16 merupakan rancangan antarmuka halaman “hutang pembeli” yang digunakan untuk kasir pada sistem. Pada halaman ini kasir hanya bisa memasukan data dengan form yang tersedia. Semua transaksi ini akan tercatat dan tersimpan langsung ke dalam sistem ini.

3.5 Perancangan Basis Data

Pada pembangunan sistem , terdapat basis data yang akan digunakan. Basis data pada sistem yang akan dibangun ini memiliki delapan tabel yaitu, tabel “user”, tabel “data_barang”, tabel “data_penjualan”, tabel “pdam”, tabel “listrik”, tabel “brilink”, tabel “pelanggan”, dan juga tabel “hutang”.



Gambar 3.17 Relasi Tabel

Gambar 3.17 Merupakan Perancangan Basis Data dalam sistem ini melibatkan beberapa tabel yang saling terkait untuk mengelola data transaksi dan pengguna. Tabel users berfungsi untuk menyimpan data pengguna yang dapat mengakses sistem. Dalam tabel ini, terdapat atribut role yang mengatur tingkat hak akses pengguna, yang dibagi menjadi tiga level, yaitu Pemilik, Admin, dan Kasir. Pembagian role ini penting untuk mengatur akses ke berbagai fitur dalam sistem dan menjaga keamanan data.

Tabel *brilink* digunakan untuk mencatat data transaksi setor tunai atau transfer melalui layanan BRILink. Pada tabel ini, hanya role Kasir yang diberikan hak akses untuk mencatat dan memverifikasi transaksi. Hal ini bertujuan untuk membatasi pengelolaan transaksi BRILink hanya pada pihak yang bertanggung jawab dalam proses tersebut.

Tabel *data_penjualan* mencatat data transaksi penjualan barang. Semua role (Pemilik, Admin, dan Kasir) memiliki hak akses terhadap tabel ini. Tabel ini memiliki atribut *id_barang*, yang menghubungkannya dengan tabel *barang*, sehingga informasi terkait barang yang terjual dapat dipanggil dan ditampilkan secara otomatis.

Tabel *hutang* menyimpan data pelanggan yang memiliki hutang saat melakukan transaksi penjualan. Hanya role Kasir yang memiliki hak akses untuk mencatat atau memperbarui data hutang pembeli, memastikan bahwa hanya pihak yang berwenang yang dapat menangani pembayaran dan pencatatan hutang pelanggan.

Tabel *pdam* dan *listrik* digunakan untuk mencatat data transaksi pembayaran untuk layanan PDAM dan listrik. Kedua tabel ini hanya dapat diakses oleh role Pemilik untuk memastikan bahwa transaksi terkait dengan tagihan layanan hanya dapat dikelola oleh pihak yang memiliki kewenangan penuh terhadap data ini. Setiap tabel ini memiliki atribut *id_pelanggan*, yang merujuk pada tabel *data_pelanggan* untuk memanggil informasi nama pelanggan yang terkait dengan transaksi pembayaran.

Secara keseluruhan, sistem ini memanfaatkan kontrol akses berbasis peran untuk menjaga keamanan dan integritas data. Pembagian hak akses ini memastikan bahwa setiap pengguna hanya dapat mengakses data dan fitur yang relevan dengan peran mereka, sehingga meminimalkan risiko kesalahan atau penyalahgunaan data. Penggunaan relasi antar tabel, seperti *id_barang* dan *id_pelanggan*, juga memastikan konsistensi data dan kemudahan dalam mengelola transaksi di seluruh sistem.

3.6 Perancangan Sistem dengan Google Apps Script

Google Apps Script dirancang sebagai komponen utama untuk mendukung otomatisasi dan pengelolaan data dalam sistem ini. Dengan kemampuan integrasinya yang erat dengan Google Sheets, Apps Script digunakan untuk menangani berbagai kebutuhan penting dalam sistem, seperti pembaruan stok barang, pembuatan laporan otomatis, validasi data pelanggan, dan pencatatan hutang. Salah satu rancangan utama adalah fitur pembaruan stok barang otomatis, di mana Apps Script membaca data transaksi dari tabel *data_penjualan* dan secara otomatis

memperbarui jumlah stok di tabel data_barang. Selain itu, pembuatan laporan harian yang juga akan di rekap setiap bulannya juga dirancang menggunakan Apps Script, yang secara otomatis mengolah data dari tabel data_penjualan dan brilink untuk menghasilkan laporan ringkasan transaksi harian. Laporan ini kemudian dilihat langsung oleh pemilik dalam sistem.

Untuk menjaga keakuratan data, validasi data pelanggan juga menjadi bagian penting dalam perancangan Apps Script. Skrip ini memastikan bahwa informasi pelanggan yang dicatat, terutama untuk penalangan PDAM atau listrik, telah sesuai dengan data yang ada di tabel data_pelanggan. Terakhir, Apps Script juga dirancang untuk mendukung pengelolaan hak akses pengguna berdasarkan peran (role). Berdasarkan data di tabel users, Apps Script memastikan bahwa hanya pengguna dengan hak akses tertentu yang dapat menjalankan fitur yang relevan.

Rancangan sistem ini memastikan bahwa Google Apps Script menjadi solusi utama yang mendukung efisiensi, keamanan, dan keakuratan dalam pengelolaan data toko. Dengan fitur-fitur otomatisasi yang dirancang secara real-time, sistem ini diharapkan dapat meminimalkan kesalahan manual dan meningkatkan kinerja operasional.

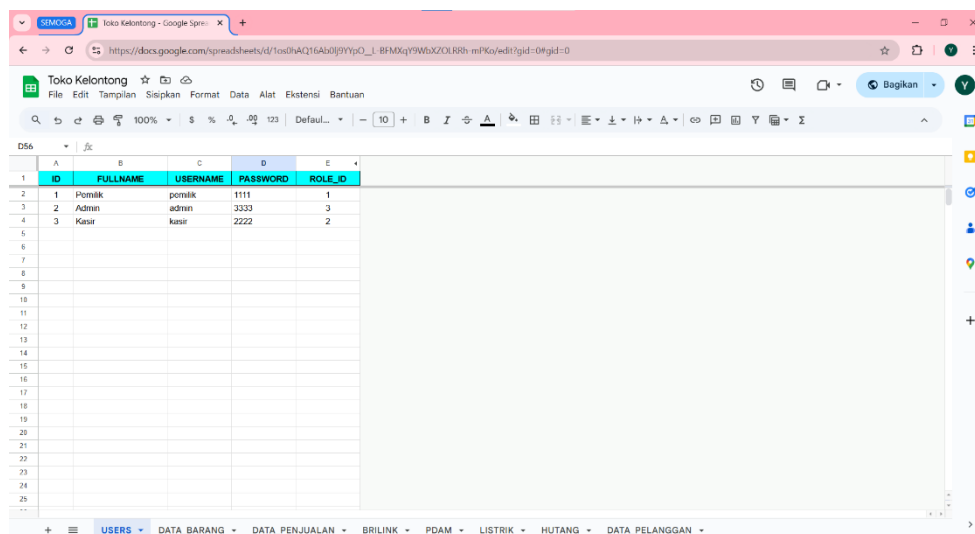
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Implementasi Basis Data

Implementasi basis data pada sistem informasi pengelolaan Toko Asih ini memanfaatkan Google Sheets sebagai media penyimpanan utama. Pemilihan Google Sheets didasarkan pada kemudahannya dalam memberikan akses kepada pengguna untuk mengelola, membaca, mengedit, dan menghapus data secara real-time. Selain itu, penggunaan Google Sheets bertujuan untuk mempermudah interaksi antara pengguna dengan sistem melalui antarmuka yang familiar dan mudah diakses. Hak akses pada Google Sheets diatur sesuai dengan peran masing-masing pengguna yang telah ditentukan sebelumnya. Berikut ini implementasi basis data sistem:



ID	FULLNAME	USERNAME	PASSWORD	ROLE_ID
1	Pomik	pomik	1111	1
2	Adnan	adnan	3333	3
3	Kasir	kasir	2222	2

Gambar 4.1 *Sheet users*

Pada Gambar 4.1 Sheet USERS dalam basis data sistem informasi Toko Asih berfungsi sebagai tempat penyimpanan informasi pengguna sistem. Setiap baris pada sheet ini merepresentasikan satu pengguna dengan atribut utama, yaitu ID sebagai identitas unik, FULLNAME untuk nama lengkap pengguna, USERNAME sebagai nama pengguna untuk login, PASSWORD sebagai kata sandi untuk otentikasi, dan ROLE_ID yang menentukan peran atau hak akses pengguna dalam sistem.

Sheet ini memiliki relasi penting dengan tabel lain dalam basis data, seperti tabel `data_barang`, di mana kolom `user_id` pada tabel tersebut merujuk ke kolom ID dalam sheet `USERS`, sehingga memungkinkan identifikasi pengguna yang mengelola data barang. Selain itu, sheet ini juga berhubungan dengan tabel-tabel transaksi seperti `listrik`, `pdam`, dan `brilink` melalui kolom `id_kasir`, yang mencatat pengguna (kasir) yang melakukan transaksi. Dengan demikian, sheet `USERS` memainkan peran sentral dalam autentikasi, otorisasi, dan pengelolaan hak akses pengguna, memastikan bahwa setiap peran memiliki akses yang sesuai ke modul sistem yang relevan.

ID	NAMA BARANG	STOCK	HARGA DASAR	HARGA JUAL	RAK	KATEGORI
59398	Kinyak Goreng 1L	11	18.500.00	18.000.00	1	Bahan Masak
aab3d	Mie Goreng	40	2.350.00	2.500.00	2	makanan
ea6b8	Airas Botol Besar	18	4.500.00	6.000.00	3	Minuman
8acaf	Makanan Ringan (Serenteng)	13	10.000.00	12.000.00	4	Makanan Ringan
7d9d0	Sampo Botol	23	14.000.00	15.000.00	5	Alat Mandi
cd9da	Sabun Sabung	11	4.500.00	5.000.00	5	Alat Mandi
2c920	Kopi Sachet	35	1.500.00	2.000.00	4	Minuman
36f88	Buku Tulis	11	2.300.00	3.000.00	6	Alat Tulis
e4151	Pensil 2B	6	4.500.00	5.500.00	6	Alat Tulis
a60ba	Pharmen (pcs)	35	150.00	200.00	0	Makanan
b7e51	Indomie rebus	60	2.350.00	2.550.00	2	makanan
f14fe	kopi abc	5	2.300.00	2.600.00	3	minuman

Gambar 4.2 Sheet `DATA_BARANG`

Pada Gambar 4.2 Sheet `DATA_BARANG` dalam sistem informasi Toko Asih berfungsi untuk mencatat data master barang yang tersedia di toko. Setiap baris dalam sheet ini merepresentasikan satu jenis barang dengan atribut utama, yaitu ID sebagai identitas unik barang, `NAMA_BARANG` untuk nama barang, `STOCK` yang menunjukkan jumlah stok barang yang tersedia, `HARGA_DASAR` sebagai harga pokok barang, `HARGA_JUAL` untuk harga penjualan barang kepada pelanggan, `RAK` yang menunjukkan lokasi barang di toko, dan `KATEGORI` untuk mengelompokkan barang berdasarkan jenisnya.

Sheet ini memiliki relasi penting dengan tabel `data_penjualan`, di mana atribut `NAMA_BARANG` digunakan untuk mencatat barang yang terjual dalam setiap transaksi penjualan. Relasi ini memungkinkan sistem untuk memotong jumlah stok secara otomatis berdasarkan transaksi yang terjadi, sehingga pengelolaan stok menjadi lebih efektif. Selain itu, `data_barang` juga memiliki relasi dengan tabel `users` melalui `id_admin` yang menunjukkan admin yang bertanggung jawab atas pengelolaan data barang.

