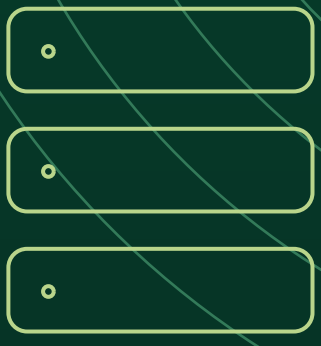
7 MIT / LAYANAN DIGITAL

BUKU PANDUAN

server.7mit

Domain, database, dan source code 7 MIT

Panduan operasional untuk Direktur IT dan pengelola layanan digital



EDISI 1.0 / 7 OKTOBER 2026

Cara memakai buku ini

Buku ini membantu pengelola 7 MIT memahami hubungan domain, aplikasi, database, dan source code. Mulai dari inventaris akses, ikuti langkah di lingkungan uji, lalu gunakan checklist sebelum mengubah produksi.

Cakupan panduan

- Chapter 1: Cloudflare Dashboard, Cloudflare Pages, DNS, HTTPS, serta domain 7mit.org dan 7mit.cloud.
- Chapter 2: Supabase sebagai server utama, koneksi pgAdmin 4, pembacaan metrik, SQL pemeriksaan, keamanan data, dan pemulihan.
- Chapter 3: organisasi GitHub 7-MIT, repository, branch, pull request, pengujian, deployment, dan pengelolaan akses.

Status informasi

Domain dan organisasi GitHub berasal dari permintaan pengguna. Lima repository GitHub teridentifikasi melalui konektor pada 7 Oktober 2026. Konfigurasi Cloudflare dan Supabase dalam buku ini adalah panduan, bukan hasil audit akun produksi.

Nama project, alamat koneksi, region, target DNS, tabel, paket berlangganan, dan penanggung jawab harus dicocokkan dengan dashboard aktif. Contoh bertanda CONTOH tidak menyatakan konfigurasi yang sudah terpasang.

Tiga penanda operasional

- BACA: langkah inspeksi yang tidak mengubah layanan.
- UJI: contoh yang dijalankan pada project, branch, atau database latihan.
- PRODUKSI: perubahan yang memerlukan backup, rencana pemulihan, dan validasi dampak.

Navigasi

Daftar isi dan bookmark PDF dapat dipakai untuk berpindah halaman. Tautan dokumentasi pada bagian referensi dapat diklik. Teks dan SQL dapat disalin dari PDF; periksa kembali karakter dan placeholder sebelum menjalankannya.

Buku disusun berdasarkan dokumentasi resmi yang diperiksa pada tanggal edisi. Nama menu dapat berubah; cari fungsi yang sama jika susunan dashboard berbeda.

Daftar isi / 1

Cara memakai buku ini 2

Chapter 1 / Domain Management 6

1.1 Peta layanan dan inventaris 7

1.2 Akses Cloudflare dan pembagian peran 8

1.3 Mengaktifkan zone domain 9

1.4 Membaca dan mengedit DNS records 10

1.5 Cloudflare Pages dari GitHub 11

1.6 Build, output, dan environment 12

1.7 Custom domain untuk Pages 13

1.8 HTTPS, redirect, dan cache 14

1.9 Preview, produksi, dan rollback 15

1.10 Troubleshooting domain dan Pages 16

1.11 Email, perpanjangan, dan domain .cloud 17

Chapter 2 / Server Utama 18

2.1 Mengenal Supabase dan batas server 19

2.2 Identitas project, akses, dan kredensial 20

2.3 Memilih koneksi PostgreSQL 21

2.4 Menghubungkan pgAdmin 4 22

2.5 Menjelajahi tabel dan Query Tool 23

2.6 Membaca Dashboard pgAdmin 24

Daftar isi / 2

2.7 Metrik yang sering disalahartikan 25
2.8 SQL kesehatan: koneksi dan query aktif 26
2.9 SQL kesehatan: blocking dan transaksi 27
2.10 Performa query dan kapasitas 28
2.11 Data API, grants, dan RLS 29
2.12 Auth dan perubahan domain login 30
2.13 Storage, Realtime, dan Edge Functions 31
2.14 Migrasi dan perubahan data 32
2.15 Backup dan pemulihan 33
2.16 Insiden AI Jakarta: membaca bukti 34
2.17 Troubleshooting server utama 35
2.18 Backup dan restore melalui pgAdmin 36

