**TUGAS LAPORAN**

**BUSINESS INTELLIGENCE**

**APLIKASI PEMBERI REKOMENDASI TINDAKAN KEPADA DEBITUR**

****

**Nama : Richard Tan**

**NPM : 221510048**

**Dosen : Saut Pintubipar Saragih, S.Kom., M.Si.**

PROGRAM STUDI Sistem Informasi

Fakultas Teknik dan Komputer

UNIVERSITAS PUTERA BATAM

**2024/2025**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc203657648)

[DAFTAR GAMBAR 4](#_Toc203657649)

[DAFTAR TABEL 6](#_Toc203657650)

[BAB I 7](#_Toc203657651)

[PENDAHULUAN 7](#_Toc203657652)

[1.1 Latar Belakang 7](#_Toc203657653)

[1.2 Identifikasi Masalah 9](#_Toc203657654)

[1.3 Rumusan Masalah 9](#_Toc203657655)

[1.4. Tujuan penelitian 9](#_Toc203657656)

[BAB II 10](#_Toc203657657)

[KAJIAN PUSTAKA 10](#_Toc203657658)

[2.1 Bussines Intelligence (BI) 10](#_Toc203657659)

[2.2 Kreditur dan Debitur 11](#_Toc203657660)

[2.3 Sistem Pemberi Tindakan kepada Debitur 12](#_Toc203657661)

[2.3.1 Phyton 12](#_Toc203657662)

[2.3.2 Pandas 13](#_Toc203657663)

[2.3.3 Numpy 13](#_Toc203657664)

[2.3.4 Matplotlib 14](#_Toc203657665)

[2.3.5 Jupyter Notebook 14](#_Toc203657666)

[2.3.6 Microsoft Excel 15](#_Toc203657667)

[2.3.7 HTML 15](#_Toc203657668)

[2.3.8 CSS 16](#_Toc203657669)

[2.4 Aplikasi Web Dan Arsitekturnya 16](#_Toc203657670)

[2.4.1 Definisi Aplikasi Website 16](#_Toc203657671)

[2.5 Arsitektur Aplikasi Web (Frontend, Backend, Database) 17](#_Toc203657672)

[2.5.1 Frontend 17](#_Toc203657673)

[2.5.2 Backend 17](#_Toc203657674)

[2.5.3 Database 18](#_Toc203657675)

[2.6 Pengembangan Frontend 18](#_Toc203657676)

[2.6.1 React.js 18](#_Toc203657677)

[2.6.2 Komponen 18](#_Toc203657678)

[2.6.3 JSX 19](#_Toc203657679)

[2.6.4 State 19](#_Toc203657680)

[2.6.5 Props(Properties) 19](#_Toc203657681)

[2.6.6 Virtual DOM 20](#_Toc203657682)

[2.7 Metode Pengumpulan Data 20](#_Toc203657683)

[2.7.1 Alat 21](#_Toc203657684)

[2.7.2 Teknik 21](#_Toc203657685)

[2.8 Visual Studio Code 22](#_Toc203657686)

[2.9 Xampp 22](#_Toc203657687)

[2.10 User Interface 23](#_Toc203657688)

[BAB III 24](#_Toc203657689)

[METODOLOGI 24](#_Toc203657690)

[3.1 Desain Penelitian 24](#_Toc203657691)

[3.2 Sumber Data 46](#_Toc203657692)

[3.2.1 Data Sekunder 47](#_Toc203657693)

[BAB IV 48](#_Toc203657694)

[HASIL DAN PEMBAHASAN 48](#_Toc203657695)

[4.1 Hasil Aplikasi Web 48](#_Toc203657696)

[4.2 Pembahasan 51](#_Toc203657697)

[4.3 Flow Aplikasi 70](#_Toc203657698)

[4.4 Codingan Aplikasi 74](#_Toc203657699)

[BAB V 80](#_Toc203657700)

[KESIMPULAN DAN SARAN 80](#_Toc203657701)

[DAFTAR PUSTAKA 82](#_Toc203657702)

# DAFTAR GAMBAR

[Gambar 3. 1Desain Metodologi 23](#_Toc203657703)

[Gambar 3. 2Data Raw 25](#_Toc203657704)

[Gambar 3. 3 Data Cleansing 28](#_Toc203657705)

[Gambar 3. 4 Data Transformation 30](#_Toc203657706)

[Gambar 3. 5 Visualisasi Tujuan Kredit 31](#_Toc203657707)

[Gambar 3. 6 Visualisasi Kategori Kendaraan Produktif 31](#_Toc203657708)

[Gambar 3. 7 Visualisasi Kategori Kendaraan Konsumtif 32](#_Toc203657709)

[Gambar 3. 8 Visualisasi Tenor Per Keterangan Bidang Usaha 32](#_Toc203657710)

[Gambar 3. 9 Visualisai DP Per Keterangan Bidang Usaha 33](#_Toc203657711)

[Gambar 3. 10 Visualisasi Nilai Kontrak Per Tenor 33](#_Toc203657712)

[Gambar 3. 11 Visualisasi Status Debitur 34](#_Toc203657713)

[Gambar 3. 12 Tujuan Status Debitur Telat 34](#_Toc203657714)

[Gambar 3. 13 Desain Aliran Sistem Informasi 38](#_Toc203657715)

[Gambar 3. 14 Dashboard 40](#_Toc203657716)

[Gambar 3. 15 Daftar Debitur Status Hijau 40](#_Toc203657717)

[Gambar 3. 16 Daftar Debitur Status Kuning 41](#_Toc203657718)

[Gambar 3. 17 Daftar Debitur Status Merah 41](#_Toc203657719)

[Gambar 3. 18 Daftar Semua Debitur 42](#_Toc203657720)

[Gambar 3. 19 Daftar Debitur Diproses 42](#_Toc203657721)

[Gambar 3. 25 Nav Bar 49](#_Toc203657722)

[Gambar 4. 1 Halaman Dashboard 47](#_Toc203657723)

[Gambar 4. 2 Status Hijau 47](#_Toc203657724)

[Gambar 4. 3 Status Kuning 48](#_Toc203657725)

[Gambar 4. 4 Status Merah 48](#_Toc203657726)

[Gambar 4. 5 Halaman Daftar Debitur 49](#_Toc203657727)

[Gambar 4. 6 Filter Status 49](#_Toc203657728)

[Gambar 4. 7 Log History Debitur 50](#_Toc203657729)

[Gambar 4. 8 Daftar Debitur Diproses 50](#_Toc203657730)

[Gambar 4. 9 Tampilan Dashboard 50](#_Toc203657731)

[Gambar 4. 10 Tampilan Daftar Debitur Terbaru 52](#_Toc203657732)

[Gambar 4. 11 Tampilan Daftar Debitur Status Hijau 53](#_Toc203657733)

[Gambar 4. 12 Tampilan Daftar Debitur Status Kuning 56](#_Toc203657734)

[Gambar 4. 13 Tampilan Daftar Debitur Status Merah 60](#_Toc203657735)

[Gambar 4. 14 Tampilan Halaman Daftar Debitur 63](#_Toc203657736)

[Gambar 4. 15 Tampilan Halaman Log History Debitur 67](#_Toc203657737)

[Gambar 4. 16 Tampilan Halaman Daftar Debitur Diproses 68](#_Toc203657738)

[Gambar 4. 17 Dashboard 70](#_Toc203657739)

[Gambar 4. 18 Daftar Debitur Status Hijau 70](#_Toc203657740)

[Gambar 4. 19 Daftar Debitur Status Kuning 71](#_Toc203657741)

[Gambar 4. 20 Daftar Debitur Status Merah 72](#_Toc203657742)

[Gambar 4. 21 Log History Debitur 72](#_Toc203657743)

[Gambar 4. 22 Halaman Daftar Debitur 72](#_Toc203657744)

[Gambar 4. 23 Code Dashboard 73](#_Toc203657745)

[Gambar 4. 24 Code Daftar Debitur 74](#_Toc203657746)

[Gambar 4. 25 Code DB 75](#_Toc203657747)

[Gambar 4. 26 Code Proses Tindakan 76](#_Toc203657748)

[Gambar 4. 27 Debitur Proses 77](#_Toc203657749)

[Gambar 4. 28 Log History 78](#_Toc203657750)

# DAFTAR TABEL

[Tabel 4. 1 51](#_Toc203657751)

[Tabel 4. 2 52](#_Toc203657752)

[Tabel 4. 3 55](#_Toc203657753)

[Tabel 4. 4 58](#_Toc203657754)

[Tabel 4. 5 62](#_Toc203657755)

[Tabel 4. 6 65](#_Toc203657756)

[Tabel 4. 7 66](#_Toc203657757)

[Tabel 4. 8 68](#_Toc203657758)

# BAB I

# PENDAHULUAN

## Latar Belakang

Dalam industri pembiayaan, kemampuan untuk mengidentifikasi risiko keterlambatan pembayaran secara cepat menjadi faktor krusial dalam menjaga stabilitas operasional perusahaan. PT Capella Multidana menghadapi kendala dalam melakukan deteksi dini terhadap debitur yang mengalami atau berpotensi mengalami keterlambatan pembayaran. Ketiadaan sistem pemantauan secara langsung menyebabkan keterlambatan informasi yang dibutuhkan oleh manajemen, sehingga proses pengambilan keputusan tidak dapat dilakukan secara tepat waktu.

Menanggapi situasi tersebut, tim pengembang melakukan studi kasus dengan merancang dan mengimplementasikan sistem berbasis Business Intelligence (BI) yang dirancang untuk memberikan peringatan dan rekomendasi secara langsung kepada manajemen, berdasarkan status pembayaran dari masing-masing debitur. Data yang digunakan dalam pengembangan sistem ini diperoleh langsung dari internal PT Capella Multidana melalui metode dokumentasi. Dataset tersebut berisi lebih dari 22.000 entri debitur bermasalah dalam rentang waktu 2000 hingga 2025, yang disimpan dalam format Excel. Informasi yang termuat mencakup nama debitur, ID pelanggan, durasi kontrak, tanggal mulai dan akhir kredit, rincian pembayaran, deskripsi kronologi, harga kendaraan, serta status kontrak yang bersangkutan. (Lilis Nurellisa\*1, n.d.)

Agar sistem dapat berfungsi secara akurat, tahap awal dilakukan dengan membersihkan data menggunakan Microsoft Excel, termasuk menghapus data duplikat, memperbaiki format tanggal, dan menyelaraskan nilai-nilai numerik. Proses transformasi data selanjutnya dijalankan menggunakan Python, guna menyesuaikan struktur data menjadi lebih seragam dan siap untuk dianalisis lebih lanjut. Setelah data diproses, hasilnya disajikan dalam bentuk dashboard interaktif yang dapat diakses oleh manajemen untuk memantau status risiko keterlambatan dari setiap debitur.

Sistem ini mengklasifikasikan status debitur ke dalam tiga kelompok besar berdasarkan durasi keterlambatan, yaitu **Hijau** (1–14 hari), **Kuning** (15–30 hari), dan **Merah** (lebih dari 30 hari). Masing-masing kategori ini selanjutnya dibagi lagi berdasarkan tingkat risiko — rendah, menengah, dan tinggi — yang ditentukan berdasarkan pola keterlambatan, riwayat pembayaran, dan faktor relevan lainnya.

Pada kategori **Hijau**, tindakan yang diberikan bervariasi. Untuk risiko rendah dan menengah, sistem menyarankan pengiriman Surat Peringatan 1 (SP1) disertai notifikasi melalui SMS atau email. Jika risiko tergolong tinggi, sistem akan mempercepat respons melalui pengiriman SP1 dengan intensitas notifikasi lebih tinggi, termasuk melalui telepon.

Untuk kategori **Kuning**, debitur yang mengalami keterlambatan antara 15 hingga 30 hari akan dikenakan tindakan berbeda. Risiko rendah akan ditangani dengan SP2 dan pengingat melalui telepon tanpa perlu inspeksi langsung. Pada tingkat menengah, inspeksi fisik oleh kolektor menjadi wajib. Sementara pada risiko tinggi, sistem menyarankan eskalasi dengan SP2 lebih awal (misalnya pada hari ke-10 keterlambatan) dan inspeksi segera.

Sedangkan pada kategori **Merah**, yang menunjukkan keterlambatan di atas 30 hari, penanganan dilakukan secara lebih tegas. Debitur berisiko rendah masih diberikan toleransi berupa masa tenggang tambahan sebelum tindakan lebih lanjut, sambil dilakukan analisis kasus. Untuk risiko menengah, proses penarikan kendaraan langsung dimulai dan pemberitahuan dikirim ke tim eksekusi. Dalam kasus risiko tinggi, sistem akan menjalankan proses penarikan secara tepat waktu atau bahkan lebih cepat jika terdapat indikasi negatif lainnya.

Pengembangan sistem dilakukan dalam bentuk aplikasi web dengan memadukan berbagai teknologi. Python digunakan untuk proses pengolahan data dan integrasi dengan database SQL. Untuk membangun antarmuka pengguna (frontend), digunakan HTML, CSS, JavaScript, dan Bootstrap. Sementara itu, backend dikembangkan menggunakan PHP. Integrasi antara visualisasi data dan sistem notifikasi dalam aplikasi ini memungkinkan manajemen untuk melakukan pemantauan kondisi pembayaran debitur secara lebih tanggap dan berdasarkan data terkini. (Hendro Poerbo Prasetiya 1), n.d.)

## Identifikasi Masalah

1. Banyak debitur telat bayar tidak terdeteksi secara dini (lebih awal)

## Rumusan Masalah

1. Bagaimana merancang dan mengembangkan sistem kendali data debitur yang terlambat membayar?
2. Bagaimana merancang dan mengembangkan system yang menghasilkan notifikasi data debitur gagal bayar atau terlambat bayar serta keterangan status debitur secara real time?

## 1.4. Tujuan penelitian

1. Tidak tersedia sistem peringatan yang sistematis untuk membantu manajemen dalam mengidentifikasi debitur yang telah melewati jatuh tempo pembayaran.
2. menghasilkan notifikasi data debitur gagal bayar atau terlambat bayar serta keterangan status debitur secara real time.

# BAB II

# KAJIAN PUSTAKA

## Bussines Intelligence (BI)

Business Intelligence (BI) merupakan sebuah proses yang memiliki peranan penting dalam mendukung pengambilan keputusan bisnis. Proses ini mencakup kegiatan pengumpulan, integrasi, analisis, serta pemanfaatan data yang berasal dari berbagai sumber dalam organisasi. Tujuan utama dari BI adalah untuk menghasilkan informasi yang akurat, relevan, dan dapat diandalkan sebagai dasar dalam merumuskan kebijakan strategis perusahaan.(Sulistyoningsih et al., n.d.)

Memasuki era Industri 4.0, lingkungan bisnis mengalami peningkatan kompleksitas dan dinamika. Perusahaan dihadapkan pada beragam tantangan, seperti pesatnya perkembangan teknologi, intensitas persaingan global yang semakin tinggi, serta perubahan dalam preferensi dan perilaku konsumen. Kondisi ini menuntut perusahaan untuk terus berinovasi, khususnya dalam proses pengambilan keputusan yang cepat dan tepat.(Kumalasari Subroto & Endaryati, 2021)

Business Intelligence menjadi salah satu solusi untuk membantu perusahaan menghadapi tantangan tersebut. Melalui penerapan BI, data yang dimiliki dapat diolah menjadi informasi yang bernilai guna mendukung manajemen dalam mengambil keputusan secara cepat, tepat sasaran, dan berbasis data nyata. Kemampuan ini sangat penting, mengingat dinamika pasar yang bergerak cepat, di mana keterlambatan atau kesalahan dalam pengambilan keputusan dapat menimbulkan risiko kerugian yang signifikan. dalam konteks bisnis modern, perusahaan harus mampu merespons dinamika pasar dengan sigap. Oleh karena itu, pemanfaatan sistem BI menjadi sangat penting, karena mampu menyajikan informasi menyeluruh tentang kondisi internal perusahaan, tren pasar, hingga perilaku konsumen secara real-time.(Andalusi & Irfanudin, 2021)

## Kreditur dan Debitur

Kreditur merupakan pihak yang memberikan pinjaman atau fasilitas pembiayaan kepada pihak lain yang membutuhkan. Kreditur bisa berasal dari lembaga keuangan seperti bank, perusahaan leasing, koperasi, ataupun individu, yang memiliki otoritas untuk menyalurkan dana berdasarkan ketentuan dan persyaratan tertentu. Dalam hal ini, kreditur berhak untuk memperoleh kembali dana yang telah dipinjamkan, termasuk bunga atau biaya lain yang telah disepakati sebelumnya.

Sebaliknya, debitur adalah pihak yang menerima pinjaman dari kreditur dan memiliki tanggung jawab untuk melunasi pinjaman tersebut sesuai dengan ketentuan dalam perjanjian yang telah disetujui kedua belah pihak. Debitur dapat berupa perseorangan, kelompok, maupun badan usaha yang menggunakan fasilitas pembiayaan tersebut untuk keperluan konsumsi atau kegiatan produktif, seperti pembelian aset, permodalan usaha, maupun investasi.

Interaksi antara kreditur dan debitur didasarkan pada suatu perjanjian kredit yang mencakup unsur-unsur utama seperti jumlah pinjaman, jangka waktu pelunasan (tenor), tingkat bunga, serta syarat-syarat pembayaran. Apabila debitur tidak dapat memenuhi kewajibannya sesuai dengan isi perjanjian, maka kreditur berwenang untuk melakukan tindakan tertentu, seperti menagih pembayaran, memberikan surat peringatan, hingga melakukan penyitaan terhadap agunan, tergantung pada tingkat keterlambatan yang terjadi.(Sukemi Kamto Sudibyo, n.d.)

Dalam konteks laporan ini, perhatian utama difokuskan pada perilaku debitur, terutama yang berkaitan dengan keterlambatan dalam pembayaran cicilan. Selain itu, dibahas pula langkah-langkah yang diambil oleh pihak kreditur, dalam hal ini PT Capella Multidana, melalui pengembangan sistem pemberi rekomendasi tindakan berbasis Business Intelligence untuk mengidentifikasi serta menangani keterlambatan pembayaran secara terstruktur dan real-time.

## Sistem Pemberi Tindakan kepada Debitur

Sistem rekomendasi tindakan kepada debitur merupakan perangkat pendukung yang sangat penting dalam sektor keuangan, khususnya dalam upaya pengelolaan dan mitigasi risiko kredit sejak tahap awal. Aplikasi ini dirancang untuk membantu pihak manajemen dalam mengambil keputusan yang akurat berdasarkan data dan pola perilaku pembayaran debitur. Kehadiran sistem ini memungkinkan perusahaan untuk melakukan intervensi dini sebelum keterlambatan pembayaran berkembang menjadi kredit bermasalah yang berpotensi merugikan.(Aisyah & Purba, n.d.)

sistem ini berperan dalam meningkatkan kewaspadaan manajemen terhadap potensi risiko kerugian finansial, terutama yang berkaitan dengan pembiayaan bermasalah. Dalam konteks ini, aplikasi tidak hanya menjalankan fungsi pemantauan, tetapi juga melakukan analisis terhadap riwayat transaksi debitur guna mengidentifikasi pola-pola yang mengindikasikan ketidakteraturan atau kecenderungan keterlambatan dalam pembayaran.

Dengan memanfaatkan data historis, aplikasi dapat mendeteksi indikasi awal seperti keterlambatan yang terjadi berulang kali, jumlah pembayaran yang tidak konsisten, atau penurunan frekuensi pembayaran. Setelah pola-pola tersebut terdeteksi, sistem secara otomatis memberikan rekomendasi tindakan yang perlu diambil oleh manajemen atau tim penagihan. Langkah-langkah yang disarankan dapat berupa pengiriman surat peringatan, kunjungan lapangan oleh petugas penagihan, hingga pertimbangan untuk penarikan jaminan, bergantung pada tingkat risiko yang teridentifikasi. Dengan pendekatan ini, aplikasi rekomendasi memungkinkan perusahaan untuk merespons secara proaktif, cepat, dan berbasis data terhadap potensi permasalahan pembayaran dari debitur. Strategi ini tentu memberikan manfaat besar dalam menjaga stabilitas arus kas perusahaan serta mencegah kerugian jangka panjang akibat terjadinya kredit macet.(Lukman Hasim, n.d.)

### Phyton

Python merupakan bahasa pemrograman tingkat tinggi yang dikembangkan dengan tujuan utama menyederhanakan proses penulisan dan pembacaan kode. Sebagai bahasa yang bersifat interpreted dan serbaguna (*general-purpose*), Python dapat dimanfaatkan dalam berbagai bidang, mulai dari pengembangan aplikasi web, otomatisasi tugas, pemrosesan data, kecerdasan buatan (AI), hingga analitik data. Keunggulan utamanya terletak pada sintaksis yang ringkas dan mudah dimengerti, sehingga sesuai digunakan oleh pemula maupun pengembang berpengalaman.(1Kelly Hermanto, n.d.)

Python juga didukung oleh komunitas yang sangat aktif serta memiliki ribuan pustaka siap pakai, seperti Pandas, NumPy, Matplotlib, dan TensorFlow, yang mempercepat dan mempermudah proses pengembangan aplikasi. Berkat fleksibilitas dan kapabilitasnya yang luas, Python telah menjadi salah satu bahasa pemrograman paling diminati secara global(Riziq sirfatullah Alfarizi et al., 2023)

### Pandas

Pandas adalah pustaka dalam Python yang dirancang untuk menyederhanakan proses manipulasi dan analisis data. Pustaka ini menyediakan struktur data, khususnya *Series* untuk data satu dimensi dan *DataFrame* untuk data dua dimensi mirip tabel. Melalui Pandas, pengguna dapat melakukan berbagai operasi data seperti membaca dan menulis file (CSV, Excel, JSON), membersihkan data dari nilai kosong maupun duplikat, mengubah tipe data, menyaring baris atau kolom tertentu, hingga melakukan analisis statistik dasar. Pandas sangat populer di kalangan praktisi *data science*, *machine learning*, dan analis bisnis karena kemampuannya dalam mengelola. Oleh karena itu, Pandas menjadi salah satu pustaka utama yang perlu dikuasai dalam proyek-proyek pengolahan dan eksplorasi data

### Numpy

NumPy (*Numerical Python*) adalah pustaka dasar dalam Python yang dikembangkan khusus untuk kebutuhan komputasi ilmiah dan numerik. Pustaka ini menawarkan struktur array multidimensi (*ndarray*), serta menyediakan berbagai fungsi matematis tingkat lanjut untuk memungkinkan pemrosesan data secara vektorisasi tanpa perulangan eksplisit. Dengan fitur seperti *broadcasting*, operasi vektor, dan manajemen memori yang optimal, NumPy menjadi fondasi bagi pustaka lain dalam analisis data dan pembelajaran mesin seperti Pandas, Matplotlib, SciPy, hingga TensorFlow. NumPy sangat krusial dalam ekosistem Python karena mampu menangani data dalam jumlah besar, dan telah menjadi standar utama dalam pemrograman numerik. Banyak proyek riset dan rekayasa mengandalkan NumPy untuk perhitungan numerik presisi tinggi yang membutuhkan kinerja optimal.

### Matplotlib

Matplotlib merupakan pustaka visualisasi data dalam Python yang digunakan untuk membuat berbagai jenis grafik dan diagram. Pustaka ini sangat diminati oleh analis data, ilmuwan data, maupun peneliti karena kemampuannya menyajikan visualisasi yang informatif serta berkualitas tinggi. Dengan Matplotlib, pengguna dapat membangun beragam grafik seperti diagram batang, grafik garis, pie chart, histogram, scatter plot, dan visualisasi lainnya. Sering kali Matplotlib digunakan bersama Pandas untuk menampilkan hasil pengolahan data secara visual, sehingga lebih mudah dipahami dan dianalisis. Salah satu kelebihan utama Matplotlib adalah fleksibilitasnya dalam kustomisasi tampilan grafik, mulai dari warna, ukuran, label, hingga tata letak secara keseluruhan. Oleh karena itu, Matplotlib menjadi salah satu alat visualisasi utama dalam bidang data science, statistik, dan analitik bisnis

### Jupyter Notebook

Jupyter Notebook adalah format dokumen interaktif yang mendukung alur kerja komputasi ilmiah dengan mengintegrasikan kode program, hasil eksekusi, dan narasi penjelas dalam satu dokumen yang dapat dijalankan ulang (reproducible). Format ini memfasilitasi transparansi dalam penyampaian metode komputasi serta memungkinkan validasi oleh pihak lain, yang berkontribusi pada peningkatan kolaborasi dan kredibilitas dalam dunia akademik. Jupyter tidak hanya mendukung Python, tetapi juga kompatibel dengan lebih dari 50 bahasa pemrograman lainnya melalui sistem kernel, dan dapat diakses melalui browser baik secara lokal maupun dari server jarak jauh. Dokumen Jupyter disimpan dalam format terbuka .ipynb dan dapat dibagikan melalui berbagai alat seperti nbconvert, nbviewer, serta Binder, yang memungkinkan distribusi dalam bentuk statis maupun interaktif. Saat ini, Jupyter Notebook banyak digunakan dalam kegiatan akademik, pengajaran, serta penyusunan buku digital, menjadikannya perangkat penting dalam era keterbukaan ilmu pengetahuan.(Asyrofi & Asyrofi, 2023)

### Microsoft Excel

Microsoft Excel merupakan salah satu aplikasi dalam paket Microsoft Office yang dirancang secara khusus untuk mengelola data dalam bentuk angka, tabel, maupun grafik. Excel berperan sebagai lembar kerja digital (spreadsheet) yang memberikan kemudahan bagi pengguna dalam melakukan berbagai jenis perhitungan matematis, pengolahan data statistik, serta analisis data tingkat lanjut dengan bantuan rumus, fungsi, dan fitur otomatis lainnya.(Satria et al., n.d.)

Aplikasi ini telah menjadi pilihan utama di berbagai sektor, termasuk keuangan, akuntansi, pendidikan, dan manajemen proyek. Excel juga dilengkapi dengan beragam alat visualisasi data, seperti grafik batang, grafik garis, dan diagram lingkaran, yang memungkinkan penyajian informasi menjadi lebih jelas, menarik, dan mudah dipahami.

Microsoft Excel dikembangkan dan didistribusikan oleh Microsoft Corporation untuk sistem operasi Windows dan Mac OS. Aplikasi ini dibekali dengan kemampuan kalkulasi serta pembuatan grafik yang mendukung dalam pengolahan data numerik. Berkat fungsionalitasnya yang luas, Excel telah menjadi salah satu perangkat penting dalam pengelolaan dan penyajian data secara profesional.

### HTML

HTML (HyperText Markup Language) adalah bahasa markup standar yang digunakan untuk menyusun struktur dasar halaman web. HTML berfungsi untuk membangun elemen-elemen seperti teks, gambar, tautan, formulir, serta multimedia, sehingga konten dapat ditampilkan secara sistematis melalui browser. Versi terbarunya, HTML5, memperkenalkan elemen-elemen semantik seperti <header>, <footer>, <article>, dan <section> yang meningkatkan pemahaman konten oleh mesin pencari maupun alat bantu pembaca layar.

Dalam jurnal Modern Web Design: Utilizing HTML5, CSS3, and Responsive Techniques, HTML5 mengatasi keterbatasan versi sebelumnya dengan menghadirkan dukungan elemen multimedia seperti <audio> dan <video> tanpa perlu plugin tambahan. Selain itu, penggunaan elemen semantik pada HTML5 juga mendukung pengembangan web yang lebih terstruktur dan ramah pengguna.

### CSS

CSS (Cascading Style Sheets) adalah bahasa yang digunakan untuk mengatur tampilan visual halaman web yang dibangun dengan HTML. CSS memungkinkan pengaturan warna, tipografi, tata letak, serta animasi secara konsisten dan terpisah dari struktur konten. Versi terbaru, CSS3, memperkenalkan sejumlah fitur lanjutan seperti Flexbox, Grid Layout, media queries, serta efek transisi dan animasi.

CSS3 memberikan fleksibilitas dalam mendesain tampilan web yang responsif terhadap berbagai ukuran layar dan jenis perangkat. Fitur media queries mendukung tampilan adaptif, sedangkan properti seperti display: flex dan grid-template-columns mempercepat proses pembangunan tata letak kompleks tanpa perlu ketergantungan pada JavaScript tambahan.

## Aplikasi Web Dan Arsitekturnya

### Definisi Aplikasi Website

Aplikasi web merupakan jenis perangkat lunak yang dijalankan melalui peramban (browser) dan diakses melalui jaringan, baik itu internet maupun intranet. Pengguna dapat berinteraksi langsung dengan aplikasi tanpa perlu memasang program secara lokal pada perangkat mereka. Dengan hanya mengandalkan koneksi internet dan web browser, layanan seperti surat elektronik, pengelolaan dokumen, transaksi online, hingga sistem informasi akademik dapat diakses dengan mudah.

aplikasi web adalah program yang disimpan di server dan dikirimkan kepada pengguna melalui jaringan internet untuk kemudian diakses melalui antarmuka browser. Pandangan ini sejalan dengan yang dikemukakan dalam Jurnal Media Informatika (2020), yang menjelaskan bahwa aplikasi web merupakan perangkat lunak yang dapat diakses melalui web browser menggunakan jaringan seperti internet maupun intranet. Selaras dengan itu, mendefinisikan aplikasi web sebagai sebuah sistem aplikasi yang memanfaatkan teknologi browser untuk menjalankannya dan diakses melalui jaringan komputer.(Alfeno et al., n.d.)

