

# **RANCANGAN FITUR DAN DESAIN SISTEM**

## **POS BERBASIS LARAVEL (SINGLE OUTLET)**

Fokus: Offline-first POS + Approval Supervisor, Smart Inventory, Receipt Maker

Payment: Cash dan E-Wallet (UI Only)

Role: Kasir dan Supervisor

Versi: 1.0

Tanggal: 20 Januari 2026

## Riwayat Revisi

Versi	Tanggal	Perubahan	Penanggung jawab
1.0	20 Januari 2026	Dokumen awal: scope, rancangan database, flow program, API, dan milestone.	-

## Daftar Isi

Catatan: jika daftar isi belum terisi, lakukan Update Field pada aplikasi pengolah dokumen.

### 1. Ruang Lingkup dan Batasan

Dokumen ini mendefinisikan rancangan matang untuk sistem POS berbasis Laravel dengan batasan berikut:

- Single outlet (satu toko).
- Role hanya dua: Kasir dan Supervisor (management).
- Pembayaran mendukung Cash dan E-Wallet; untuk tahap pertama E-Wallet hanya tampilan UI (tanpa integrasi backend ke payment gateway).
- Fitur pembeda yang wajib: Offline-first POS dengan mekanisme approval, Smart Inventory (reorder recommendation), dan Receipt Maker (template struk).

### Tujuan

Menyediakan POS yang tetap dapat beroperasi saat koneksi internet terputus, memiliki kontrol fraud melalui approval supervisor, dan menghasilkan rekomendasi pengadaan stok yang praktis.

### Asumsi Operasional

- Perangkat kasir: browser modern (tablet/desktop) yang menjalankan PWA.
- Database utama MySQL.
- Sistem dapat digunakan oleh satu atau beberapa perangkat pada outlet yang sama; potensi konflik stok saat offline tetap diantisipasi.
- Invoice final ditetapkan oleh server (untuk mencegah bentrok nomor transaksi antar perangkat).

## 2. Fitur yang Akan Dibuat

### 2.1 RBAC Minimal (Kasir dan Supervisor)

Kontrol akses disederhanakan menjadi dua role. Selain pembatasan menu, beberapa aksi memerlukan approval supervisor (PIN + alasan) agar tetap aman walau offline.

Matriks Akses (ringkas):

Aksi	Kasir	Supervisor
Buat transaksi, scan barcode, checkout	Ya	Ya

Diskon <= limit	Ya	Ya
Diskon > limit (approval)	Request	Approve
Price override (ubah harga item)	Tidak	Approve
Void transaksi (dengan alasan)	Ya (terbatas)	Ya
Stock adjustment / opname	Tidak	Ya
Kelola produk & template struk	Tidak	Ya

## 2.2 Pembayaran (Cash dan E-Wallet UI Only)

- Cash: input uang diterima, hitung kembalian.
- E-Wallet (UI only): kasir memilih metode E-Wallet, menampilkan instruksi/QR placeholder, mengisi reference (opsional), lalu menandai pembayaran sebagai dicatat.
- Untuk tahap pertama tidak ada callback/konfirmasi gateway; status payment dicatat sebagai RECORDED.

Kebijakan pembayaran untuk tahap awal: transaksi dianggap selesai (PAID) saat kasir menekan konfirmasi pembayaran.

## 2.3 Stock Management (Ledger)

Stok dikelola berbasis ledger (stock\_movements) agar setiap perubahan stok memiliki jejak audit. Nilai on\_hand dapat disimpan sebagai cache (stock\_items) untuk performa.

- SALE\_OUT: stok keluar saat transaksi PAID.
- RETURN\_IN: stok masuk saat return (V1).
- ADJUSTMENT: koreksi opname oleh supervisor.
- SYNC\_CORRECTION: koreksi akibat konflik/rekonsiliasi saat sync.

## 2.4 Offline-first POS + Sync

- POS dibangun sebagai PWA: aset dicache via service worker.
- Katalog, settings, dan cache stok disimpan di IndexedDB.
- Transaksi offline disimpan ke local queue dan didorong (push) saat online kembali.
- Server menerapkan idempotency untuk mencegah duplikasi saat retry.

Approval supervisor harus berfungsi offline dengan verifikasi PIN (hash) dan pencatatan reason.

## 2.5 Smart Inventory

- Menghitung reorder point dan suggested reorder quantity.
- Dasar perhitungan: rata-rata penjualan 7 hari, lead time, dan safety stock.
- Output: daftar produk yang perlu dibeli beserta jumlah yang disarankan.