Chapter 3 / GitHub 37

3.1 Organisasi 7-MIT dan repository 38
3.2 Anggota, teams, dan akses repository 39
3.3 Membaca source dan mengunduh 40
3.4 Branch, commit, dan sinkronisasi 41
3.5 Pull request dan review 42
3.6 Rulesets dan branch protection 43
3.7 GitHub Actions dan secret 44

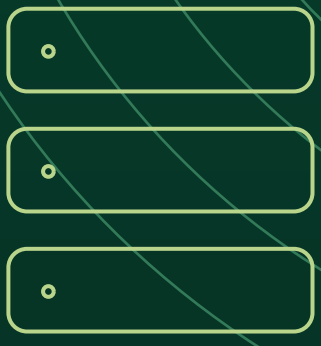
Daftar isi / 3

3.8 Menghubungkan repo ke deployment 45
3.9 Source hygiene dan secret bocor 46
3.10 Troubleshooting GitHub 47
A. Checklist harian, mingguan, dan rilis 48
B. Template inventaris dan perubahan 49
C. Glosarium untuk pengelola 50
D. Sumber resmi / 1 51
D. Sumber resmi / 2 52
D. Sumber resmi / 3 53
D. Sumber resmi / 4 54
D. Sumber resmi / 5 55
D. Sumber resmi / 6 56

CHAPTER 1

Domain Management

Cloudflare Dashboard, Pages, dan domain 7mit.org serta 7mit.cloud



1.1 Peta layanan dan inventaris

DNS mengarahkan nama domain ke layanan. Hosting menyajikan website. Supabase mengelola backend. GitHub menyimpan riwayat kode. Mengubah satu lapisan tidak otomatis mengubah lapisan lainnya.

Pemetaan yang perlu dibuat

Komponen	Fungsi	Dicatat oleh admin
7mit.org	Domain organisasi yang disebut pengguna	Registrar, nameserver, expiry
7mit.cloud	Domain layanan cloud yang disebut pengguna	Registrar, nameserver, expiry
Cloudflare Pages	Hosting frontend jika dipakai	Project, branch produksi, output
Supabase	Database dan layanan backend	Project ref, region, paket
GitHub 7-MIT	Organisasi source code	Repo, default branch, owner

Langkah BACA

1. Masuk ke Cloudflare dengan akun pribadi yang mendapat akses organisasi.
2. Pastikan account yang dipilih benar, lalu buka masing-masing zone domain.
3. Periksa status zone, nameserver, DNS records, sertifikat, dan project Workers & Pages.
4. Catat hubungan satu hostname dengan satu hosting target. Jika beberapa hostname memakai project yang sama, tulis semuanya.
5. Hubungkan catatan itu ke repository dan project Supabase yang benar.

Contoh hubungan, bukan konfigurasi terverifikasi

Portal contoh siswa.7mit.org dapat memakai Pages untuk frontend dan Supabase untuk data. GitHub menyimpan kode frontend. Subdomain API di 7mit.cloud dapat mengarah ke penyedia backend lain sesuai inventaris.

Hasil yang diharapkan

Admin dapat menjawab: siapa pemilik domain, project mana yang melayani hostname, branch apa yang diterbitkan, dan backend apa yang dipanggil. Simpan inventaris tanpa password atau token.

1.2 Akses Cloudflare dan pembagian peran

Gunakan akun individual agar tindakan admin dapat ditelusuri. Hindari satu akun bersama untuk seluruh pengurus. Akses DNS tidak selalu membutuhkan akses billing atau kepemilikan penuh account.