Adapun keunggulan dari aplikasi web meliputi kemudahan akses dari berbagai lokasi, tidak memerlukan proses instalasi pada perangkat pengguna, proses pembaruan aplikasi yang hanya dilakukan di sisi server, serta kompatibilitas dengan berbagai jenis perangkat seperti desktop, tablet, maupun ponsel pintar. Beberapa contoh aplikasi web yang umum digunakan dalam kehidupan sehari-hari antara lain Gmail, Google Docs, serta platform e-commerce seperti Shopee dan Tokopedia.(Maulida Solihat et al., 2021; Permata Sari, n.d.)

## Arsitektur Aplikasi Web (Frontend, Backend, Database)

### Frontend

Frontend merupakan bagian dari aplikasi web yang secara langsung berinteraksi dengan pengguna. Bagian ini mencakup berbagai elemen antarmuka seperti halaman masuk (login), formulir input, tabel data, menu navigasi, serta komponen visual lainnya. Dalam pengembangannya, frontend umumnya dibangun menggunakan teknologi seperti HTML, CSS, dan JavaScript. Untuk menciptakan tampilan yang interaktif dan responsif, pengembang sering memanfaatkan framework modern seperti React, Vue.js, atau Angular.(Emirzaki et al., n.d.)

penerapan arsitektur micro-frontend dapat secara signifikan mempercepat waktu build proyek web—dari sebelumnya sekitar 30–50 menit menjadi hanya 1–2 menit. Namun demikian, pendekatan ini menyebabkan waktu pemuatan awal aplikasi sedikit lebih lama, dengan peningkatan sekitar 2–3 detik dibandingkan arsitektur monolitik.

### Backend

Backend merupakan bagian dari aplikasi yang menjalankan logika utama sistem. Komponen ini bertugas menangani permintaan yang dikirimkan oleh pengguna melalui antarmuka frontend, melakukan validasi, memproses data, serta memberikan respon yang relevan. Selain itu, backend berperan sebagai penghubung antara antarmuka pengguna dan basis data. Beberapa bahasa pemrograman yang sering digunakan untuk membangun backend antara lain PHP, Python, Java, serta JavaScript dengan lingkungan runtime Node.js.(Pangestika & Dirgahayu, n.d.)

penerapan backend berbasis REST API menggunakan Node.js telah berhasil diimplementasikan pada aplikasi layanan transaksi elektronik, di mana proses komunikasi antara frontend dan backend berjalan dengan baik melalui pertukaran data dalam format JSON.

### Database

Database merupakan sistem yang digunakan untuk menyimpan berbagai data yang diperlukan dalam aplikasi web. Seluruh informasi penting, seperti data pengguna, transaksi, konten, hingga pengaturan sistem, tersimpan di dalam database. Beberapa sistem manajemen basis data (Database Management System atau DBMS) yang umum digunakan meliputi MySQL, PostgreSQL, MongoDB, dan SQLite. Komponen backend bertugas untuk berkomunikasi langsung dengan database dalam proses penyimpanan maupun pengambilan data sesuai kebutuhan pengguna.

membangun backend sistem manajemen aset menggunakan MySQL. Mereka menekankan bahwa penggunaan teknologi seperti Node.js, yang mendukung pendekatan non-blocking, memungkinkan sistem menangani banyak permintaan secara bersamaan tanpa mengorbankan kinerja aplikasi.

## Pengembangan Frontend

### React.js

React merupakan pustaka JavaScript yang dirancang untuk membangun antarmuka pengguna (user interface) secara modular. Dikembangkan oleh Facebook sejak tahun 2013, React berfokus pada elemen tampilan (view) dalam kerangka kerja Model-View-Controller (MVC), dan mendukung pengembangan komponen-komponen yang dapat digunakan kembali. Pendekatan berbasis komponen ini menjadikan React sangat ideal untuk membangun aplikasi berskala besar dengan kebutuhan perubahan data yang dinamis(Murti et al., n.d.)

### Komponen

Komponen merupakan elemen paling mendasar dalam React yang merepresentasikan bagian antarmuka pengguna (UI) yang bersifat independen, seperti tombol, formulir, atau tampilan halaman tertentu. Komponen dapat dibangun menggunakan dua pendekatan, yaitu sebagai functional component maupun class component. Namun, saat ini pendekatan berbasis fungsi lebih banyak digunakan karena sintaksnya yang lebih ringkas dan mudah dipahami.

React adalah pustaka JavaScript yang digunakan untuk membangun komponen UI yang dapat digunakan kembali. React telah menjadi pilihan utama dalam pengembangan antarmuka berbasis arsitektur Model-View-Controller (MVC), di mana React berperan dalam menyusun bagian view. Pendekatan ini mendorong pengembang untuk merancang UI yang bersifat modular, dapat digunakan kembali, dan mudah untuk diuji.

### JSX

JSX merupakan sintaks khusus yang memungkinkan penulisan kode menyerupai HTML di dalam file JavaScript. Dengan menggunakan JSX, pengembang dapat mendeskripsikan tampilan antarmuka secara lebih jelas, sekaligus membuat struktur kode menjadi lebih mudah dibaca karena mirip dengan HTML pada umumnya. JSX digambarkan sebagai “ekstensi sintaks JavaScript” yang membuat penulisan kode menjadi lebih ringkas dan mudah dipahami, sehingga mempermudah proses pembuatan dan pemahaman komponen dalam pengembangan React.

### State

State merupakan data lokal yang dimiliki oleh suatu komponen dan dapat mengalami perubahan seiring berjalannya waktu. Setiap kali terjadi perubahan pada state, React akan secara otomatis melakukan proses render ulang pada komponen tersebut, sehingga tampilan antarmuka selalu mencerminkan kondisi data terkini. Sebagai contoh, interaksi pengguna seperti pengisian formulir atau perubahan status pada tombol akan direkam melalui state. pengelolaan state dalam React memiliki peran yang sangat krusial untuk memastikan konsistensi antara data dan tampilan, terutama pada aplikasi yang bersifat dinamis dan interaktif.

### Props(Properties)

Props merupakan data yang dikirimkan dari komponen induk ke komponen anak dalam React. Data ini bersifat hanya-baca (read-only), sehingga mendukung prinsip alur data satu arah (one-way data binding) yang memudahkan pengelolaan data serta mengurangi risiko terjadinya perubahan yang tidak diinginkan. Sebuah artikel dalam jurnal ReactJS menyoroti bahwa penggunaan props sangat penting dalam menjaga modularitas komponen serta memastikan keamanan data selama proses pengembangan aplikasi.

### Virtual DOM

Virtual DOM (Virtual Document Object Model) merupakan representasi tiruan dari struktur antarmuka pengguna (UI) yang disimpan dalam memori. Konsep ini diterapkan oleh pustaka JavaScript seperti React untuk meningkatkan proses pembaruan tampilan pada halaman web. Daripada langsung memodifikasi DOM asli di browser—yang dikenal cukup lambat dan membutuhkan banyak sumber daya—Virtual DOM bekerja dengan mencatat perubahan terlebih dahulu pada versi virtualnya. Setelah itu, React akan melakukan proses diffing, yakni membandingkan versi Virtual DOM terbaru dengan versi sebelumnya untuk mengidentifikasi perubahan yang terjadi. Hanya perubahan yang relevan dan diperlukan saja yang kemudian diterapkan ke DOM asli.

Tahapan ini dikenal sebagai reconciliation.Pendekatan ini secara signifikan meningkatkan performa aplikasi, karena mengurangi interaksi langsung dengan DOM fisik—yang secara tradisional menjadi salah satu faktor utama menurunnya kecepatan aplikasi web.

## Metode Pengumpulan Data

Metode pengumpulan yang digunakan untuk melakukan project ini memakai metode dokumentasi terhadap dokumen yang telah diberikan oleh perusahaan PT Capella Multidana sebagai sumber informasi projek ini Metode ini dilakukan dengan cara mengakses dan menelaah laporan yang telah disusun dan disimpan oleh pihak perusahaan. Data yang digunakan diperoleh dalam bentuk file digital (Excel) yang berisi data data debitur terkait PT Capella Multidana (Darudiato et al., n.d.)

Dengan demikian, metode dokumentasi menjadi pendekatan dalam pengumpulan data untuk penelitian ini, mengingat ketersediaan data dari perusahaan sudah mencukupi kebutuhan informasi yang akan dianalisis, baik dari segi volume, isi, maupun keabsahannya sebagai sumber data yang dapat dipertanggungjawabkan.(Rizky et al., 2021)

### **Alat**

1. Microsoft Excel

Aplikasi spreadsheet yang dikembangkan oleh Microsoft. Ini digunakan untuk mengelola, menganalisis, dan memvisualisasikan data dalam bentuk tabel. Data disimpan dalam bentuk tabel dengan kolom (A-U) dan baris yang terstruktur. Setiap kolom mewakili variabel tertentu (misalnya, NAMA DEBITUR, JENIS PEMBIAYAAN, TGL JTO, dll.), sementara setiap baris mewakili entri data individu.dan setiap sheet mewakili untuk setiap dealer yang berbeda

### Teknik

1. Data Cleansing

Pembersihan data dilakukan untuk memastikan data bersih, valid, dan konsisten. Ini mencakup perbaikan data kosong, penghapusan duplikat, standarisasi penulisan teks seperti nama pendidikan atau status pernikahan, dan pengecekan tipe data agar sesuai, misalnya umur dan penghasilan harus dalam format numerik. Tanpa pembersihan, hasil analisis bisa menyesatkan.

1. Transfromasi Data

Transformasi data adalah proses mengubah data mentah ke dalam bentuk yang lebih terstruktur, terstandar, atau bermakna untuk dianalisis lebih lanjut. Transformasi juga mencakup normalisasi atau standarisasi nilai numerik agar memiliki skala yang seragam, agar data dapat lebih muda digunakan untuk penelitian

1. Visualisasi Data

Visualisasi data bertujuan untuk menyajikan informasi dalam bentuk grafis agar mudah dipahami. Visualisasi membantu dalam penyajian hasil analisis kepada pemangku kepentingan. Visualisasi data adalah proses menyajikan data dalam bentuk grafis atau visual, seperti grafik, diagram, atau peta, dengan tujuan untuk memudahkan pemahaman pola, tren, dan hubungan antarvariabel dalam data. Dengan visualisasi, informasi kompleks dapat disampaikan dengan cara yang lebih cepat dan intuitif dibandingkan hanya menggunakan tabel angka, Visualisasi sangat berguna dalam pengambilan keputusan berbasis data karena mampu menyoroti hal-hal penting secara visual dan langsung.

## Visual Studio Code

Visual Studio Code (VS Code) merupakan editor kode sumber yang dikenal karena sifatnya yang ringan, cepat, namun tetap memiliki kemampuan yang kuat. Aplikasi ini mendukung instalasi pada tiga sistem operasi utama—Windows, macOS, dan Linux—sehingga memberikan fleksibilitas bagi para pengembang untuk bekerja di platform yang mereka pilih. Secara default, VS Code telah dilengkapi dengan fitur IntelliSense, debugging, dan potongan kode (snippet) untuk bahasa seperti JavaScript, TypeScript, dan Node.js.

Meski demikian, nilai unggul dari VS Code terletak pada Marketplace ekstensi yang dimilikinya. Hanya dengan satu klik, pengguna dapat menambahkan dukungan menyeluruh untuk berbagai bahasa pemrograman, seperti C/C++, C#, Java, Python, PHP, Go, hingga ekosistem modern termasuk .NET, Unity, dan Docker.

Fitur lain yang sering mendapat apresiasi dari pengguna adalah integrasi Git yang sudah tertanam, terminal internal, serta opsi untuk menyesuaikan tema dan pintasan keyboard sesuai preferensi. Seluruh fitur ini dirancang untuk mendukung alur kerja pengembangan perangkat lunak, tanpa mengorbankan kinerja sistem. Berkat kombinasi antara portabilitas tinggi, ekosistem ekstensi yang sangat kaya, dan antarmuka pengguna yang ramah, VS Code telah menjadi salah satu editor pilihan utama baik bagi pengembang profesional maupun pemula. (Presman, 2010)

## Xampp

XAMPP merupakan perangkat lunak yang dirancang untuk menjalankan situs web berbasis PHP secara lokal di komputer. Aplikasi ini hadir sebagai paket instalasi lengkap yang mencakup beberapa komponen inti, di antaranya Apache sebagai server web, MySQL sebagai sistem manajemen basis data, serta bahasa pemrograman PHP dan Perl yang didukung secara langsung.

Melalui XAMPP, para pengembang dapat membangun, menguji, dan mengelola situs web atau aplikasi berbasis web tanpa perlu terhubung ke internet ataupun menyewa layanan hosting. Hal ini sangat bermanfaat, khususnya dalam tahap awal pengembangan dan pengujian proyek. XAMPP memudahkan pengguna—terutama bagi mereka yang masih pemula—dalam mempelajari pemrograman web serta pengelolaan basis data, berkat pengaturan yang sederhana dan mudah digunakan.

## User Interface

User Interface (UI) merupakan media atau sarana yang memungkinkan pengguna untuk berinteraksi dengan sistem komputer atau aplikasi. UI mencakup seluruh elemen visual yang dapat dilihat dan digunakan oleh pengguna, seperti tombol, menu, ikon, skema warna, hingga tata letak layar, yang semuanya dirancang untuk menjalankan fungsi-fungsi tertentu dalam sistem. Sebagaimana dijelaskan oleh Jamilah & Padmasari (n.d.), UI sering dikaitkan dengan bidang Human-Computer Interaction (HCI) karena memiliki peran penting dalam menjembatani komunikasi antara manusia dan komputer.

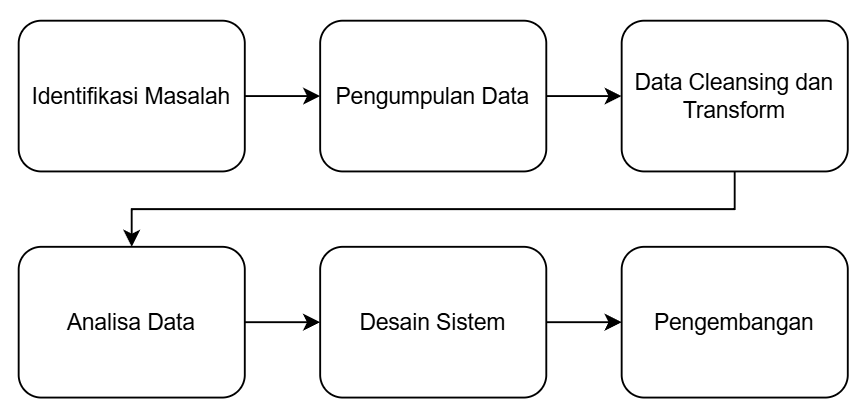
Secara sederhana, User Interface adalah bagian dari sistem—baik perangkat keras maupun perangkat lunak—yang mengatur penyusunan tampilan aplikasi agar mudah dioperasikan dan enak dipandang. Kehadiran UI yang dirancang dengan baik berkontribusi besar dalam menciptakan pengalaman penggunaan yang menyenangkan.

Selain itu, UI dapat dipahami sebagai representasi visual dari proses User Experience (UX), yakni keseluruhan interaksi pengguna saat berhadapan dengan sistem. Oleh karena itu, tampilan yang terstruktur, mudah dimengerti, serta responsif, merupakan hasil dari perencanaan UX yang matang dan terlihat nyata melalui elemen-elemen UI.

# BAB III

# METODOLOGI

## Desain Penelitian



Gambar 3. 1Desain Metodologi

Penelitian ini menggunakan pendekatan Research and Development (R&D) dengan studi kasus pada PT Capella Multidana, sebuah perusahaan yang bergerak di bidang pembiayaan kendaraan. Pendekatan ini dipilih karena tujuan utama penelitian adalah mengembangkan sistem rekomendasi tindakan kepada debitur berbasis Business Intelligence (BI), yang berfungsi untuk memberikan notifikasi keterlambatan pembayaran serta mendukung pengambilan keputusan oleh pihak manajemen. Model R&D yang digunakan melibatkan tahapan sistematis mulai dari identifikasi permasalahan, pengumpulan data, pembersihan dan transformasi data, analisis data, hingga desain dan implementasi sistem berbasis dashboard interaktif.(Oktaviani & Harahap, 2022)

1. Identifikasi Masalah: Berdasarkan hasil observasi terhadap proses kerja di PT Capella Multidana, ditemukan bahwa perusahaan belum memiliki sistem yang mampu memberikan notifikasi secara langsung ketika debitur mengalami keterlambatan pembayaran. Kondisi ini menyebabkan informasi penting tidak segera sampai ke pihak manajemen, sehingga keterlambatan dalam penanganan kasus debitur berisiko menjadi sulit dihindari.

Ketiadaan sistem klasifikasi berbasis data membuat penilaian terhadap tingkat keterlambatan dan profil risiko debitur menjadi lambat dan kurang terstruktur. Sebagian besar analisis masih bergantung pada laporan statis, tanpa dukungan visualisasi real-time yang dapat menggambarkan kondisi lapangan secara aktual. Hal ini berdampak pada tidak konsistennya tindakan yang diambil oleh tim penagihan karena tidak ada acuan sistematis terhadap status keterlambatan dan tingkat risiko masing-masing debitur.

Selain itu, perusahaan juga belum memiliki alat bantu yang mampu memberikan rekomendasi tindakan secara otomatis berdasarkan kombinasi antara lama keterlambatan dan klasifikasi risiko (rendah, menengah, tinggi). Padahal, kebutuhan untuk membedakan penanganan antara debitur dengan keterlambatan ringan dan mereka yang telah masuk kategori kritis menjadi sangat penting agar sumber daya perusahaan dapat diarahkan secara lebih terarah.

Permasalahan ini diperkuat oleh data historis dari tahun 2003 hingga 2025 yang menunjukkan banyaknya debitur bermasalah yang tidak teridentifikasi sejak awal. Tanpa sistem yang dapat mengelompokkan status debitur ke dalam kategori seperti *Hijau*, *Kuning*, dan *Merah*, serta memberikan arahan tindakan seperti SP1, SP2, hingga penarikan kendaraan, maka manajemen kesulitan dalam mengatur prioritas intervensi.

Oleh karena itu, dibutuhkan sebuah sistem berbasis Business Intelligence yang dapat mengintegrasikan data historis debitur, memproses dan menyajikannya dalam bentuk dashboard interaktif, serta memberikan rekomendasi tindakan secara otomatis sesuai dengan klasifikasi risiko dan durasi keterlambatan. Sistem ini diharapkan dapat membantu perusahaan dalam melakukan deteksi dini dan mengambil tindakan yang sesuai secara konsisten dan terarah.

1. Pengumpulan Data: Teknik pengumpulan data yang digunakan dalam penelitian ini adalah metode dokumentasi internal, yaitu dengan memanfaatkan data digital yang diperoleh langsung dari pihak PT Capella Multidana. Data tersebut merupakan arsip internal perusahaan yang mencakup informasi terkait status pembayaran, identitas debitur, jenis pembiayaan, tenor, tanggal jatuh tempo, hingga status keterlambatan. Seluruh data tersedia dalam format Microsoft Excel (.xlsx), dan digunakan sebagai dasar utama untuk pengembangan sistem Business Intelligence dalam penelitian ini. Dokumen yang diperoleh bersifat data sekunder dan telah direkap oleh tim administrasi perusahaan, sehingga dianggap memiliki validitas yang cukup tinggi. Akses terhadap data diberikan untuk keperluan akademik melalui komunikasi internal tim yang terlibat dalam proyek ini.

Analisis data dalam proyek ini dilakukan melalui tahapan berikut:

**Data Raw (Data Mentah)**

Gambar 3. 2Data Raw

Data Berisikan: Contract Number, Nama Konsumen, ID Pelanggan, Contract Terms, Contract Value, 1st DueDate, Last DueDate, Monthly Installments, Balance Receivables, Receivables NotYetDue, OD (01–30), OD Angsuran (01–30), OD (31–60), OD Angsuran (31–60), OD (≥61), OD Angsuran (≥61), Overdue Jumlah Hari, Overdue Angsuran, Kolektabilitas, OD Pokok (01–30), OD Bunga (01–30), OD Pokok (31–60), OD Bunga (31–60), OD Pokok (≥61), OD Bunga (≥61), Last Payment Installments, Last Payment DueDate, Last Payment TrxDate, Last Payment Value, Vehicles Category, Vehicles Type, Vehicles Number, Chassis Number, Engine Number, Salesman, Supervisor, Customer Address (ID), Kelurahan (ID), Kecamatan (ID), Kabupaten (ID), Customer Address (Domicile), Kelurahan (Domicile), Kecamatan (Domicile), Kabupaten (Domicile), WO Acc, WO Mkt, Tarik / Tebus, Tgl Tarik, Tgl Tebus, OTR, DP, Biaya Administrasi Lainnya, Keterangan Bidang Usaha (KetBidUs), Tujuan, Addm Addb, Flat, Efek

Data mentah merupakan hasil awal yang diperoleh langsung dari PT Capella Multidana sebelum melalui proses pembersihan (*cleansing*) dan transformasi. Data ini berfungsi sebagai sumber utama dalam menyusun sistem *Business Intelligence* untuk klasifikasi dan analisis keterlambatan pembayaran debitur.

Dataset berisi **berbagai atribut rinci** yang menggambarkan kondisi kontrak kredit, status pembayaran, dan profil pelanggan secara menyeluruh. Beberapa kategori utama dalam raw data ini antara lain:

1. **Identitas Debitur dan Kontrak Kredit**
   * Contract Number, Nama Konsumen, ID Pelanggan, Contract Terms (durasi tenor), dan Contract Value
   * Tanggal penting seperti 1st DueDate, Last DueDate, serta nilai Monthly Installments
2. **Status Pembayaran dan Piutang**
   * Kolom Balance Receivables, Receivables NotYetDue, dan OD (Overdue) yang dibagi berdasarkan rentang hari keterlambatan seperti:
     + OD (01–30), OD (31–60), OD (>=61)
     + Baik pokok maupun bunga (OD Pokok, OD Bunga) dan angsuran
   * Overdue Jumlah Hari untuk menghitung keterlambatan spesifik
   * LastPayment Installments, LastPayment Value, LastPayment DueDate, dan Trx Date
3. **Kolektabilitas dan Kategori Status**
   * Status Kolektabilitas seperti *LANCAR* atau *Dalam Perhatian Khusus*
   * Data ini digunakan sebagai salah satu dasar awal dalam sistem penilaian keterlambatan
4. **Informasi Kendaraan**
   * Vehicles Category, Vehicles Type, Chassis Number, Engine Number, dan Vehicle Number
5. **Alamat dan Wilayah Debitur**
   * Alamat pelanggan dalam format ID maupun domisili (Kelurahan, Kecamatan, Kabupaten)
6. **Data Tambahan Operasional**
   * Seperti Salesman, Supervisor, WO Account, status Tarik/Tebus, Tgl Tarik, OTR, DP, biaya administrasi lainnya, serta variabel keuangan tambahan (Addm Addb, Flat, Efek)

**Data Cleaning (Pembersihan Data):**

Proses *data cleansing* merupakan tahap penting dalam pengolahan data yang bertujuan untuk memastikan bahwa data yang digunakan dalam sistem telah bebas dari informasi yang tidak relevan, redundan, maupun tidak sesuai dengan tujuan analisis. Dalam penelitian ini, data cleansing dilakukan terhadap dataset yang diperoleh dari PT Capella Multidana untuk menyederhanakan struktur data dan menghilangkan kolom-kolom yang tidak dibutuhkan dalam proses klasifikasi keterlambatan debitur dan pemberian rekomendasi tindakan.

Berdasarkan hasil evaluasi awal terhadap struktur dataset, terdapat sebanyak **42 kolom** yang dinilai tidak memiliki kontribusi signifikan terhadap analisis utama. Kolom-kolom tersebut terdiri dari berbagai informasi seperti identitas kendaraan (Chassis Number, Engine Number, Vehicles Number), informasi administratif (Customer Address, Kelurahan, Kecamatan, Kabupaten baik versi ID maupun domisili), data historis kolektabilitas dan overdue (OD, OD Angsuran, OD Pokok, OD Bunga), serta informasi operasional lainnya seperti Salesman, Supervisor, WO Acc, Tarik/Tebus, Tgl Tarik, dan Efek.

Penghapusan dilakukan dengan pertimbangan bahwa:

* Beberapa kolom bersifat **redundan** dan memiliki turunan nilai yang sudah direpresentasikan dalam atribut utama lainnya.
* Kolom seperti OTR, Flat, Addm Addb, dan Efek tidak memberikan pengaruh terhadap penentuan status keterlambatan maupun profil risiko.
* Beberapa kolom seperti Contract Number, Last Due Date, dan Receivables Not Yet Due bersifat administratif dan tidak relevan untuk klasifikasi sistem warna (Hijau, Kuning, Merah).
* Informasi wilayah (Kelurahan, Kecamatan, Kabupaten) tidak digunakan dalam perhitungan keterlambatan atau rekomendasi sistem, sehingga dapat diabaikan dalam konteks ini.

Dengan menghilangkan kolom-kolom yang tidak diperlukan, struktur data menjadi lebih ringkas dan lebih mudah diolah dalam proses transformasi dan analisis. Proses ini juga mempercepat eksekusi sistem saat melakukan klasifikasi debitur, serta meningkatkan keakuratan output karena hanya variabel penting yang dipertahankan untuk membentuk dasar pengambilan keputusan.

Selanjutnya, dataset yang telah dibersihkan digunakan dalam tahapan transformasi data, visualisasi, serta pengembangan fitur notifikasi dan rekomendasi tindakan yang diotomatisasi berdasarkan tingkat keterlambatan dan risiko.

Gambar 3. 3 Data Cleansing

Data Berisikan : Nama Kons,Id Pelanggan, Contract Terms, Contract Value,1st DueDate, Monthly Installments, LastPayment Installments, LastPayment DueDate, LastPayment TrxDate, LastPayment Value, Vehicles Category, DP , Biaya Adm Lainnya, KetBidUs ,Tujuan

**Data Transformation (Transformasi Data):**

Setelah proses pembersihan data selesai dilakukan, tahap selanjutnya adalah **transformasi data**, yaitu proses penyesuaian nilai dan struktur data agar sesuai dengan kebutuhan sistem dan siap digunakan dalam analisis. Transformasi ini bertujuan untuk menyederhanakan format data, meningkatkan keterbacaan, dan menambahkan informasi baru yang diperlukan dalam proses klasifikasi risiko dan penentuan status keterlambatan debitur.

Beberapa proses transformasi yang dilakukan antara lain:

Pertama, dilakukan **penyesuaian format** pada beberapa kolom numerik dan tanggal yang sebelumnya tidak konsisten. Perubahan format dilakukan pada kolom Down Payment (DP), Nilai Kontrak, Biaya Angsuran, Nominal Last Payment, Tanggal Awal Kontrak, Tanggal Jatuh Tempo (Tgl Jto Last Payment), dan Tanggal Pembayaran Terakhir (Tgl Last Payment) agar seluruh data memiliki format seragam dan dapat dikenali oleh sistem secara otomatis saat dianalisis.

Kedua, dilakukan **penyederhanaan rentang tenor** kredit. Awalnya, tenor memiliki variasi dari 6 hingga 72 bulan. Namun, untuk mempermudah segmentasi dan analisis, rentang tersebut dikurangi menjadi antara 12 hingga 36 bulan, yang merupakan kisaran paling dominan dalam data dan paling relevan untuk pengambilan keputusan.

Ketiga, sistem menambahkan **kolom perhitungan hari keterlambatan** yang dihitung berdasarkan selisih antara tanggal jatuh tempo dan tanggal pembayaran terakhir. Kolom ini berguna untuk mengukur berapa hari keterlambatan yang dialami oleh masing-masing debitur, atau mengidentifikasi apakah debitur masih membayar tepat waktu.

Keempat, ditambahkan **kolom status** yang secara otomatis mengklasifikasikan kondisi debitur ke dalam kategori *lancar* atau *telat* berdasarkan hasil perhitungan hari keterlambatan. Kategori ini menjadi dasar dalam sistem untuk menentukan apakah debitur akan dimasukkan ke dalam profil risiko dan status warna tertentu (Hijau, Kuning, Merah).

Transformasi ini menjadi fondasi penting dalam membangun sistem klasifikasi otomatis berbasis Business Intelligence yang bertujuan memberikan peringatan serta rekomendasi tindakan kepada manajemen secara real-time.