## 2.6 Receipt Maker

- Supervisor dapat mengelola template struk (header/body/footer) berbasis placeholder.
- Kasir dapat preview, print, dan reprint struk.
- Receipt menyimpan template version agar hasil reprint konsisten.

# 3. Arsitektur Sistem

## 3.1 Komponen

- Backend: Laravel (API + business rules), MySQL.

- Frontend POS: PWA (Inertia/Vue/React atau Blade/Alpine), IndexedDB untuk offline storage.
- Sync Engine: modul di frontend untuk push/pull dan resolusi konflik dasar.

### 3.2 Prinsip Integritas Data

- Checkout PAID harus atomik: sales, sale\_items, payments, stock\_movements disimpan dalam satu transaksi DB.
- Invoice final ditetapkan server saat online/sync (menghindari bentrok).
- Idempotency key: device\_id + local\_txn\_uuid.

## 4. Rancangan Database (MySQL)

Bagian ini menjabarkan rancangan tabel inti, kolom penting, indeks, dan aturan konsistensi untuk kebutuhan fitur yang diminta.

### 4.1 Ringkasan Entitas

- User & Settings: users, app\_settings.
- Catalog: products.
- Sales & Payments: sales, sale\_items, payments.
- Inventory: stock\_items (cache), stock\_movements (ledger).
- Approval & Audit: approvals, audit\_logs.
- Receipt: receipt\_templates, receipt\_print\_logs (opsional).
- Smart inventory: inventory\_recommendations.
- Offline sync: sync\_batches, sync\_idempotency\_keys.

#### users

Menyimpan akun aplikasi untuk role Kasir dan Supervisor. Supervisor memiliki PIN untuk approval (hash).

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
name	VARCHAR	Nama tampilan
username	VARCHAR	UNIQUE, untuk login
password_hash	VARCHAR	Hash password
role	ENUM	CASHIER   SUPERVISOR
supervisor_pin_hash	VARCHAR	Nullable; hanya supervisor
is_active	TINYINT	1 aktif, 0 nonaktif
created_at	DATETIME	-
updated_at	DATETIME	-

Indeks/Constraint penting:

- UNIQUE(username)

#### app\_settings

Key-value settings untuk konfigurasi operasional POS.

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
key	VARCHAR	UNIQUE; contoh: discount_limit_percent
value	TEXT/JSON	Nilai setting

updated_at	DATETIME	-
------------	----------	---

Indeks/Constraint penting:

- UNIQUE(key)

## products

Master produk. Untuk tahap awal: satu produk satu barcode/sku.

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
name	VARCHAR	Nama produk
sku	VARCHAR	UNIQUE, nullable
barcode	VARCHAR	UNIQUE, nullable
price	DECIMAL(12,2)	Harga jual
cost	DECIMAL(12,2)	Harga modal (opsional disembunyikan dari kasir)
uom	VARCHAR	Default pcs
is_active	TINYINT	Aktif/nonaktif
created_at	DATETIME	-
updated_at	DATETIME	-

Indeks/Constraint penting:

- UNIQUE(barcode)
- UNIQUE(sku)
- INDEX(name)

## sales

Header transaksi. Mendukung offline dengan local\_txn\_uuid; invoice final oleh server.

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
server_invoice_no	VARCHAR	UNIQUE, nullable; diisi server
local_txn_uuid	CHAR(36)	UNIQUE; dibuat di client
status	ENUM	DRAFT   PENDING_PAYMENT   PAID   VOID   SYNC_FAILED
cashier_id	BIGINT	FK users
subtotal	DECIMAL(12,2)	-
discount_total	DECIMAL(12,2)	-
tax_total	DECIMAL(12,2)	-
grand_total	DECIMAL(12,2)	-
paid_total	DECIMAL(12,2)	-
change_total	DECIMAL(12,2)	-
occurred_at	DATETIME	Waktu transaksi di device
synced_at	DATETIME	Nullable; waktu berhasil sync
created_at	DATETIME	-
updated_at	DATETIME	-

Indeks/Constraint penting:

- UNIQUE(local\_txn\_uuid)
- UNIQUE(server\_invoice\_no)

- INDEX(status)
- INDEX(occurred\_at)

### sale\_items

Item transaksi. Menyimpan snapshot nama produk untuk menjaga histori.