ID	NAMA_BARANG	JUMLAH_BARANG	TOTAL	TANGGAL_TRANSAKSI
5b80d1	Parmen (pcs), Minyak Goreng 1L, Mie Goreng	8	24.000,00	2024-9-10 19:28:44
1844c	Pensil 2B, Minyak Goreng 1L	10	117.500,00	2024-8-10 22:21:18
1b9e08	Minyak Goreng 1L, Kaps Sachet, Sampo Botol	8	61.000,00	2024-10-08 19:54:36
cbec0	Aqua Botol Besar, Makanan Ringan (Serenteng)	5	48.000,00	2024-10-08 19:59:13
fb395	Sabun Batang, Kopi Sachet, Parmen (pcs)	6	9.600,00	2024-10-08 20:00:35
a91a0	Buku Tulis, Pensil 2B	3	11.500,00	2024-11-09 20:02:24
e9a0d	Minyak Goreng 1L, Mie Goreng	6	30.500,00	2024-11-09 20:10:29
84538	Makanan Ringan (Serenteng), Sampo Botol	3	42.000,00	2024-11-09 20:10:49
73a6a	Aqua Botol Besar	2	12.000,00	2024-11-09 21:58:35
5f88f	Pensil 2B, Buku Tulis	4	12.000,00	2024-11-10 19:27:42
41153	Sabun Batang, Sampo Botol, Aqua Botol Besar	4	41.000,00	2024-11-10 19:28:18
5b80d1	Parmen (pcs), Minyak Goreng 1L, Mie Goreng	8	24.000,00	2024-11-10 19:28:44
1844c	Pensil 2B, Minyak Goreng 1L	10	117.500,00	2024-11-10 22:21:18
91800	Minyak Goreng 1L	-3	-54.000,00	2024-11-11 14:53:42
1ae53	Aqua Botol Besar, Sabun Batang	4	22.000,00	2024-11-11 15:09:28
1ae53	Aqua Botol Besar, Sabun Batang	2,2	22.000,00	2024-11-11 15:09:28
f6881	Mie Goreng, Sampo Botol	5	37.500,00	2024-11-21 12:14:32
765c8	Minyak Goreng 1L, Pensil 2B	3	29.000,00	2024-11-21 12:38:28

Gambar 4.3 Sheet Data_Penjualan

Pada Gambar 4.3 Sheet DATA_PENJUALAN dalam sistem informasi Toko Asih berfungsi untuk mencatat data transaksi penjualan barang secara detail. Setiap baris pada sheet ini merepresentasikan satu transaksi, dengan atribut utama meliputi ID sebagai identitas unik untuk transaksi, NAMA_BARANG sebagai nama barang yang dijual, JUMLAH_BARANG yang menunjukkan kuantitas barang yang terjual, TOTAL sebagai total harga penjualan dalam transaksi tersebut, dan TANGGAL_TRANSAKSI yang mencatat waktu transaksi dilakukan.

Sheet ini berelasi langsung dengan tabel data_barang melalui atribut NAMA_BARANG, yang merujuk pada barang yang telah tercatat sebelumnya di dalam tabel barang, sehingga memungkinkan pelacakan stok dan harga. Selain itu, atribut id_hutang pada sheet ini terhubung dengan tabel HUTANG, yang mencatat informasi terkait transaksi kredit, seperti nama pelanggan, nominal hutang, keterangan, status pembayaran, dan tanggal transaksi. Jika atribut id_hutang terisi, berarti transaksi tersebut dilakukan secara kredit, sedangkan jika kosong, transaksi dilakukan secara tunai.

ID	NAMA_PELANGGAN	NO_HP	NOMINAL	JENIS_TRANSAKSI	TANGGAL_TRANSAKSI
5b49e	dimas	081288831757	540.000.00	Tarik Tunai	2024-08-23 09:36:42
46f60	Onik	08122736534	750.000.00	Setor Tunai	2024-10-30 01:02:28
b6b23	Yogi	08128738518	230.000.00	Setor Tunai	2024-10-30 01:03:02
600b3	lucky	08125346888	120.000.00	Tarik Tunai	2024-10-30 01:03:24
b6c48	jesse poliak	08142325672	328.000.00	Setor Tunai	2024-11-02 09:24:16
81177	Vitalik	081387358223	240.000.00	Setor Tunai	2024-11-08 20:13:30
88c88	Juko S	08138075376	100.000.00	Tarik Tunai	2024-11-11 09:20:41
77211	Juko P	081342573542	150.000.00	Setor Tunai	2024-11-11 09:21:04
45acc	jesse poliak	08137831577	300.000.00	Tarik Tunai	2024-11-11 14:58:54

Gambar 4.4 Sheet BRILINK

Pada Gambar 4.4 Sheet BRILINK dalam sistem informasi Toko Asih berfungsi untuk mencatat transaksi layanan Brilink yang dilakukan oleh toko. Setiap baris pada sheet ini merepresentasikan satu transaksi dengan atribut utama, yaitu ID sebagai identitas unik transaksi, NAMA_PELANGGAN sebagai nama pelanggan yang melakukan transaksi, NO_HP untuk nomor kontak pelanggan, NOMINAL sebagai jumlah uang yang ditransaksikan, JENIS_TRANSAKSI yang menunjukkan jenis transaksi (misalnya, Tarik Tunai atau Setor Tunai), dan TANGGAL_TRANSAKSI sebagai waktu transaksi dilakukan.

Meskipun sheet ini mencatat nama pelanggan dan nomor HP, data tersebut tidak memiliki relasi langsung dengan tabel DATA_PELANGGAN di dalam struktur basis data. Informasi pelanggan pada sheet BRILINK disimpan secara mandiri, terpisah dari tabel DATA_PELANGGAN, sehingga setiap transaksi Brilink berdiri sendiri tanpa keterkaitan formal dengan data pelanggan lain dalam sistem.

Namun, sheet ini tetap berelasi dengan tabel USERS melalui atribut id_kasir yang mencatat kasir atau pengguna sistem yang bertanggung jawab atas transaksi Brilink tersebut. Relasi ini memungkinkan pelacakan transaksi berdasarkan pengguna sistem. Dengan pendekatan ini, sheet BRILINK mendukung pencatatan dan pelaporan transaksi Brilink secara independen, tanpa mengandalkan tabel pelanggan, namun tetap dapat dimonitor oleh pengguna yang bertugas.

ID	NAMA_PELANGGAN	NOMINAL	KETERANGAN	TANGGAL_TRANSAKSI
810a8	Yovan	1,000,000.00	Lunas	2024-10-13 17:40:10
a5834	Dimas	750,000.00	Hutang	2024-10-13 17:40:22
88a7e	Harip	750,000.00	Hutang	2024-10-13 17:40:40
4a660	Ozek	750,000.00	Lunas	2024-10-13 17:40:54
a9633	Rindho	112,000.00	Hutang	2024-11-03 14:25:41
6a979	Peppy	132,000.00	Hutang	2024-11-08 20:16:44
994af	Yovan	5,455,000.00	Hutang	2024-11-09 21:52:29
1bb05	Pollak	467,000.00	Lunas	2024-11-10 22:24:51
694de	Harip	556,000.00	Lunas	2024-11-11 14:55:49

Gambar 4.5 Sheet PDAM

Pada Gambar 4.5 Sheet PDAM dalam sistem informasi Toko Asih digunakan untuk mencatat transaksi pembayaran layanan PDAM. Setiap baris dalam sheet ini merepresentasikan satu transaksi dengan atribut utama, yaitu ID sebagai identitas unik transaksi, NAMA_PELANGGAN untuk mencatat nama pelanggan yang melakukan transaksi, NOMINAL yang menunjukkan jumlah tagihan yang harus dibayar, KETERANGAN yang mencatat status pembayaran (misalnya, "Lunas" atau "Hutang"), dan TANGGAL_TRANSAKSI yang mencatat waktu transaksi dilakukan.

Nama pelanggan yang tercatat di kolom NAMA_PELANGGAN dalam sheet PDAM diambil dari tabel DATA_PELANGGAN, yang berfungsi untuk menyimpan informasi dasar pelanggan, seperti nama dan nomor HP. Oleh karena itu, data NAMA_PELANGGAN di sheet PDAM harus sesuai dengan yang ada di tabel DATA_PELANGGAN, sehingga transaksi pembayaran dapat terhubung langsung dengan informasi pelanggan yang tercatat dalam database.

Dengan adanya hubungan ini, sistem dapat memastikan bahwa setiap transaksi pembayaran PDAM yang tercatat di sheet PDAM berhubungan dengan pelanggan yang sudah terdaftar di tabel DATA_PELANGGAN, mempermudah pelacakan pembayaran, serta memperbarui status pembayaran pelanggan, apakah sudah lunas atau masih memiliki hutang. Relasi ini juga memungkinkan pemilik atau kasir untuk memverifikasi data pelanggan yang melakukan pembayaran dan mengelola status tagihan secara lebih efisien.

ID	NAMA_PELANGGAN	NOMINAL	KETERANGAN	TANGGAL_TRANSAKSI
0f8b9	Ozek	540,000.00	Lunas	2024-10-13 17:41:29
4fc1d	Hanip	5,455,000.00	Hutang	2024-10-13 17:41:39
74173	Yovan	19,000,000.00	Hutang	2024-10-13 17:41:47
81e4a	Dimas	350,000.00	Hutang	2024-10-30 01:17:18
0e972	Yovan	5,455,000.00	Lunas	2024-11-09 21:52:50
88863	Polak	120,000.00	Hutang	2024-11-11 09:23:59
c90e8	Ozek	250,000.00	Hutang	2024-11-11 14:57:16

Gambar 4.6 Sheet Listrik

Pada Gambar 4.6 Sheet LISTRIK dalam sistem informasi Toko Asih digunakan untuk mencatat transaksi pembayaran layanan Listrik. Setiap baris dalam sheet ini merepresentasikan satu transaksi dengan atribut utama, yaitu ID sebagai identitas unik transaksi, NAMA_PELANGGAN untuk mencatat nama pelanggan yang melakukan transaksi, NOMINAL yang menunjukkan jumlah tagihan yang harus dibayar, KETERANGAN yang mencatat status pembayaran (misalnya, "Lunas" atau "Hutang"), dan TANGGAL_TRANSAKSI yang mencatat waktu transaksi dilakukan.

Nama pelanggan yang tercatat di kolom NAMA_PELANGGAN dalam sheet LISTRIK diambil dari tabel DATA_PELANGGAN, yang berfungsi untuk menyimpan informasi dasar pelanggan, seperti nama dan nomor HP. Oleh karena itu, data NAMA_PELANGGAN di sheet PDAM harus sesuai dengan yang ada di tabel DATA_PELANGGAN, sehingga transaksi pembayaran dapat terhubung langsung dengan informasi pelanggan yang tercatat dalam database.

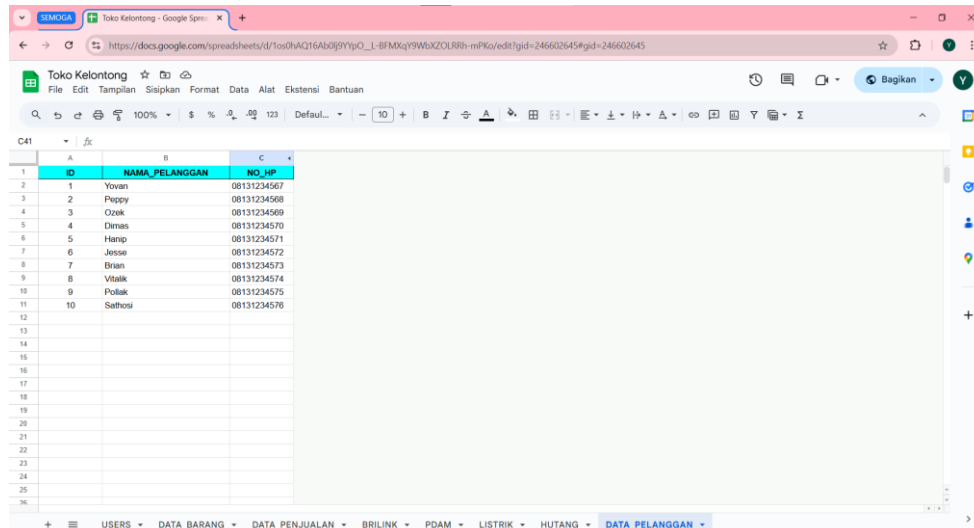
Dengan adanya hubungan ini, sistem dapat memastikan bahwa setiap transaksi pembayaran LISTRIK yang tercatat di sheet LISTRIK berhubungan dengan pelanggan yang sudah terdaftar di tabel DATA_PELANGGAN, mempermudah pelacakan pembayaran, serta memperbarui status pembayaran pelanggan, apakah sudah lunas atau masih memiliki hutang. Relasi ini juga memungkinkan pemilik atau kasir untuk memverifikasi data pelanggan yang melakukan pembayaran dan mengelola status tagihan secara lebih efisien.