Gambar 3. 4 Data Transformation

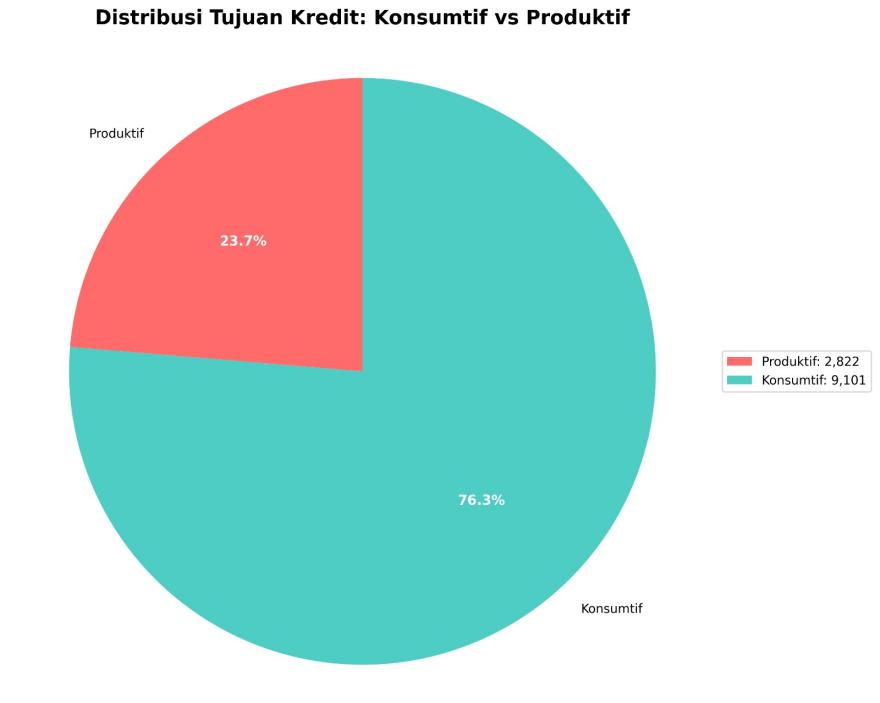
Data Berisikan : Nama Debitur, Id Payment, DP, Tenor, Nilai Kontrak, Tanggal Awal Kontrak, Biaya Angsuran Per Bulan, Angsuran Last Payment, Tgl Jto Last Payment, Tgl Last Payment, Perhitungan Hari Telat, Status, Nominal Last Payment, Kategori Kendaraan, Keterangan Bidang Usaha, Tujuan

**Analisis Deskriptif dan Visualisasi Business Intelligence:**

* + Menganalisis distribusi keterlambatan berdasarkan jenis pembiayaan, bidang usaha, serta tenor kredit.
  + Mengidentifikasi segmen debitur dengan risiko tinggi berdasarkan histori keterlambatan.

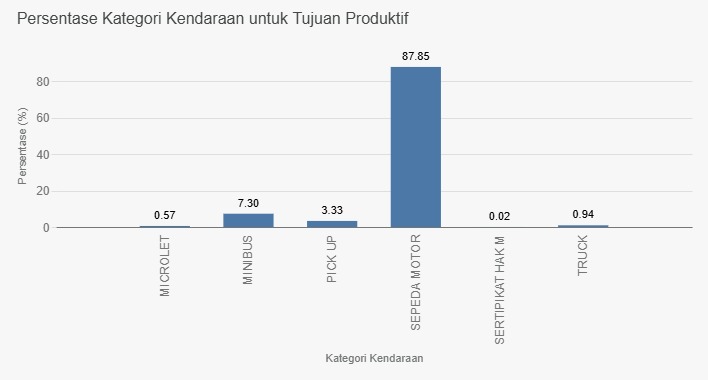
Menggunakan diagram batang, pie chart, dan scatter plot untuk menampilkan insight seperti proporsi keterlambatan, nilai kontrak berdasarkan tenor, serta hubungan antara jenis usaha dengan kecenderungan keterlambatan.

* Tujuan Kredit



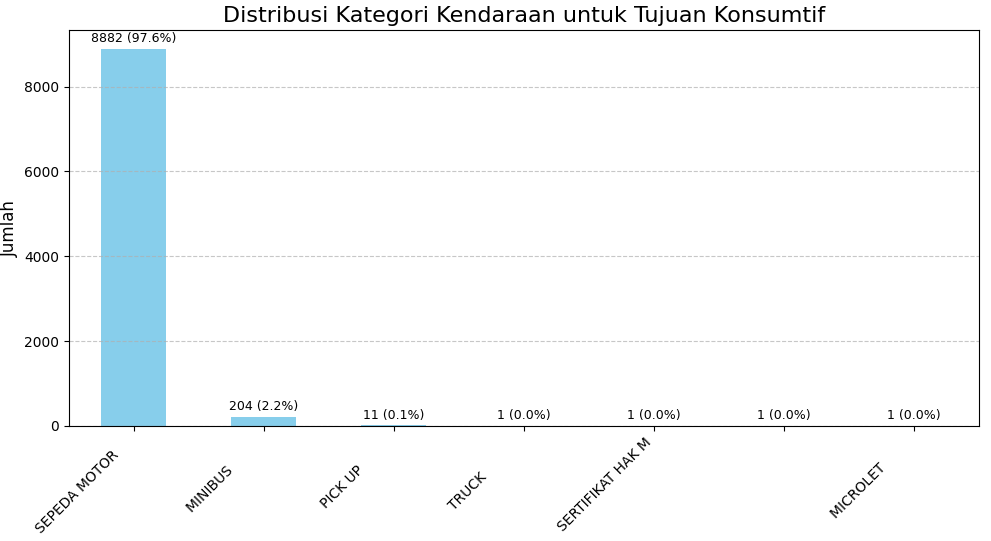
Gambar 3. 5 Visualisasi Tujuan Kredit

* Kategori Kendaraan Produktif



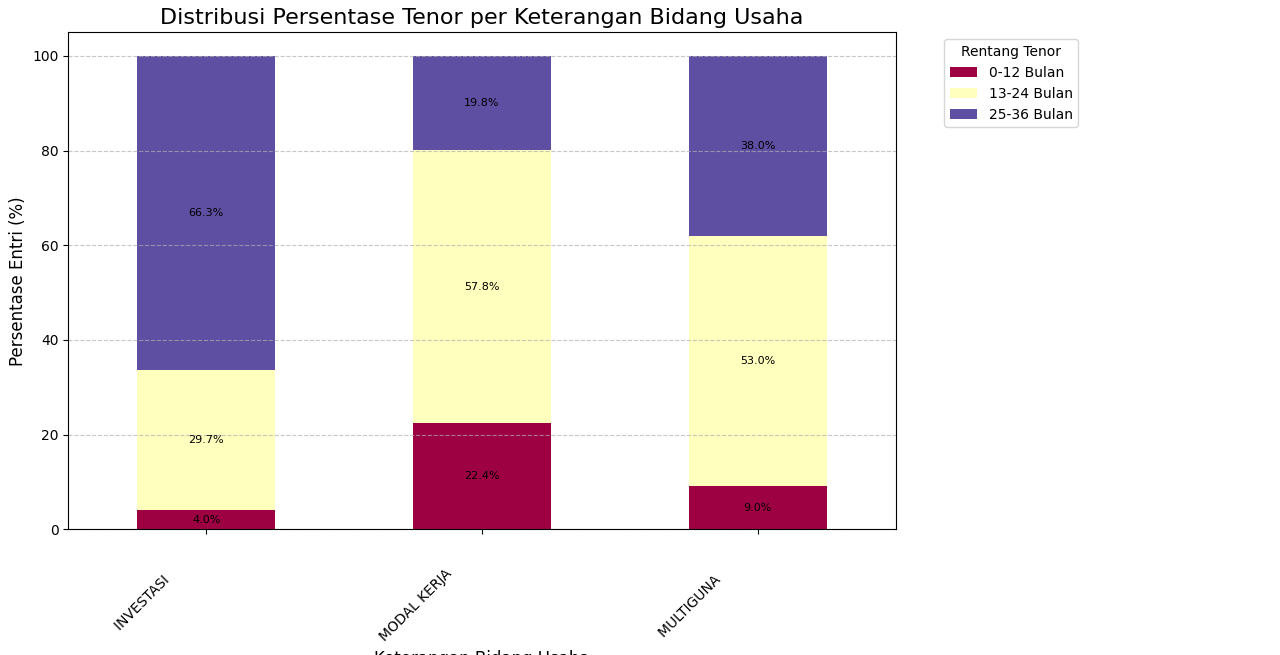
Gambar 3. 6 Visualisasi Kategori Kendaraan Produktif

* Kategori Kendaraan Konsumtif



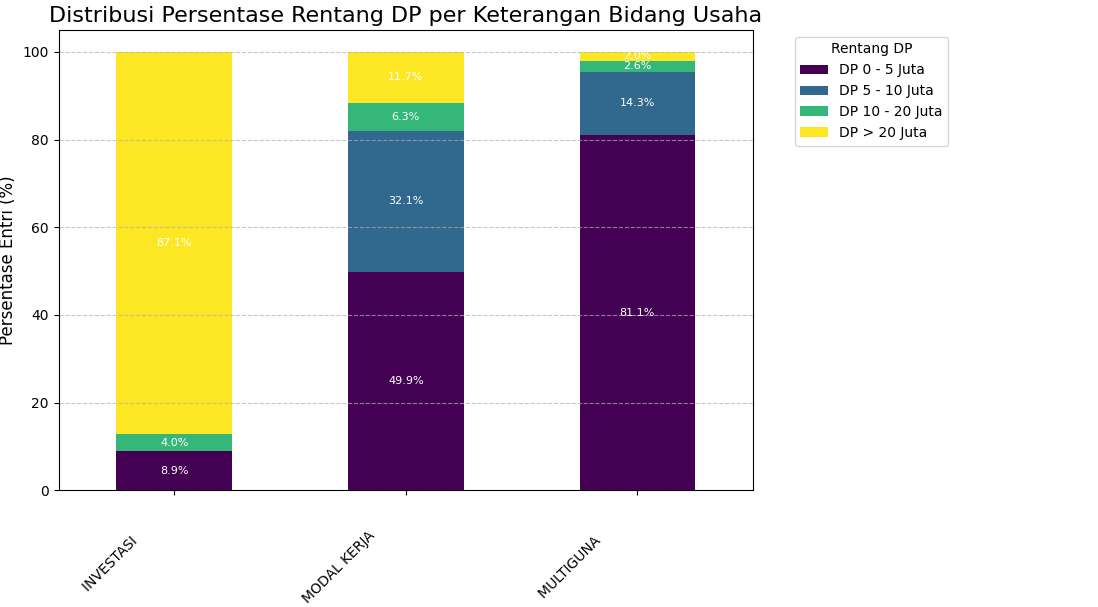
Gambar 3. 7 Visualisasi Kategori Kendaraan Konsumtif

* Tenor Per Keterangan Bidang Usaha



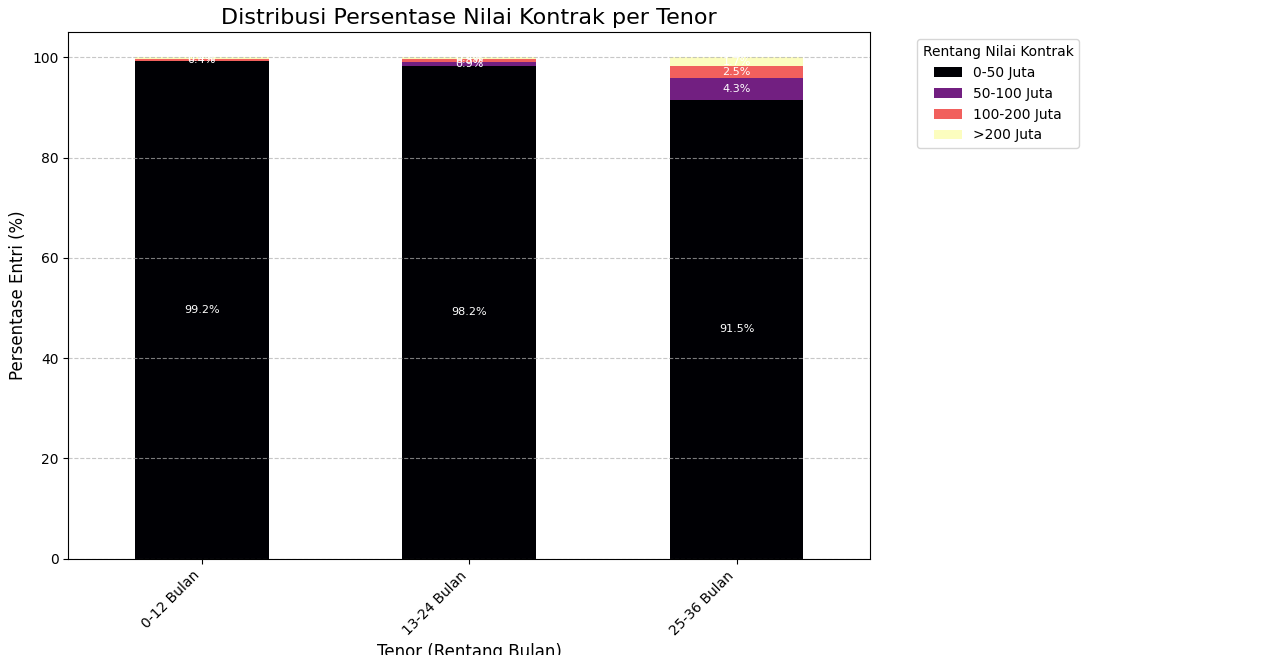
Gambar 3. 8 Visualisasi Tenor Per Keterangan Bidang Usaha

* DP Per Keterangan Bidang Usaha



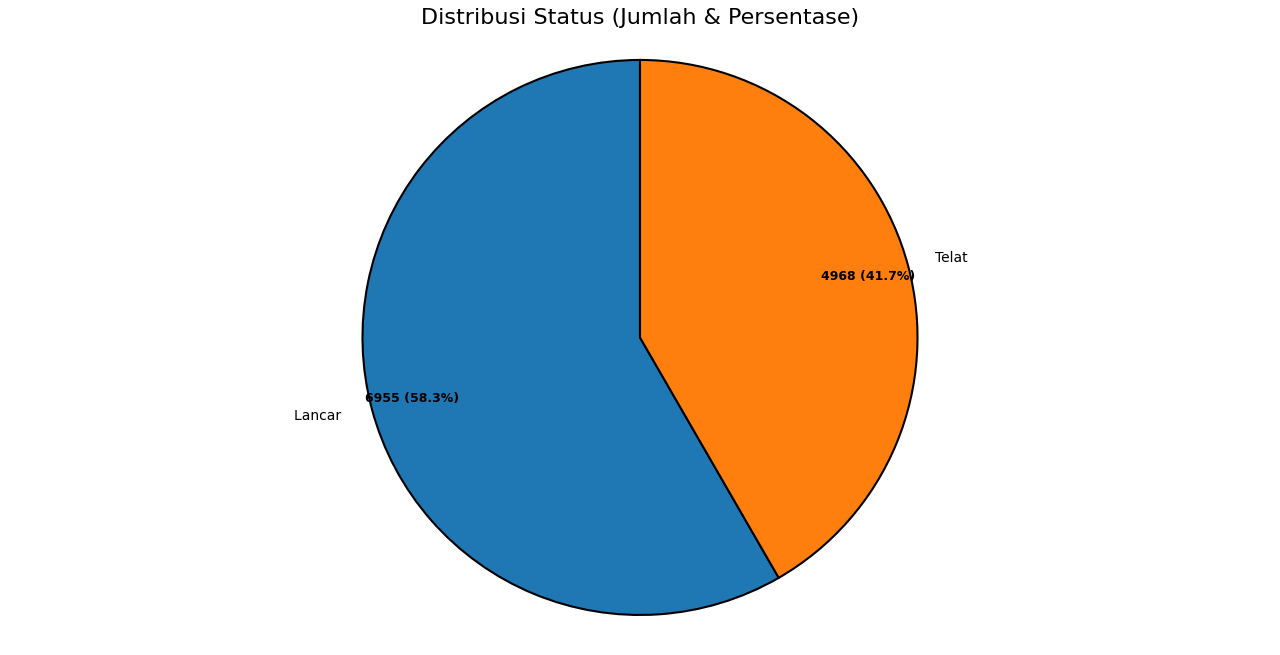
Gambar 3. 9 Visualisai DP Per Keterangan Bidang Usaha

* Nilai Kontrak Per Tenor



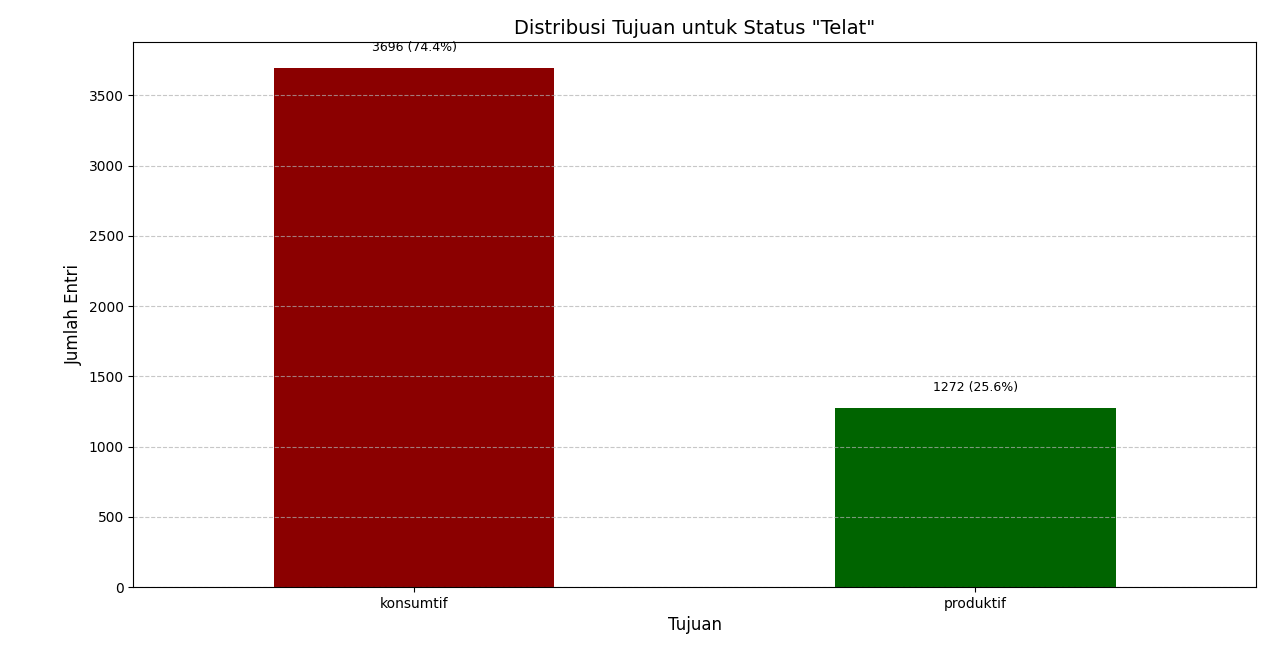
Gambar 3. 10 Visualisasi Nilai Kontrak Per Tenor

* Status Debitur



Gambar 3. 11 Visualisasi Status Debitur

* Tujuan Status Debitur Telat



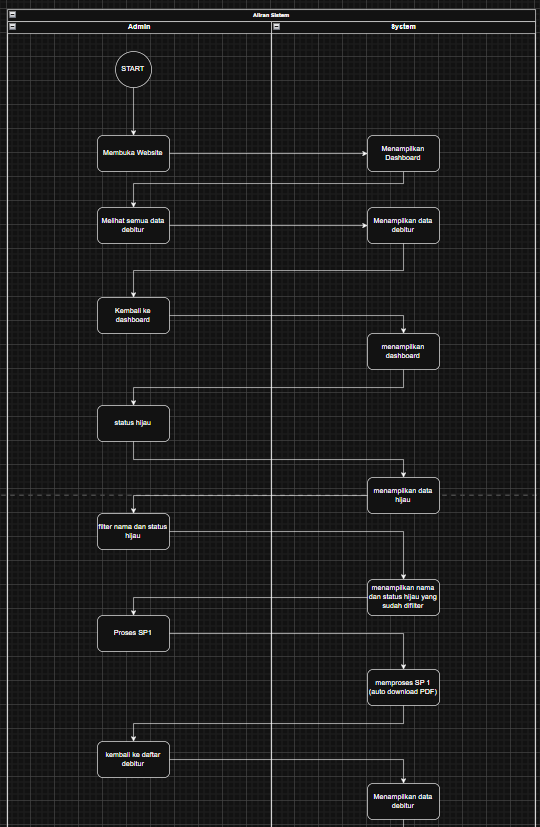
Gambar 3. 12 Tujuan Status Debitur Telat

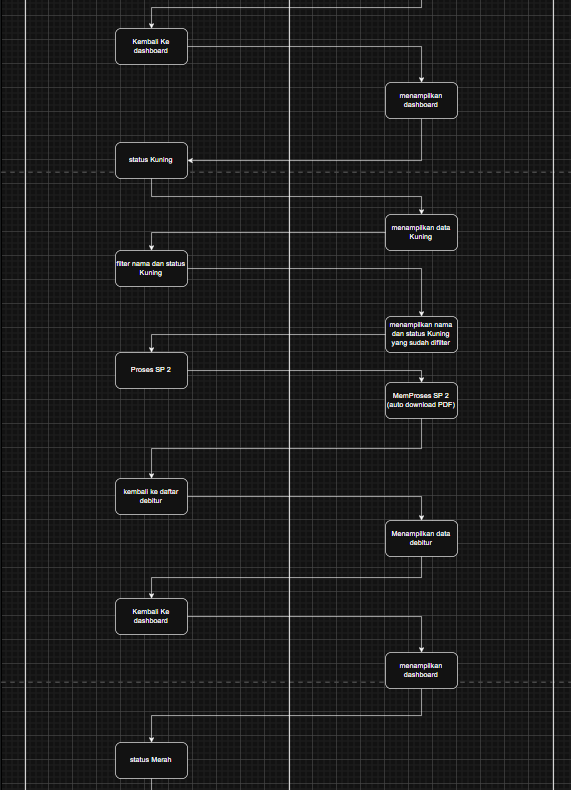
1. Analisis Data:

* **Python**: Python berperan sebagai bahasa pemrograman utama dalam proses pengolahan data. Bahasa ini dipilih karena memiliki sintaks yang ringkas, fleksibel, serta dilengkapi dengan berbagai pustaka yang mendukung kebutuhan data science dan analisis bisnis secara optimal.
* **Pandas**: Pandas adalah pustaka dalam Python yang berfungsi untuk mengelola dan menganalisis data dalam format tabular (DataFrame). Penggunaan Pandas memudahkan berbagai tugas seperti pembersihan data, penghapusan entri duplikat, penanganan nilai kosong, serta transformasi data. NumPy merupakan salah satu pustaka esensial dalam Python yang dirancang untuk mendukung perhitungan numerik. Library ini menyediakan struktur data berupa array multidimensi (ndarray) yang memungkinkan pengolahan data dalam skala besar lebih cepat dibandingkan struktur standar seperti list. Dalam konteks analisis data sparepart, NumPy memfasilitasi proses perhitungan matematika, operasi berbasis vektor, serta berbagai transformasi dan manipulasi angka, termasuk penghitungan total nilai.
* **Numpy:** NumPy merupakan pustaka Python yang dirancang untuk melakukan komputasi numerik dan mengolah data dalam bentuk array. Pustaka ini memiliki peran penting dalam bidang data science, machine learning, dan komputasi ilmiah karena kemampuannya dalam memproses data secara jauh lebih cepat dibandingkan dengan struktur data standar Python seperti list, khususnya saat menangani data berukuran besar.
* **Matplotlib**: Matplotlib dimanfaatkan untuk menyusun berbagai bentuk grafik dan visualisasi data. Hasil visualisasi ini mempermudah manajemen dalam mengidentifikasi pola pengeluaran sparepart, penyebaran kategori harga, serta kecenderungan pemakaian sparepart secara umum.
* **Jupyter Notebook**: Jupyter Notebook digunakan sebagai media interaktif dalam menjalankan skrip Python. Platform ini memungkinkan penulisan kode, tampilan hasil eksekusi, dan penyisipan penjelasan naratif dalam satu dokumen yang terintegrasi dan mudah dipahami.
* **Microsoft Excel**: Microsoft Excel dimanfaatkan pada tahap awal sebagai alat untuk membuka dan meninjau struktur data dari laporan mutasi sparepart. Excel berperan sebagai sarana pendukung awal sebelum data dianalisis lebih lanjut menggunakan Python.

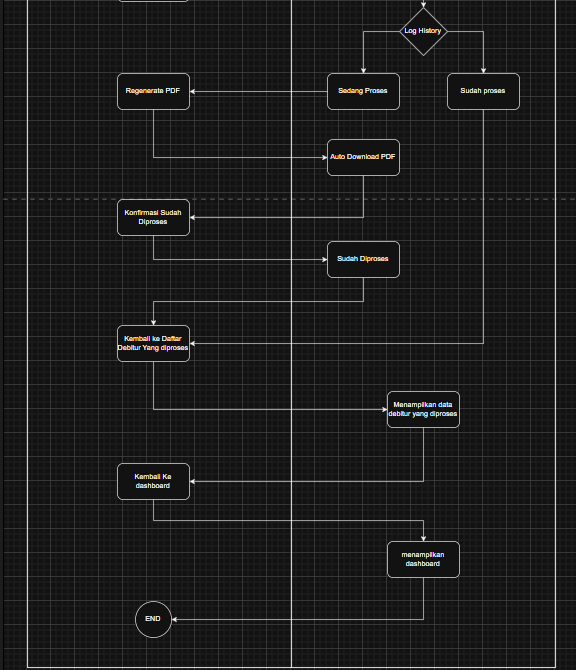
1. Desain Sistem: sistem yang dikembangkan dirancang dengan dua komponen utama yang saling melengkapi, yaitu:

Desain aliran sistem informasi sistem notifikasi gagal bayar









Gambar 3. 13 Desain Aliran Sistem Informasi

**Memulai Aplikasi**

Admin membuka website sistem informasi.Sistem akan merespons dengan menampilkan dashboard utama.

**Melihat Data Debitur**

Admin memilih untuk melihat semua data debitur.Sistem akan menampilkan seluruh data debitur.Setelah itu, admin bisa kembali ke dashboard.

**Memproses Debitur Berstatus Hijau**

Admin memilih kategori status hijau.Sistem akan menampilkan daftar debitur dengan status hijau.Admin melakukan filter berdasarkan nama dan status hijau.Sistem menampilkan nama-nama debitur dengan status hijau yang sudah difilter.Admin kemudian memberikan tindakan, misalnya monitoring ringan.Sistem akan melakukan konfirmasi terhadap tindakan tersebut.Setelah selesai, admin kembali ke daftar debitur, lalu ke dashboard.

**Memproses Debitur Berstatus Kuning**

Dari dashboard, admin memilih status kuning.Sistem menampilkan data debitur berstatus kuning.Admin memfilter berdasarkan nama dan status.Sistem menampilkan hasil filter.Admin memberikan tindakan, misalnya peringatan ataufollow up yang lebih ketat.Sistem mengonfirmasi tindakan tersebut.Admin kembali ke daftar debitur, lalu ke dashboard.

**Memproses Debitur Berstatus Merah**

Admin memilih status merah dari dashboard.Sistem menampilkan data debitur dengan status merah.Admin memfilter berdasarkan nama dan status.Sistem menampilkan hasil filter.Admin memberikan tindakan lanjutan, misalnyapenagihan langsung atau pelimpahan ke kolektor.Sistem memberikan konfirmasi atas tindakan tersebut.Admin kembali ke daftar debitur, lalu ke dashboard.