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
sale_id	BIGINT	FK sales
product_id	BIGINT	FK products
product_name_snapshot	VARCHAR	Nama saat transaksi
unit_price	DECIMAL(12,2)	Harga per unit
qty	DECIMAL(12,3)	Mendukung berat
discount_amount	DECIMAL(12,2)	Diskon nominal per line
line_total	DECIMAL(12,2)	Total line setelah diskon
created_at	DATETIME	-

Indeks/Constraint penting:

- INDEX(sale\_id)
- INDEX(product\_id)

### payments

Pencatatan pembayaran. E-Wallet tahap awal hanya UI (status RECORDED).

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
sale_id	BIGINT	FK sales
method	ENUM	CASH   EWALLET
amount	DECIMAL(12,2)	Nominal
reference	VARCHAR	Nullable; catatan/ref ewallet
status	ENUM	RECORDED
created_at	DATETIME	-

Indeks/Constraint penting:

- INDEX(sale\_id)

### stock\_items

Cache on\_hand per produk untuk performa. Sumber kebenaran tetap stock\_movements.

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
product_id	BIGINT	FK products, UNIQUE
on_hand	DECIMAL(12,3)	Stok tersedia
updated_at	DATETIME	-

Indeks/Constraint penting:

- UNIQUE(product\_id)

### stock\_movements

Ledger perubahan stok. qty\_delta negatif untuk keluar, positif untuk masuk.

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
product_id	BIGINT	FK products
type	ENUM	SALE_OUT   RETURN_IN   ADJUSTMENT   SYNC_CORRECTION
qty_delta	DECIMAL(12,3)	Delta stok
ref_type	VARCHAR	sale   adjustment   sync
ref_id	VARCHAR	sale_id atau local_txn_uuid
note	VARCHAR	Nullable
created_by	BIGINT	FK users
created_at	DATETIME	-

Indeks/Constraint penting:

- INDEX(product\_id, created\_at)
- INDEX(ref\_type, ref\_id)

### approvals

Mencatat approval supervisor untuk aksi sensitif (diskon override, price override, void, dll).

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
action	ENUM	DISCOUNT_OVERRIDE   PRICE_OVERRIDE   VOID   REFUND
sale_id	BIGINT	Nullable; FK sales
requested_by	BIGINT	FK users (kasir)
approved_by	BIGINT	FK users (supervisor)
reason	VARCHAR	Alasan wajib
payload_json	JSON	Detail old/new, item, percent, dll
occurred_at	DATETIME	Waktu kejadian (device)
created_at	DATETIME	-

Indeks/Constraint penting:

- INDEX(sale\_id)
- INDEX(action, occurred\_at)

### audit\_logs

Event log untuk aksi penting (stock adjustment, login, void, perubahan template, dll).

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
actor_id	BIGINT	FK users
event	VARCHAR	Nama event
entity_type	VARCHAR	Nullable
entity_id	VARCHAR	Nullable
meta_json	JSON	Nullable
occurred_at	DATETIME	-

created_at	DATETIME	-
------------	----------	---

Indeks/Constraint penting:

- INDEX(actor\_id, occurred\_at)
- INDEX(event)

### receipt\_templates

Template struk yang dikelola supervisor. Template version disimpan untuk konsistensi reprint.

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
name	VARCHAR	Nama template
version	INT	Nomor versi
is_active	TINYINT	1 aktif
template_json	JSON	Struktur header/body/footer
created_by	BIGINT	FK users
created_at	DATETIME	-

Indeks/Constraint penting:

- INDEX(is\_active)

### receipt\_print\_logs (opsional)

Log pencetakan/reprint untuk audit operasional.

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
sale_id	BIGINT	FK sales
template_id	BIGINT	FK receipt_templates
printed_by	BIGINT	FK users
printed_at	DATETIME	-

Indeks/Constraint penting:

- INDEX(sale\_id, printed\_at)

### inventory\_recommendations

Hasil perhitungan smart inventory (reorder recommendation).

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
product_id	BIGINT	FK products
avg_daily_sales_7d	DECIMAL(12,3)	-
avg_daily_sales_30d	DECIMAL(12,3)	-
lead_time_days	INT	Default 3
safety_stock	DECIMAL(12,3)	Default 0
reorder_point	DECIMAL(12,3)	-
suggested_reorder_qty	DECIMAL(12,3)	-
computed_at	DATETIME	-

Indeks/Constraint penting:



- INDEX(product\_id, computed\_at)

### sync\_batches

Mencatat batch sync dari device (untuk tracing dan debugging).

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
device_id	VARCHAR	ID unik device/browser
batch_uuid	CHAR(36)	UNIQUE
pushed_at	DATETIME	-
status	ENUM	RECEIVED   PROCESSED   FAILED
error_message	TEXT	Nullable

Indeks/Constraint penting:

- UNIQUE(batch\_uuid)
- INDEX(device\_id, pushed\_at)

### sync\_idempotency\_keys

Menjamin idempotency pemrosesan transaksi saat sync (anti-duplikasi).