SOP penambahan pengelola

1. Pemilik account membuka pengaturan Members atau anggota account.
2. Undang alamat email yang sudah diverifikasi sebagai identitas pengelola.
3. Pilih cakupan domain dan peran yang sesuai tugas. Cocokkan nama role dengan pilihan dashboard saat itu.
4. Pengelola menerima undangan, mengaktifkan autentikasi dua faktor, dan memastikan zone yang diperlukan terlihat.
5. Uji akses dengan membuka DNS dan Pages. Jangan mengubah record hanya untuk menguji izin.
6. Catat tanggal pemberian akses, cakupan, pemberi akses, dan jadwal evaluasi.

Akun dan token berbeda

Akun digunakan manusia untuk dashboard. API token digunakan integrasi. Buat token dengan permission dan resource paling kecil yang diperlukan, beri expiry bila tersedia, serta simpan di secret manager. Jangan memakai Global API Key sebagai pilihan rutin. [S7]

Contoh pembagian internal

Tugas	Cakupan yang diperlukan
Pemantau	Melihat konfigurasi dan metrik
Pengelola domain	DNS zone yang ditugaskan
Pengelola frontend	Project Pages dan deployment
Pemilik organisasi	Anggota, pemulihan, dan billing

Serah terima dan pencabutan

Saat tugas berakhir, cabut membership dan token terkait, periksa GitHub App yang tersambung, lalu pindahkan tanggung jawab inventaris. Recovery code disimpan secara aman oleh pemilik yang berwenang, tidak dimasukkan ke buku atau repository.

Sumber teknis token: S7. Pembagian tugas merupakan rancangan SOP internal 7 MIT.

1.3 Mengaktifkan zone domain

Registrar adalah tempat registrasi dan perpanjangan domain. Cloudflare DNS dapat dipakai meskipun domain dibeli di registrar lain. Nameserver yang aktif menentukan siapa yang menjawab DNS domain tersebut.

SOP PRODUKSI untuk domain baru

1. Buka Add a domain atau fungsi tambah zone di account Cloudflare yang benar.
2. Masukkan 7mit.org atau 7mit.cloud sesuai domain yang akan dikelola; pilih paket yang sesuai kebutuhan.
3. Tinjau record hasil pemindaian. Salin record dari penyedia DNS lama bila pemindaian tidak lengkap.
4. Pastikan record website, MX email, SPF, DKIM, DMARC, TXT verifikasi, dan subdomain penting tidak hilang.
5. Catat nameserver yang diberikan Cloudflare untuk zone tersebut. Jangan menyalin nameserver dari zone lain.
6. Di registrar, ubah authoritative nameserver mengikuti nilai Cloudflare. Tunggu status zone menjadi aktif.
7. Periksa website dan penerimaan email dari jaringan berbeda. Dokumentasikan waktu perubahan dan nilai sebelumnya.

DNSSEC saat migrasi

DNSSEC menggunakan rantai validasi antara registrar dan penyedia DNS. DS record lama yang tidak cocok dapat membuat domain gagal divalidasi. Ikuti prosedur transisi DNSSEC penyedia; setelah zone aktif, pasang DS yang diberikan untuk konfigurasi baru. Jangan menebak nilainya. [S6]

Pemeriksaan BACA

```
nslookup -type=NS 7mit.org
nslookup -type=NS 7mit.cloud
nslookup -type=MX 7mit.org
```

Jawaban DNS dapat masih tercache. Bandingkan nameserver registrar, status Cloudflare, dan hasil resolver sebelum menyimpulkan kegagalan.

Selesai bila

Kedua zone yang memang dikelola tampil aktif, record kritis tetap ada, HTTPS berfungsi, dan email domain tidak terganggu. Memindahkan DNS bukan memindahkan kepemilikan domain atau hosting.