**Melihat Riwayat dan Status Proses**

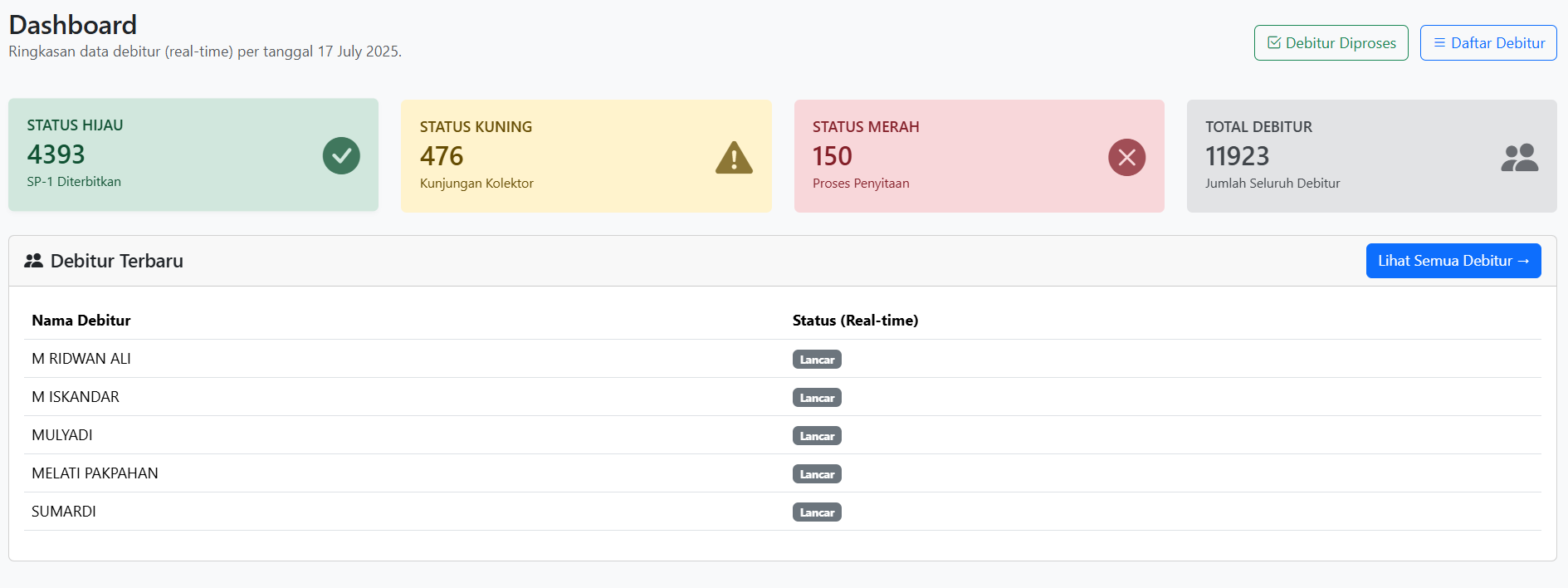
Untuk mengelola dan memantau tindak lanjut yang telah dilakukan, admin dapat mengakses menu "Daftar Debitur Yang Diproses", di mana sistem akan menampilkan daftar seluruh debitur yang sedang atau sudah berada dalam suatu proses. Guna melihat detail lebih lanjut, admin memilih opsi "Log History" pada debitur yang diinginkan. Sistem kemudian akan menampilkan status riwayatnya; jika statusnya adalah "Sedang Proses", admin diberikan kewenangan untuk membuat ulang dokumen dengan memilih "Regenerate PDF" yang akan memicu unduhan PDF secara otomatis. Setelah PDF berhasil diunduh, atau jika statusnya sudah "Sudah Proses", langkah selanjutnya bagi admin adalah melakukan "Konfirmasi Sudah Diproses" untuk menyelesaikan tahap tersebut. Sesudah menyelesaikan verifikasi ini, admin memiliki fleksibilitas untuk kembali ke daftar debitur yang diproses ataupun langsung kembali ke dashboard utama.

**Selesai**

Proses kembali ke dashboard dan sistem siap untuk digunakan kembali.Proses berakhir.

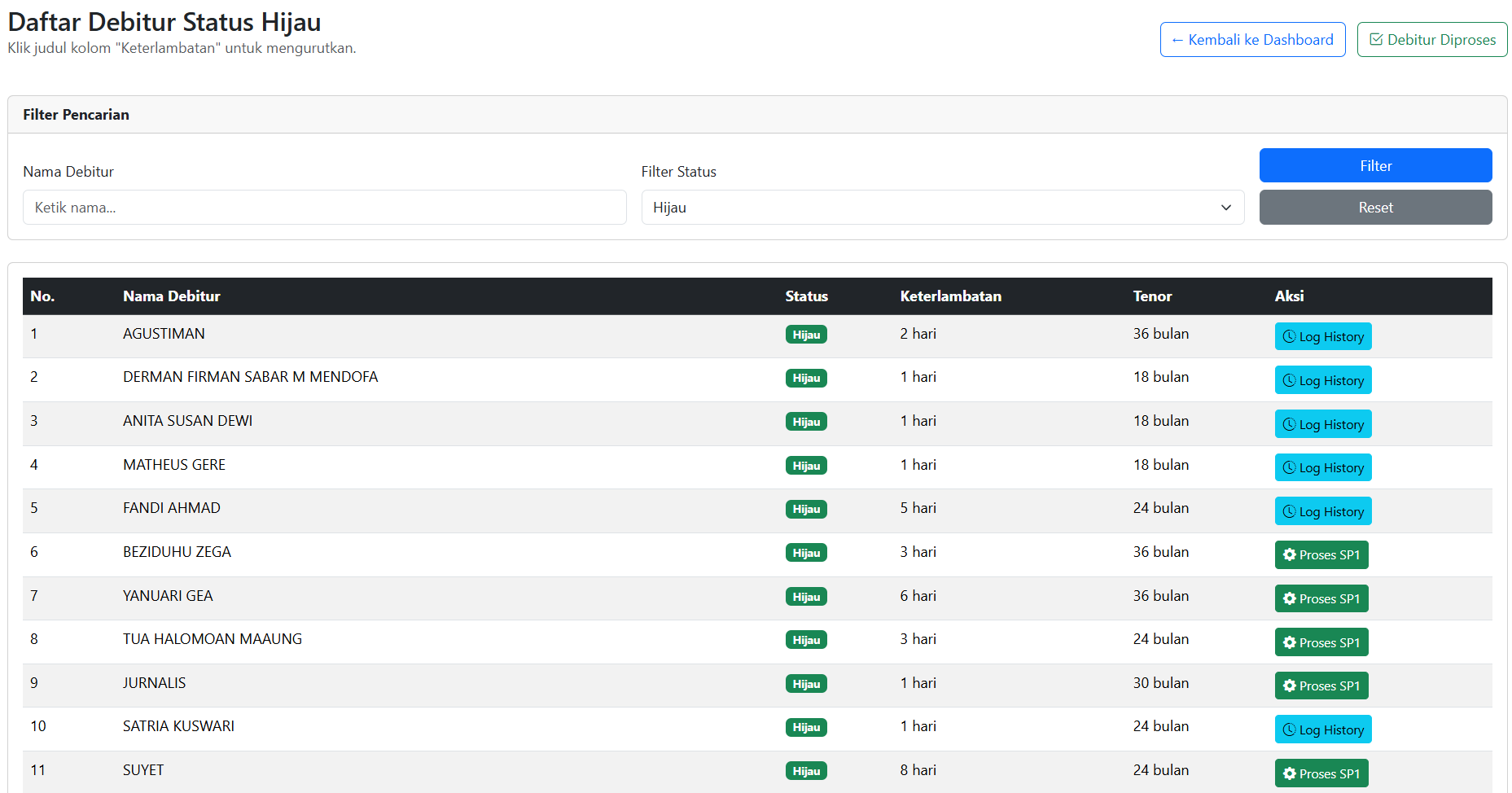
Desain User Interface

1. Dashboard



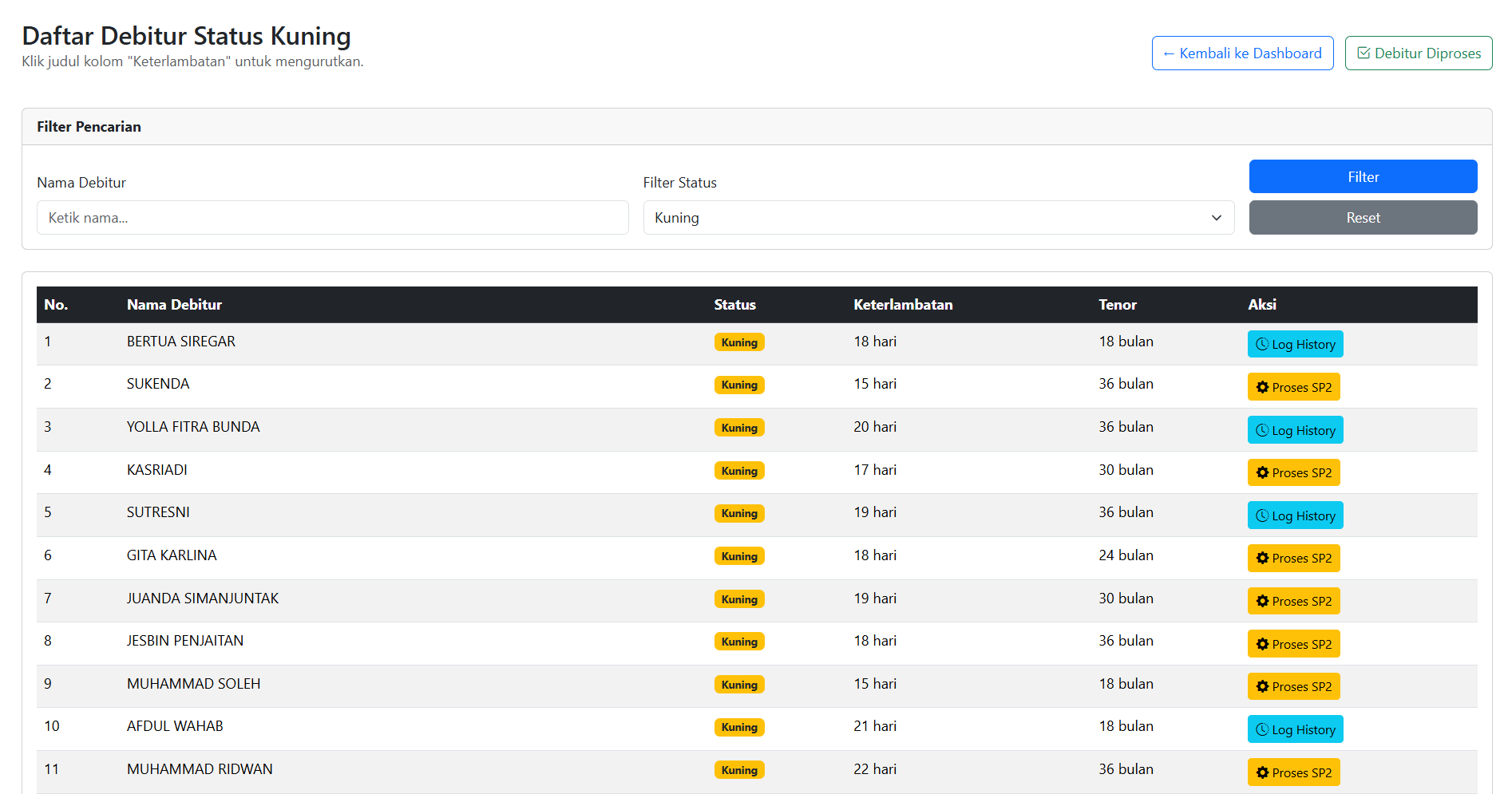
Gambar 3. 14 Dashboard

1. Daftar Debitur Status Hijau



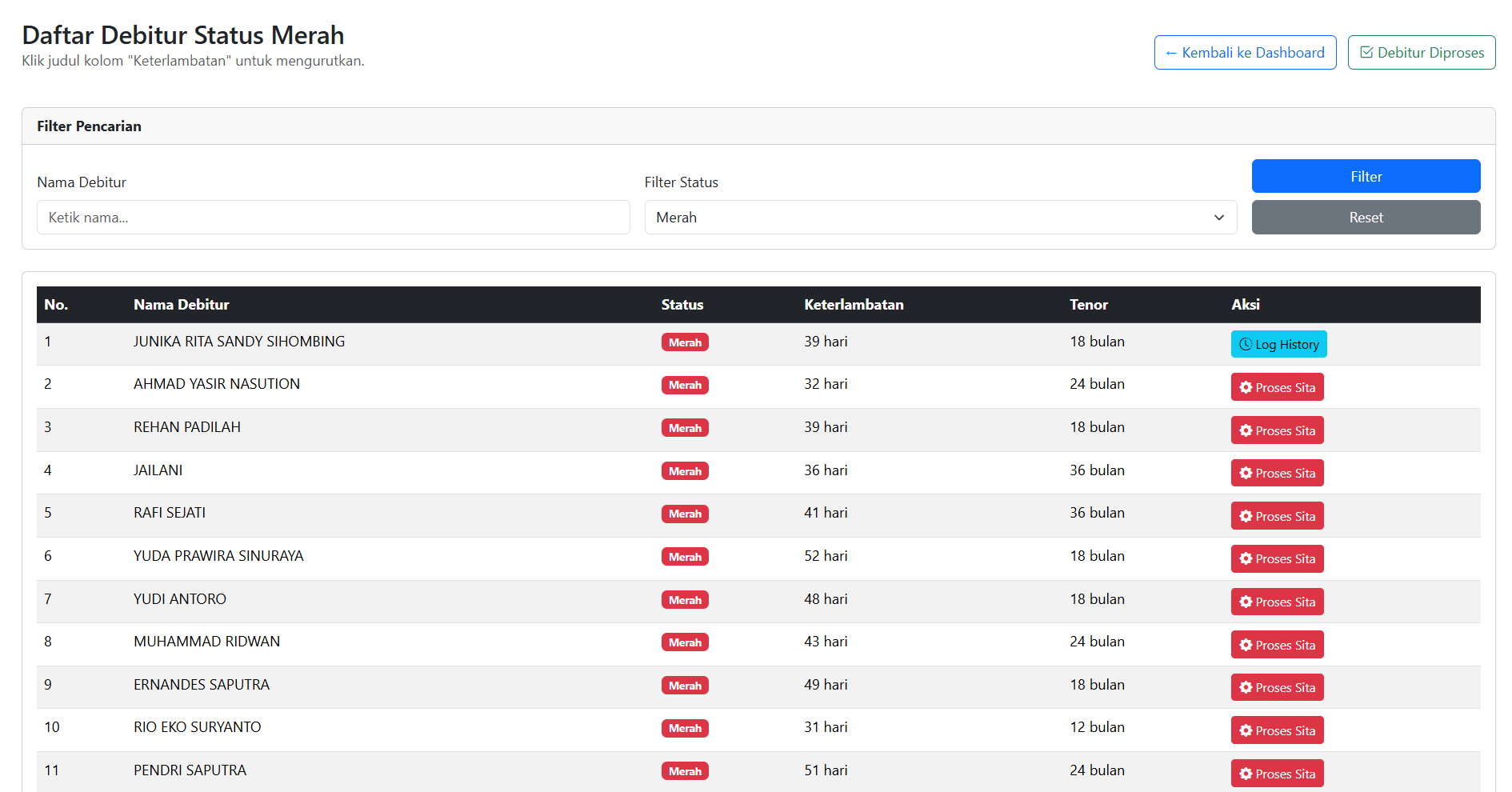
Gambar 3. 15 Daftar Debitur Status Hijau

1. Daftar Debitur Status Kuning



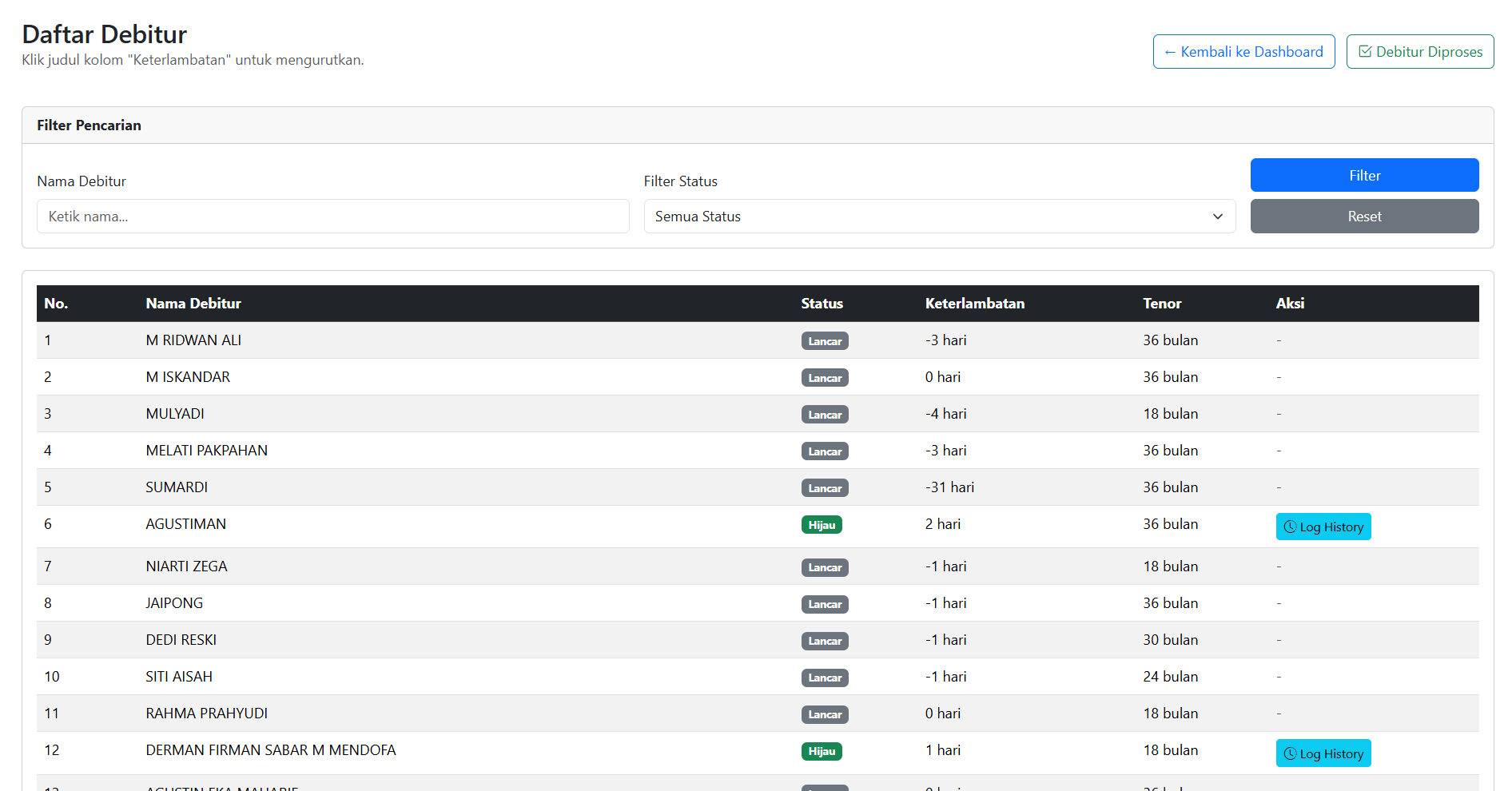
Gambar 3. 16 Daftar Debitur Status Kuning

1. Daftar Debitur Status Merah



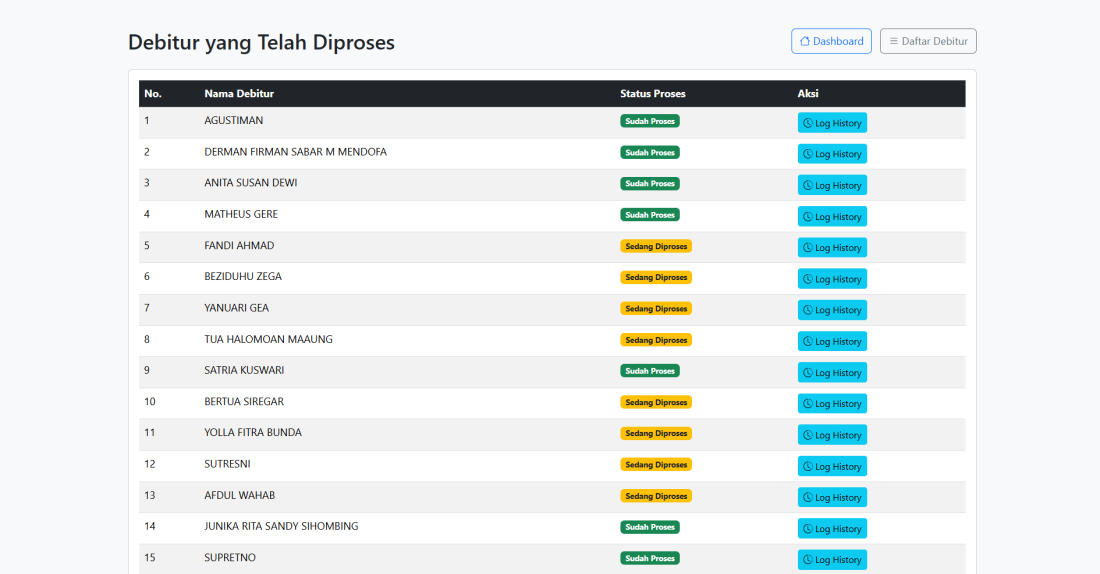
Gambar 3. 17 Daftar Debitur Status Merah

1. Daftar Semua Debitur



Gambar 3. 18 Daftar Semua Debitur

1. Daftar Debitur Diproses



Gambar 3. 19 Daftar Debitur Diproses

Pada awal membuka aplikasi, manajemen akan langsung diarahkan ke halaman dashboard yang menampilkan ringkasan data seperti jumlah debitur berstatus hijau, kuning, dan merah, total keseluruhan debitur, serta daftar debitur terbaru. Dari sini, manajemen dapat mengklik salah satu kategori status untuk melihat daftar debitur yang lebih rinci sesuai dengan warnanya. Pada halaman daftar debitur ini, disajikan informasi seperti nama debitur, status, lama keterlambatan, tenor, dan sebuah kolom "Aksi" yang bersifat dinamis. Untuk debitur yang belum ditindaklanjuti, kolom tersebut akan menampilkan tombol tindakan seperti "Proses SP1", "Proses SP2", atau "Proses Sita", yang jika diklik akan membawa manajemen ke halaman Konfirmasi Tindakan untuk melihat detail aksi sebelum memprosesnya. Namun, untuk debitur yang tindakannya sudah pernah diproses, tombol "Aksi" akan berubah menjadi "Log History". Saat tombol ini diklik, sistem akan menampilkan halaman Log History Debitur, yang berisi catatan riwayat tindakan seperti "PDF surat sudah pernah digenerate untuk debitur ini" dan memberikan status konfirmasi "Sudah Proses". Setelah melakukan tindakan atau melihat riwayat, manajemen dapat kembali ke halaman daftar debitur, di mana mereka juga bisa memanfaatkan fitur Filter Pencarian untuk mencari debitur berdasarkan nama dan memfilter berdasarkan status yang diinginkan.

1. Pengembangan Sistem: Pengembangan sistem dalam penelitian ini dilakukan untuk membangun sebuah aplikasi Business Intelligence berbasis web yang dapat memberikan rekomendasi tindakan terhadap debitur berdasarkan tingkat keterlambatan dan profil risiko. Sistem dirancang untuk mempermudah manajemen dalam memantau kondisi pembayaran debitur, serta memberikan peringatan dan tindakan secara otomatis dan real-time.

Sistem ini dikembangkan dengan pendekatan Research and Development (R&D), yang dimulai dari pengumpulan dan pembersihan data, dilanjutkan dengan transformasi data, klasifikasi risiko, hingga pengujian fungsionalitas sistem melalui dashboard interaktif.

Beberapa tahapan penting dalam proses pengembangan sistem meliputi:

* Integrasi Data

Data yang telah melalui tahap cleansing dan transformasi diimpor ke dalam basis data SQL. Data ini berisi informasi penting seperti nama debitur, ID pelanggan, nilai kontrak, jumlah angsuran, tanggal pembayaran terakhir, DP, serta hasil perhitungan hari keterlambatan.

* Penentuan Kategori Risiko dan Status Warna

Sistem melakukan klasifikasi otomatis terhadap status pembayaran debitur berdasarkan durasi keterlambatan dan kriteria risiko. Klasifikasi ini dibagi ke dalam tiga warna utama:

Hijau: Keterlambatan 1–14 hari

Kuning: Keterlambatan 15–30 hari

Merah: Keterlambatan lebih dari 30 hari

Setiap warna kemudian dikaitkan dengan tingkat risiko (rendah, menengah, tinggi) untuk menentukan jenis tindakan yang disarankan.

* Rekomendasi Tindakan

Berdasarkan status warna dan tingkat risiko, sistem memberikan rekomendasi tindakan yang berbeda. Misalnya, SP1 untuk keterlambatan ringan, SP2 dengan inspeksi untuk keterlambatan menengah, hingga rekomendasi penyitaan kendaraan untuk keterlambatan berat dan risiko tinggi.

* Perancangan Antarmuka (Front-End)

Antarmuka sistem dikembangkan menggunakan kombinasi teknologi HTML, CSS, JavaScript, dan Bootstrap. Tampilan dirancang agar user-friendly dan memungkinkan pengguna (admin) untuk melakukan filter data, melihat daftar debitur berdasarkan status, serta memberikan dan mengonfirmasi tindakan secara langsung.

* Pemrograman Backend dan Visualisasi

Backend sistem dibangun menggunakan PHP, yang mengatur koneksi ke database SQL serta mengeksekusi logika sistem seperti klasifikasi status, pemberian warna, dan notifikasi. Proses transformasi data dan penghitungan statistik dilakukan dengan Python, dan visualisasi grafik dibangun menggunakan chart dan tabel interaktif.

* Dashboard Real-Time

Sistem menyediakan dashboard utama yang menampilkan ringkasan status seluruh debitur, grafik keterlambatan, dan total rekomendasi tindakan yang sudah atau belum diambil. Admin dapat menavigasi ke daftar debitur berdasarkan status (Hijau, Kuning, Merah) untuk mengelola kasus masing-masing secara langsung.

## Sumber Data

Data yang digunakan dalam pengembangan sistem rekomendasi tindakan kepada debitur ini merupakan data sekunder yang diperoleh langsung dari PT Capella Multidana. Data disediakan oleh pihak internal perusahaan dalam bentuk file digital berformat Excel dan berisi informasi historis mengenai riwayat pembiayaan dan keterlambatan pembayaran debitur selama periode tahun 2000 hingga 2025.

Sumber data ini terdiri atas lebih dari 22.000 baris entri yang mencakup berbagai atribut penting seperti nama debitur, ID pelanggan, tanggal awal dan akhir kontrak kredit, jenis tenor, jumlah angsuran, tanggal dan nilai pembayaran terakhir, status kontrak, kronologi kasus, serta informasi kendaraan seperti nomor plat, nomor rangka, nomor mesin, harga motor baru, dan uang muka. Data juga mencakup kategori pembiayaan dan deskripsi kondisi pembayaran yang relevan untuk pemetaan risiko keterlambatan.

Seluruh informasi tersebut telah direkap oleh pihak administrasi perusahaan sebagai bagian dari sistem pencatatan transaksi dan pelaporan internal. Oleh karena itu, data ini dianggap representatif dan valid untuk digunakan dalam proses perancangan dan pengembangan sistem Business Intelligence.

Sumber data ini menjadi fondasi utama dalam membangun sistem klasifikasi status debitur (Hijau, Kuning, Merah) berdasarkan durasi keterlambatan dan tingkat risiko (rendah, menengah, tinggi). Dari data inilah, proses lanjutan seperti pembersihan, transformasi, visualisasi, hingga perancangan sistem notifikasi dan rekomendasi otomatis dilakukan untuk mendukung pengambilan keputusan yang lebih responsif oleh manajemen perusahaan.

### **Data Sekunder**

Data sekunder yang digunakan dalam penelitian ini diperoleh melalui kerja sama langsung dengan pihak internal PT Capella Multidana. Dokumen yang disediakan berbentuk file Excel, berisi data historis yang telah dikumpulkan dan dikelola oleh perusahaan sebagai bagian dari proses operasional mereka. Data ini mencakup catatan debitur bermasalah dari tahun 2000 hingga 2025, dengan total lebih dari 22.000 entri.

Informasi dalam dataset tersebut mencakup berbagai variabel penting, antara lain nama debitur, **(**ID pelanggan, tenor kredit, tanggal awal dan akhir kontrak, nominal angsuran, jumlah pembayaran terakhir, tanggal pembayaran terakhir, status kontrak, serta catatan kronologi keterlambatan. Selain itu, data juga memuat detail kendaraan yang dibiayai, seperti nomor plat, nomor rangka, nomor mesin, harga motor baru, serta jumlah uang muka**)**.