Kolom	Tipe	Constraint/Catatan
id	BIGINT	PK
key	VARCHAR	UNIQUE; device_id:local_txn_uuid
ref_type	VARCHAR	sale
ref_id	BIGINT	sale_id
created_at	DATETIME	-

Indeks/Constraint penting:

- UNIQUE(key)

## 4.2 Aturan Konsistensi (Wajib)

- Jika sale status berubah menjadi PAID, maka harus ada minimal 1 payment dan stock\_movements tipe SALE\_OUT untuk setiap sale\_item.
- stock\_items.on\_hand adalah cache; setiap update harus berasal dari penjumlahan ledger (atau dari delta yang tervalidasi).
- Approval harus memiliki: approved\_by (supervisor), reason, dan payload\_json yang memadai untuk audit.
- Transaksi offline selalu memakai local\_txn\_uuid; server membuat server\_invoice\_no saat sync.

## 5. Alur Program (Program Flow)

Bagian ini menjelaskan alur utama dari perspektif aplikasi POS (PWA) dan backend.

### 5.1 Initial Data Load (Online)

- Kasir login (online).
- POS melakukan pull: products, app\_settings, stock\_items, receipt template aktif.

- Data disimpan ke IndexedDB untuk kebutuhan offline.

## 5.2 Login Offline

- Offline login hanya diizinkan jika device pernah login online sebelumnya (trust-on-first-use).
- Supervisor approval offline: verifikasi PIN menggunakan hash yang tersimpan lokal.
- Jika device baru dan belum pernah login online, offline login ditolak.

## 5.3 Flow Transaksi Penjualan

### A. Buat Cart

- Scan barcode atau cari produk.
- Tambah item ke cart; ubah qty.
- Hitung subtotal dan total berjalan.

### B. Diskon dan Approval

Aturan: diskon  $\leq$  discount\_limit\_percent bisa langsung; diskon di atas limit membutuhkan approval supervisor (PIN + reason).

- Kasir memasukkan diskon di UI.
- Jika melewati limit: tampil modal approval.
- Supervisor input PIN + reason; sistem membuat record approvals (lokal jika offline, server saat sync).
- Diskon diterapkan, total dihitung ulang.

### C. Checkout Cash

- Kasir memilih CASH, input uang diterima.
- Sistem menghitung kembalian.
- Sale di-set PAID dan disimpan (lokal jika offline).
- Sistem membuat payments dan stock\_movements SALE\_OUT.
- Receipt di-render dari template aktif lalu dicetak.

### D. Checkout E-Wallet (UI only)

- Kasir memilih EWALLET.
- UI menampilkan instruksi/QR placeholder dan input reference (opsional).
- Kasir menekan konfirmasi; payment dicatat sebagai RECORDED.
- Sale di-set PAID; stok keluar dicatat sama seperti cash.

## 5.4 Receipt Rendering, Print, dan Reprint

- Template aktif diambil dari receipt\_templates (atau cache IndexedDB saat offline).
- Placeholder di-resolve menggunakan data sale dan sale\_items.
- Reprint mengambil template\_version yang tersimpan pada transaksi (atau dicatat via receipt\_print\_logs bila diaktifkan).

## 5.5 Offline Storage dan Sync

- Saat offline, transaksi disimpan ke IndexedDB: sales\_local, payments\_local, approvals\_local, stock\_movements\_local.
- POS menambahkan transaksi ke sync\_queue.
- Saat online, POS melakukan push batch ke /api/sync/push.

- Server memproses dengan idempotency key dan mengembalikan mapping local\_txn\_uuid -> server invoice + sale\_id.
- POS menandai transaksi lokal sebagai synced dan menyimpan server\_invoice\_no.

Aturan konflik stok (minimal):

- Jika allow\_negative\_stock = false: transaksi yang membuat stok negatif ditandai SYNC\_FAILED atau memerlukan tindakan supervisor.
- Jika allow\_negative\_stock = true: transaksi tetap diproses; stok dapat menjadi negatif, dan supervisor menerima alert.

## 5.6 Stock Adjustment (Supervisor)

- Supervisor memilih produk, memasukkan qty delta (positif/negatif) dan alasan.
- Sistem membuat stock\_movements tipe ADJUSTMENT dan audit log.
- Cache stock\_items.on\_hand diperbarui.

## 6. Desain API (Minimal)

Endpoint berikut dirancang untuk memenuhi kebutuhan offline sync, catalog pull, dan pengelolaan template receipt.