ID	NAMA_PELANGGAN	NOMINAL	KETERANGAN	TANGGAL_TRANSAKSI	STATUS
b652	DIMAS	250,000.00	hutang beli beras 60 kg	2024-10-13 17:46:00	Hutang
bbdad	lucky arif mustofa	13,000.00	gula 1 kg	2024-10-13 20:51:07	Hutang
e38fc	hans Armstrong	45,000.00	mie rebus 15 biji	2024-10-22 15:08:13	Hutang
970fa	vitaik	2,500.00	mie goreng	2024-10-30 01:20:59	Hutang
884b	jesse poltak	54,300.00	kopi abc dan gula 3 kg	2024-10-30 01:21:46	Lunas
4d5a2	hans mustofa	100,000.00	hutang shampo	2024-11-11 15:00:02	Lunas
b652	DIMAS	250,000.00	hutang beli beras 60 kg	2024-10-13 17:46:00	Hutang

Gambar 4.7 Sheet Hutang

Pada Gambar 4.7 Sheet HUTANG digunakan untuk mencatat transaksi hutang pelanggan, dengan atribut utama seperti ID sebagai identitas unik transaksi, NAMA_PELANGGAN yang mencatat nama pelanggan yang melakukan hutang, NOMINAL yang menunjukkan jumlah hutang, KETERANGAN yang menjelaskan jenis transaksi hutang (misalnya, jenis barang yang dibeli secara kredit), TANGGAL_TRANSAKSI yang mencatat waktu transaksi dilakukan, dan STATUS yang menunjukkan apakah hutang sudah lunas atau masih tertunggak.

Nama pelanggan pada sheet HUTANG diinput secara manual dan tidak terhubung langsung dengan tabel DATA_PELANGGAN. Oleh karena itu, meskipun nama pelanggan tercatat di sheet ini, tidak ada relasi langsung antara data pelanggan yang ada di HUTANG dan DATA_PELANGGAN dalam basis data. Hal ini membuat sheet HUTANG berfungsi sebagai pencatatan hutang yang bersifat independen dari data pelanggan yang ada dalam sistem.



ID	NAMA_PELANGGAN	NO_HP
1	Yovan	08131234567
2	Peggy	08131234568
3	Ozek	08131234569
4	Dimas	08131234570
5	Hanip	08131234571
6	Jesse	08131234572
7	Brian	08131234573
8	Vitalik	08131234574
9	Polak	08131234575
10	Sathosi	08131234576

Gambar 4.8 Sheet Data_Pelanggan

Pada Gambar 4.8 menunjukkan implementasi dari sheet data_pelanggan, yang digunakan untuk menyimpan informasi pelanggan terkait layanan PDAM dan listrik. Sheet ini mencakup kolom ID sebagai pengidentifikasi unik setiap pelanggan, Nama_Pelanggan untuk mencatat nama lengkap pelanggan, dan No_HP sebagai informasi kontak. Data ini mempermudah identifikasi dan pengelolaan layanan pelanggan, memungkinkan pemilik untuk mengakses informasi kontak pelanggan secara cepat dalam transaksi atau layanan terkait PDAM dan listrik.

4.1.2 Implementasi Sistem

Implementasi sistem adalah tahap di mana sistem yang telah dirancang sebelumnya akan diterapkan sesuai dengan pedoman yang telah disusun, sehingga sistem siap digunakan dengan fungsi yang diharapkan dan dapat diuji oleh pengguna. Pada sistem “Toko Asih” ini, implementasi dilakukan menggunakan *google apps script*. Bagian ini akan menjelaskan dan merinci bentuk implementasi dari sistem tersebut sebagai berikut.

a. Form Login User

Sistem login pada aplikasi Toko Kelontong berfungsi sebagai gerbang autentikasi untuk mengelola akses pengguna ke dalam sistem. Pada tampilan login, pengguna diminta memasukkan username dan password yang telah didaftarkan sebelumnya pada tabel USERS di basis data. Setelah pengguna memasukkan kredensial, sistem akan mencocokkan data tersebut dengan kolom USERNAME dan PASSWORD pada tabel USERS. Jika validasi berhasil, pengguna akan diberikan akses sesuai peran mereka yang diatur melalui kolom ROLE_ID.

Basis data ini dirancang untuk memastikan integrasi yang erat antara autentikasi pengguna dan manajemen aktivitas di dalam sistem. Kolom ID pada tabel USERS digunakan sebagai kunci relasi utama yang menghubungkan pengguna dengan aktivitas operasional lainnya, seperti transaksi pada tabel listrik, pdam, dan brilink, serta manajemen stok barang melalui tabel data_barang. Setiap transaksi yang dilakukan oleh pengguna akan mencatat kolom id_kasir atau user_id, yang mengacu pada kolom ID di tabel USERS, untuk memastikan bahwa setiap tindakan dapat dilacak ke pengguna tertentu.

Sistem ini memastikan hanya pengguna yang berwenang, sesuai peran yang tercatat di ROLE_ID, dapat mengakses modul atau fungsi tertentu. Dengan demikian, sistem login ini tidak hanya memberikan keamanan data tetapi juga mendukung pengelolaan hak akses yang terstruktur. Tampilan antarmuka sederhana seperti yang ditampilkan di Gambar 4.9.

Gambar 4.9 *Form Login User*

Script Untuk Login dapat dilihat pada Gambar 4.10.

```
function login(username, password) {
  const table = openSheet().getSheetByName("USERS")
  const find_row =
table.getRange("C2:C").createTextFinder(username).findNext()
  if (find_row) {
    const row = find_row.getRow()
    const values = table.getRange(`A${row}:E${row}`).getValues()

    if (values[0][3] === password) {
      return {
        status: true,
        user: {
          id: values[0][0],
          fullname: values[0][1],
          username: values[0][2],
          role_id: values[0][4]
        }
      }
    }
  }
}
```

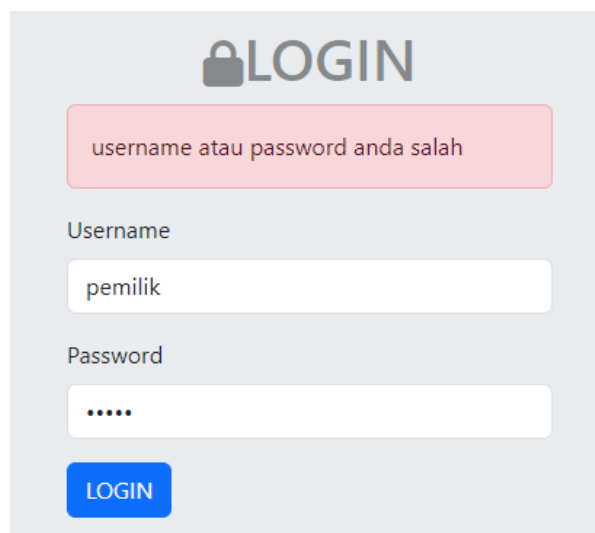
```

    }
  }
}
return {
  status: false,
  message: "username atau password anda salah"
}
}
return {
  status: false,
  message: "username atau password anda salah"
}
}
}

```

Gambar 4.10 Script Login

Halaman Form Login User digunakan oleh aktor pemilik, admin, dan kasir untuk mengakses sistem. Pengguna perlu memasukkan username dan password yang akan diverifikasi lebih lanjut. Jika verifikasi gagal, misalnya username atau password yang dimasukkan salah, sistem akan menampilkan pemberitahuan kesalahan, seperti yang terlihat pada Gambar 4.11



The image shows a login form with a grey background. At the top, there is a lock icon followed by the word "LOGIN" in a large, bold, grey font. Below this, a red rectangular box contains the error message "username atau password anda salah" in a smaller, dark font. Underneath the error message, there are two input fields: "Username" with the text "pemilik" entered, and "Password" with five dots representing a masked password. At the bottom of the form, there is a blue button with the word "LOGIN" in white capital letters.

Gambar 4.11 Tampilan Pemberitahuan Kesalahan

b. Halaman Pengelolaan PDAM

Halaman PDAM dalam sistem informasi Toko Asih dirancang untuk menampilkan data transaksi pembayaran PDAM kepada pemilik dan kasir. Namun, pengelolaan data PDAM, seperti penambahan, pengubahan, dan penghapusan data, hanya dapat dilakukan oleh pemilik. Sistem ini mengatur akses halaman berdasarkan peran pengguna, di mana hanya pengguna dengan ROLE_ID yang menunjukkan status pemilik yang dapat melakukan tindakan pengelolaan data.

Data pada halaman PDAM diambil dari Sheet PDAM, yang mencatat detail transaksi pembayaran, termasuk ID transaksi, NAMA_PELANGGAN, NOMINAL, KETERANGAN status pembayaran, dan TANGGAL_TRANSAKSI. Nama pelanggan yang tercatat diambil dari Sheet DATA_PELANGGAN, yang berfungsi sebagai repositori utama data pelanggan, sehingga memastikan setiap transaksi terhubung dengan pelanggan yang terdaftar.

Relasi antara Sheet PDAM dan Sheet DATA_PELANGGAN mempermudah pemilik untuk memverifikasi dan mengelola pembayaran pelanggan secara efisien. Selain itu, halaman ini juga memastikan bahwa pengguna sistem (kasir dan pemilik) yang melakukan transaksi dapat dipantau melalui kolom id_kasir, yang berelasi dengan Sheet USERS. Implementasi ini mendukung transparansi dan kendali akses sesuai dengan peran pengguna, memastikan data dikelola secara aman dan terorganisir. Tampilan antarmuka sederhana seperti yang ditampilkan di Gambar 4.12.

ID	Nama Pelanggan	Nominal	Tanggal Transaksi	Keterangan	Action
4a660	Ozek	Rp 0,00	2024-10-13 17:40:54	Lunas	[Edit] [Delete]
810a8	Yovan	Rp 0,00	2024-10-13 17:40:10	Lunas	[Edit] [Delete]
88e7e	Hanip	Rp 10.000.000,00	2024-10-13 17:40:40	Hutang	[Edit] [Delete]
a5834	Dimas	Rp 750.000,00	2024-10-13 17:40:22	Hutang	[Edit] [Delete]

Gambar 4.12 Halaman Pengelolaan PDAM

Script untuk mengambil data dari Sheet PDAM dapat dilihat pada Gambar 4.13

```
function getDataPDAM() {
  const table = openSheet().getSheetByName("PDAM")
  const values = table.getRange("A2:E").getValues().filter(row =>
!row.every(cell => cell == ""))
  const result = values.map(value => ({
    ID: value[0],
    NAMA_PELANGGAN: value[1],
    NOMINAL: value[2],
    KETERANGAN: value[3],
    TANGGAL_TRANSAKSI: value[4],
  }))
  return {
```

```

    status: true,
    data: result
  }
}

```

Gambar 4.13 Script Mengambil Data PDAM

Script Untuk Menampilkan Data PDAM ke sistem dapat dilihat pada Gambar 4.14

```

async function getDataPDAM() {
  return new Promise((resolve, reject) => {
    google.script.run
      .withSuccessHandler((response) => resolve(response))
      .withFailureHandler((err) => reject(console.log(err)))
      .getDataPDAM()
  })
}

```

Gambar 4.14 Script Menampilkan Data PDAM

Pada halaman PDAM ini terdapat menu *add* data untuk menambahkan data PDAM. Jika data pelanggan tidak terdapat dalam data base maka data tidak akan muncul dan perlu penambahan di data base. Tampilan implementasi add data PDAM dapat dilihat pada Gambar 4.15.

Gambar 4.15 Form Add Data PDAM

Script Untuk Menambahkan data PDAM dapat dilihat pada Gambar 4.16

```

function addDataPDAM(data) {
  try {
    const { pelanggan, nominal, keterangan } = JSON.parse(data)
    const table = openSheet().getSheetByName("PDAM")
    const row = table.getLastRow() + 1

    let jenisKeterangan
    if (keterangan == 1) {

```

```

        jenisKeterangan = "Lunas"
    } else {
        jenisKeterangan = "Hutang"
    }

    table.getRange(`A${row}:E${row}`).setValues([[
        createRandomId(),
        pelanggan,
        nominal,
        jenisKeterangan,
        generateTanggal()
    ]])

    return {
        status: false,
        message: "Data has been successfully saved."
    }
} catch(err) {
    return {
        status: false,
        message: err.message
    }
}
}
}

```

Gambar 4.16 *Script Add Data PDAM*

c. Halaman Pengelolaan Listrik

Halaman Listrik dalam sistem informasi Toko Asih dirancang untuk menampilkan data transaksi pembayaran listrik kepada pemilik dan kasir. Namun, akses untuk pengelolaan data listrik, seperti penambahan, pengubahan, dan penghapusan data, hanya diberikan kepada pemilik. Sistem memastikan keamanan data dengan membatasi akses pengelolaan tersebut berdasarkan `ROLE_ID` yang menunjukkan status pemilik, sehingga hanya pengguna dengan hak akses pemilik yang dapat melakukan pengelolaan.