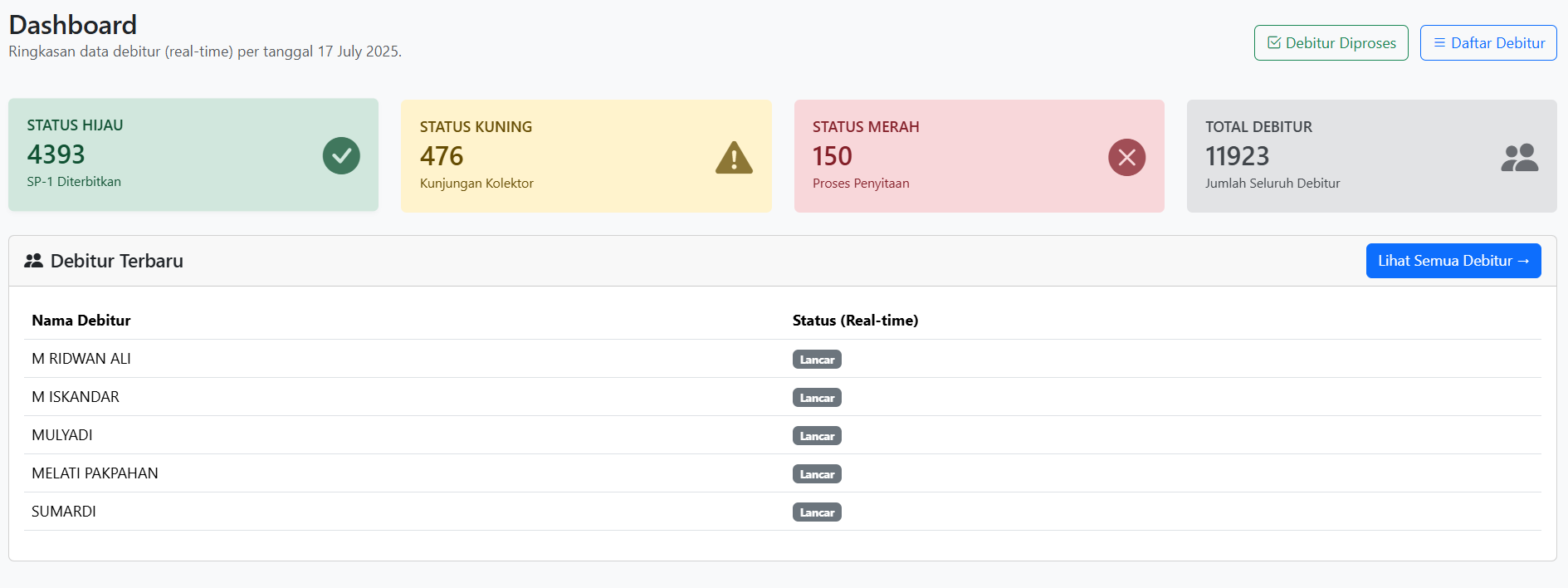
Data sekunder ini berfungsi sebagai bahan utama dalam proses pengolahan, pembersihan, transformasi, serta visualisasi yang menjadi dasar dalam merancang sistem Business Intelligence. Dengan data ini, sistem yang dikembangkan dapat membagi status debitur berdasarkan kategori warna (Hijau, Kuning, Merah) serta memberikan rekomendasi tindakan yang disesuaikan dengan tingkat risiko keterlambatan (rendah, menengah, tinggi).

Penggunaan data sekunder ini dinilai tepat karena bersumber langsung dari catatan aktual perusahaan yang telah terdokumentasi dan tervalidasi, sehingga dapat memberikan gambaran yang akurat mengenai pola keterlambatan pembayaran debitur di lapangan.

# BAB IV

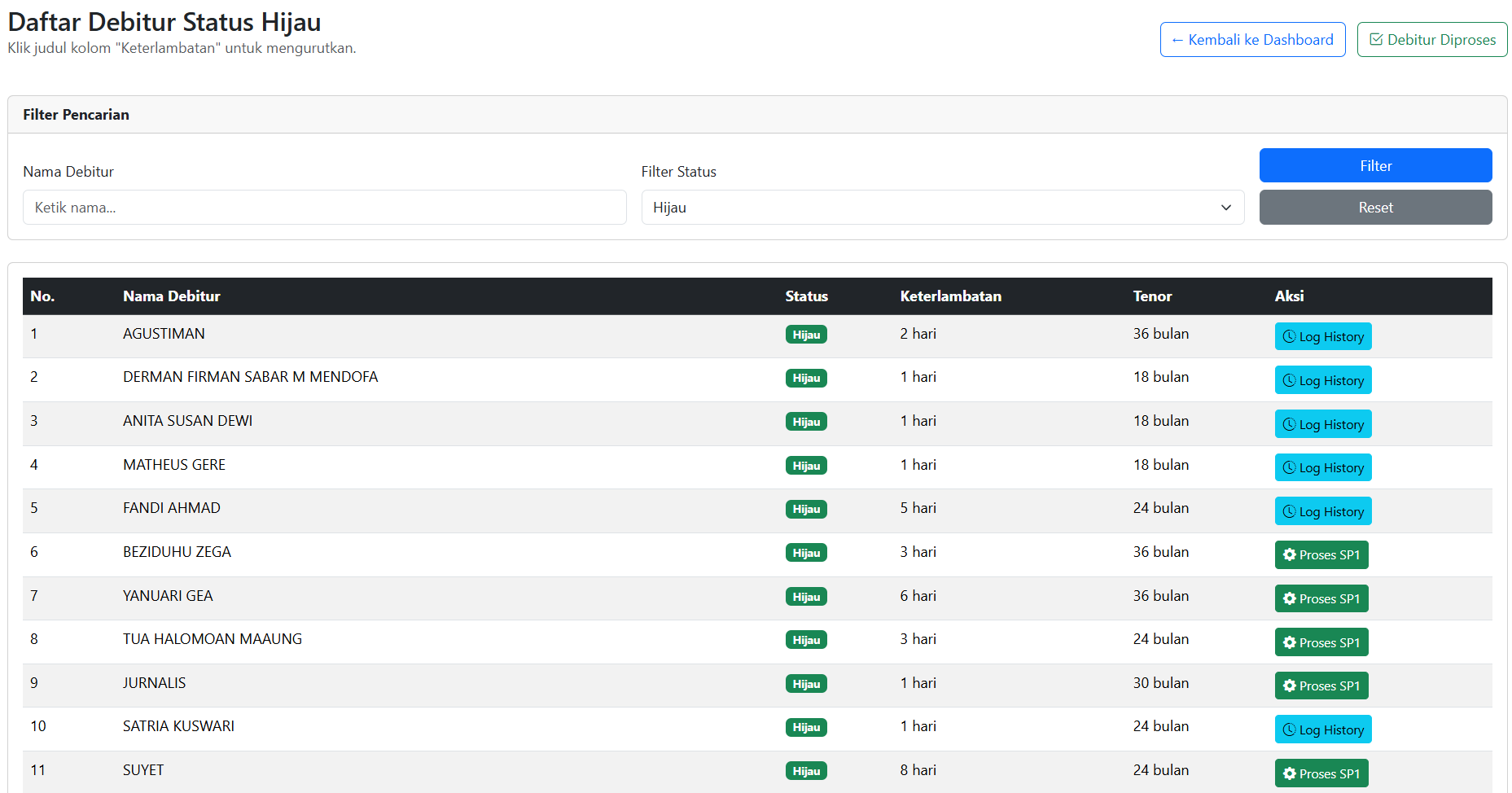
# HASIL DAN PEMBAHASAN

## Hasil Aplikasi Web

1. Halaman Dashboard

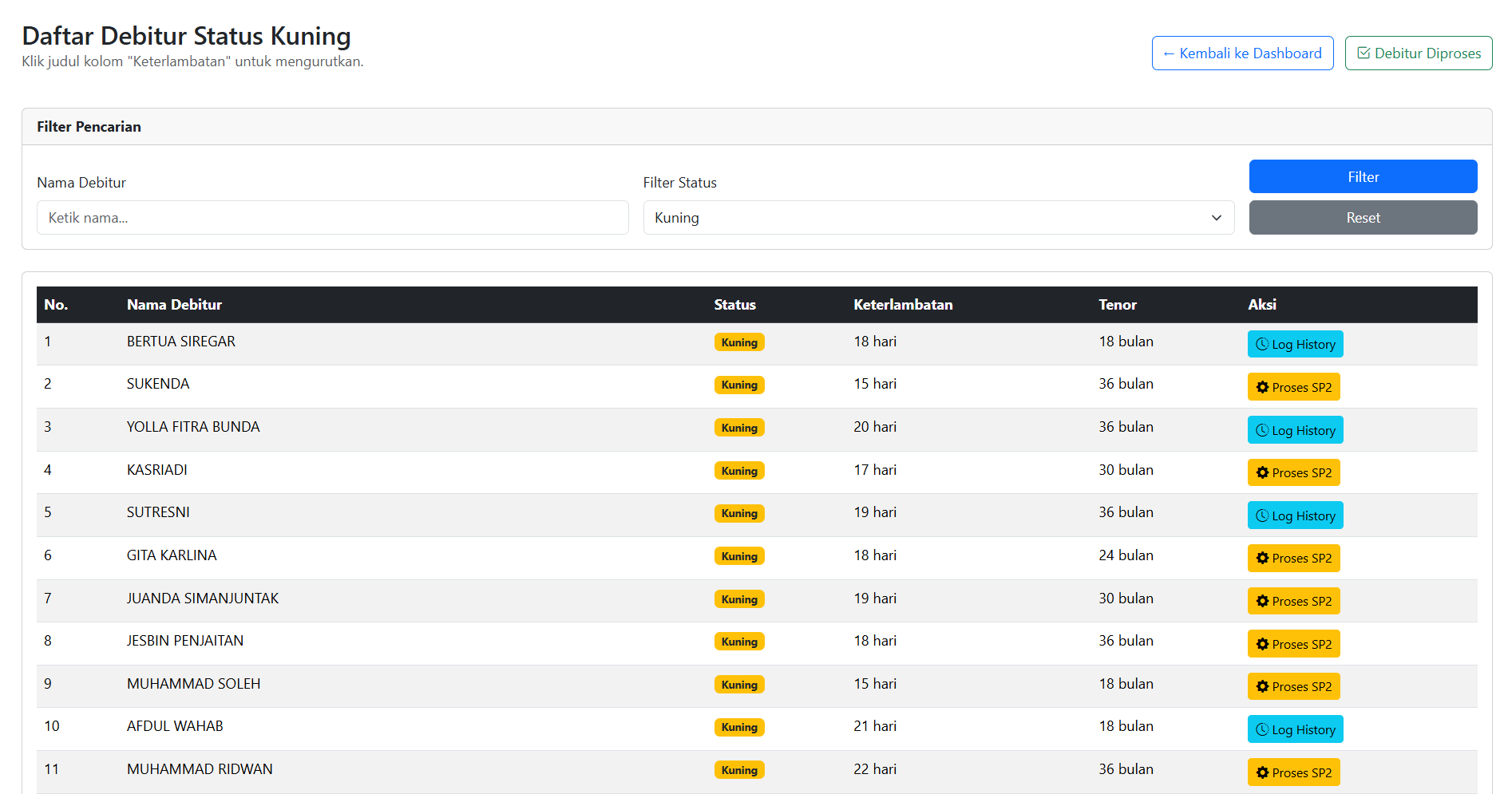
Gambar 4. 1 Halaman Dashboard

1. Halaman Daftar Debitur Status Hijau

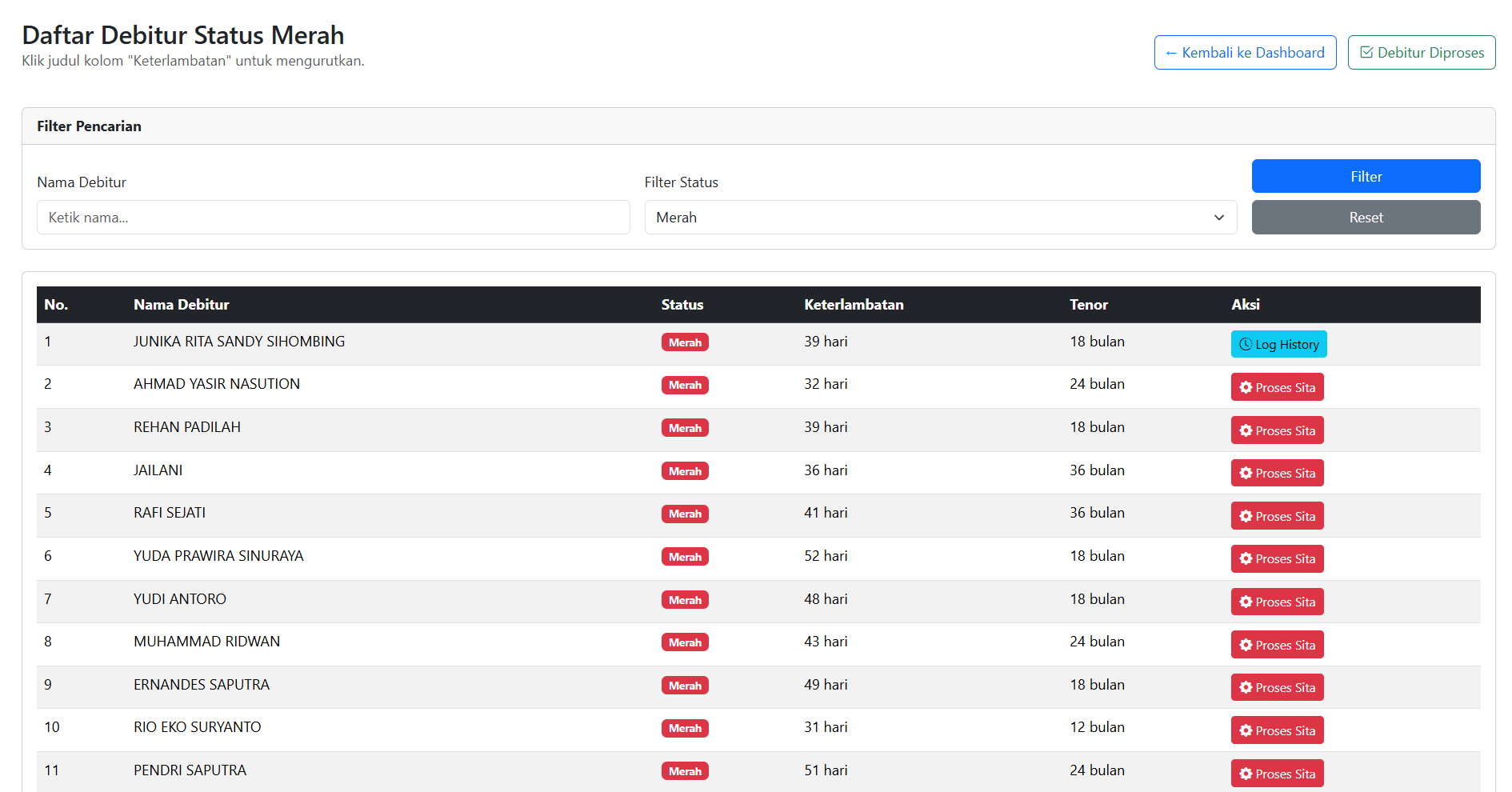


Gambar 4. 2 Status Hijau

1. Halaman Daftar Debitur Status Kuning

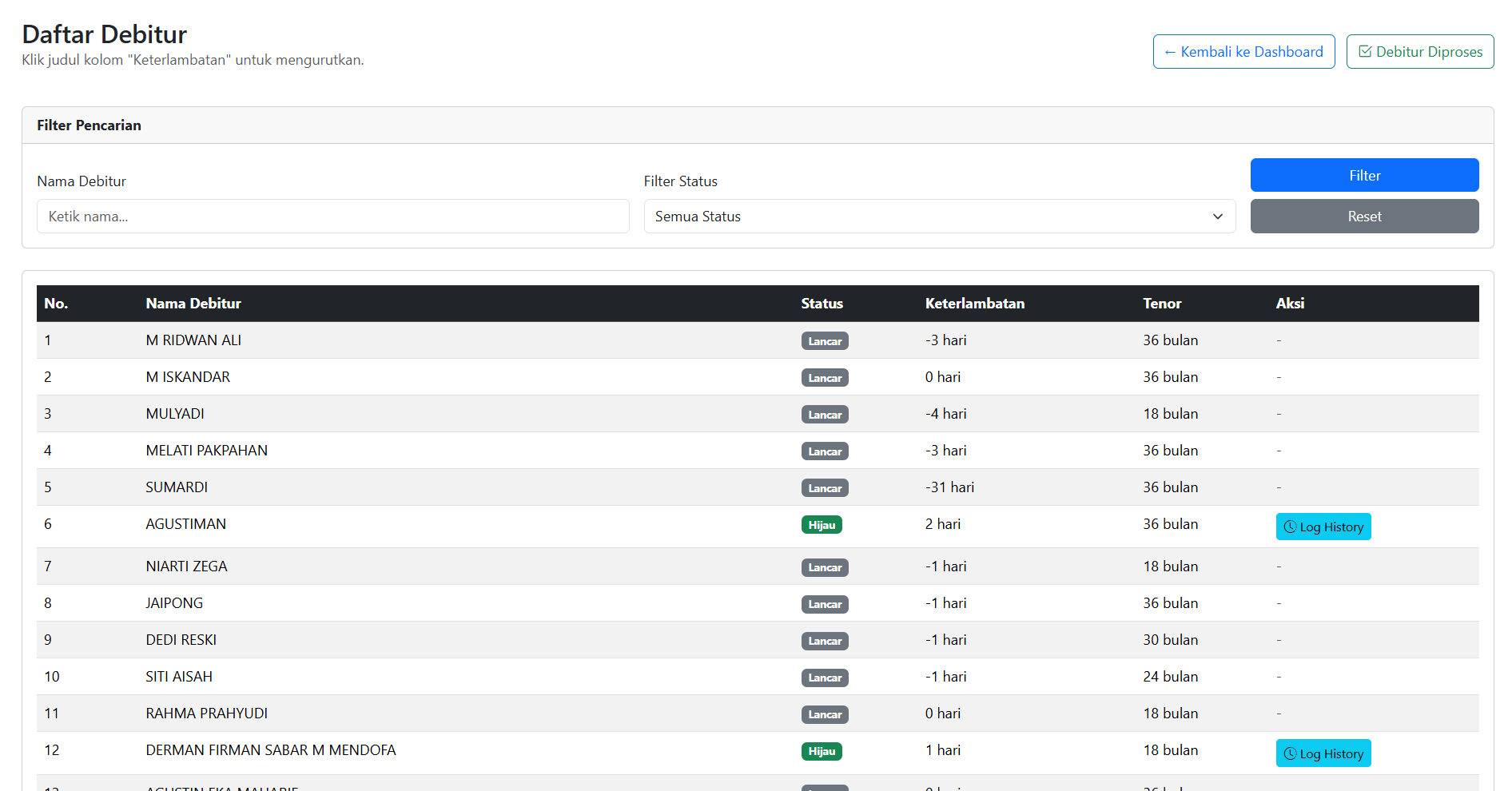


Gambar 4. 3 Status Kuning

1. Halaman Daftar Debitur Status Merah

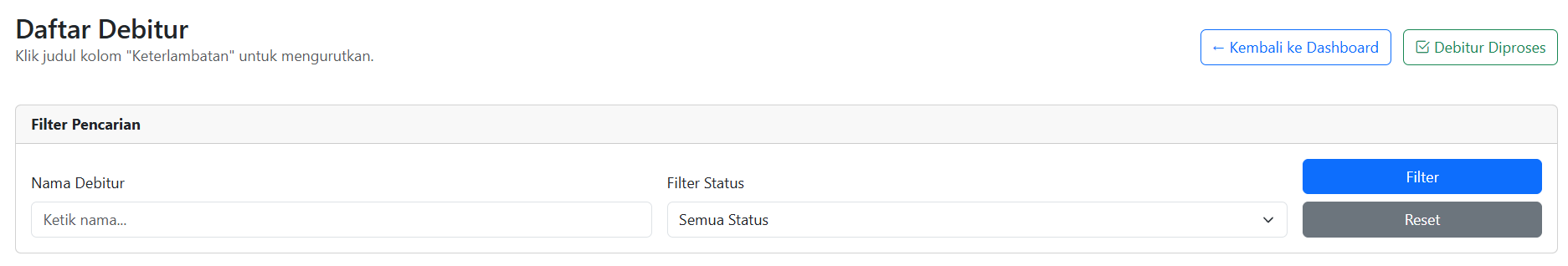
Gambar 4. 4 Status Merah

1. Halaman Daftar Debitur



Gambar 4. 5 Halaman Daftar Debitur

1. Nav Bar



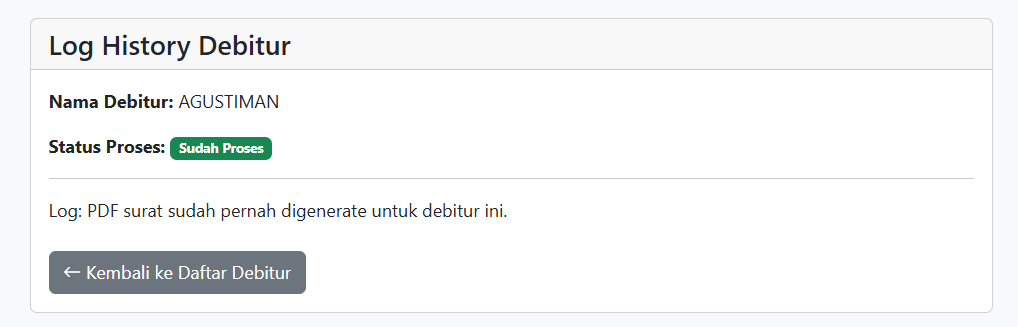
Gambar 3. 25 Nav Bar

1. Filter Status



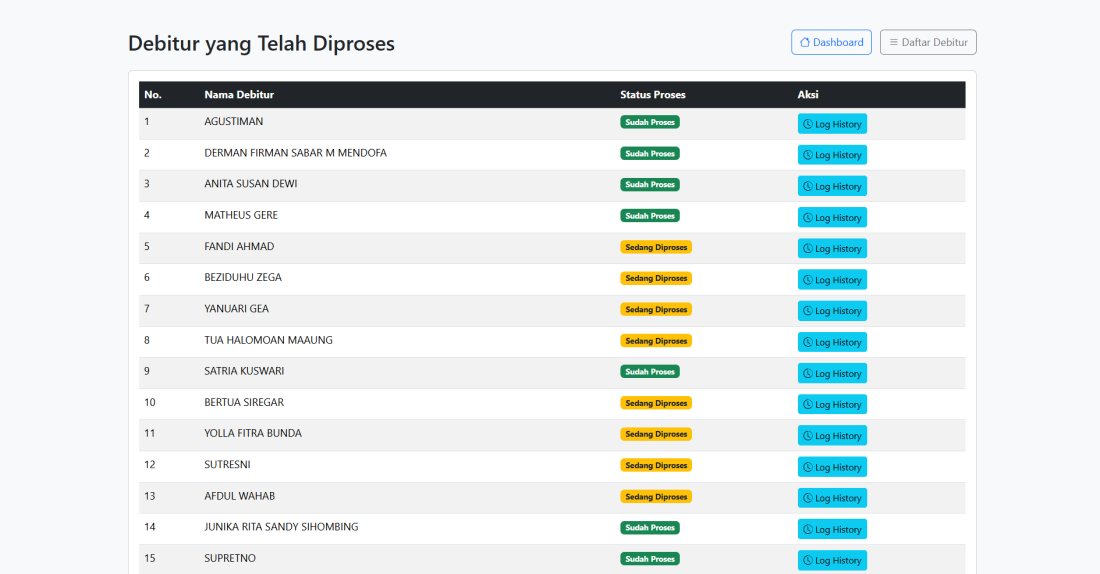
Gambar 4. 6 Filter Status

1. Log History Debitur



Gambar 4. 7 Log History Debitur

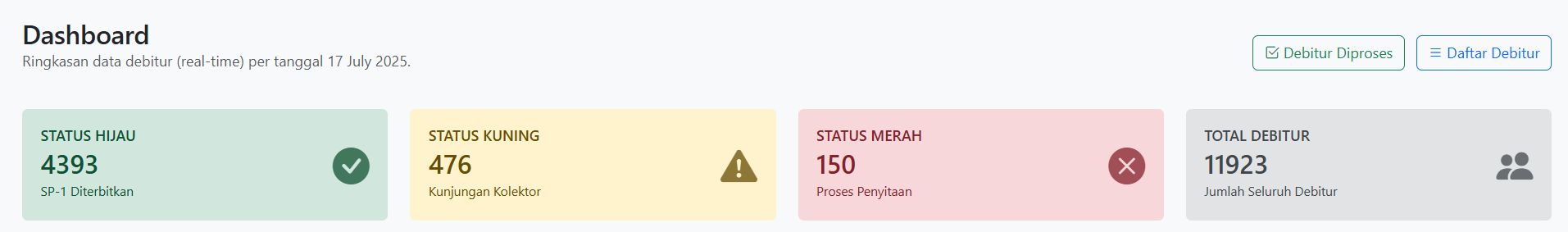
1. Daftar Debitur Diproses



Gambar 4. 8 Daftar Debitur Diproses

## Pembahasan

1. Tampilan pada bagian dashboard

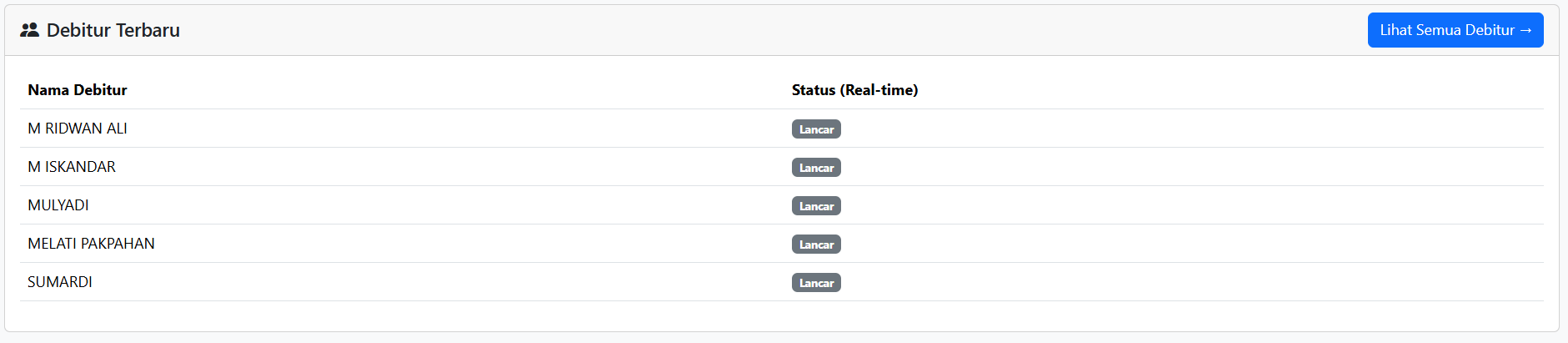
****

Gambar 4. 9 Tampilan Dashboard

|  |  |
| --- | --- |
| Properti | Keterangan |
|  | Judul halaman yang menunjukkan ini adalah tampilan utama (dashboard) untuk melihat **ringkasan data debitur** secara **real-time**. Juga ditampilkan tanggal update data terakhir (15 Juli 2025). |
|  | Jumlah debitur dengan status "Hijau". Ini berarti ada 4393 debitur yang SP-1 (Surat Peringatan 1) telah diterbitkan. Status hijau umumnya mengindikasikan tahapan awal proses penagihan atau debitur yang baru saja diberikan peringatan. |
|  | Jumlah debitur dengan status "Kuning". Ada 476 debitur yang sedang dalam tahap kunjungan kolektor, yang berarti kolektor sedang atau akan melakukan kunjungan ke debitur tersebut untuk penagihan atau konfirmasi. Status ini bisa mengindikasikan bahwa SP-1 tidak diindahkan atau perlu tindak lanjut lebih lanjut. |
|  | Jumlah debitur dengan status "Merah". Ada 150 debitur yang sedang dalam proses penyitaan adalah tahapan paling serius, di mana aset debitur mungkin akan disita untuk melunasi utang. Status merah menunjukkan bahwa upaya penagihan sebelumnya tidak berhasil. |
|  | Kotak informasi berwarna abu-abu dengan jumlah **11923**, menunjukkan total keseluruhan debitur yang terdata dalam sistem baik yang statusnya hijau, kuning maupun merah. Ikon orang menandakan entitas debitur. |
|  | Tombol atau indikator yang kemungkinan besar mengarahkan pengguna untuk melihat daftar debitur yang sedang dalam proses penagihan atau penanganan (mungkin termasuk semua status: hijau, kuning, merah, atau status lain yang sedang berjalan). |
|  | Tombol atau indikator yang kemungkinan besar mengarahkan pengguna untuk melihat seluruh daftar debitur yang ada dalam sistem, tanpa filter berdasarkan status. |