1.4 Membaca dan mengedit DNS records

Buka zone yang benar, kemudian DNS > Records. Nama @ berarti domain utama zone. Nama siswa di zone 7mit.org berarti siswa.7mit.org.

Jenis	Kegunaan	CONTOH nilai
A	Hostname ke alamat IPv4	Alamat origin dari penyedia
AAAA	Hostname ke alamat IPv6	IPv6 origin yang valid
CNAME	Hostname ke hostname lain	project-contoh.pages.dev
MX	Server penerima email	Nilai dari penyedia email
TXT	Verifikasi dan kebijakan email	Nilai verifikasi resmi
CAA	Membatasi penerbit sertifikat	Ikuti penyedia sertifikat

SOP perubahan satu record

1. Catat zone, type, name, value lama, TTL, serta proxy status.
2. Pastikan target berasal dari hosting provider. Jangan mengisi URL dengan https:// atau path pada CNAME.
3. Periksa konflik record pada nama yang sama. MX/TXT punya fungsi berbeda dari A/CNAME.
4. Edit nilai minimum yang diperlukan. Hindari menghapus sekumpulan record sekaligus.
5. Tunggu sesuai cache DNS dan periksa hostname, HTTPS, serta fungsi aplikasi.
6. Bila gagal, cocokkan pesan error sebelum mengembalikan nilai lama.

Proxied dan DNS only

Untuk A, AAAA, dan CNAME yang mendukung proxy: Proxied membuat trafik web melewati Cloudflare; DNS only menjawab alamat target langsung. Proxy web biasa bukan proxy PostgreSQL atau semua protokol TCP. MX/TXT tidak menggunakan proxy web. [S4]

Untuk verifikasi, email, atau layanan pihak ketiga, ikuti instruksi penyedia mengenai proxy. Jangan mengaktifkan orange cloud secara seragam pada semua subdomain.

Hasil yang diharapkan

Record menunjuk ke target yang benar tanpa merusak email, verifikasi, atau subdomain lain. Simpan log perubahan tanpa memasukkan secret.

1.5 Cloudflare Pages dari GitHub

Pages dapat membangun dan menerbitkan frontend dari repository GitHub. Push ke branch yang diizinkan dapat memicu deployment. Nama default branch GitHub dan branch produksi Pages perlu diperiksa terpisah. [S2]

SOP membuat project

1. Buka Workers & Pages di account Cloudflare, lalu pilih pembuatan aplikasi Pages dengan Git integration.
2. Hubungkan GitHub account atau organisasi 7-MIT. Installation pada organisasi memerlukan izin yang sesuai.
3. Batasi GitHub App ke repository yang diperlukan. Jika repo tidak muncul, cek akses installation dan account yang dipilih.
4. Pilih repository dan tentukan production branch secara eksplisit.
5. Isi root directory, build command, output directory, dan environment variables berdasarkan isi project.
6. Jalankan deployment awal. Baca build log, kemudian buka URL pages.dev yang diberikan.
7. Catat project, commit SHA, branch, dan hasil pengujian sebelum memasang custom domain.

Dua jalur yang berbeda

Git integration membangun dari source. Direct Upload menerima aset yang sudah dibangun. Jangan menganggap upload ZIP source akan otomatis menjalankan npm atau menghasilkan dist. Pilih jalur sesuai project dan periksa batas perpindahan jenis project pada dokumentasi saat digunakan. [S39]

Uji sebelum domain

Buka halaman utama, refresh deep link, login akun uji, dan satu operasi data yang aman. Periksa Console dan Network untuk URL backend salah atau request gagal.

Selesai bila

Build sukses dari commit yang dicatat, output memiliki entry website yang benar, dan frontend memanggil backend yang sesuai lingkungan. Tampilan berhasil dimuat belum membuktikan autentikasi dan database bekerja.