Data pada halaman Listrik bersumber dari Sheet Listrik, yang mencatat setiap transaksi pembayaran listrik, termasuk `ID` transaksi, `NAMA_PELANGGAN`, `NOMINAL` tagihan, `KETERANGAN` status pembayaran (misalnya, "Lunas" atau "Hutang"), dan `TANGGAL_TRANSAKSI`. Nama pelanggan yang tercatat pada sheet ini diambil dari Sheet `DATA_PELANGGAN`, sehingga memastikan setiap transaksi terhubung dengan pelanggan yang terdaftar secara formal. Relasi antara Sheet Listrik dan Sheet `DATA_PELANGGAN` mendukung integritas data pelanggan dan memudahkan pemilik dalam melacak status pembayaran pelanggan.

Selain itu, halaman ini mencatat pengguna yang melakukan transaksi melalui kolom `id_kasir`, yang berelasi dengan Sheet `USERS`, sehingga pemilik dapat memantau aktivitas transaksi berdasarkan pengguna sistem. Implementasi ini memberikan kontrol penuh kepada

pemilik untuk mengelola data listrik, memastikan akurasi pencatatan, serta memberikan akses yang sesuai dengan peran pengguna dalam sistem. Tampilan antarmuka sederhana seperti yang ditampilkan di Gambar 4.17.

ID	Nama Pelanggan	Nominal	Keterangan	Tanggal Transaksi	Action
0f8b9	Ozek	Rp 540.000,00	Hutang	2024-10-13 17:41:29	
4fc1d	Hanip	Rp 5.455.000,00	Hutang	2024-10-13 17:41:39	
74173	Yovan	Rp 19.000.000,00	Lunas	2024-10-13 17:41:47	

Gambar 4.17 Halaman Pengelolaan Listrik

Script untuk mengambil data dari Sheet LISTRIK dapat dilihat pada Gambar 4.18

```
function getDataListrik() {
  const table = openSheet().getSheetByName("LISTRIK")
  const values = table.getRange("A2:E").getValues().filter(row =>
!row.every(cell => cell == ""))
  const result = values.map(value => ({
    ID: value[0],
    NAMA_PELANGGAN: value[1],
    NOMINAL: value[2],
    KETERANGAN: value[3],
    TANGGAL_TRANSAKSI: value[4]
  }))

  return {
    status: true,
    data: result
  }
}
```

Gambar 4.18 Script Mengambil Data Listrik

Script untuk menampilkan data listrik ke sistem dapat dilihat pada Gambar 4.19

```
async function getDataListrik() {
  return new Promise((resolve, reject) => {
    google.script.run
      .withSuccessHandler((response) => resolve(response))
      .withFailureHandler((err) => reject(console.log(err)))
      .getDataListrik()
  })
}
```

Gambar 4.19 Script Menampilkan Data Listrik

Pada halaman listrik ini terdapat menu *add* data untuk menambahkan data listrik. Jika data pelanggan tidak terdapat dalam data base maka data tidak akan muncul dan perlu penambahan di data base. Tampilan implementasi add data listrik dapat dilihat pada Gambar 4.20.

ID	Nama Pelanggan	Tanggal Transaksi	Action
0f8b9	Ozek	2024-10-13 17:41:29	[Edit] [Delete]
4fc1d	Hanip	2024-10-13 17:41:39	[Edit] [Delete]
74173	Yovan	Rp 19.000.000,00 Lunas	2024-10-13 17:41:47 [Edit] [Delete]

Gambar 4.20 Form Add Data Listrik

Script Untuk Menambahkan data Listrik dapat dilihat pada Gambar 4.21

```
function addDataListrik(data) {
  try {
    const { pelanggan, nominal, keterangan } = JSON.parse(data)
    const table = openSheet().getSheetByName("LISTRIK")
    const row = table.getLastRow() + 1

    let jenisKeterangan
    if (keterangan == 1) {
      jenisKeterangan = "Lunas"
    } else {
      jenisKeterangan = "Hutang"
    }

    table.getRange(`A${row}:E${row}`).setValues([[
      createRandomId(),
      pelanggan,
      nominal,
      jenisKeterangan,
      generateTanggal()
    ]])

    return {
      status: false,
      message: "Data has been successfully saved."
    }
  }
}
```

```
} catch(err) {  
  return {  
    status: false,  
    message: err.message  
  }  
}  
}
```

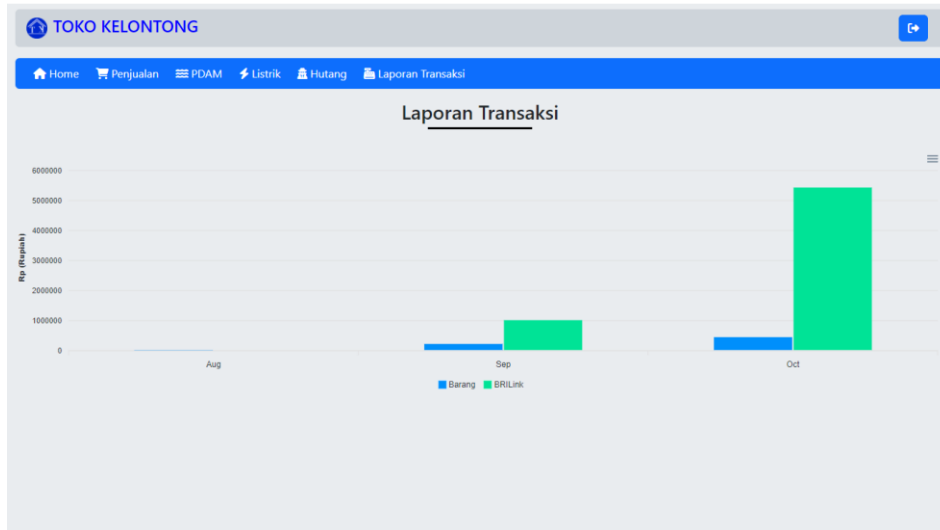
Gambar 4.21 Script *Add Data Listrik*

d. Halaman Laporan Transaksi

Halaman Laporan Transaksi dalam sistem informasi Toko Asih dirancang khusus untuk pemilik toko sebagai alat untuk memantau rekapitulasi transaksi bulanan, mencakup transaksi penjualan barang dan transaksi BRILink. Pada halaman ini, data transaksi disajikan dalam bentuk grafik batang, di mana grafik batang berwarna hijau merepresentasikan total transaksi BRILink, sedangkan grafik batang berwarna biru menunjukkan total transaksi penjualan barang dalam satu bulan.

Data yang ditampilkan pada halaman ini bersumber dari Sheet DATA_PENJUALAN untuk transaksi penjualan barang dan Sheet BRILINK untuk transaksi BRILink. Dari Sheet DATA_PENJUALAN, sistem mengambil informasi seperti TANGGAL_TRANSAKSI, jumlah barang yang terjual, dan total pendapatan. Sementara itu, data pada Sheet BRILINK mencakup TANGGAL_TRANSAKSI dan nominal total transaksi. Sistem mengelompokkan data berdasarkan bulan, sehingga grafik yang ditampilkan memberikan pemilik visualisasi yang mudah dipahami mengenai tren penjualan dan transaksi lainnya.

Halaman ini sepenuhnya diakses oleh pemilik melalui hak akses khusus yang diatur berdasarkan ROLE_ID pada Sheet USERS. Implementasi halaman laporan ini mempermudah pemilik dalam menganalisis performa toko, membuat keputusan strategis, dan memantau kontribusi setiap jenis transaksi terhadap pendapatan toko secara keseluruhan. Tampilan antarmuka sederhana seperti yang ditampilkan di Gambar 4.22.



Gambar 4.22 Laporan Transaksi

Script untuk mengambil data laporan transaksi dapat dilihat pada Gambar 4.23

```
function getDataLaporanTransaksi() {
    const penjualan_barang =
    openSheet().getSheetByName("DATA_PENJUALAN").getRange("A2:E").getValues()
    .filter(row => !row.every(cell => cell == ""))
    const brilink =
    openSheet().getSheetByName("BRILINK").getRange("A2:F").getValues().filter
    (row => !row.every(cell => cell == ""))

    const months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
    'Sep', 'Oct', 'Nov', 'Dec'];
    const result = {
        bulan: [],
        penjualan: [
            { name: 'Barang', data: [] },
            { name: 'BRILink', data: [] }
        ]
    };
};
```

Gambar 4.23 Script Mengambil Laporan Transaksi

e. Halaman Data Barang

Halaman Data Barang dalam sistem informasi Toko Asih dirancang untuk memberikan admin kendali penuh terhadap data master barang. Admin dapat menambahkan barang baru, mengubah informasi barang yang sudah ada, serta menghapus barang yang tidak lagi relevan. Data barang yang dikelola mencakup beberapa atribut utama, yaitu ID barang sebagai identitas unik, nama barang, stok barang, harga dasar, harga jual, rak, dan kategori barang. Data ini tersimpan pada Sheet DATA_BARANG, yang berfungsi sebagai pusat penyimpanan dan pengelolaan informasi barang.


```

return {
  status: true,
  data: result
}
}

```

Gambar 4.25 Script Mengambil Data Barang

Script untuk menampilkan data barang ke sistem dapat dilihat pada Gambar 4.26

```

async function getDataBarang() {
  return new Promise((resolve, reject) => {
    google.script.run
      .withSuccessHandler((response) => resolve(response))
      .withFailureHandler((err) => reject(console.log(err)))
      .getDataBarang()
  })
}

```

Gambar 4.26 Script Menampilkan Data Barang

Pada halaman data barang ini terdapat menu *add* data untuk menambahkan data barang. Form *add* data barang akan menambahkan data baru atau bisa juga pengadaan data yang sudah ada. Jika melakukan pengadaan barang pada form add data nama barang harus sama dengan yang sudah terdapat pada data base *google sheet*. Tampilan implementasi add data barang dapat dilihat pada Gambar 4.27.

Gambar 4.27 Form Add Data Barang

Script untuk menambahkan data barang dapat dilihat pada Gambar 4.28

```

function addDataBarang(data) {
  try {
    const { namaBarang, stockBarang, hargaDasarBarang, hargaJualBarang,
    rakBarang, kategoriBarang } = JSON.parse(data)
    const table = openSheet().getSheetByName("DATA_BARANG")

```

```

const findBarang =
table.getRange("B:B").createTextFinder(namaBarang).findNext()
if (findBarang) {
  const rowBarang = findBarang.getRow()
  let stock = table.getRange(`C${rowBarang}`).getValue()
  stock = parseInt(stock) + parseInt(stockBarang)

  table.getRange(`C${rowBarang}:G${rowBarang}`).setValues([[
    stock,
    hargaDasarBarang,
    hargaJualBarang,
    rakBarang, kategoriBarang
  ]])

  return {
    status: true,
    message: "Data barang berhasil di input."
  }
}

```

Gambar 4.28 Script *Add Data Barang*

f. Halaman Melihat Harga Jual dan Stock Barang

Halaman Melihat Harga Jual dan Stok Barang dalam sistem informasi Toko Asih dirancang untuk memudahkan kasir dalam mengakses informasi terkait harga jual dan stok barang yang tersedia di toko. Fitur ini memungkinkan kasir untuk melihat data barang tanpa harus mengubah atau mengelola informasi lainnya. Halaman ini hanya menampilkan informasi yang relevan untuk transaksi, yaitu ID barang, nama barang, stok barang, dan harga jual.

Informasi tersebut diambil langsung dari Sheet DATA_BARANG dalam basis data sistem, yang menyimpan data utama mengenai barang yang tersedia di toko. Dengan menggunakan ID barang sebagai referensi, kasir dapat dengan cepat mencari dan menampilkan data barang terkait. Stok yang ditampilkan menunjukkan jumlah barang yang tersedia untuk dijual, sementara harga jual menunjukkan harga barang yang akan dikenakan kepada pelanggan.

Sistem ini membantu kasir dalam melakukan pengecekan harga dan stok secara efisien saat melayani pelanggan, memastikan bahwa transaksi penjualan berjalan dengan lancar dan sesuai dengan informasi yang ada di sistem. Halaman ini tidak memungkinkan kasir untuk mengubah data barang, karena hanya berfungsi untuk melihat informasi yang sudah tercatat.