Tabel 4. 1

1. Tampilan Daftar Debitur Terbaru

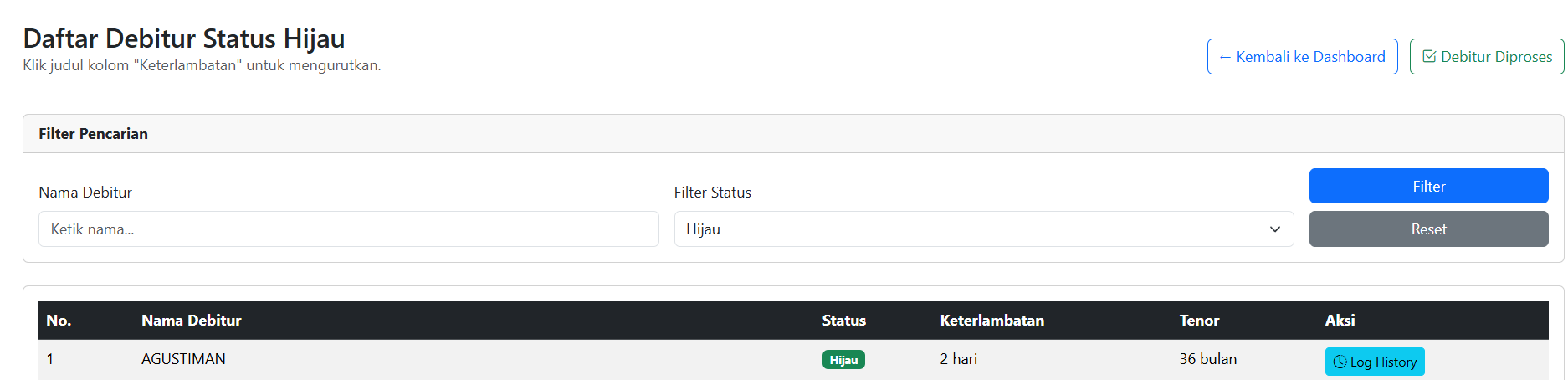
****

Gambar 4. 10 Tampilan Daftar Debitur Terbaru

|  |  |
| --- | --- |
| Properti | Keterangan |
|  | Menampilkan nama-nama debitur beserta status real-time mereka seperti “Lancar”. |
|  | Ini adalah **judul kolom status** yang menunjukkan status kredit (misalnya “Lancar”). Data diperbarui secara otomatis atau saat di-refresh. |
|  | indikator status yang bersifat informatif. Umumnya akan berubah jika status debitur berubah (misalnya menjadi “Merah” untuk macet). |
|  | Berguna jika pengguna ingin melihat data lebih luas dari daftar yang ditampilkan secara ringkas di halaman utama. |
|  | Daftar ini merupakan bagian dari **preview ringkas**, biasanya hanya menampilkan 5 data paling baru. Lengkapnya bisa dilihat lewat tombol “Lihat Semua Debitur”. |

Tabel 4. 2

1. Tampilan Daftar Debitur Status Hijau

****

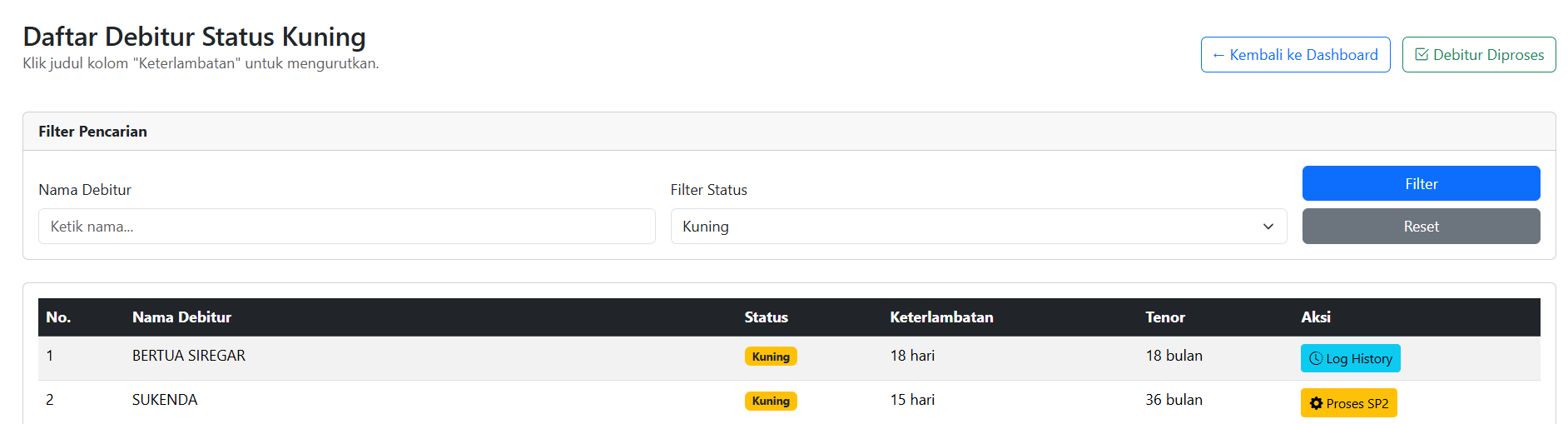
****

Gambar 4. 11 Tampilan Daftar Debitur Status Hijau

|  |  |
| --- | --- |
| Properti | Keterangan |
|  | Halaman ini menampilkan daftar debitur dengan status hijau (status yang paling baik dalam sistem penilaian). Ini digunakan untuk memantau debitur yang masih dalam kondisi pembayaran yang baik. |
|  | Menandakan bagian pencarian/filter. Pengguna dapat memfilter daftar berdasarkan nama debitur atau status keterlambatan. |
|  | Tombol ini digunakan untuk menandai bahwa debitur sedang dalam proses penanganan atau evaluasi lebih lanjut. Umumnya berfungsi sebagai penanda status internal. |
|  | Digunakan untuk kembali ke halaman utama yang menampilkan seluruh daftar debitur tanpa filter tertentu. |
|  | Tombol ini mengarahkan pengguna kembali ke dashboard utama aplikasi. |
|  | Kolom input teks yang memungkinkan pengguna mencari debitur berdasarkan nama. Fungsionalitas ini memudahkan pencarian secara spesifik. |
|  | Menu dropdown yang berisi beberapa pilihan status debitur:   1. **Semua Status**: Menampilkan semua debitur. 2. **Hijau**: Debitur dengan keterlambatan sangat rendah atau tidak ada keterlambatan. 3. **Kuning**: Debitur mulai menunggak dalam jangka pendek. 4. **Merah**: Debitur memiliki keterlambatan serius. 5. **Lancar**: Debitur dalam kondisi pembayaran lancar. |
|  | Mengaktifkan filter berdasarkan input nama dan status yang dipilih. Menyaring daftar debitur sesuai kriteria. |
|  | Menghapus filter yang telah diatur dan menampilkan kembali semua data tanpa penyaringan. |
|  | Menampilkan data debitur dengan kolom berikut:   1. **No.**: Nomor urut. 2. **Nama Debitur**: Nama pelanggan atau debitur. 3. **Status**: Status keterlambatan (Hijau, Kuning, Merah). 4. **Keterlambatan**: Lama keterlambatan dalam hari. 5. **Tenor**: Lama periode pinjaman dalam bulan. 6. **Aksi**: Tombol tindakan lanjutan. |
|  | Contoh data debitur yang ditampilkan berdasarkan filter:   1. Nama: AGUSTIMAN 2. Status: Hijau 3. Keterlambatan: 2 hari 4. Tenor: 36 bulan |
|  | Tombol untuk memulai proses SP1 (Surat Peringatan 1), biasanya diberikan untuk debitur yang mulai menunggak. |
|  | Menampilkan riwayat aktivitas atau perubahan status debitur yang bersangkutan. |

Tabel 4. 3

1. Tampilan Daftar Debitur Status Kuning

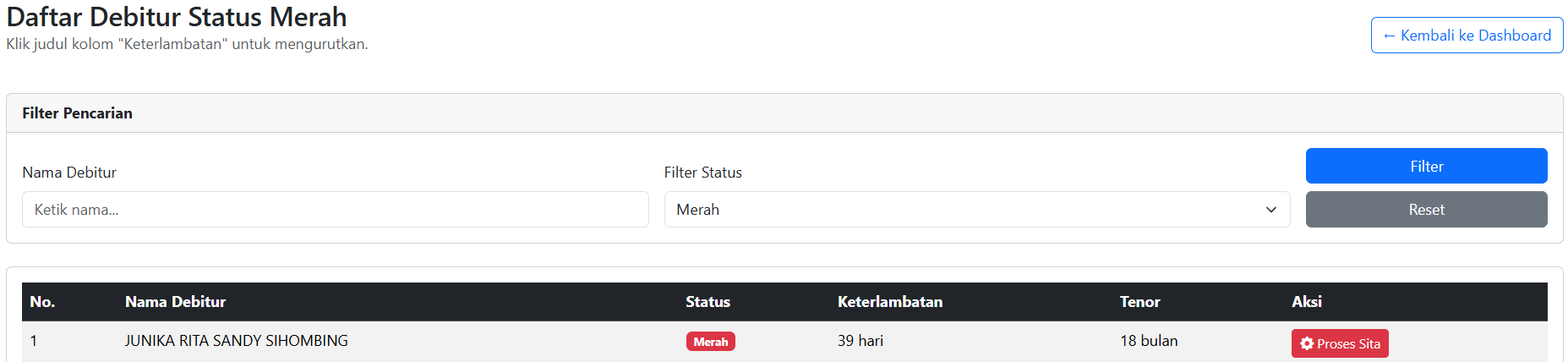


Gambar 4. 12 Tampilan Daftar Debitur Status Kuning

|  |  |
| --- | --- |
| Properti | Keterangan |
|  | Daftar debitur yang memiliki **status keterlambatan Kuning**. Status ini mengindikasikan bahwa debitur telah menunggak pembayaran dalam jumlah **menengah**, misalnya antara 10 hingga 30 hari. |
|  | Bagian filter yang memungkinkan pengguna memfilter daftar debitur berdasarkan nama atau status untuk kemudahan pencarian data. |
|  | Salah satu atau beberapa debitur telah/akan diproses dalam tahap penanganan lebih lanjut. Ini bersifat sebagai indikator administratif internal. |
|  | Navigasi untuk kembali ke halaman utama daftar seluruh debitur, tanpa terbatas pada status tertentu. |
|  | Navigasi untuk kembali ke dashboard utama aplikasi, tempat berbagai menu dan fitur utama sistem tersedia. |
|  | Kolom input teks tempat pengguna dapat mengetik nama debitur untuk melakukan pencarian secara langsung dan cepat dalam daftar. |
|  | Dropdown untuk memilih status keterlambatan debitur yang ingin ditampilkan, dengan pilihan:   1. **Semua Status**: Menampilkan semua status. 2. **Hijau**: Pembayaran lancar (tidak menunggak). 3. **Kuning**: Menunggak sedang (10–30 hari). 4. **Merah**: Menunggak berat (lebih dari 30 hari). 5. **Lancar**: Pembayaran sesuai jadwal, tidak ada keterlambatan. |
|  | Memproses pencarian/filter berdasarkan input nama dan status keterlambatan yang dipilih. Sistem akan menampilkan hasil yang sesuai. |
|  | Menghapus semua filter yang sudah diterapkan dan menampilkan kembali seluruh daftar tanpa penyaringan. |
|  | Menampilkan daftar debitur dalam format tabel dengan kolom:   1. **No.**: Nomor urut data. 2. **Nama Debitur**: Nama pelanggan/debitur. 3. **Status**: Status keterlambatan (dalam hal ini “Kuning”). 4. **Keterlambatan**: Jumlah hari keterlambatan. 5. **Tenor**: Jangka waktu pinjaman. 6. **Aksi**: Tombol-tombol tindakan lebih lanjut. |
|  | Contoh data debitur yang masuk dalam status kuning, seperti:  **\* BETRIA SIREGAR**: Keterlambatan 18 hari, tenor 18 bulan. |
|  | Memulai atau menjalankan **Surat Peringatan 2 (SP2)** kepada debitur yang menunggak lebih lanjut (biasanya setelah SP1 tidak ditindaklanjuti). Indikator adanya tindakan tegas yang akan dilakukan. |
|  | Menampilkan riwayat aktivitas atau perubahan status debitur, seperti perubahan status keterlambatan, tindakan sebelumnya, atau komunikasi yang pernah dilakukan. |

Tabel 4. 4

1. Tampilan Daftar Debitur Status Merah

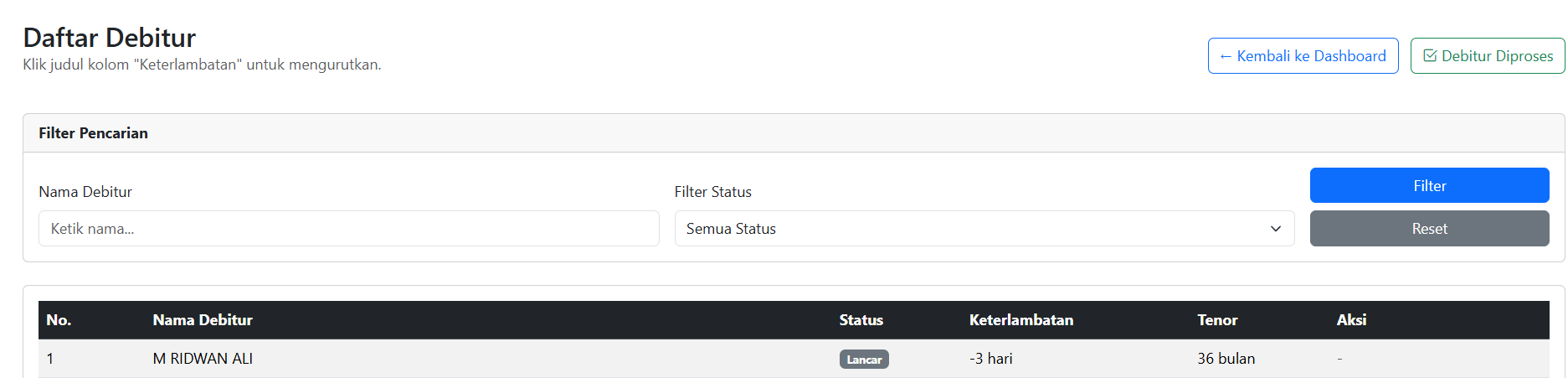


Gambar 4. 13 Tampilan Daftar Debitur Status Merah

|  |  |
| --- | --- |
| Properti | Keterangan |
|  | Halaman ini menampilkan **daftar debitur dengan status Merah**, yaitu debitur yang mengalami **keterlambatan pembayaran serius (lebih dari 30 hari)** dan termasuk dalam kategori berisiko tinggi. |
|  | Komponen antarmuka untuk menyaring daftar debitur berdasarkan nama dan status keterlambatan. Memudahkan pengguna menemukan data yang spesifik. |
|  | Indikator atau tombol untuk menunjukkan bahwa debitur tertentu sedang atau telah dalam tahap proses penanganan lebih lanjut oleh petugas. |
|  | Tombol navigasi untuk kembali ke halaman utama yang menampilkan semua daftar debitur tanpa penyaringan berdasarkan status. |
|  | Navigasi ke dashboard utama aplikasi, tempat pengguna dapat mengakses menu dan modul lainnya. |
|  | Kolom teks untuk memasukkan nama debitur sebagai kriteria pencarian dalam daftar. |
|  | Menu dropdown untuk memilih filter berdasarkan status keterlambatan debitur, pilihan meliputi:   1. **Semua Status**: Menampilkan semua data. 2. **Hijau**: Keterlambatan rendah atau tidak ada keterlambatan. 3. **Kuning**: Keterlambatan sedang. 4. **Merah**: Keterlambatan berat (biasanya di atas 30 hari). 5. **Lancar**: Tidak ada keterlambatan, pembayaran tepat waktu. |
|  | Menjalankan fungsi pencarian atau filter berdasarkan input nama dan status keterlambatan. Digunakan untuk mempersempit data yang ditampilkan. |
|  | Menghapus filter yang telah diterapkan dan mengembalikan tampilan daftar ke kondisi awal (menampilkan semua data). |
|  | Menampilkan informasi debitur dengan format tabel yang terdiri dari:   1. **No.**: Nomor urut. 2. **Nama Debitur**: Nama debitur. 3. **Status**: Warna merah yang menunjukkan keterlambatan berat. 4. **Keterlambatan**: Jumlah hari keterlambatan (contoh: 39 hari). 5. **Tenor**: Lama waktu pinjaman (contoh: 18 bulan). 6. **Aksi**: Tombol aksi lanjutan terhadap debitur tersebut. |
|  | Contoh data debitur yang ditampilkan:   1. Nama: **JUNIKA RITA SANDY SHOMBING** 2. Status: **Merah** 3. Keterlambatan: **39 hari** 4. Tenor: **18 bulan** |
|  | Memulai **tindakan penyitaan** terhadap debitur yang telah gagal membayar dalam jangka waktu lama dan telah melewati proses peringatan sebelumnya. Ini merupakan tindakan hukum atau administratif terakhir. |
|  | Menampilkan riwayat tindakan atau perubahan status pada akun debitur, misalnya histori peringatan, log pembayaran, atau catatan interaksi sebelumnya. |

Tabel 4. 5

1. Tampilan Halaman Daftar Debitur

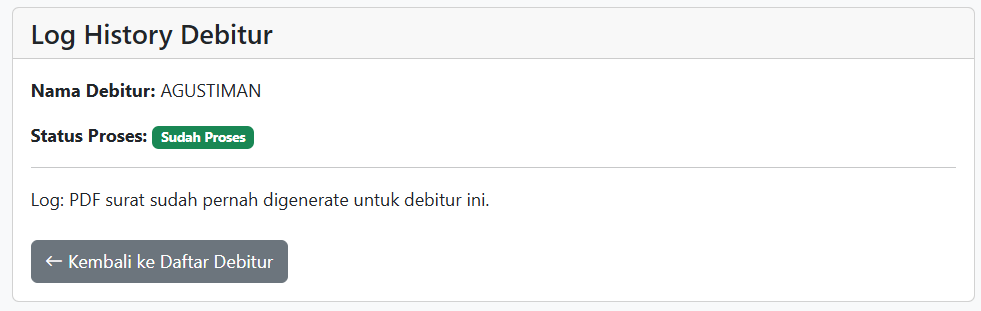


Gambar 4. 14 Tampilan Halaman Daftar Debitur

|  |  |
| --- | --- |
| Properti | Keterangan |
|  | Seluruh data debitur tanpa filter status tertentu. Halaman ini berfungsi sebagai tampilan umum yang mengakomodasi semua status: Hijau, Kuning, Merah, dan Lancar. |
|  | Komponen yang memungkinkan pengguna menyaring data debitur berdasarkan nama dan status untuk mempercepat pencarian data spesifik. |
|  | Menunjukkan bahwa salah satu debitur atau beberapa debitur sedang dalam tahap pemrosesan oleh pihak pengelola. |
|  | Tombol navigasi untuk mengakses daftar semua debitur, tidak terbatas pada status tertentu. |
|  | Untuk kembali ke halaman dashboard utama aplikasi. |
|  | Mengetik nama debitur guna melakukan pencarian langsung berdasarkan nama. |
|  | Dropdown yang memungkinkan pengguna memilih status tertentu untuk menampilkan debitur berdasarkan kategori status keterlambatan:  **Semua Status**: Tampilkan semua status tanpa filter.  **Hijau**: Pembayaran lancar / keterlambatan sangat rendah.  **Kuning**: Keterlambatan sedang.  **Merah**: Keterlambatan berat / risiko tinggi.  **Lancar**: Pembayaran sesuai jadwal. |
|  | Menjalankan proses filter berdasarkan nama debitur dan status yang dipilih dari dropdown. |
|  | Menghapus filter dan mengembalikan tampilan daftar debitur ke kondisi awal (semua data ditampilkan). |
|  | Menampilkan informasi detail debitur dalam format tabel dengan kolom:  **No.**: Nomor urut.  **Nama Debitur**: Nama lengkap debitur.  **Status**: Status pembayaran (Hijau, Kuning, Merah, Lancar).  **Keterlambatan**: Jumlah hari keterlambatan.  **Tenor**: Jangka waktu pinjaman.  **Aksi**: Tombol tindakan lanjutan. |
|  | Status dengan keterlambatan rendah (0–7 hari), menunjukkan kondisi aman. |
|  | Menunjukkan keterlambatan sedang (sekitar 8–30 hari). Perlu diperhatikan. |
|  | Status dengan keterlambatan tinggi (>30 hari). Masuk kategori risiko tinggi, bisa memicu tindakan lanjut.. |
|  | Status terbaik: tidak ada keterlambatan sama sekali. Debitur membayar tepat waktu. |
|  | |  | | --- | | Digunakan saat debitur berstatus Hijau mulai menunjukkan tanda keterlambatan. Tindakan awal berupa **Surat Peringatan 1**. |  |  | | --- | |  | |
|  | Digunakan untuk debitur berstatus Kuning yang sudah mendapatkan SP1 namun belum melakukan pembayaran. Lanjut ke **Surat Peringatan 2**. |
|  | Tindakan akhir terhadap debitur berstatus Merah, jika tidak merespons SP1 & SP2. Proses hukum berupa penyitaan aset. |
|  | Menampilkan seluruh riwayat aktivitas debitur, termasuk status sebelumnya, tindakan peringatan yang diberikan, serta histori pembayaran. |

Tabel 4. 6

1. Tampilan Halaman Log History Debitur

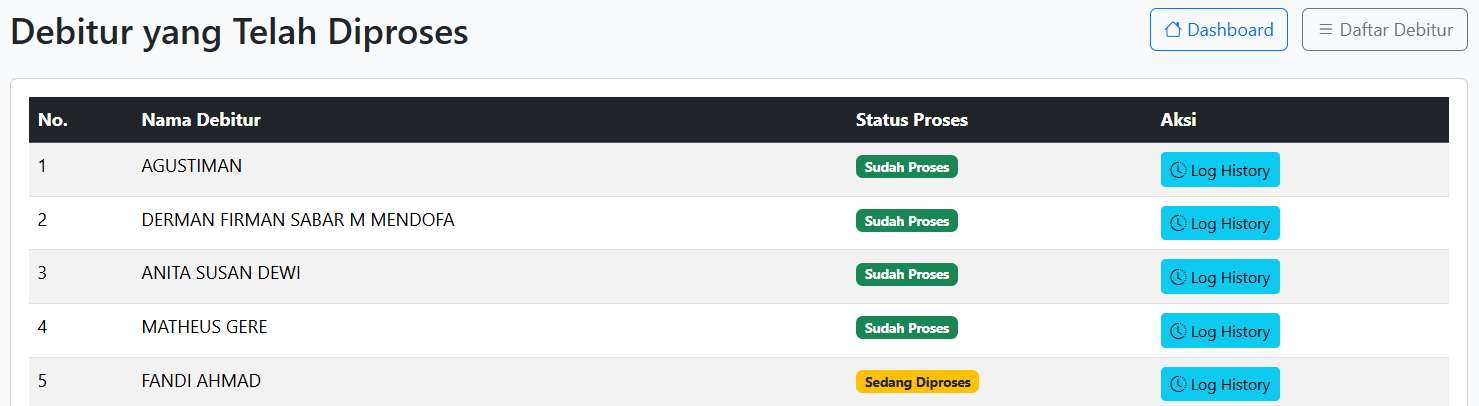


Gambar 4. 15 Tampilan Halaman Log History Debitur

|  |  |
| --- | --- |
| Properti | Keterangan |
|  | Riwayat aktivitas atau log tindakan yang pernah dilakukan terhadap debitur tertentu. Halaman ini khusus untuk satu debitur dan menunjukkan status proses serta histori dokumen. |
|  | Nama lengkap debitur yang sedang ditinjau. Dalam contoh ini, debitur yang dipilih adalah **AGUSTIMAN**. Data yang ditampilkan bersifat spesifik untuk individu ini. |
|  | Proses administratif atau tindakan terhadap debitur telah dilakukan. Indikator bahwa surat atau dokumen seperti SP1, SP2, atau lainnya telah dibuat dan di-generate (dalam format PDF, sesuai dengan log yang tertera). |
|  | Proses administratif atau tindakan terhadap debitur telah dilakukan. Indikator bahwa surat atau dokumen seperti SP1, SP2, atau lainnya telah dibuat dan di-generate (dalam format PDF, sesuai dengan log yang tertera). |

Tabel 4. 7

1. Tampilan Halaman Daftar Debitur Diproses



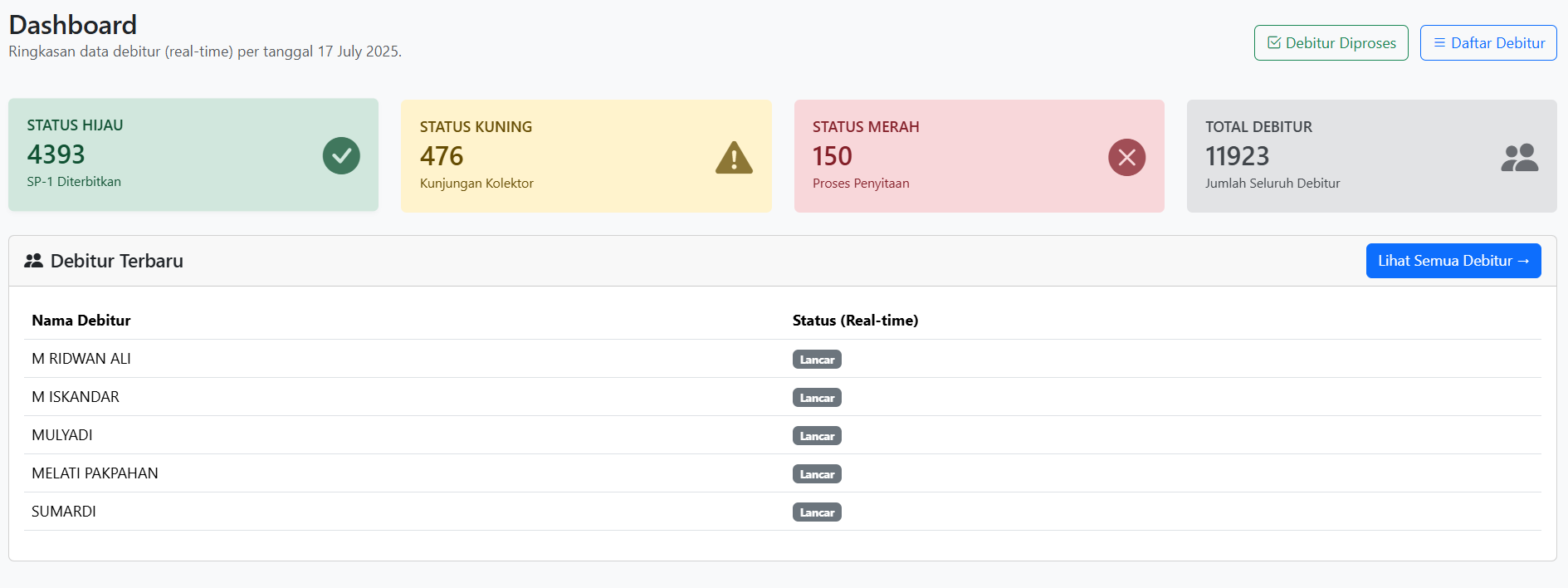
Gambar 4. 16 Tampilan Halaman Daftar Debitur Diproses

|  |  |
| --- | --- |
| Properti | Keterangan |
|  | Daftar seluruh debitur yang status prosesnya sudah berjalan, baik yang sedang diproses maupun sudah selesai diproses. |
|  | Tombol navigasi untuk kembali ke halaman utama dashboard aplikasi. |
|  | Tombol navigasi untuk kembali ke halaman daftar seluruh debitur (termasuk yang belum diproses). |
|  | Tabel yang menampilkan daftar debitur yang telah diproses. Tabel memiliki kolom:   1. **No.**: Nomor urut. 2. **Nama Debitur**: Nama lengkap debitur. 3. **Status Proses**: Menunjukkan status terkini dari proses debitur. 4. **Aksi**: Berisi tombol untuk melihat riwayat proses/log debitur. |
|  | Status proses menandakan bahwa proses terhadap debitur sudah selesai dilakukan. |
|  | Status proses menunjukkan bahwa saat ini debitur sedang dalam tahap penanganan atau pemrosesan. |
|  | Tombol aksi yang mengarahkan pengguna ke halaman riwayat/log aktivitas proses yang dilakukan terhadap debitur. |

Tabel 4. 8

## Flow Aplikasi

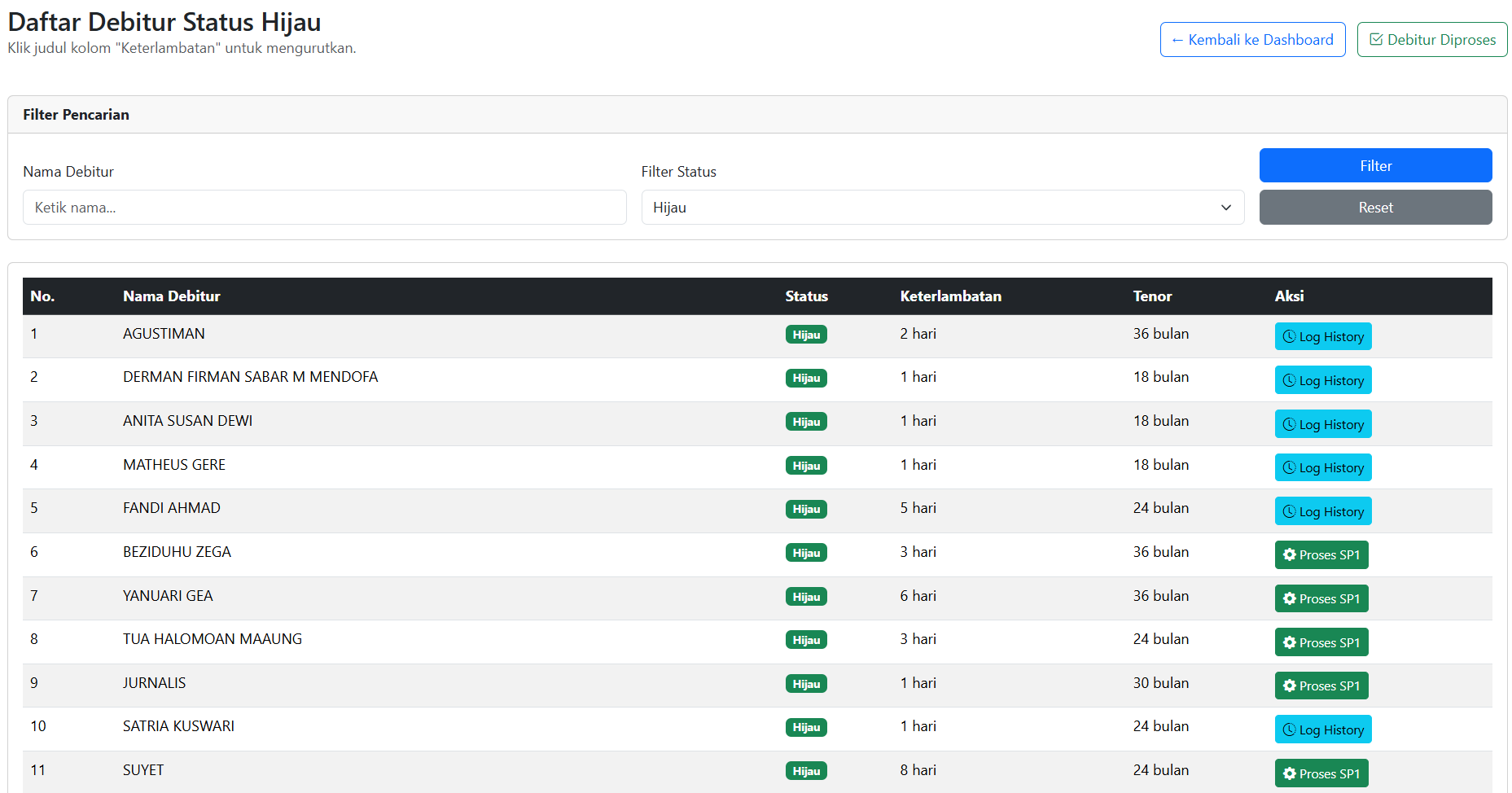
1. Alur Tampilan Dashboard Debitur



Gambar 4. 17 Dashboard

Saat pertama kali membuka aplikasi, manajemen akan disambut dengan halaman **Dashboard** utama yang menyajikan ringkasan data debitur secara *real-time*. Halaman ini menampilkan jumlah debitur dalam **Status Hijau** dengan rekomendasi tindakan penerbitan SP-1 , jumlah debitur dalam **Status Kuning** yang memerlukan Kunjungan Kolektor, dan jumlah debitur dalam **Status Merah** yang masuk dalam Proses Penyitaan. Selain itu, ditampilkan juga total keseluruhan debitur yang terdaftar dalam sistem dan sebuah daftar singkat yang memperlihatkan debitur terbaru dengan status pembayaran lancar. Dari halaman ini, pengguna dapat langsung menekan salah satu kotak status untuk menavigasi dan melihat daftar debitur yang lebih rinci sesuai kategori yang dipilih.