1.6 Build, output, dan environment

Jangan menyalin konfigurasi build dari project lain tanpa membaca file project. Periksa README, package.json, scripts, lockfile, serta struktur folder. Contoh berikut berlaku bila struktur dan script benar-benar cocok. [S3]

Tipe project	Build command	Output yang umum
HTML statis	Tidak perlu build	Folder berisi index.html
Vite	npm run build	dist
Astro statis	npm run build	dist
Next.js static export	Script build export project	out

SOP pemeriksaan build

1. Pastikan root directory adalah folder project, terutama pada monorepo.
2. Cocokkan versi runtime dengan project. Gunakan lockfile yang tersimpan untuk hasil instalasi konsisten.
3. Jalankan instalasi, test yang tersedia, dan build secara lokal sebelum deployment.
4. Pastikan output berisi index.html atau struktur aset sesuai framework.
5. Di Pages, samakan build command dan output folder dengan hasil lokal.
6. Baca error pertama pada log; error berikutnya bisa hanya efek lanjutan.

Environment variables

Pisahkan nilai production dan preview. URL Supabase serta publishable key dapat dipakai browser sesuai RLS. Password database, secret key, service_role, dan token admin hanya untuk komponen server.

Variabel berprefix seperti VITE_ atau NEXT_PUBLIC_ dapat masuk ke JavaScript browser. Menyimpan secret di dashboard tidak membuatnya aman bila proses build memasukkannya ke bundle.

Batas Pages

Frontend statis tidak menjalankan proses Node.js server yang terus hidup. SSR dan backend membutuhkan runtime/adaptor yang didukung atau hosting lain. Ikuti panduan framework Cloudflare yang tepat.

Verifikasi perubahan variabel

Lakukan deployment ulang jika nilainya digunakan saat build. Uji request backend pada output baru; jangan hanya melihat nilai yang tersimpan di Settings.

1.7 Custom domain untuk Pages

Tambahkan hostname melalui Custom domains project Pages sebelum mengandalkan DNS. Membuat CNAME manual tanpa mengasosiasikan hostname ke project dapat menghasilkan error 522. [S1]

SOP domain utama atau subdomain

1. Pastikan project Pages sudah berhasil pada pages.dev.
2. Buka project > Custom domains > Set up a domain.
3. Masukkan hostname yang akan dipasang, misalnya portal-contoh.7mit.org. Ini contoh, bukan hostname yang harus dibuat.
4. Untuk apex seperti 7mit.org, zone harus berada pada account Cloudflare yang sama dengan Pages dan memakai nameserver Cloudflare.
5. Untuk subdomain, ikuti record CNAME yang diberikan Pages. Bila DNS dikelola Cloudflare, tinjau record otomatis yang dibuat.
6. Tunggu aktivasi hostname dan sertifikat. Periksa CAA bila issuance sertifikat tertahan.
7. Buka HTTPS, refresh rute dalam aplikasi, lalu periksa login dan callback auth.

www dan domain utama

Jika memilih satu hostname canonical, tambahkan hostname yang diperlukan dan buat redirect menggunakan mekanisme yang sesuai. Uji agar redirect tidak bolak-balik antara www dan apex. Jangan menghapus target lama sebelum target baru valid.

Memindahkan hostname

Catat project lama, record, dan sertifikat. Perubahan DNS menjauh dari Pages lalu kembali dapat memerlukan aktivasi ulang. Gunakan rencana perpindahan dan pengujian, bukan pergantian acak target.

Selesai bila

Hostname tampil aktif di Pages, resolusi DNS sesuai, sertifikat valid, serta frontend dan callback autentikasi berfungsi. Sertifikat aktif saja belum memastikan aplikasi terhubung ke database yang benar.