Tampilan implementasi dari fitur Melihat Harga Jual dan Stok Barang dapat dilihat pada Gambar 4.29, yang menunjukkan antarmuka yang dirancang khusus untuk kasir agar dapat mengakses informasi ini dengan mudah dan cepat.

TOKO KELONTONG

Home Data Barang Penjualan BriLink PDAM Listrik Hutang

Data Barang

10 entries per page Search:

ID	Nama Barang	Stock	Harga Jual
3da29	mie sedap	16	Rp 2.700,00
3da30	Aqua	2	Rp 6.000,00
6b4a8	gula 1 kg	10	Rp 13.400,00
a4f97	indomie goreng	42	Rp 2.550,00
a839c	Nivea	25	Rp 24.000,00
cce29	kopi abc	10	Rp 2.500,00

Showing 1 to 6 of 6 entries

Gambar 4.29 Halaman Melihat Harga Jual dan Stock Barang

Script untuk menampilkan harga jual dan stock barang dapat dilihat pada Gambar 4.30

```
function showTableDataBarang(data) {
  const role_id = getUserRoleId()
  if (datatable) datatable.clear().destroy();
  $('#myTable').empty() //mengosongkan table
  datatable = $("#myTable").DataTable({
    data: data.map((item) => [
      item.ID,
      item.NAMA_BARANG,
      item.STOCK,
      item.HARGA_DASAR,
      item.HARGA_JUAL,
      item.RAK,
      item.KATEGORI,
      item.ID
    ]),
  });
}
```

Gambar 4.30 Script Menampilkan Harga Jual dan Stock Barang

g. Halaman Transaksi Penjualan Barang

Halaman Penjualan Barang dalam sistem informasi Toko Asih dirancang untuk memudahkan kasir dalam memproses transaksi penjualan barang. Pada halaman ini, kasir dapat memilih barang yang akan dijual kepada pelanggan berdasarkan data yang sudah tercatat dalam Sheet DATA_BARANG. Hanya barang-barang yang terdaftar di sheet tersebut yang dapat dipilih dan diinput oleh kasir, sehingga memastikan bahwa transaksi hanya mencatat barang yang tersedia di toko.

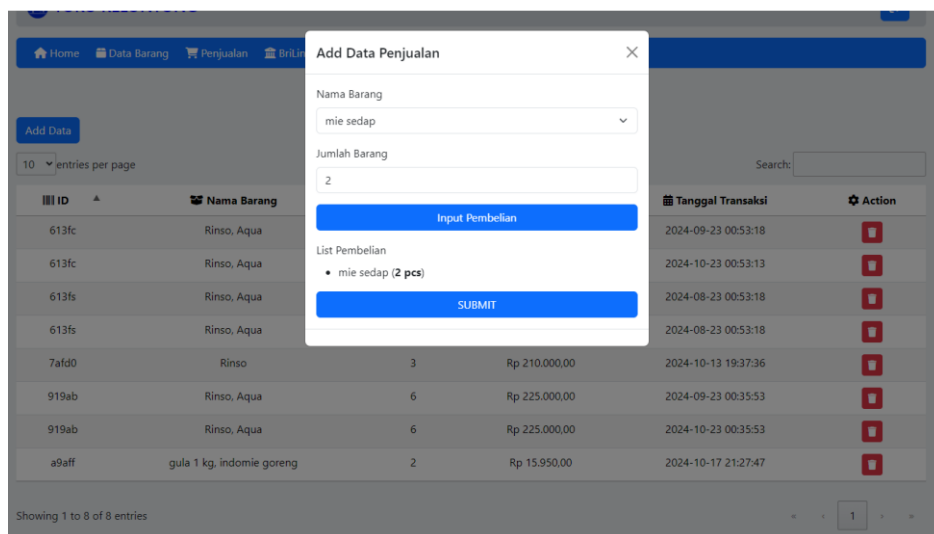
Setelah memilih barang yang akan dijual, kasir dapat menginput jumlah barang yang dibeli pelanggan. Sistem akan secara otomatis menghitung total harga penjualan berdasarkan harga jual yang tercatat di database dan jumlah barang yang dipilih. Transaksi penjualan ini kemudian

dicatat dalam Sheet DATA_PENJUALAN, yang mencakup informasi terkait ID transaksi, nama barang, jumlah barang, total harga, dan tanggal transaksi.

Halaman ini juga memungkinkan kasir untuk memilih apakah transaksi dilakukan secara tunai atau kredit. Jika transaksi dilakukan dengan kredit, maka informasi terkait hutang pelanggan akan tercatat di Sheet HUTANG, menghubungkan transaksi dengan saldo hutang yang harus dibayar oleh pelanggan di kemudian hari. Namun, jika transaksi dilakukan secara tunai, tidak ada informasi hutang yang dicatat.

Dengan adanya fitur ini, kasir dapat memproses transaksi penjualan secara cepat dan akurat, serta memastikan bahwa setiap transaksi tercatat dengan baik dalam sistem. Halaman Penjualan Barang juga memiliki kontrol akses sehingga hanya kasir yang berwenang yang dapat mengakses dan menggunakan halaman ini.

Tampilan implementasi dari halaman Penjualan Barang dapat dilihat pada Gambar 4.31, yang menunjukkan antarmuka yang dirancang untuk memudahkan kasir dalam mengelola data penjualan secara efisien.



Gambar 4.31 Penjualan Barang

Script untuk menambahkan data transaksi dapat dilihat pada Gambar 4.32

```
function addDataPenjualan(data) {
  const table = openSheet().getSheetByName("DATA_PENJUALAN")
  const row = table.getLastRow() + 1

  const namaBarang = data.list.map(item => item.nama).join(', ')
  const jumlahBarang = data.list.reduce((acc, current) => acc +
  current.jumlah, 0)
  const totalPenjualan = data.list.reduce((acc, current) => acc +
  current.total, 0)

  data.list.map(item => {
```

```

    return reduceDataBarangByName(item.nama, item.jumlah);
  })

  table.getRange(`A${row}:E${row}`).setValues([[
    createRandomId(),
    namaBarang,
    jumlahBarang,
    totalPenjualan,
    generateTanggal()
  ]])

  return {
    status: true,
    message: "Data has been successfully saved."
  }
}

```

Gambar 4.32 Script *Add* Data Penjualan Barang

Script untuk menampilkan data transaksi ke sistem dapat dilihat pada Gambar 4.33

```

async function getDataPenjualan() {
  return new Promise((resolve, reject) => {
    google.script.run
      .withSuccessHandler((response) => resolve(response))
      .withFailureHandler((err) => reject(console.log(err)))
      .getDataPenjualan()
  })
}

```

Gambar 4.33 Script Menampilkan Data Penjualan Barang

h. Halaman Transaksi BRILink

Halaman BRILink dalam sistem informasi Toko Asih diimplementasikan untuk memfasilitasi kasir dalam melayani transaksi tarik tunai atau setor tunai melalui layanan BRILink. Pada halaman ini, kasir dapat mengakses menu "Add Data" untuk menambahkan transaksi BRILink baru, yang akan mencatat informasi terkait pelanggan dan transaksi yang dilakukan. Data yang dimasukkan mencakup ID transaksi, nama pelanggan, nomor handphone, nominal transaksi, jenis transaksi (tarik tunai atau setor tunai), serta tanggal transaksi.

Informasi yang tercatat pada halaman ini disimpan dalam Sheet BRILINK dalam basis data. ID transaksi yang dihasilkan berfungsi sebagai identitas unik untuk setiap transaksi, sementara data lainnya, seperti nama pelanggan dan nomor handphone, dicatat untuk keperluan dokumentasi transaksi. Meskipun data pelanggan tersebut tercatat dalam transaksi BRILink, data ini tidak berhubungan langsung dengan tabel DATA_PELANGGAN, yang menyimpan informasi pelanggan secara lebih umum dalam sistem. Hal ini membuat data pelanggan pada transaksi BRILink bersifat independen dan hanya terkait dengan transaksi yang bersangkutan.

Transaksi BRILink ini juga berelasi dengan Sheet USERS, melalui id_kasir, yang mencatat kasir atau pengguna sistem yang melakukan transaksi tersebut. Relasi ini memungkinkan pelacakan transaksi berdasarkan kasir yang bertanggung jawab, memberikan kontrol dan akurasi dalam pencatatan data transaksi.

Dengan implementasi sistem ini, kasir dapat dengan mudah memasukkan transaksi BRILink yang terkait dengan pelanggan secara real-time, dan sistem akan otomatis mencatat semua informasi transaksi di Sheet BRILINK. Hal ini memastikan proses pencatatan yang efisien dan terstruktur, serta memungkinkan pemantauan transaksi dengan mudah oleh pihak yang berwenang. Tampilan implementasi dari Transaksi BRILink dapat dilihat pada Gambar 4.34, yang menunjukkan antarmuka yang mudah digunakan oleh kasir untuk memproses transaksi ini.

ID	Nama Pelanggan	No HP	Nominal	Jenis Transaksi	Tanggal Transaksi	Action
03246	Yovan Yovan	081222333444	Rp 500.000,00	Setor Tunai	2024-09-22 23:45:45	[Edit] [Delete]
5b49a	dimas	081286831757	Rp 540.000,00	Tarik Tunai	2024-09-23 00:36:42	[Edit] [Delete]
f0b1c	DIMAS	081823063536	Rp 5.455.000,00	Setor Tunai	2024-10-13 17:46:19	[Edit] [Delete]

Gambar 4.34 Transaksi BRILink

Script untuk mengambil data BRILink pada sheet BRILINK dapat dilihat pada Gambar 4.35

```
function getDataBRILink() {
  const table = openSheet().getSheetByName("BRILINK")
  const values = table.getRange("A2:F").getValues().filter(row =>
!row.every(cell => cell == ""))
  const result = values.map(value => ({
    ID: value[0],
    NAMA_PELANGGAN: value[1],
    NO_HP: value[2],
    NOMINAL: value[3],
    JENIS_TRANSAKSI: value[4],
    TANGGAL_TRANSAKSI: value[5]
  }))

  return {
    status: true,
  }
}
```

```

data: result
}
}

```

Gambar 4.35 Script Mengambil Data BRILink

Script untuk menampilkan data BRILink ke sistem dapat dilihat pada Gambar 4.36

```

async function getDataBRILink() {
  return new Promise((resolve, reject) => {
    google.script.run
      .withSuccessHandler((response) => resolve(response))
      .withFailureHandler((err) => reject(console.log(err)))
      .getDataBRILink()
  })
}

```

Gambar 4.36 Script Menampilkan Data BRILink

Menu add data digunakan kasir untuk menambahkan data transaksi baru. Setelah memilih menu ini, kasir akan diarahkan ke halaman form *add data* untuk menambahkan transaksi baru. Pada halaman add data BRILink kasir dapat mengisi form nama pelanggan, nomor handphone, nominal, dan dapat memilih jenis transaksi. Tampilan implementasi add data BRILink dapat dilihat pada Gambar 4.37.

Gambar 4.37 Form Add Data BRILink

Script Untuk menambahkan data BRILink dapat dilihat pada Gambar 4.38

```

function editDataBRILinkById(data) {
  try {
    const { idBRILink, nominal, jenisTransaksi } = JSON.parse(data)
    const table = openSheet().getSheetByName("BRILINK")
    const row = table.getRange("A:A").createTextFinder(idBRILink).findNext().getRow()
    let jenis
  }
}

```

```
if (jenisTransaksi == 1) {  
    jenis = "Tarik Tunai"  
} else {  
    jenis = "Setor Tunai"  
}  
  
table.getRange(`D${row}:E${row}`).setValues([[  
    nominal,  
    jenis  
]])
```

Gambar 4.38 Script Add Data BRILink

i. Halaman Transaksi PDAM

Halaman Transaksi PDAM dalam sistem informasi Toko Asih dirancang untuk memudahkan kasir dalam melayani pelanggan yang akan membayar hutang penalangan PDAM. Pada halaman ini, kasir dapat mengakses dan melakukan pembaruan informasi terkait transaksi pembayaran PDAM, terutama keterangan status pembayaran, yang mencakup status seperti "Lunas" atau "Hutang". Kasir dapat mengubah keterangan status pembayaran ini sesuai dengan kondisi aktual transaksi pelanggan.

Data yang dikelola pada halaman ini akan tercatat dalam Sheet PDAM dalam basis data. Setiap transaksi yang dicatat berisi informasi utama seperti ID transaksi, nama pelanggan, nominal pembayaran, keterangan yang menunjukkan status pembayaran (lunas atau hutang), dan tanggal transaksi. Dengan adanya fasilitas untuk mengubah status pembayaran, kasir dapat memperbarui informasi secara dinamis berdasarkan transaksi yang terjadi, baik itu pembayaran lunas ataupun pembayaran sebagian.