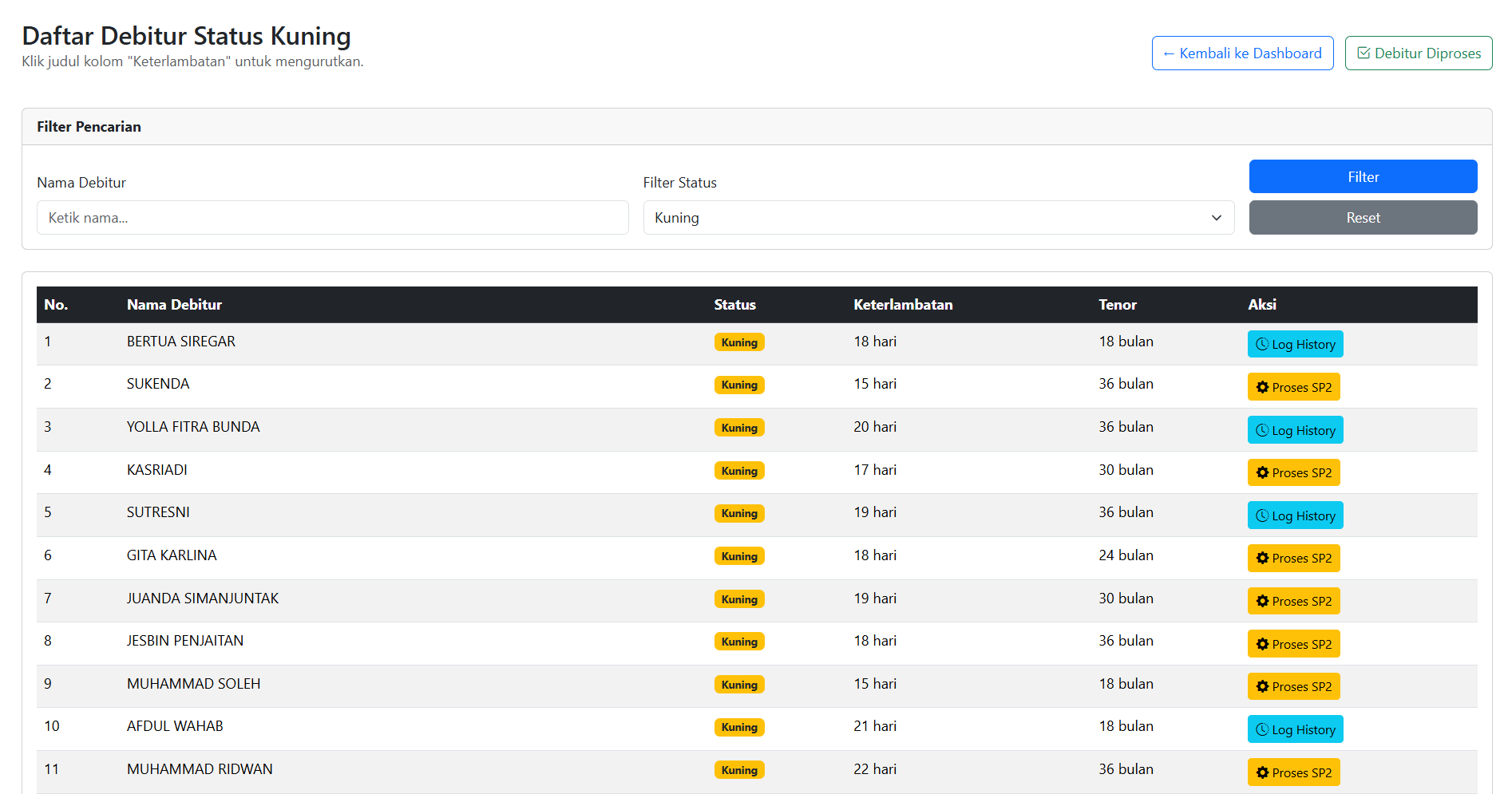
1. Alur Daftar Debitur Status Hijau



Gambar 4. 18 Daftar Debitur Status Hijau

Setelah memilih kategori Status Hijau dari dashboard, pengguna akan diarahkan ke halaman **Daftar Debitur Status Hijau**. Halaman ini secara spesifik menampilkan daftar debitur yang memiliki keterlambatan pembayaran ringan (1–14 hari). Informasi rinci yang disajikan untuk setiap debitur mencakup nama, status, lama keterlambatan, dan tenor pinjaman. Pengguna juga dapat memanfaatkan fitur **Filter Pencarian** untuk mencari debitur tertentu berdasarkan nama, sehingga mempercepat proses identifikasi. Pada kolom **Aksi**, terdapat tombol **Proses SP1** yang berfungsi sebagai langkah awal penanganan. Dengan menekan tombol ini, pengguna dapat langsung memulai proses pengiriman Surat Peringatan 1 kepada debitur yang bersangkutan, sesuai dengan prosedur yang telah ditetapkan untuk kategori risiko ini.

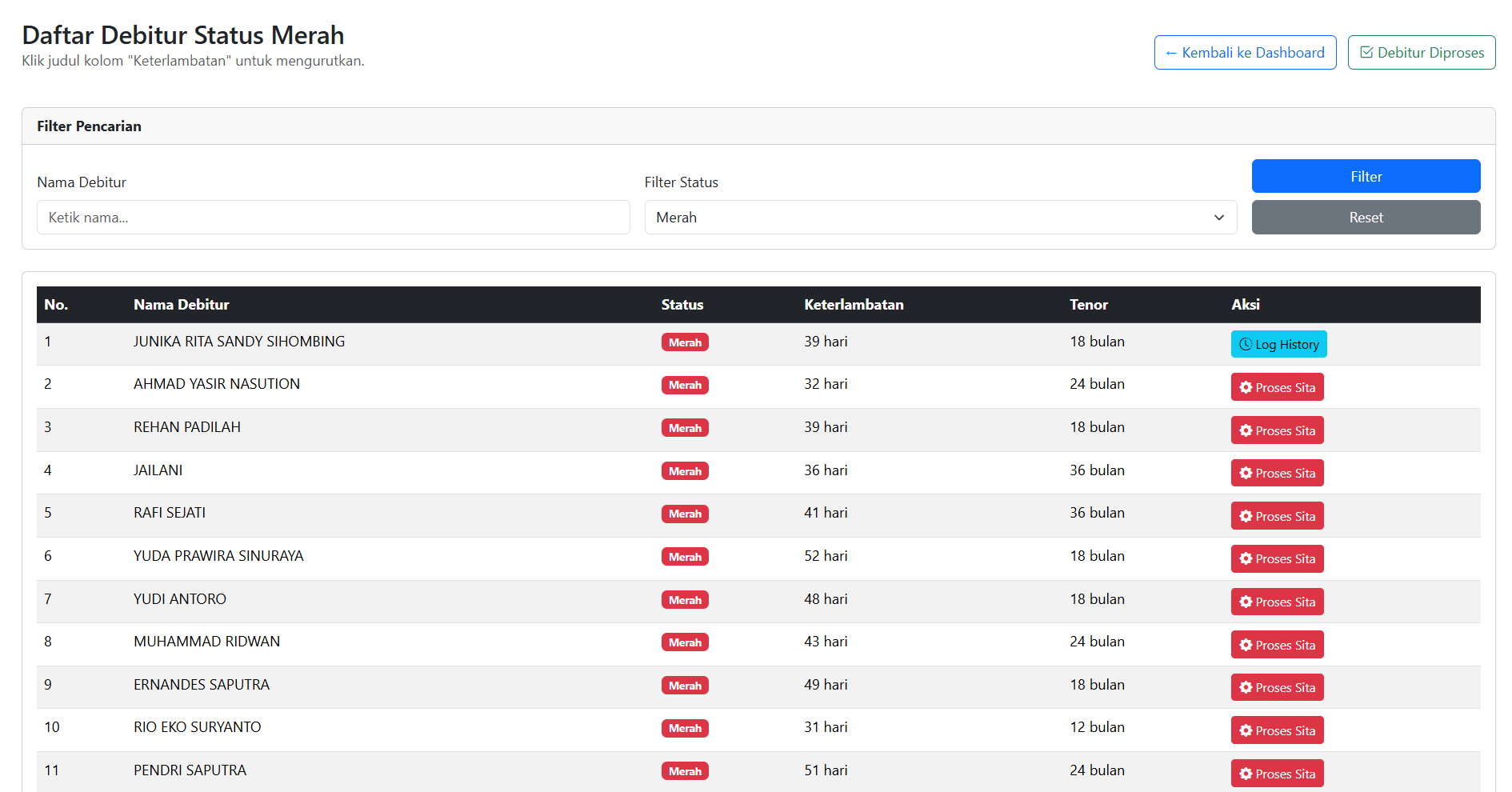
1. Alur Daftar Debitur Status Kuning



Gambar 4. 19 Daftar Debitur Status Kuning

Untuk debitur dengan tingkat risiko menengah, alurnya dimulai ketika pengguna memilih kategori **Status Kuning** dari dashboard. Pengguna kemudian akan diarahkan ke halaman **Daftar Debitur Status Kuning** , yang berisi data debitur dengan keterlambatan pembayaran antara 15 hingga 30 hari. Seperti halaman sebelumnya, di sini disajikan informasi nama debitur, status, lama keterlambatan, tenor, serta fitur filter berdasarkan nama. Perbedaan utamanya terletak pada kolom **Aksi**, di mana tombol yang tersedia adalah **Proses SP2**. Tindakan ini merupakan eskalasi dari peringatan sebelumnya dan menandakan bahwa debitur memerlukan penanganan lebih lanjut, seperti pengiriman Surat Peringatan 2.

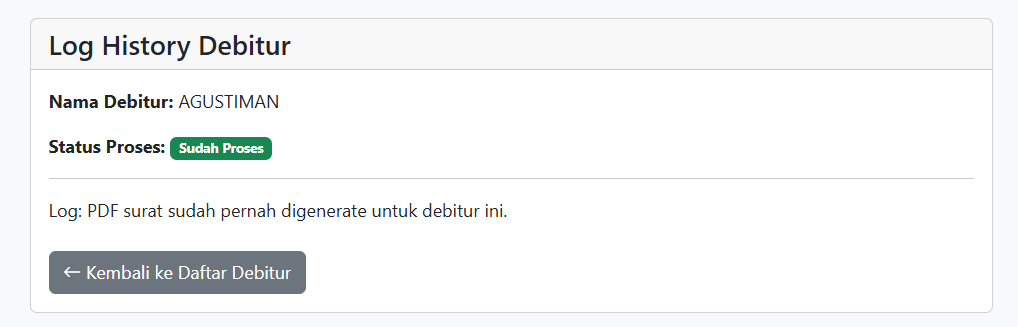
1. Alur Daftar Debitur Status Merah



Gambar 4. 20 Daftar Debitur Status Merah

Alur untuk penanganan debitur berisiko tinggi dimulai saat pengguna memilih kategori **Status Merah** pada dashboard. Pengguna akan diarahkan ke halaman **Daftar Debitur Status Merah**, yang menampilkan debitur dengan keterlambatan pembayaran kritis, yaitu lebih dari 30 hari. Halaman ini menyajikan data serupa dengan kategori lainnya, namun tindakan yang tersedia pada kolom **Aksi** adalah **Proses Sita**. Tombol ini merupakan tindakan paling tegas dalam alur penagihan, yang berfungsi untuk memulai prosedur penyitaan aset atau kendaraan dari debitur yang bersangkutan. Fitur filter juga tetap tersedia untuk memudahkan pencarian debitur spesifik dalam kategori kritis ini.

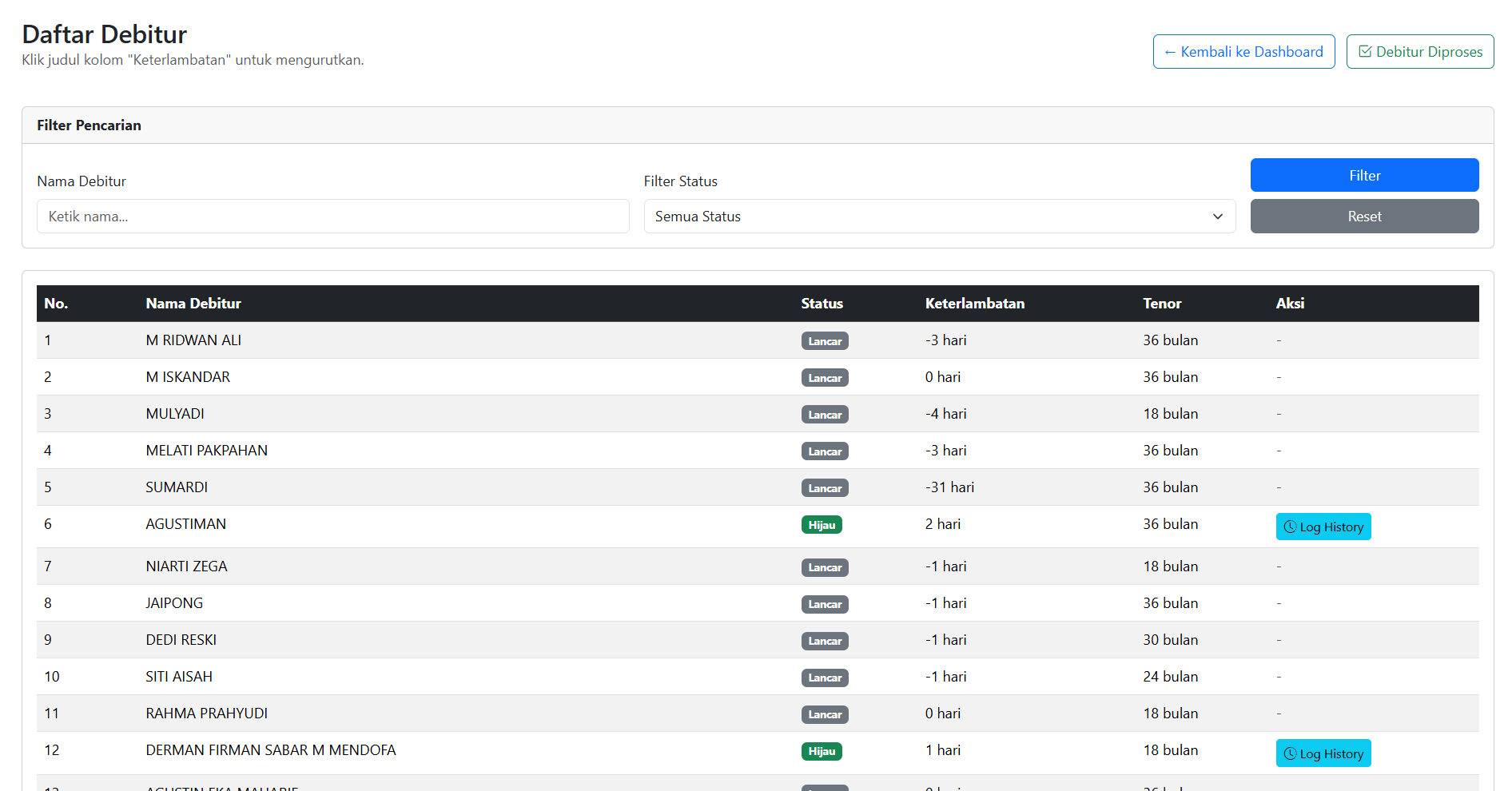
1. Alur Log History Debitur



Gambar 4. 21 Log History Debitur

pengguna akan melihat sebuah tabel yang berisi daftar para debitur. Untuk debitur yang sebelumnya sudah pernah ditangani atau diproses, pada kolom "Aksi" akan tersedia tombol **Log History**. Ketika pengguna mengklik tombol "Log History" untuk debitur tertentu (dalam contoh ini, AGUSTIMAN), sistem akan mengarahkan mereka ke halaman **Log History Debitur**. Di halaman inilah pengguna dapat melihat informasi spesifik seperti nama debitur, status terakhir yaitu **Sudah Proses**, serta catatan riwayat tindakan yang relevan seperti "PDF surat sudah pernah digenerate untuk debitur ini". Dari halaman ini, alur selanjutnya adalah pengguna dapat menekan tombol **Kembali ke Daftar Debitur** untuk kembali ke halaman daftar sebelumnya.

1. Alur Halaman Daftar Debitur

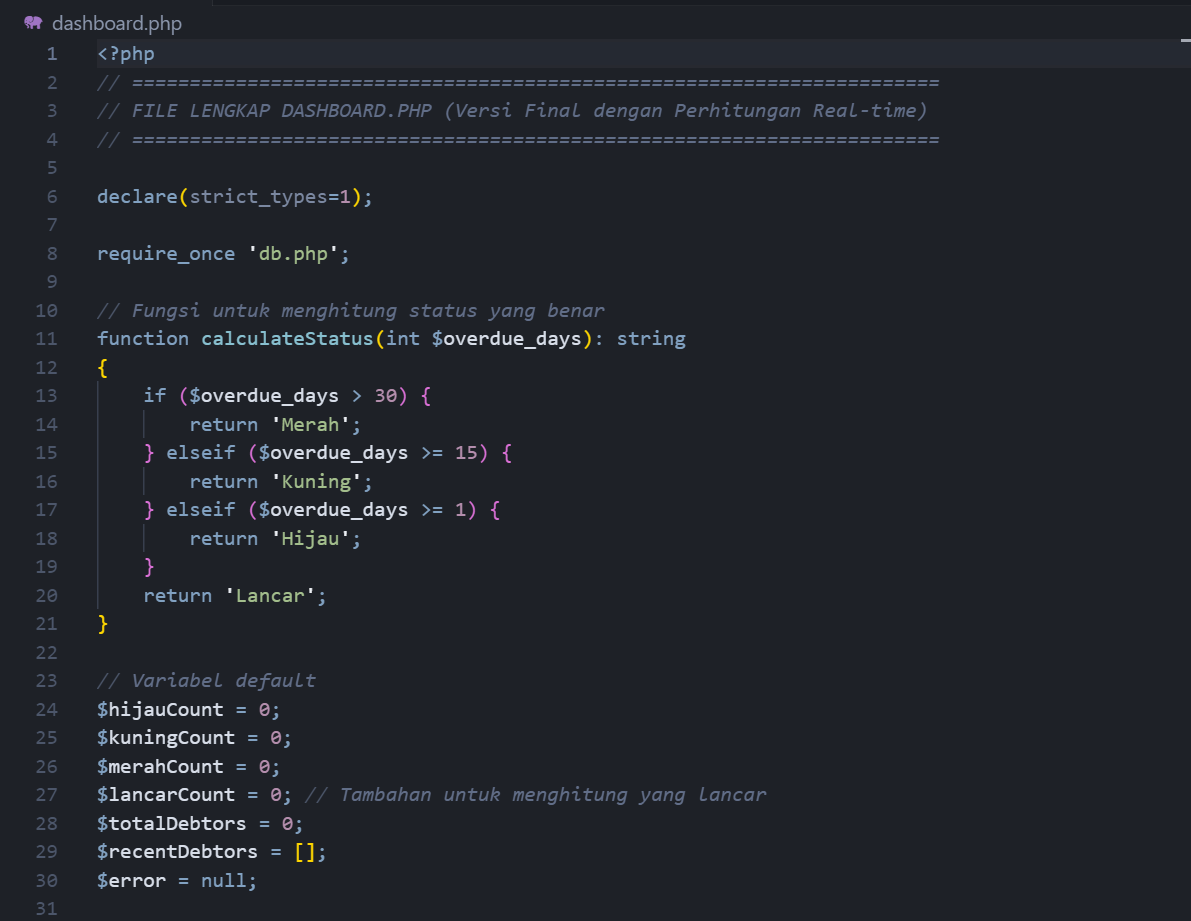


Gambar 4. 22 Halaman Daftar Debitur

Selain melihat daftar per kategori, pengguna juga dapat mengakses halaman **Daftar Debitur** untuk melihat seluruh data debitur yang ada di dalam sistem. Halaman ini dapat diakses melalui tombol "Lihat Semua Debitur" pada dashboard. Halaman ini menampilkan gabungan dari semua status, mulai dari yang berstatus **'Lancar'** (ditandai dengan keterlambatan 0 atau negatif), hingga yang berstatus **'Hijau'**, **'Kuning'**, dan **'Merah'**. Fitur **Filter Pencarian** di sini menjadi sangat fleksibel, memungkinkan pengguna untuk mencari berdasarkan nama atau memfilter berdasarkan status spesifik yang diinginkan dari *dropdown* 'Filter Status'. Dengan demikian, pengguna dapat melakukan analisis menyeluruh atau fokus pada kategori tertentu dari satu halaman terpusat.

## Codingan Aplikasi

1. Dashboard



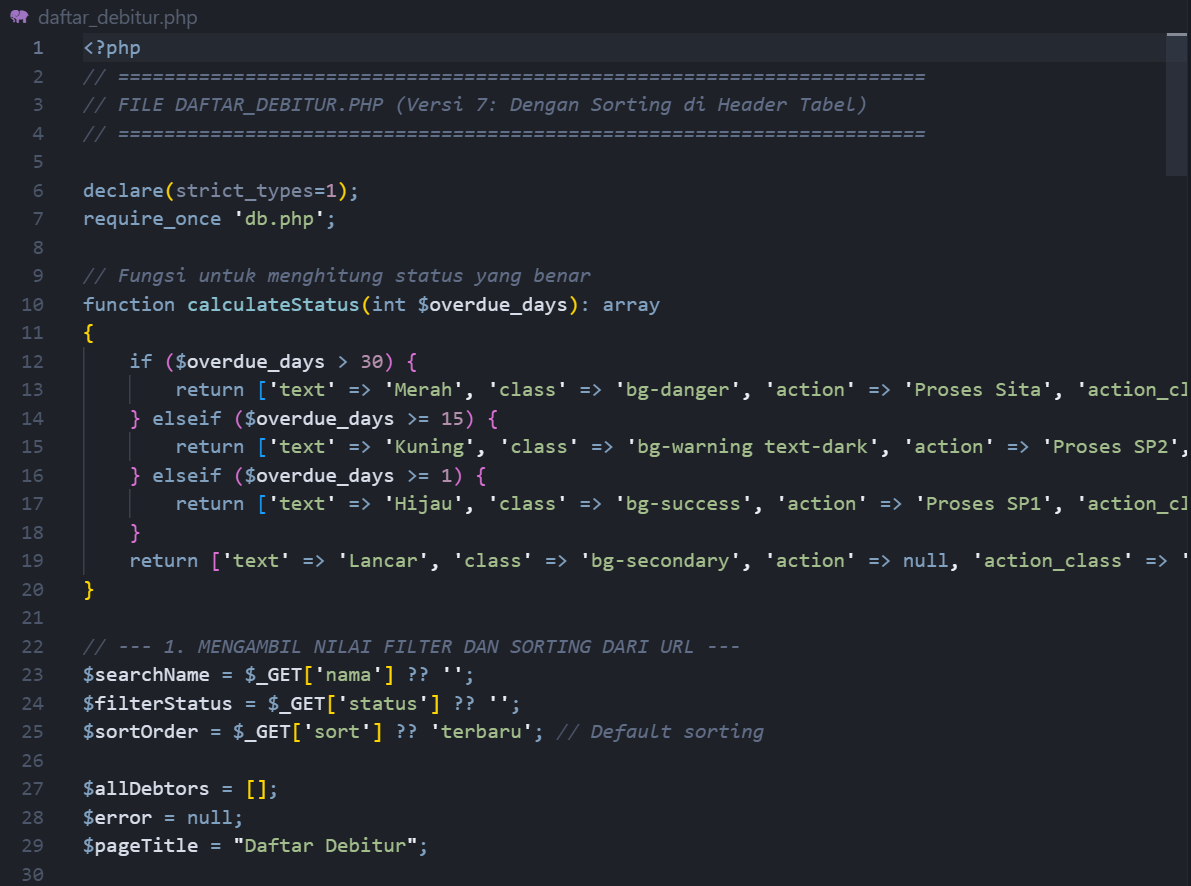
Gambar 4. 23 Code Dashboard

dashboard.php merupakan file utama untuk menampilkan halaman dashboard aplikasi. Kode ini mengintegrasikan logika backend PHP dengan struktur frontend HTML untuk menyajikan data secara dinamis.

Bagian inti dari logika ini adalah fungsi calculateStatus(), yang secara *real-time* menentukan status debitur ('Merah', 'Kuning', 'Hijau', atau 'Lancar') berdasarkan jumlah hari keterlambatan ($overdue\_days). Status 'Hijau' diberikan untuk keterlambatan 1 hari atau lebih, 'Kuning' untuk 15 hari atau lebih, dan 'Merah' untuk lebih dari 30 hari.

Saat halaman dimuat, skrip akan mengambil seluruh data keterlambatan dari database, kemudian melakukan iterasi untuk menghitung jumlah total debitur pada setiap kategori status menggunakan *switch statement*. Hasil perhitungan ini (misalnya $hijauCount, $kuningCount) kemudian disajikan secara visual pada kartu-kartu ringkasan di bagian atas halaman. Selain itu, kode ini juga mengambil lima debitur terbaru untuk ditampilkan dalam tabel 'Debitur Terbaru', di mana status setiap debitur juga dihitung secara *real-time* dan diberi label warna yang sesuai menggunakan kelas CSS dari Bootstrap. Keseluruhan proses ini dibungkus dalam blok try-catch untuk menangani jika terjadi kesalahan pada koneksi database.

