CredBar  
  
[**Cara Melakukan Pembayaran QRIS**](https://iuran.org/docs/pembayaran#cara-melakukan-pembayaran-qris)

* Pilih invoice yang ingin dibayar
* Klik tombol "Bayar Sekarang"
* Sistem akan menampilkan kode QR yang dapat dipindai
* Buka aplikasi e-wallet atau mobile banking Anda (seperti GoPay, OVO, DANA, LinkAja, atau aplikasi mobile banking)
* Pindai kode QR yang ditampilkan
* Konfirmasi jumlah pembayaran di aplikasi Anda
* Selesaikan pembayaran sesuai instruksi di aplikasi
* Setelah pembayaran berhasil, sistem akan otomatis memperbarui status pembayaran Anda
* Anda akan menerima notifikasi dan bukti pembayaran melalui email
* Nominal saldo akan masuk ke Saldo Aktif

<https://iuran.org/>



Itu logo kita bagi 2 elemen jadi elemen serpti c itu kita buat cred atau C naya nanti dan garis | itu jadi B atau Bar unutk pada animasinya  
  
  
  
**📅 Timeline Development CredBar (2 Bulan)**

**🟢 Bulan 1 — Fondasi & Core System**

**Minggu 1 — Setup Proyek**

✅ **Tugas:**

* Buat repo credbar-backend, credbar-web, credbar-mobile.
* Setup backend Go (Gin + GORM + Postgres driver).
* Setup docker-compose.yml untuk Postgres + pgAdmin.
* Definisikan model GORM (User, Iuran, Debt, Payment, Transaction).
* Buat koneksi DB + migrasi otomatis.
* Endpoint awal /ping.

🎯 **Output:**

* Backend berjalan, terkoneksi ke DB.
* Bisa migrate tabel otomatis.
* Semua project sudah punya struktur folder rapi.

**Minggu 2 — Authentication & User Management**

✅ **Tugas:**

* Backend: endpoint register, login, GET /me.
* Password hashing pakai bcrypt.
* JWT auth middleware + role-based (admin, user).
* Web: halaman login & register (Next.js + Tailwind).
* Mobile: halaman login (React Native + Expo).
* Simpan token (web: cookie, mobile: AsyncStorage).

🎯 **Output:**

* User bisa daftar & login di web & mobile.
* JWT dipakai untuk akses API.
* Role admin & user dibedakan.

**Minggu 3 — Core Fitur Iuran**

✅ **Tugas:**

* Backend: endpoint CRUD Iuran (buat, list, update, delete).
* Admin bisa buat iuran untuk semua user.
* Web: dashboard admin untuk buat iuran & lihat daftar iuran user.
* Mobile: user bisa lihat daftar iuran + status (paid/unpaid).

🎯 **Output:**

* Admin bisa buat iuran rutin.
* User bisa melihat iuran miliknya.
* Status iuran tersimpan di DB.

**Minggu 4 — Core Fitur Debt**

✅ **Tugas:**

* Backend: endpoint CRUD Debt (buat hutang, list, update status).
* Web: admin bisa input hutang user & lihat daftar hutang.
* Mobile: user bisa lihat hutang miliknya.
* Tambah validasi status (paid/unpaid).

🎯 **Output:**

* Hutang sudah bisa dikelola penuh.
* Web & mobile bisa menampilkan data hutang.
* DB menyimpan data hutang per user.

**🔵 Bulan 2 — Payment, Transaction & Deployment**

**Minggu 5 — Payment System**

✅ **Tugas:**

* Backend: endpoint POST /payments.
* Relasi pembayaran ke iuran/debt.
* Simpan bukti pembayaran (string/file path).
* Update status iuran/hutang setelah pembayaran diverifikasi admin.
* Web: form verifikasi pembayaran admin.
* Mobile: user bisa submit pembayaran.

🎯 **Output:**

* User bisa input pembayaran.
* Admin bisa verifikasi & ubah status iuran/hutang.
* Data pembayaran masuk ke DB.

**Minggu 6 — Transaction Logging**

✅ **Tugas:**

* Backend: buat tabel & API Transaction (log semua aktivitas finansial).
* Tambahkan auto-log setiap ada pembayaran, iuran, atau hutang baru.
* Web: admin bisa lihat daftar transaksi user.
* Mobile: user bisa lihat riwayat transaksi.

🎯 **Output:**

* Semua transaksi tercatat otomatis.
* Web & mobile bisa menampilkan riwayat transaksi.

**Minggu 7 — Dockerization & Testing**

✅ **Tugas:**

* Buat Dockerfile untuk backend, frontend web.
* Buat docker-compose untuk dev (backend + db + web).
* Testing end-to-end: login → buat iuran → bayar → transaksi tercatat.
* Tambahkan seed data untuk testing.
* Dokumentasi API (Swagger/Postman collection).

🎯 **Output:**

* Semua service bisa dijalankan via Docker.
* Testing end-to-end berhasil.
* Dokumentasi API tersedia.

**Minggu 8 — Deployment & Polishing**

✅ **Tugas:**

* Deploy backend + Postgres ke VPS/Cloud (misal DigitalOcean/Render).
* Deploy web ke Vercel/Netlify.
* Publish mobile app via Expo (atau build APK untuk testing).
* Tambahkan fitur tambahan: export CSV transaksi, reminder iuran (email/notifikasi).
* Finalisasi dokumentasi (README + guide install).
* Buat presentasi portfolio (slide/demo video).

🎯 **Output:**

* Aplikasi full (web, mobile, backend) live online.
* Bisa dipresentasikan sebagai portfolio.
* Dokumentasi lengkap.

**⏳ Ringkasan Timeline**

* **Bulan 1:** Setup, Auth, Iuran, Debt. (fondasi sistem selesai)
* **Bulan 2:** Payment, Transaction, Docker, Deployment. (aplikasi production-ready)

**Branch Utama**

1. **main**
   * Branch stabil → berisi release versi final yang sudah siap dipakai.
   * Hanya boleh merge dari release atau hotfix.
2. **develop**
   * Branch utama development.
   * Semua fitur baru akan masuk ke sini dulu sebelum akhirnya di-*release*.

**🌿 Branch Pendukung**

1. **feature/\***
   * Untuk mengembangkan fitur spesifik.
   * Contoh:
     + feature/authentication → fitur login/register
     + feature/habit-tracking → fitur pencatatan kebiasaan
     + feature/calendar → integrasi jadwal
     + feature/nutrition-calculator → perhitungan kalori
2. **release/\***
   * Dibuat saat mau rilis versi stabil.
   * Contoh: release/v1.0.0
   * Setelah stabil → merge ke main dan develop.
3. **hotfix/\***
   * Untuk memperbaiki bug darurat di main.
   * Contoh: hotfix/fix-login-error.