Tampilan implementasi dari Transaksi PDAM dapat dilihat pada Gambar 4.39, yang menggambarkan antarmuka yang digunakan kasir untuk melakukan pelayanan pembayaran PDAM dan mengubah informasi terkait status pembayaran.

ID	Nama Pelanggan	Nominal	Tanggal Transaksi	Keterangan	Action
4a660	Ozek	Rp 0,00	2024-10-13 17:40:54	Lunas	[Edit]
810a8	Yovan	Rp 0,00	2024-10-13 17:40:10	Lunas	[Edit]
88e7e	Hanip	Rp 10.000.000,00	2024-10-13 17:40:40	Hutang	[Edit]
a5834	Dimas	Rp 750.000,00	2024-10-13 17:40:22	Hutang	[Edit]

Gambar 4.39 Transaksi Data PDAM

Script untuk menampilkan transaksi PDAM dapat dilihat pada Gambar 4.40

```

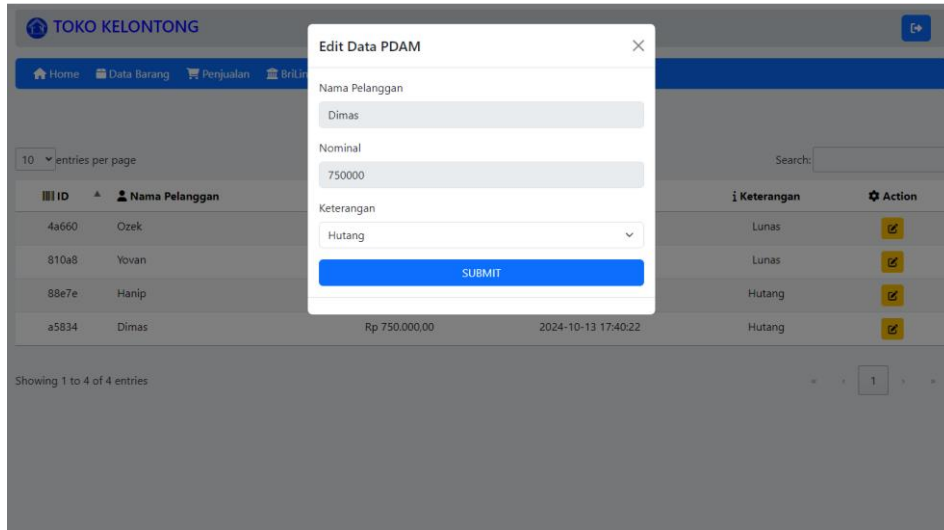
async function showPDAM() {
  hiddenHome()

  const role_id = getUserRoleId()
  if (role_id == 2) {
    document.getElementById("add-data-button").hidden = true
  } else {
    document.getElementById("add-data-button").hidden = false
  }
}

```

Gambar 4.40 Script Transaksi PDAM

Pada informasi Action digunakan oleh kasir untuk mengubah informasi keterangan untuk pelanggan yang sudah membayar. Tampilan implementasi action transaksi PDAM dapat dilihat pada Gambar 4.41.



Gambar 4.41 Tampilan *Action* PDAM

Script untuk mengubah informasi keterangan PDAM dapat dilihat pada Gambar 4.42

```
function editDataPDAMById(data) {
  try {
    const { idPDAM, nominal, keterangan } = JSON.parse(data)
    const table = openSheet().getSheetByName("PDAM")
    const row =
      table.getRange("A:A").createTextFinder(idPDAM).findNext().getRow()

    let jenis
    if (keterangan == 1) {
      jenis = "Lunas"
    } else {
      jenis = "Hutang"
    }

    table.getRange(`C${row}:D${row}`).setValues([[
      nominal,
      jenis
    ]])
  }
}
```

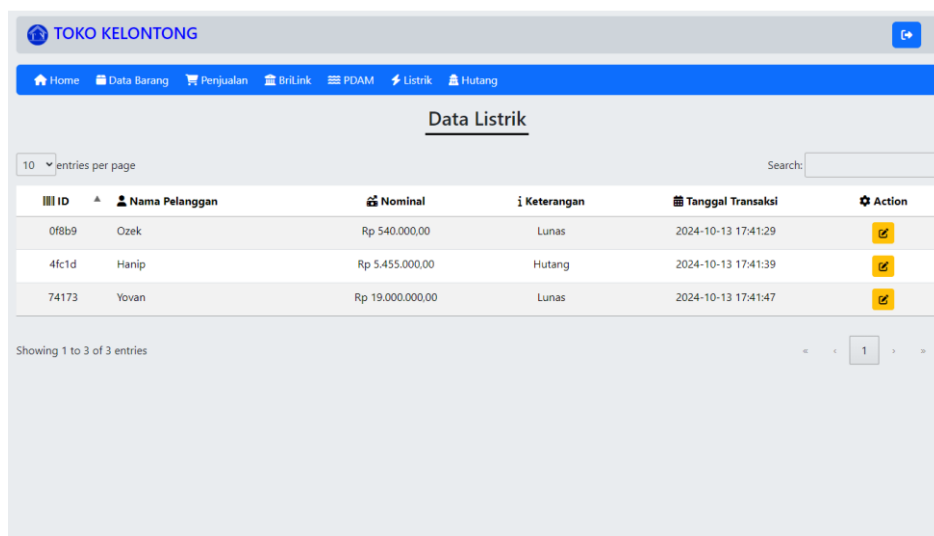
Gambar 4.42 *Script Action* PDAM




j. Halaman Transaksi Listrik

Halaman Transaksi Listrik dalam sistem informasi Toko Asih dirancang untuk memudahkan kasir dalam melayani pelanggan yang akan membayar hutang penalangan Listrik. Pada halaman ini, kasir dapat mengakses dan melakukan pembaruan informasi terkait transaksi pembayaran Listrik, terutama keterangan status pembayaran, yang mencakup status seperti "Lunas" atau "Hutang". Kasir dapat mengubah keterangan status pembayaran ini sesuai dengan kondisi aktual transaksi pelanggan.

Data yang dikelola pada halaman ini akan tercatat dalam Sheet Listrik dalam basis data. Setiap transaksi yang dicatat berisi informasi utama seperti ID transaksi, nama pelanggan, nominal pembayaran, keterangan yang menunjukkan status pembayaran (lunas atau hutang), dan tanggal transaksi. Dengan adanya fasilitas untuk mengubah status pembayaran, kasir dapat memperbarui informasi secara dinamis berdasarkan transaksi yang terjadi, baik itu pembayaran lunas ataupun pembayaran sebagian.

Tampilan implementasi dari Transaksi Listrik dapat dilihat pada Gambar 4.43, yang menggambarkan antarmuka yang digunakan kasir untuk melakukan pelayanan pembayaran Listrik dan mengubah informasi terkait status pembayaran.



ID	Nama Pelanggan	Nominal	Keterangan	Tanggal Transaksi	Action
0f8b9	Ozek	Rp 540.000,00	Lunas	2024-10-13 17:41:29	
4fc1d	Hanip	Rp 5.455.000,00	Hutang	2024-10-13 17:41:39	
74173	Yovan	Rp 19.000.000,00	Lunas	2024-10-13 17:41:47	

Gambar 4.43 Transaksi Data Listrik

Script untuk menampilkan transaksi Listrik dapat dilihat Gambar 4.44

```

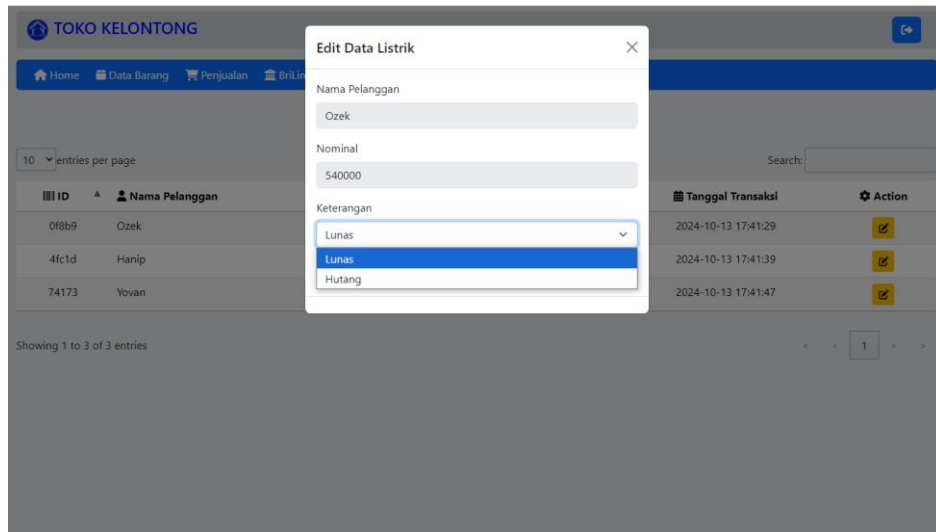
async function showListrik() {
  hiddenHome()

  const role_id = getUserRoleId()
  if (role_id == 2) {
    document.getElementById("add-data-button").hidden = true
  } else {
    document.getElementById("add-data-button").hidden = false
  }
}

```

Gambar 4.44 Script Transaksi Listrik

Pada informasi Action digunakan oleh kasir untuk mengubah informasi keterangan untuk pelanggan yang sudah membayar. Tampilan implementasi action transaksi listrik dapat dilihat pada Gambar 4.45.



Gambar 4.45 Tampilan *Action* Listrik

Script untuk mengubah informasi keterangan listrik dapat dilihat pada Gambar 4.44 Gambar 4.46

```
function editDataListrikById(data) {
  try {
    const { idListrik, keterangan } = JSON.parse(data)
    const table = openSheet().getSheetByName("LISTRIK")
    const row = table.getRange("A:A").createTextFinder(idListrik).findNext().getRow()

    let jenis
    if (keterangan == 1) {
      jenis = "Lunas"
    } else {
      jenis = "Hutang"
    }

    table.getRange(`D${row}`).setValues([[
      jenis
    ]])
  }
}
```

Gambar 4.46 Script *Action* Listrik

k. Halaman Data Hutang

Halaman Hutang dalam sistem informasi Toko Asih dirancang untuk membantu kasir mencatat dan mengelola hutang pelanggan terkait dengan transaksi penjualan barang. Pada halaman ini, kasir dapat menambah data hutang pelanggan dengan menggunakan menu "Add data", yang memungkinkan mereka untuk mencatat transaksi kredit di mana pelanggan belum membayar penuh atau melakukan pembayaran secara cicilan. Informasi yang ditampilkan di

halaman ini mencakup ID transaksi, nama pelanggan, nominal hutang, keterangan yang menjelaskan rincian transaksi hutang (seperti jenis barang yang dibeli secara kredit), dan tanggal transaksi yang mencatat waktu terjadinya transaksi.

Data yang dimasukkan pada halaman ini akan disimpan dalam Sheet Hutang dalam basis data, yang berfungsi sebagai tempat untuk menyimpan catatan terkait hutang pelanggan. Halaman ini memungkinkan kasir untuk mencatat informasi secara lengkap dan jelas mengenai status hutang, sehingga sistem dapat dengan mudah melacak transaksi yang masih memiliki saldo hutang yang harus dilunasi oleh pelanggan.

Tampilan implementasi dari Transaksi Hutang dapat dilihat pada Gambar 4.47, yang menggambarkan antarmuka yang digunakan kasir untuk melakukan pencatatan data hutang transaksi barang.