### 6.1 Auth

- POST /api/auth/login
- POST /api/auth/logout
- GET /api/me

### 6.2 Catalog & Settings

- GET /api/catalog/products
- GET /api/catalog/settings
- GET /api/stock (opsional)

### 6.3 Sync

- POST /api/sync/push
- GET /api/sync/pull?since=2026-01-20T00:00:00

Contoh payload sync push (ringkas):

```
{
  "device_id": "POS-DEVICE-01",
  "batch_uuid": "1d2b3c4d-...",
  "sales": [
    {
      "local_txn_uuid": "...",
      "occurred_at": "2026-01-20T10:15:12",
      "cashier_username": "kasir1",
      "items":
[{"barcode":"899...", "qty":2, "unit_price":12000, "discount_amount":0}],
      "payments": [{"method":"CASH", "amount":24000}],
      "approvals": []
    }
  ]
}
```

## 6.4 Receipt Templates

- GET /api/receipts/active-template
- POST /api/receipts/templates (Supervisor)
- PUT /api/receipts/templates/{id}/activate (Supervisor)

## 6.5 Smart Inventory

- GET /api/inventory/recommendations (Supervisor)

## 7. Smart Inventory: Perhitungan dan Scheduler

Smart inventory menghitung reorder point dan suggested reorder qty berdasarkan penjualan historis dan parameter lead time serta safety stock.

Rumus:

```
avg_daily_sales_7d = total_qty_terjual_7_hari / 7
reorder_point = (avg_daily_sales_7d * lead_time_days) + safety_stock
suggested_reorder_qty = max(0, reorder_point - on_hand)
```

Scheduler: jalankan job harian (mis. 00:10) untuk menghitung semua produk aktif dan menyimpan hasil ke inventory\_recommendations.

## 8. Keamanan, Audit, dan Ketahanan Offline

- Supervisor PIN disimpan sebagai hash (bukan plaintext).
- Approval event wajib menyimpan reason dan payload detail (old/new).
- Idempotency key mencegah transaksi dubel saat sync retry.
- Jika offline login diaktifkan, batasi hanya device yang pernah login online (trust-on-first-use) dan simpan token terenkripsi bila memungkinkan.
- Audit log untuk stock adjustment dan void.

## 9. Strategi Testing (Minimal)

- Unit test: perhitungan total sale, diskon, dan rumus smart inventory.
- Integration test: checkout menghasilkan sales, payments, stock\_movements konsisten dalam satu transaksi DB.
- Offline test: transaksi dibuat saat offline, masuk queue, lalu tersinkron saat online tanpa duplikasi.
- Approval test: diskon > limit memerlukan PIN supervisor dan tercatat di approvals.
- Receipt test: placeholder ter-render benar, reprint konsisten dengan template\_version.

## 10. Milestone Implementasi

- M1 - Setup proyek, auth dua role, settings dasar (discount limit, template aktif).
- M2 - Katalog produk (barcode search) + stok cache + ledger dasar.
- M3 - POS cart + checkout CASH + EWALLET (UI only) + receipt print.
- M4 - Offline-first PWA: service worker, IndexedDB, sync push/pull + idempotency.
- M5 - Approval supervisor (PIN) untuk diskon override/price override + audit log.
- M6 - Smart inventory job + halaman rekomendasi reorder (supervisor).
- M7 - Hardening: conflict handling, SYNC\_FAILED flow, reprint invoice final setelah sync.

## Lampiran A. Contoh Struktur template\_json Receipt

```
{
  "paper": "80mm",
  "header": [
    {"type": "text", "value": "{store_name}", "align": "center", "bold": true},
    {"type": "text", "value": "Invoice: {invoice_no}", "align": "left"},
    {"type": "text", "value": "{date} Cashier: {cashier_name}", "align": "left"}
  ],
  "body": [
    {"type": "items_table", "columns": ["name", "qty", "price", "total"]},
    {"type": "divider"},
    {"type": "kv", "key": "Subtotal", "value": "{subtotal}"},
    {"type": "kv", "key": "Discount", "value": "{discount_total}"},
    {"type": "kv", "key": "Grand Total", "value": "{grand_total}", "bold": true}
  ],
  "footer": [
    {"type": "kv", "key": "Payment", "value": "{payment_method}"},
    {"type": "kv", "key": "Paid", "value": "{paid_total}"},
    {"type": "kv", "key": "Change", "value": "{change_total}"},
    {"type": "text", "value": "Thank you", "align": "center"}
  ]
}
```