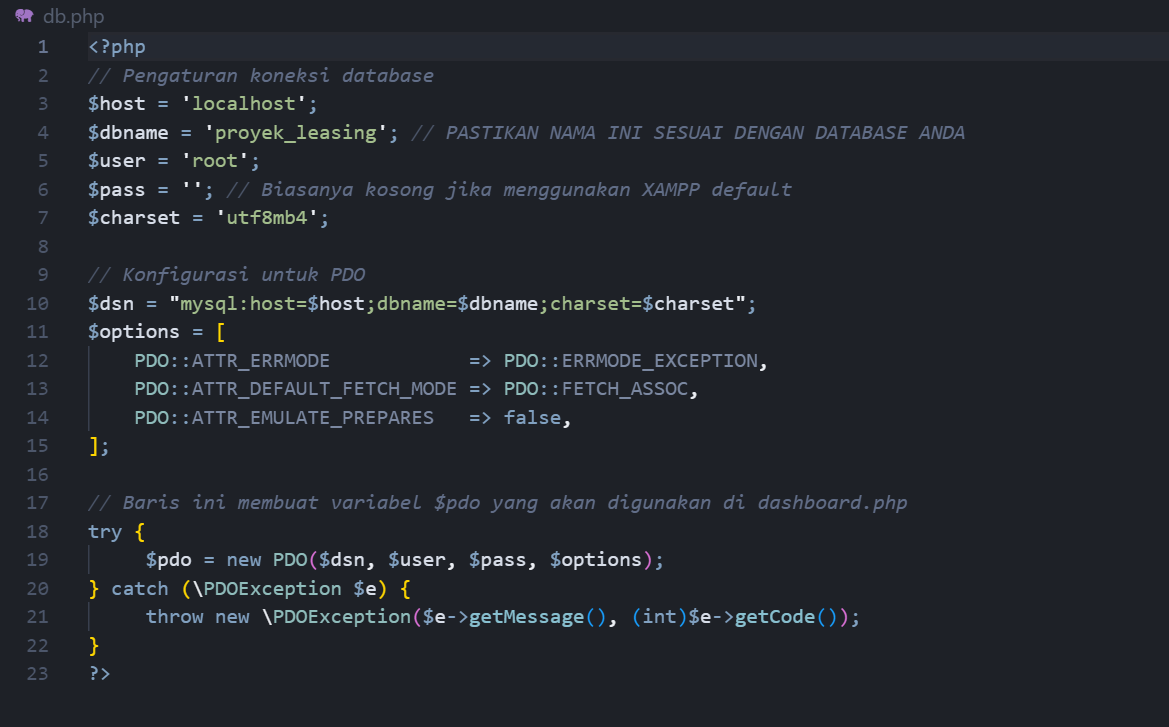
1. Daftar Debitur



Gambar 4. 24 Code Daftar Debitur

daftar\_debitur.php sebuah halaman dinamis yang berfungsi untuk menampilkan, memfilter, dan mengurutkan daftar semua debitur. Kode ini mengambil parameter dari URL ($\_GET) untuk menerapkan filter berdasarkan nama (nama) dan status (status), serta untuk mengurutkan data (sort). Sebuah fitur utama adalah bagaimana skrip ini membangun *query* SQL secara dinamis untuk menyertakan pencarian nama dan klausa pengurutan (ORDER BY). Penting untuk dicatat bahwa penyaringan berdasarkan status (Hijau, Kuning, dll.) dilakukan di sisi PHP setelah data diambil dari database; skrip akan melakukan iterasi pada hasil dan menerapkan logika dari fungsi calculateStatus. Fungsi calculateStatus pada halaman ini telah disempurnakan untuk mengembalikan sebuah *array* yang berisi teks status, kelas CSS untuk *badge* dan tombol, serta teks untuk tombol aksi (Proses SP1, Proses SP2, dll.). Informasi ini kemudian digunakan di bagian HTML untuk secara dinamis merender setiap baris tabel dengan warna status dan tombol aksi yang sesuai, menciptakan antarmuka yang interaktif dan responsif.

1. Database

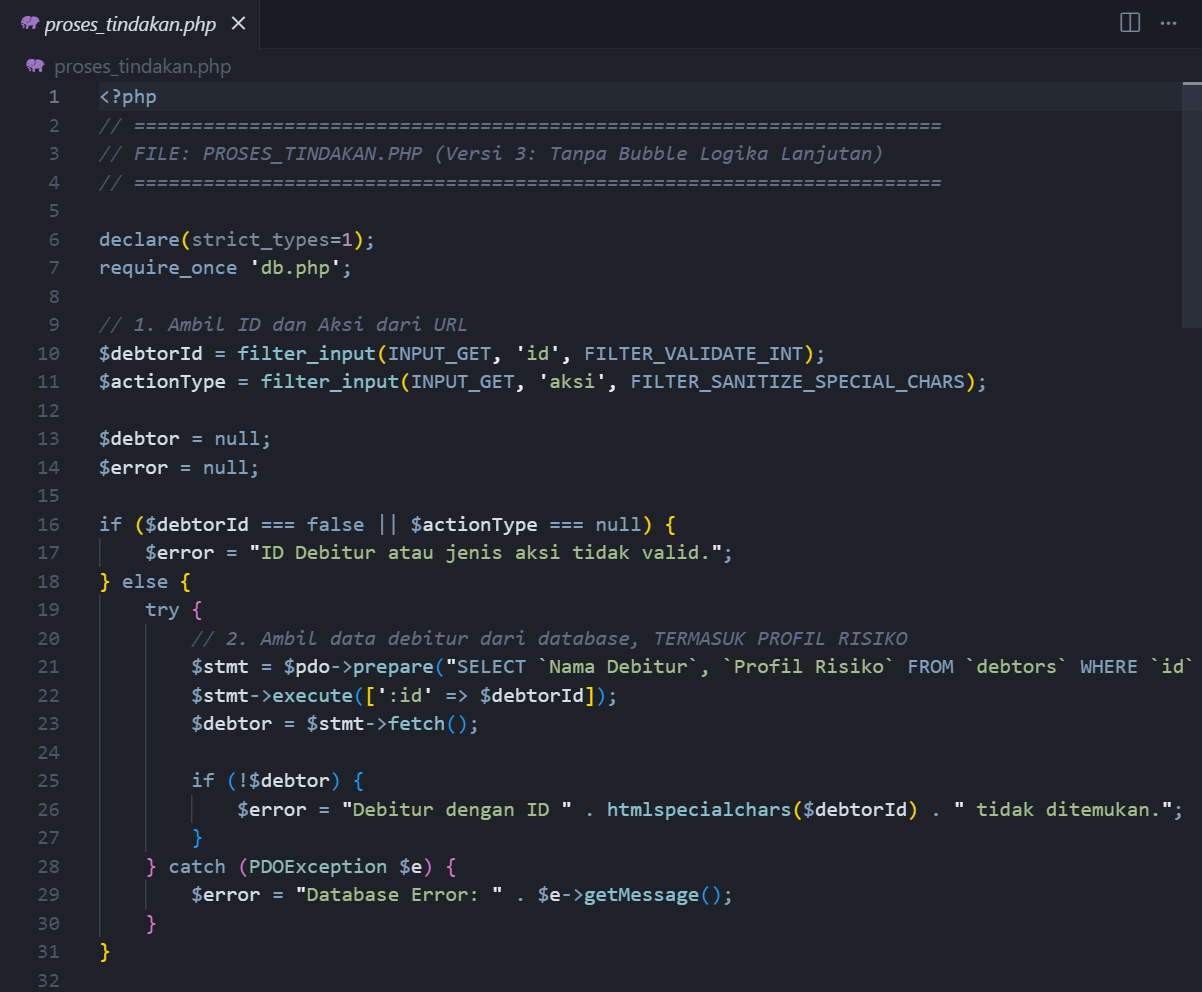


Gambar 4. 25 Code DB

sebuah file konfigurasi yang memegang peranan krusial dalam menghubungkan aplikasi dengan database **MySQL**. File ini mendefinisikan semua parameter koneksi yang diperlukan, seperti *host*, nama *database* (proyek\_leasing), *user*, dan *password*.

Koneksi dibuat menggunakan **PDO (PHP Data Objects)**, yang merupakan antarmuka modern dan aman untuk mengakses database di PHP. Beberapa opsi penting dikonfigurasi untuk koneksi PDO, di antaranya adalah PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION untuk memastikan setiap kesalahan database akan memicu *exception* yang bisa ditangani, dan PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC agar hasil *query* dikembalikan dalam bentuk *array* asosiatif yang mudah digunakan. Seluruh proses koneksi dibungkus dalam blok try-catch untuk menangani kegagalan koneksi secara efektif. Hasil akhir dari file ini adalah variabel $pdo, yang kemudian diimpor (require\_once) oleh file lain seperti dashboard.php dan daftar\_debitur.php untuk menjalankan semua operasi database.

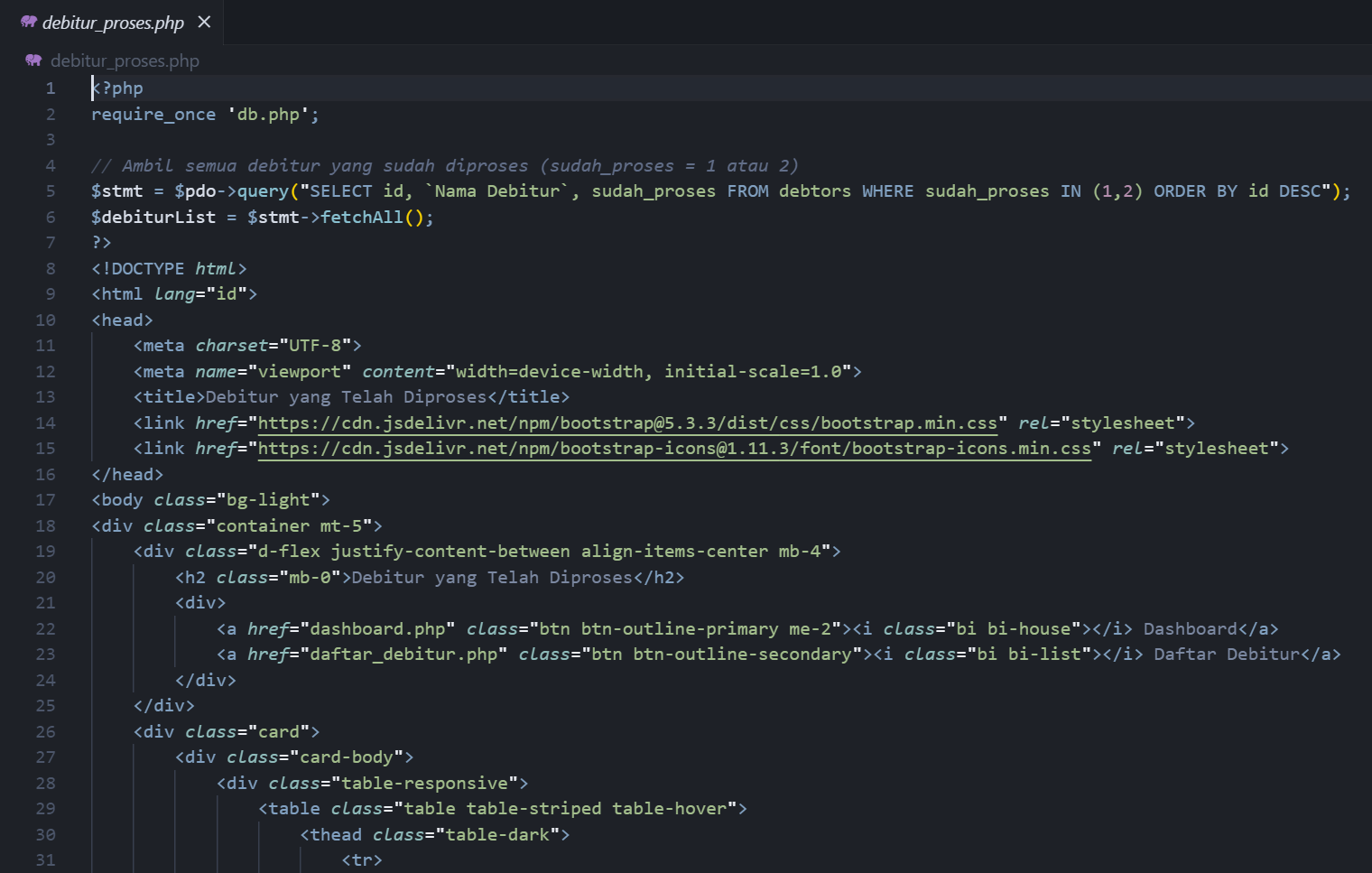
1. Proses Tindakan



Gambar 4. 26 Code Proses Tindakan

proses\_tindakan.php merupakan skrip *backend* untuk halaman "Konfirmasi Tindakan". Skrip ini dirancang agar dapat digunakan kembali (*reusable*) untuk semua jenis tindakan (Hijau, Kuning, dan Merah).Pertama, kode ini secara aman mengambil ID debitur dan jenis aksi dari parameter URL menggunakan filter\_input. Dengan menggunakan ID tersebut, skrip kemudian mengambil data spesifik debitur, seperti nama dan profil risiko, dari database. Bagian penting dari logika ini adalah *switch statement* yang menentukan deskripsi tindakan yang sesuai berdasarkan jenis aksi yang diterima. Misalnya, jika aksinya adalah 'Hijau', deskripsi akan menjadi "memproses Surat Peringatan 1 (SP1)". Informasi yang telah diambil dari database dan deskripsi tindakan yang telah ditentukan ini kemudian ditampilkan di dalam template HTML untuk menyajikan halaman konfirmasi yang informatif kepada pengguna.

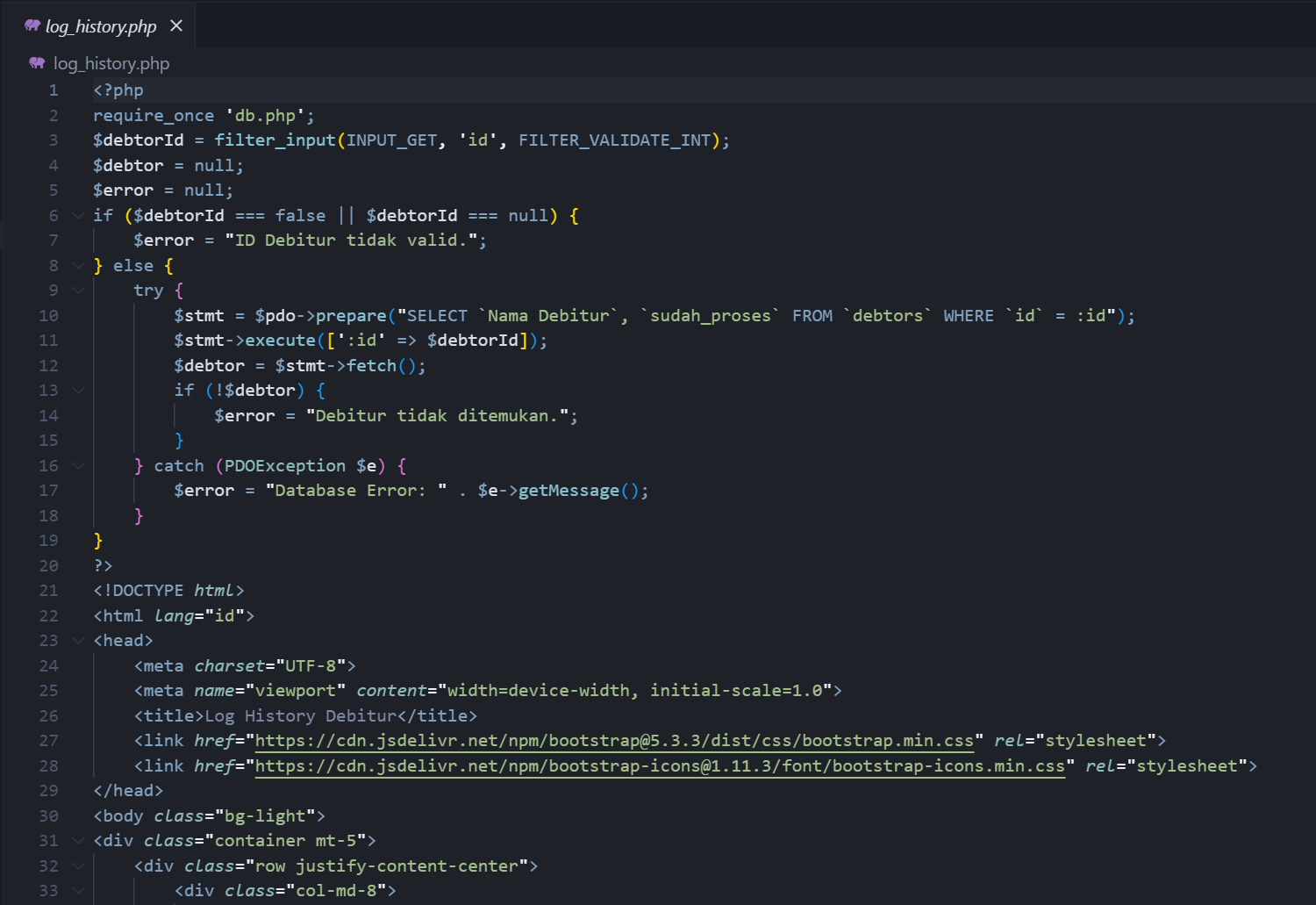
1. Debitur Proses



Gambar 4. 27 Debitur Proses

menampilkan daftar **debitur yang telah diproses**, baik yang sedang diproses (sudah\_proses = 2) maupun yang telah selesai diproses (sudah\_proses = 1). Di bagian atas, file db.php di-*require* untuk menghubungkan ke database. Kemudian dilakukan query ke tabel debtors untuk mengambil data semua debitur yang nilai sudah\_proses-nya adalah 1 atau 2. Data yang diambil meliputi id, Nama Debitur, dan sudah\_proses, lalu disimpan dalam array $debiturList. Setelah itu, bagian HTML digunakan untuk membuat tampilan antarmuka. Halaman menggunakan **Bootstrap** untuk mempercantik tampilan dengan gaya modern dan responsif. Di bagian atas halaman terdapat judul "Debitur yang Telah Diproses" dan dua tombol navigasi: satu untuk menuju halaman dashboard (dashboard.php) dan satu lagi untuk kembali ke daftar seluruh debitur (daftar\_debitur.php). Data yang sudah diambil dari database ditampilkan dalam bentuk tabel yang terdiri dari kolom **Nomor**, **Nama Debitur**, **Status Proses**, dan **Aksi**. Jika tidak ada debitur yang memenuhi kriteria, maka akan muncul pesan bahwa tidak ada debitur yang sudah diproses. Jika ada, maka data akan ditampilkan baris demi baris. Status proses ditampilkan dengan badge warna-warni: **warna kuning** untuk "Sedang Diproses" dan **warna hijau** untuk "Sudah Proses". Terakhir, di kolom aksi tersedia tombol kecil untuk melihat riwayat log (log\_history.php) berdasarkan id debitur terkait.

1. Log History



Gambar 4. 28 Log History

Halaman log\_history.php dibuat untuk menampilkan **riwayat dan status proses debitur** berdasarkan ID yang dikirim melalui URL (contoh: log\_history.php?id=3). Pertama-tama, file db.php dipanggil untuk memastikan koneksi database tersedia. Selanjutnya, sistem mencoba mengambil nilai ID dari URL menggunakan filter\_input() dengan validasi agar hanya menerima angka. Jika ID tidak valid atau kosong, maka akan muncul pesan error bahwa "ID Debitur tidak valid". Jika ID valid, maka dilakukan query ke database menggunakan PDO prepare untuk mengambil data debitur dengan ID tersebut. Jika data ditemukan, maka informasi debitur akan ditampilkan seperti nama dan status prosesnya. Namun jika data tidak ditemukan atau ada kesalahan query, maka pesan error akan ditampilkan seperti "Debitur tidak ditemukan" atau "Database Error". Setelah itu, tampilan halaman dimuat menggunakan Bootstrap. Di dalamnya terdapat informasi detail nama debitur, status proses (dengan badge berwarna), serta tombol aksi tambahan **"Konfirmasi Sudah Diproses"** dan **"Regenerate PDF"** yang muncul jika status debitur belum selesai (sudah\_proses != 1). Tombol **"Konfirmasi Sudah Diproses"** akan mengirim data melalui AJAX (tanpa reload halaman) ke server untuk memperbarui status debitur di database menjadi "1" (selesai diproses). Setelah berhasil, tampilan badge akan berubah menjadi hijau (Sudah Proses) dan tombol akan hilang.

Sementara itu, tombol **"Regenerate PDF"** digunakan untuk membuat ulang dokumen surat (biasanya SP1, SP2, atau penyitaan) dalam bentuk file PDF. Proses ini juga dilakukan melalui AJAX dan hasilnya akan langsung diunduh oleh pengguna. Jika proses berhasil, akan muncul notifikasi bahwa PDF berhasil digenerate ulang. Jika gagal, akan muncul notifikasi error. Di bagian bawah file, terdapat kode PHP tambahan yang menangani proses POST jika ada permintaan untuk mengonfirmasi status melalui tombol AJAX sebelumnya.

# BAB V

# KESIMPULAN DAN SARAN

Berdasarkan hasil analisis dan pengembangan sistem rekomendasi tindakan kepada debitur berbasis Business Intelligence (BI) pada PT Capella Multidana, dapat disimpulkan bahwa sistem ini mampu memberikan kontribusi nyata dalam meningkatkan efisiensi proses pemantauan keterlambatan pembayaran. Sistem yang dibangun berhasil mengklasifikasikan debitur ke dalam tiga kategori risiko, yaitu hijau, kuning, dan merah, berdasarkan analisis data keterlambatan yang diperoleh dari file internal perusahaan. Klasifikasi ini memungkinkan manajemen untuk lebih cepat mengenali debitur yang berpotensi bermasalah dan mengambil tindakan yang sesuai secara proaktif. Melalui dashboard interaktif dan fitur filter yang tersedia, sistem mampu menyajikan informasi secara real-time dan visual, sehingga proses pengambilan keputusan menjadi lebih tepat sasaran dan berbasis data. Pemanfaatan teknologi seperti Python, Pandas, React.js, serta MySQL dalam proses pengolahan dan penyajian data, turut memperkuat akurasi dan kemudahan penggunaan sistem secara keseluruhan. Selain itu, integrasi fitur notifikasi dan rekomendasi otomatis juga menjadi nilai tambah yang mendukung tim manajemen dan penagihan dalam menangani debitur secara cepat dan efektif.

Namun demikian, pengembangan sistem ini masih memiliki ruang untuk penyempurnaan. Oleh karena itu, beberapa saran yang dapat diajukan antara lain adalah perlunya integrasi sistem secara langsung dengan basis data operasional perusahaan agar proses analisis dapat berjalan secara otomatis tanpa perlu input manual berupa file Excel. Selain itu, pengembangan fitur lanjutan seperti pelacakan histori tindakan debitur, penjadwalan pengingat otomatis, serta laporan berkala juga akan memberikan manfaat lebih bagi pengambilan keputusan strategis. Selanjutnya, pelatihan teknis bagi pengguna akhir juga penting untuk memastikan bahwa seluruh fitur sistem dapat dimanfaatkan secara optimal oleh manajemen dan staf operasional. Untuk ke depannya, penerapan metode analitik yang lebih canggih seperti machine learning juga dapat dipertimbangkan guna meningkatkan kemampuan sistem dalam memprediksi risiko secara lebih akurat dan adaptif terhadap perubahan data historis. Dengan demikian, sistem ini diharapkan dapat terus dikembangkan dan menjadi solusi jangka panjang dalam mendukung pengelolaan risiko kredit yang lebih terstruktur, efisien, dan berbasis teknologi informasi.

# DAFTAR PUSTAKA

1Kelly Hermanto, 2Darrius Salim, 3Bryan Wu, 4Odelia Regina Salim,  5Ruby Belinda Gunadi. (n.d.). *Penggunaan Python Untuk Menganalisis Pola Penyebaran Covid-19 Di Masa  Pandemi*.

Aisyah, S., & Purba, W. (n.d.). *APLIKASI SISTEM PENDUKUNG KEPUTUSAN ANALISIS KELAYAKAN PEMBERIAN KREDIT MENGGUNAKAN METODE SAW PADA PERUSAHAAN LEASING*.

Alfeno, S., Saptono, A., & Pamungkas, A. B. (n.d.). *PERANCANGAN ANTAR MUKA APLIKASI MONITORING PEMBAYARAN ANGSURAN KENDARAAN BERMOTOR BERBASIS ANDROID PADA PT BENTARA SINERGIES MULTIFINANCE*.

Andalusi, R., & Irfanudin, A. M. (2021). PENGARUH KUALITAS PELAYANAN DAN KUALITAS PRODUK TERHADAP KEPUASAN PELANGGAN PERUSAHAAN PEMBIAYAAN (LEASING) STUDI KASUS PT. BCA FINANCE KANTOR PUSAT PONDOK INDAH. *Jurnal ARASTIRMA Fakultas Ekonomi Program Studi Manajemen UNPAM*, *1*(2), 204–215.

Asyrofi, R. R., & Asyrofi, R. (2023). IMPLEMENTASI APLIKASI JUPYTER NOTEBOOK SEBAGAI ANALISIS KRETERIA PLAGIASI DENGAN TEKNIK SIMANTIK. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *8*(2), 627–637. https://doi.org/10.29100/jipi.v8i2.3699

Darudiato, S., Sigit, ;, Santoso, W., & Wiguna, ; Setiady. (n.d.). *Business Intelligence: Konsep dan Metode (Suparto Darudiato; dkk)*.

Emirzaki, M., Ajie, H., & Nurhidayat, D. (n.d.). *PENGEMBANGAN MODUL FRONT-END WEBSITE SISTEM MANAJEMEN ASET UNIT PELAYANAN TEKNIK TEKNOLOGI INFORMASI DAN KOMUNIKASI UNIVERSITAS NEGERI JAKARTA*.

Hendro Poerbo Prasetiya 1), M. S. 2). (n.d.). *PEMANFAATAN BUSINESS INTELLIGENCE  DI PERGURUAN TINGGI*.

Kumalasari Subroto, V., & Endaryati, E. (2021). *BUSINESS INTELLIGENCE DAN KESUKSESAN BISNIS di ERA DIGITAL*. *1*(2), 41–47. http://journal.stiestekom.ac.id/index.php/dinamikapage41

Lilis Nurellisa\*1, D. F. (n.d.). *ANALISIS REKOMENDASI CALON DEBITUR MOTOR PADA PT.XYZ  MENGGUNAKAN ALGORITMA C4.5*.

Lukman Hasim. (n.d.). *PENERAPAN BUSINESS INTELLIGENCE PADA MANAJEMEN DASHBOARD  REPORT PERUSAHAAN ASURANSI*.

Maulida Solihat, N., Studi Teknik Informatika, P., Sumbawa Jln Raya Olat Maras, U., Alang, B., Hulu, M., Sumbawa, K., & Tenggara Barat, N. (2021). REKAYASA APLIKASI CENTER RUMAH KOST BERBASIS WEB DI KABUPATEN SUMBAWA. *Jurnal Manajemen Informatika & Sistem Informasi)*, *4*(2). http://e-journal.stmiklombok.ac.id/index.php/misi

Murti, S. K., Informatika, J., Industri, T., Sujarwo Badan, A., & Informasi, S. (n.d.). *Membangun Antarmuka Pengguna Menggunakan ReactJs untuk Modul Manajemen Pengguna*.

Oktaviani, D., & Harahap, L. (2022). ANALISIS PENGENDALIAN INTERNAL TERHADAP PIUTANG LEASE: STUDI KASUS PADA PERUSAHAAN PEMBIAYAAN PT. FIF. *RELEVAN*, *3*(1), 28–40.

Pangestika, R., & Dirgahayu, R. T. (n.d.). *Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta*.

Permata Sari, A. (n.d.). RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN TALENT FILM BERBASIS APLIKASI WEB. *Jurnal Informatika Terpadu*, *6*(1), 29–37. https://journal.nurulfikri.ac.id/index.php/JIT

Riziq sirfatullah Alfarizi, M., Zidan Al-farish, M., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING. In *Karimah Tauhid* (Vol. 2, Issue 1).

Rizky, S. A., Yesputra, R., & Santoso, S. (2021). PREDIKSI KELANCARAN PEMBAYARAN CICILAN CALON DEBITUR DENGAN METODE K-NEAREST NEIGHBOR. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, *7*(2), 195–202. https://doi.org/10.33330/jurteksi.v7i2.1078

Satria, A., Umar Kholid, A., Muttaqin, D., Prasnowo, H., Rizki Maulana, M., Farhan, M., Ramdani, M., Faren Rajendra Ratosila, M., Muhammad Yahya, P., & Suwandi, R. (n.d.). *MENGENAL APA ITU MICROSOFT EXCEL*. https://jurnalmahasiswa.com/index.php/appa

Sukemi Kamto Sudibyo, T. S. (n.d.). *ANALISA SISTEM PENCATATAN DAN MONITORING PIUTANG LEASING*.

Sulistyoningsih, W., Nyoman, I., Anggara Wiajaya, Y., & Alam, H. S. (n.d.). Penerapan Model Business Intelligence Pada Perusahaan Retail XLT Untuk Meningkatkan Strategi Pemasaran. In *Jurnal Ilmiah Teknologi Informasi Asia* (Vol. 17, Issue 1).