ID	Nama Pelanggan	Nominal	Keterangan	Tanggal Transaksi	Action
b65f2	DIMAS	Rp 723.000,00	hutang top up dana	2024-10-13 17:46:00	[Edit] [Delete]
bbdad	lucky arif mustofa	Rp 13.000,00	gula 1 kg	2024-10-13 20:51:07	[Edit] [Delete]
e36fc	hans mustofa	Rp 45.000,00	mie rebus 15 biji	2024-10-22 15:06:13	[Edit] [Delete]

Gambar 4.47 Halaman Data Hutang

Script untuk mengambil data hutang dari sheet HUTANG dapat dilihat pada Gambar 4.48

```
function getDataHutang() {
  const table = openSheet().getSheetByName("HUTANG")
  const values = table.getRange("A2:F").getValues().filter(row =>
!row.every(cell => cell == ""))

  const result = values.filter(v => v[5] === "Hutang").map(value => ({
    ID: value[0],
    NAMA_PELANGGAN: value[1],
    NOMINAL: value[2],
    KETERANGAN: value[3],
    TANGGAL_TRANSAKSI: value[4]
  }))

  return {
    status: true,
    data: result
  }
}
```

```

}
}

```

Gambar 4.48 Script Mengambil Data Hutang

Script untuk menampilkan data hutang ke sistem dapat dilihat pada Gambar 4.49

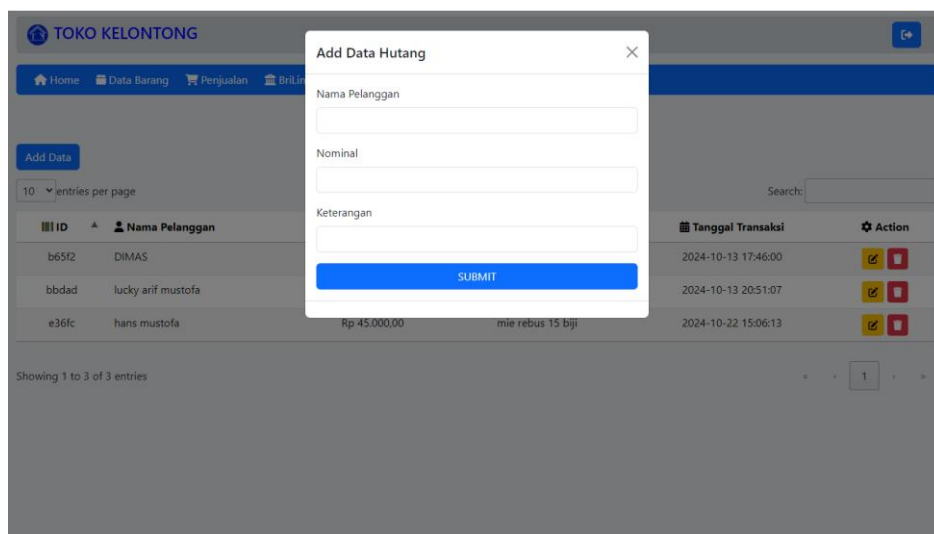
```

async function getDataHutang() {
  return new Promise((resolve, reject) => {
    google.script.run
      .withSuccessHandler((response) => resolve(response))
      .withFailureHandler((err) => reject(console.log(err)))
      .getDataHutang()
  })
}

```

Gambar 4.49 Script Menampilkan Data Hutang

Menu add data digunakan kasir untuk menambahkan data hutang baru. Setelah memilih menu ini, kasir akan diarahkan ke halaman form *add data* untuk menambahkan data baru. Pada halaman add data hutang kasir dapat mengisi form nama pelanggan, nominal, dan keterangan. Tampilan implementasi add data hutang dapat dilihat pada Gambar 4.50.



Gambar 4.50 Form Add Data Hutang

Script untuk menambahkan data hutang dapat dilihat pada Gambar 4.51

```

function addDataHutang(data) {
  try {
    const { pelanggan, nominal, keterangan } = JSON.parse(data)
    const table = openSheet().getSheetByName("HUTANG")
    const row = table.getLastRow() + 1

    table.getRange(`A${row}:F${row}`).setValues([[
      createRandomId(),
      pelanggan,
      nominal,

```

```

keterangan,
generateTanggal(),
"Hutang"
]])

```

Gambar 4.51 Script Add Data Hutang

4.2 Pembahasan

Google Apps Script yang menggunakan Google Sheets sebagai basis data menawarkan kemudahan dalam pengelolaan data dengan antarmuka yang sederhana dan dapat diakses secara real-time. Dalam studi kasus sistem informasi Toko Asih, Google Sheets memungkinkan pengguna untuk mengelola, membaca, mengedit, dan menghapus data secara mudah, bahkan dalam kolaborasi tim yang lebih besar, tanpa memerlukan perangkat lunak tambahan. Google Apps Script berfungsi untuk mengotomatiskan berbagai proses dalam sistem, seperti pembaruan data, pengelolaan stok barang, dan pelaporan transaksi secara langsung di dalam spreadsheet. Hal ini sangat berguna untuk Toko Asih, yang memiliki volume transaksi kecil hingga menengah. Meskipun efektif untuk aplikasi skala kecil, penggunaan Google Sheets sebagai basis data memiliki keterbatasan dalam hal skalabilitas dan keandalan untuk menangani volume data yang besar, seperti pada saat jumlah pelanggan atau transaksi mulai meningkat secara signifikan.

Sebaliknya, basis data SQL (relasional) seperti MySQL, PostgreSQL, atau SQLite memberikan struktur yang lebih kuat dan terorganisir dalam mengelola data yang besar dan kompleks (Elmasri & Navathe, 2015). Dalam konteks Toko Asih, jika volume transaksi dan data pelanggan meningkat, penggunaan database SQL akan sangat menguntungkan. Database SQL dirancang untuk menangani relasi antar tabel, memungkinkan normalisasi data yang lebih baik, pengelolaan transaksi yang lebih konsisten, dan performa yang lebih cepat dalam query data, terutama untuk aplikasi yang memerlukan akses data dalam jumlah besar atau kompleks (Date, 2012). Basis data SQL mendukung fitur-fitur seperti pengindeksan, pencarian yang efisien, dan keamanan tingkat lanjut untuk melindungi data, yang akan sangat penting bagi pengelolaan informasi pelanggan, stok barang, dan laporan transaksi Toko Asih.

Kelebihan Google Sheets dengan Google Apps Script antara lain adalah kemudahan penggunaan, akses real-time, dan integrasi langsung dengan produk Google lainnya, yang memungkinkan pemilik dan staf Toko Asih untuk mengelola data tanpa memerlukan infrastruktur server atau pengaturan database yang rumit. Kelemahannya, Google Sheets memiliki keterbatasan dalam kapasitas data, performa yang lebih rendah saat menangani data besar, serta keterbatasan dalam pengelolaan transaksi dan integritas data yang lebih kompleks. Misalnya, untuk transaksi yang melibatkan pembayaran secara kredit atau pelacakan stok

secara rinci, Google Sheets mungkin tidak mampu mengelola dengan efisien saat jumlah data terus berkembang.

Di sisi lain, kelebihan database SQL adalah skalabilitas, kemampuan menangani data dalam jumlah besar, serta keandalan dalam menjaga integritas data melalui fitur seperti foreign keys, constraints, dan transaksi (Kumar & Ghosh, 2016). Dalam Toko Asih, penggunaan database SQL akan mendukung performa yang lebih baik, terutama untuk menangani laporan penjualan yang lebih kompleks atau saat melibatkan banyak transaksi yang dilakukan oleh kasir di berbagai waktu. SQL juga lebih mampu dalam mengelola relasi antara data pelanggan, transaksi penjualan, dan hutang, sehingga dapat memastikan integritas data yang lebih tinggi dan transaksi yang lebih aman. Kekurangannya adalah bahwa pengelolaan database SQL memerlukan pengetahuan teknis yang lebih mendalam, serta setup dan pemeliharaan server yang lebih rumit. Selain itu, akses real-time dan kolaborasi antar pengguna tidak secepat dan sesederhana yang ada di Google Sheets, yang mungkin menjadi kendala dalam pengelolaan toko yang membutuhkan pembaruan data secara cepat oleh banyak pengguna.

Dengan demikian, pemilihan antara Google Sheets dengan Google Apps Script dan database SQL dalam studi kasus Toko Asih bergantung pada skala dan kebutuhan sistem. Google Sheets lebih cocok untuk aplikasi kecil hingga menengah dengan kebutuhan data yang sederhana dan akses real-time, sementara database SQL lebih ideal untuk aplikasi dengan volume data besar, transaksi kompleks, dan kebutuhan integritas data yang tinggi. Dalam tahap awal Toko Asih, Google Sheets dapat digunakan dengan efektif untuk pengelolaan data sederhana, namun seiring berkembangnya bisnis dan meningkatnya kompleksitas transaksi, beralih ke database SQL akan memberikan keuntungan dari segi skalabilitas dan pengelolaan data yang lebih canggih.

4.3 Pengujian Sistem

Pengujian sistem ini merupakan tahapan untuk menjelaskan proses uji coba yang telah dilakukan dalam penelitian. Sistem yang telah selesai dikembangkan diuji pada aspek input, proses, dan output dengan tujuan untuk menilai kesiapan sistem yang dibangun. Pengujian ini bertujuan untuk mengidentifikasi potensi kesalahan dalam sistem sekaligus memastikan bahwa sistem tersebut sudah berjalan sesuai dengan harapan dan kebutuhan yang dirancang.

4.3.1 Pengujian *Blackbox*

Pengujian sistem ini dilakukan dengan metode *Blackbox*, yaitu pengujian yang melibatkan semua fungsi dan fitur dalam sistem. Dalam pendekatan ini, hasil dari setiap fungsi diperiksa untuk memastikan kesesuaian kebutuhan yang diharapkan.

Pengujian Aktor Pemilik

Tabel 4.1 Pengujian Aktor Pemilik

No	Fitur/Menu	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Transaksi Penjualan Barang	Pilih menu penjualan, terus menekan tombol add data lalu pengguna mengisi form data. Setelah data tersimpan pengguna dapat menghapus data sesuai kebutuhan melalui fitur action.	Sistem akan menampilkan halaman penjualan, serta akan menampilkan form data dan menyimpan data transaksi. Kemudian data tersimpan pada basis data dan ditampilkan pada halaman data penjualan. Data penjualan yang sudah tersimpan dapat dihapus melalui fitur action.	Data barang berhasil ditambahkan dan sistem menampilkan data penjualan yang sudah ditambahkan serta menyimpan data pada basis data. Serta data barang yang sudah ditampilkan dapat diedit atau hapus melalui fitur action	Berhasil
2	Pengelolaan data PDAM dan Listrik	Pengguna memilih menu PDAM / Listrik, terus menekan tombol Add Data lalu mengisi form yang tersedia	Sistem akan menampilkan halaman PDAM/Listrik, serta akan menampilkan form add data dan akan menyimpan data.	Sistem menampilkan halaman PDAM atau Listrik dan form add data serta menyimpan data pada halaman PDAM atau listrik	Berhasil

3	Melihat Laporan Transaksi	Pilih menu laporan transaksi	Sistem akan menampilkan Laporan transaksi yang berupa bar Charts	Sistem menampilkan data laporan transaksi	Berhasil
---	---------------------------	------------------------------	--	---	----------

Pengujian Aktor Admin

Tabel 4.2 Pengujian Aktor Admin

No	Fitur/Menu	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Kelola data barang	Admin Pilih menu data barang, terus menekan tombol add data lalu pengguna mengisi form data. Lalu pengguna mengisi form yang tersedia dengan data barang yang di perlukan dan pengguna dapat melakukan tindakan lebih lanjut seperti menghapus atau mengedit melalui action.	Sistem akan menampilkan halaman data barang, serta akan menampilkan form data dan menyimpan data barang baru. Kemudian data akan tersimpan di basis data dan ditampilkan pada halaman data barang. Data barang yang tersimpan dapat diedit atau dihapus melalui fitur action.	Data barang berhasil ditambahkan dan sistem menampilkan data barang yang sudah ditambahkan. Serta data barang yang sudah ditampilkan dapat diedit atau hapus melalui fitur action	Berhasil
2	Transaksi Pengadaan barang	Admin Pilih menu data barang, terus menekan tombol add data lalu pengguna mengisi form data. Admin mengisi form data dengan	Sistem akan menampilkan halaman data barang, serta akan menampilkan form data barang. Kemudian data akan tersimpan	pengadaan barang berhasil ditambahkan dan sistem menampilkan data barang yang sudah ditambahkan	Behasil

		nama barang yang sama dengan data sebelumnya.	dan memperbarui informasi stock, harga dasar, harga jual, rak, dan kategori.		
3	Transaksi Penjualan Barang	Pilih menu penjualan, terus menekan tombol add data lalu pengguna mengisi form data. Setelah data tersimpan pengguna dapat menghapus data sesuai kebutuhan melalui fitur action.	Sistem akan menampilkan halaman penjualan, serta akan menampilkan form data dan menyimpan data transaksi. Kemudian data tersimpan pada basis data dan ditampilkan pada halaman data penjualan. Data penjualan yang sudah tersimpan dapat dihapus melalui fitur action.	Data penjualan berhasil ditambahkan dan sistem menampilkan data penjualan yang sudah ditambahkan serta menyimpan data pada basis data. Serta data penjualan dapat dihapus melalui fitur action	Berhasil
4	Melihat Harga jual dan Jumlah barang	Admin membuka halaman menu data barang.	Sistem akan menampilkan halaman data barang serta menampilkan detail informasi data barang yang berupa ID, Nama barang, Stock, Dan harga jual.	Sistem menampilkan halaman data barang serta detail informasi data barang.	Berhasil

