Oke siap, Boss Dicky! Dicatat dan langsung diamankan! Login/Sign Up itu pintu gerbangnya, wajib banget ada. Dan pembayaran? Nah, itu dia inti dari *passive income* yang Boss mau.

Ini dia update dokumennya, Kiro pasti langsung ngerti!

**PROYEK : PROJEKSYNC**

VERSI DOKUMEN: 1.1 (Update Penting dari Boss Dicky)

TANGGAL: 17 Juli 2025

AUTHOR: Gemini (AI Assistant Boss Dicky)

### 1. RINGKASAN PROYEK (THE "WHAT")

Boss Dicky mau Kiro bikin platform SaaS dashboard terintegrasi buat freelancer. Tujuannya biar freelancer bisa ngatur *schedule*, *milestone*, *timesheet/worksheet*, dan *invoice* dalam satu tempat aja. Intinya, biar hidup freelancer makin gampang, efektif, dan nggak pusing sama tetek bengek administrasi. Ini adalah jalan menuju *passive income* global, jadi Kiro harus kasih yang terbaik!

### 2. VISI & TARGET PENGGUNA (THE "WHY" & "WHO")

* **Visi:** Menjadi dashboard manajemen proyek dan administrasi yang paling intuitif dan efisien untuk freelancer di seluruh dunia, membantu mereka fokus pada kerja kreatif daripada urusan birokrasi.
* **Target Pengguna:** Individual Freelancer, konsultan independen, agensi mikro (1-2 orang) yang butuh alat all-in-one simpel tapi powerfull. Mereka cari efisiensi dan automasi.

### 3. FITUR MINIMAL VIABLE PRODUCT (MVP) - YANG HARUS ADA DI V1.0

Kiro, fokus ke **core functionality** ini dulu ya biar cepet launch dan bisa di-test drive sama Boss Dicky.

#### 3.1. Modul Proyek & Klien:

* **Buat/Edit/Hapus Proyek:** Nambahin proyek baru, edit detailnya (nama, deskripsi, klien, tanggal mulai/deadline).
* **Manajemen Klien:** Data dasar klien (nama, email). Tiap proyek harus terhubung ke satu klien.

#### 3.2. Modul Milestone:

* **Buat/Edit/Hapus Milestone:** Untuk setiap proyek, bisa bikin daftar milestone dengan nama dan deadline.
* **Update Status Milestone:** Bisa ditandai sebagai 'To Do', 'In Progress', 'Done'.

#### 3.3. Modul Timesheet/Worksheet:

* **Pencatatan Waktu Manual:** User bisa input jam kerja (start time, end time) atau durasi untuk tugas tertentu di bawah proyek/milestone. Plus, bisa nulis deskripsi singkat.
* **Ringkasan Waktu:** Dashboard simpel yang nunjukkin total jam kerja per hari/minggu/proyek.

#### 3.4. Modul Invoice:

* **Buat Invoice Manual:** User bisa bikin invoice baru.
  + Detail invoice: nomor invoice, tanggal, jatuh tempo, info klien, detail layanan/item (deskripsi, kuantitas, harga satuan).
  + Subtotal, diskon (opsional), total.
* **Lihat Daftar Invoice:** Dashboard yang nunjukkin semua invoice yang udah dibuat.
* **Status Invoice:** Tandai invoice sebagai 'Draft', 'Sent', 'Paid', 'Overdue'.

#### 3.5. Dashboard Utama (Homepage):

* Tampilan ringkas yang nunjukkin proyek aktif, milestone yang mendekati deadline, dan ringkasan invoice terbaru. Ini yang paling pertama diliat user pas login.

### 4. FITUR KRUSIAL UNTUK AKSES & MONETISASI (YANG BARU DITAMBAHKAN BOSS DICKY!)

#### 4.1. Autentikasi Pengguna:

* **Login & Sign Up:** Harus ada halaman dan tombol fungsional untuk user baru mendaftar (Sign Up) dan user lama masuk (Login). Ini jadi gerbang utama aplikasi.

#### 4.2. Sistem Pembayaran (Phase Lanjut/Setelah MVP Core Fungsional):

* **Konsep:** Setelah MVP core berfungsi, Kiro harus siap untuk mengimplementasikan sistem pembayaran berlangganan (Subscription) agar Boss Dicky bisa generate *passive income*.
* **Rencana:** Ini akan menjadi prioritas *setelah* fitur-fitur manajemen proyek inti berjalan lancar.
* **Opsi Payment Gateway (Kiro bisa riset):**
  + Internasional: Stripe, Paddle.
  + Lokal (Indonesia): Midtrans, Xendit (untuk fleksibilitas pembayaran lokal).
* **Model:** Subscription bulanan/tahunan (misal: Freemiun atau Tiered Pricing) akan kita diskusikan lebih lanjut saat siap implementasi.

### 5. TEKNOLOGI (THE "HOW")

Kiro udah jago di sini:

* **Backend:** Node.js / Express.js (atau sesuai preferensi Kiro yang paling efisien)
* **Database:** Supabase (ini udah nyaman banget kan, Kiro!)
* **Frontend:** React / Next.js (buat dashboard yang *responsive* dan *modern*)
* **Deployment:** Vercel / Netlify (buat frontend), Render / Fly.io (buat backend, kalau terpisah dari Supabase Functions)
* **Auth:** Supabase Auth (simpel, cepet, aman, dan udah termasuk UI Login/Sign Up dasar yang bisa di-*customize*)

### 6. ALUR KERJA AWAL UNTUK KIRO (THE "NEXT STEPS")

Oke, Kiro, ini *action item* pertama kamu biar langsung ngegas:

1. **Setup Proyek Awal:** Siapkan repository git, struktur folder dasar untuk backend dan frontend.
2. **Database Schema Design:** Buat skema tabel dasar di Supabase untuk:
   * users (integrasi dengan Supabase Auth)
   * clients (id, name, email, user\_id)
   * projects (id, client\_id, name, description, start\_date, deadline, status, user\_id)
   * milestones (id, project\_id, name, deadline, status)
   * timesheets (id, project\_id, milestone\_id, description, start\_time, end\_time, duration, user\_id)
   * invoices (id, project\_id, client\_id, invoice\_number, issue\_date, due\_date, total\_amount, status, user\_id)
   * invoice\_items (id, invoice\_id, description, quantity, unit\_price, total\_price)
3. **Basic Authentication & UI:** Implementasi registrasi dan login user pakai Supabase Auth. **Sediakan tombol Login/Sign Up yang jelas di UI.**
4. **API Endpoint Pertama:** Buat API dasar untuk users, clients, dan projects (CRUD - Create, Read, Update, Delete).
5. **Frontend Dashboard Mockup (Basic):** Bikin tampilan dashboard utama yang kosong tapi udah bisa login, biar Boss Dicky bisa liat wujudnya.

Boss Dicky, Kiro akan langsung kerjain ini. Kita bakal update progress-nya secara berkala ya! Ini bakal jadi SaaS yang gokil, Boss! Yuk, sikat!