Pengujian Aktor Kasir

Tabel 4.3 Pengujian Aktor Kasir

No	Fitur/Menu	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
----	------------	--------------------	-----------------------	-----------------	------------

1	Transaksi Penjualan Barang	Pilih menu penjualan, terus menekan tombol add data lalu pengguna mengisi form data. Setelah data tersimpan pengguna dapat menghapus data sesuai kebutuhan melalui fitur action.	Sistem akan menampilkan halaman penjualan, serta akan menampilkan form data dan menyimpan data transaksi. Kemudian data tersimpan pada basis data dan ditampilkan pada halaman data penjualan. Data penjualan yang sudah tersimpan dapat dihapus melalui fitur action.	Data penjualan berhasil ditambahkan dan sistem menampilkan data penjualan yang sudah ditambahkan serta menyimpan data pada basis data. Serta data penjualan dapat dihapus melalui fitur action	Berhasil
2	Melihat Harga jual dan Jumlah barang	Kasir membuka halaman menu data barang.	Sistem akan menampilkan halaman data barang serta menampilkan detail informasi data barang yang berupa ID, Nama barang, Stock, Dan harga jual.	Sistem menampilkan halaman data barang serta detail informasi data barang.	Berhasil
3	Melayani Transaksi BRILink	Pilih menu Transaksi BRILink, terus menekan tombol add data lalu pengguna mengisi form data. Setelah data tersimpan pengguna dapat melakukan tindakan lebih lanjut seperti mengedit atau menghapus	Sistem akan menampilkan halaman Data BRILink, serta akan menampilkan form data dan menyimpan data transaksi. Kemudian data tersimpan pada basis data dan ditampilkan pada halaman data BRILink.	Data BRILink berhasil ditambahkan dan sistem menampilkan data penjualan yang sudah ditambahkan serta menyimpan data pada basis data. Serta data BRILink dapat diedit dan	Berhasil

		data sesuai kebutuhan melalui fitur action.	Data transaksi yang sudah tersimpan dapat diedit atau dihapus melalui fitur action.	dihapus melalui fitur action.	
4	Mencatat Hutang Pembeli	Pilih menu Hutang, terus menekan tombol add data lalu pengguna mengisi form data. Setelah data tersimpan pengguna dapat melakukan tindakan lebih lanjut seperti mengedit atau menghapus data sesuai kebutuhan melalui fitur action.	Sistem akan menampilkan halaman Hutang, serta akan menampilkan form data dan menyimpan data transaksi. Kemudian data tersimpan pada basis data dan ditampilkan pada halaman hutang. Data hutang yang sudah tersimpan dapat diedit atau dihapus melalui fitur action.	Data Hutang ditambahkan dan sistem menampilkan data hutang yang sudah ditambahkan. Serta data hutang yang sudah ditampilkan dapat diedit atau hapus melalui fitur action.	

4.3.2 Pengujian *System Usability Scale* (SUS)

Pengujian sistem menggunakan metode *System Usability Scale* (SUS) telah dilakukan dengan melibatkan dua responden yang akan berperan sebagai pemilik toko yang melibatkan Toko Asih dan Toko Berkah Mirah. Berdasarkan hasil pengujian, diperoleh skor rata-rata SUS sebesar 72,5 yang berada pada *grade C* dengan *adjective rating* dalam kategori "*good*." Berdasarkan hasil penilaian tersebut, sistem informasi Toko Asih dinilai mudah digunakan oleh pengguna dan dapat diandalkan sebagai alat untuk mengelola operasional toko Asih secara efisien.

Tabel 4.4 Tabel Hasil Pengujian *System Usability Scale* (SUS)

No	Pertanyaan	Responden 1	Responden 2
1	Apakah Anda berencana untuk menggunakan sistem ini	4	4
2	Apakah Anda merasa sistem ini sulit untuk digunakan	2	3
3	Apakah Anda merasa sistem ini mudah digunakan	4	4
4	Apakah Anda merasa sistem ini membingungkan	2	2

5	Apakah fitur-fitur dalam sistem ini berfungsi sebagaimana mestinya	4	3
6	Apakah menurut Anda ada banyak ketidakkonsistenan atau ketidakserasian dalam sistem ini	3	2
7	Apakah Anda membutuhkan bantuan dari orang lain atau teknisi untuk menggunakan sistem ini	3	2
8	Apakah Anda merasa tidak mengalami hambatan dalam menggunakan sistem ini	3	3
9	Apakah Anda perlu waktu untuk membiasakan diri sebelum dapat menggunakan sistem ini dengan baik	3	2
10	Apakah Anda yakin orang lain dapat memahami cara menggunakan sistem ini dengan cepat	4	3
Jumlah		32	28
Nilai (Jumlah x 2.5)		80	65
Nilai Rata-rata		72,5	

BAB V

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Sistem informasi penjualan dan pengelolaan data barang untuk Toko Asih yang dikembangkan menggunakan *Google Apps Script* telah berhasil memenuhi kebutuhan yang telah diidentifikasi dalam penelitian ini. Sistem ini memberikan kemudahan dalam pengelolaan transaksi penjualan barang, layanan BRILink, serta pencatatan hutang, yang sebelumnya dilakukan secara manual. Dengan sistem ini, proses pengelolaan data di Toko Asih menjadi lebih cepat, terstruktur, dan akurat.

Hasil pengujian sistem dengan metode *Blackbox* menunjukkan bahwa semua fitur utama seperti pengelolaan data barang, transaksi penjualan, transaksi BRILink, dan pengelolaan hutang berfungsi dengan baik dan sesuai harapan. Fitur pencatatan hutang serta pengelolaan transaksi PDAM dan listrik terbukti dapat diakses dan diolah secara akurat, mengurangi potensi kesalahan yang biasa terjadi dalam pencatatan manual.

Penggunaan *Google Apps Script* yang terintegrasi dengan *Google Sheets* sebagai basis data telah terbukti sebagai platform yang cocok untuk sistem ini. Penggunaan layanan berbasis *cloud* memungkinkan pengguna mengakses data kapan saja, sehingga pencatatan dan pemantauan transaksi dapat dilakukan dengan lebih mudah dan fleksibel. Meski sistem telah memenuhi kebutuhan utama Toko Asih, masih terdapat ruang untuk perbaikan, seperti penambahan fitur notifikasi stok barang yang hampir habis, yang dapat lebih mendukung pengelolaan inventaris.

Secara keseluruhan, pengembangan sistem informasi ini telah memenuhi tujuan penelitian, yaitu menyediakan solusi digital yang lebih terstruktur dan mudah digunakan untuk pengelolaan penjualan dan data barang di Toko Asih. Diharapkan sistem ini memberikan kontribusi nyata bagi pemilik dan karyawan dalam menjalankan aktivitas toko sehari-hari dengan lebih teratur.

5.2 SARAN

Meskipun sistem informasi untuk Toko Asih telah berhasil dikembangkan dan berfungsi sesuai harapan, ada beberapa saran untuk pengembangan lebih lanjut agar sistem ini dapat memberikan manfaat yang lebih optimal. Beberapa saran untuk pengembangan sistem nantinya adalah:

1. Dapat dikembangkan dengan pengingat otomatis hutang pelanggan maupun PDAM dan listrik berupa pesan yang dikirim secara berkala.
2. Dapat dikembangkan dengan penambahan informasi lainnya pada homepage seperti stock barang ataupun jumlah transaksi hari ini.
3. Dapat dikembangkan dengan tanpa memerlukan refresh halaman setiap pingin berpindah ke menu lain.

DAFTAR PUSTAKA

- Airinei, D., & Homocianu, D. (2017). Cloud Computing Based Web Applications. Examples and Considerations on Google Apps Script. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2964756>
- Ashok, S., Assistant, T. P., Priya, M., Semi, K., Assistant, P., Sreeja, M., & Library Assistant, K. P. (2019). Application of google app scripts in email for providing current awareness services to research scholars, at central university of kerala: an evaluative study. *Ijeast.Com A Thomas, MS KPInt J Eng Appl Sci Technol, 2019*•*ijeast.Com, 4*, 313–318. <http://www.ijeast.com/papers/313-318,Tesma406,IJEAST.pdf>
- atulrofida, hanik. (2021). *Sistem Informasi Manajemen Toko Pertanian Ud Sari Bumi Berbasis Web*. <https://dspace.uii.ac.id/handle/123456789/33497>
- Date, C. J. (2012). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed.). Addison-Wesley.
- Díaz, K., Director, A., Jesús, : , & Segura, A. (2015). *External services integration for internal usage of an organization, using Google App Script*. <https://upcommons.upc.edu/handle/2117/83848>
- Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems (7th ed.). Pearson.
- Ferreira, J. (2014). Google apps script: Web application development essentials. " O'Reilly Media, Inc."
- Ganapathy, R. (2016). Learning google apps script. Packt Publishing Ltd.
- Google Developers. (n.d.). *Google Apps Script*. Google. Retrieved November 25, 2024, from <https://developers.google.com/apps-script>
- Khariono, H., Parlita, R., Kusuma, H. A., & Setyawan, D. A. (2021). PEMANFAATAN BOT TELEGRAM SEBAGAI E-LEARNING UJIAN BERBASIS FILE. *Jurnal Informatika Polinema, 7*(4), 65–72. <https://doi.org/10.33795/JIP.V7I4.696>
- Kumar, S., & Ghosh, A. (2016). SQL Databases and Data Management for Large Applications. *International Journal of Computer Applications, 139*(1), 26-30.
- Nur Aidah, F. (2023). *Eksplorasi Perangkat Lunak dan Apps Script - Dicoding Blog*. <https://www.dicoding.com/blog/eksplorasi-perangkat-lunak-dan-apps-script/>
- Petrović, N., Roblek, V., Radenković, M., & Nejković, V. (2020, October). Approach to rapid development of data-driven applications for smart cities using appsheet and apps script.

In AIIT 2020 International conference on Applied Internet and Information Technologies (pp. 77-81).

- Prabowo, Y., Aulia, S., eProceedings, R. M.-, & 2023, undefined. (2023). Pembuatan Bot Telegram Untuk Teknisi Assurance Dengan Metode Push Notification Di Pt Telkom Akses Rajawali Kota Bandung. ..*Telkomuniversity.Ac.IdY Prabowo, S Aulia, R MaulanaeProceedings of Applied Science, 2023*•... ..*Telkomuniversity.Ac.Id*. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/19370>
- Rahman, A. (2020). *Rancang Bangun Sistem Informasi Penjadwalan Menggunakan Google Apps Script Pada Politeknik Kampar*. <http://repository.uin-suska.ac.id/28053/>
- Sari, K. (2019). *PENGEMBANGAN SISTEM INFORMASI*. https://www.researchgate.net/publication/336676758_PENGEMBANGAN_SISTEM_INFORMASI
- Simandjuntak, J. P. (John). (2009). Mengatasi Kendala Dalam Penerapan Sistem Informasi Di Institusi (Studi Kasus Dalam Proses Penerapan Sistem Informasi Di Sebuah Institusi.). *Jurnal Eksplorasi Karya Sistem Informasi Dan Sains*, 2(2), 79108. <https://www.neliti.com/publications/79108/>
- Sopingi, S., Teknologi, S. W.-J. J. S. dan, & 2023, undefined. (2023). Integrasi Sistem Pembelajaran dengan Google Classroom melalui Google Apps Script. *Journal.Ummat.Ac.IdS Sopingi, S WulandariJustek: Jurnal Sains Dan Teknologi, 2023*•*journal.Ummat.Ac.Id*, 6(2), 195–206. <https://doi.org/10.31764/justek.vXiY.ZZZ>
- Vallisree Sivathanu., et al. (2023). IOT based Smart Food Grain Storage Monitoring System for Silos using Google Apps Script. *PriMera Scientific Engineering*. <https://doi.org/10.56831/PSEN-02-050>
- Xu, Q., Lin, L., & Wu, X. (2021). Implementing Controlled Digital Lending with Google Drive and Apps Script: A Case Study at the NYU Shanghai Library. *International Journal of Librarianship*, 6(1), 37–54. <https://doi.org/10.23974/IJOL.2021.VOL6.1.193>

LAMPIRAN