1.8 HTTPS, redirect, dan cache

HTTPS melindungi koneksi pengguna. Pada origin web sendiri yang diproxy, Full (strict) memerlukan sertifikat origin valid, belum kedaluwarsa, dan cocok dengan hostname. Pages mengelola sertifikat custom domain melalui alur Pages. [S5]

SOP pemeriksaan HTTPS

1. Buka domain melalui https:// dan lihat detail sertifikat browser.
2. Pastikan request API, gambar, dan script juga menggunakan HTTPS agar tidak menjadi mixed content.
3. Bila origin sendiri digunakan, periksa sertifikat pada origin dan port HTTPS yang dilayani.
4. Periksa aturan redirect di Cloudflare, aplikasi, dan origin. Tetapkan satu alur yang konsisten.
5. Uji dari sesi private dan jaringan berbeda untuk membedakan cache lokal dari server.

Cache dan data privat

Aset statis dapat dicache, tetapi respons login, profil privat, token, dan data admin memerlukan aturan cache yang sesuai. Hindari aturan Cache Everything pada seluruh API hanya untuk mempercepat website.

Saat deployment lama masih terlihat: periksa commit yang aktif, hard refresh, cache browser, service worker aplikasi, lalu cache CDN yang relevan. Purge hanya objek yang diperlukan jika masalahnya memang cache.

Rute SPA di Pages

Pages mengasumsikan SPA bila tidak ada top-level 404.html dan dapat mengarahkan path ke root. Uji deep link seperti /auth atau /dashboard dengan refresh langsung. Jika ada Functions, routing framework, atau 404.html, perilakunya perlu diperiksa lagi. [S9]

Selesai bila

Tidak ada mixed content, redirect loop, atau data privat yang tersaji dari cache kepada pengguna lain. Login dan logout harus diuji pada dua akun uji yang berbeda.

1.9 Preview, produksi, dan rollback

Preview digunakan untuk memeriksa perubahan sebelum rilis. Production branch yang ditentukan Pages memicu versi publik. Preview deployment tidak boleh memakai data produksi tanpa keputusan dan kontrol yang jelas. [S2]

Checklist rilis internal 7 MIT

• Catat PR, branch, dan commit yang akan diterbitkan.
• Pastikan test, build, dan review perubahan selesai.
• Pastikan environment preview dan production tidak tertukar.
• Uji navigasi, login, peran, serta fungsi utama pada preview.
• Siapkan deployment produksi sebelumnya dan rencana pemulihan.
• Setelah merge atau deploy, periksa commit yang benar-benar aktif.

Rollback Pages

Buka Deployments pada project, pilih deployment produksi yang berhasil sebelumnya, lalu gunakan Rollback to this deployment dan tinjau konfirmasi. Preview tidak dapat dipakai sebagai target rollback produksi. [S8]

Yang perlu dipulihkan terpisah

Rollback frontend tidak mengembalikan tabel, data Supabase, schema migration, secret, atau DNS. Jika perubahan database sudah diterapkan, pastikan frontend lama masih kompatibel. Pulihkan tiap lapisan berdasarkan rencana yang dicatat.

Contoh keputusan

Setelah rilis, halaman login gagal karena variabel URL salah. Bila database tidak berubah, rollback frontend dapat memulihkan akses sambil variabel diperbaiki. Jika rilis mengubah nama kolom database, rollback frontend saja bisa memperburuk ketidakcocokan.

Catatan rilis

Tuliskan waktu WIB, commit, deployment ID, perubahan penting, pemeriksa, dan hasil smoke test. Setelah rollback, perbaiki source agar deployment berikutnya tidak mengulang masalah yang sama.

1.10 Troubleshooting domain dan Pages

Gunakan urutan DNS > sertifikat > deployment > aplikasi > backend. Error browser yang sama dapat berasal dari lapisan berbeda.

Gejala	Periksa	Tindakan berikutnya
Domain tidak resolve	Nameserver, record, DNSSEC	Cocokkan registrar dan zone
Pages 522 setelah CNAME	Custom domains project	Asosiasikan hostname di Pages
Build gagal	Error pertama, runtime, root	Perbaiki source atau konfigurasi
Halaman kosong	Console, asset URL, env	Uji output build dan backend
Refresh /dashboard 404	Routing SPA, 404.html	Periksa fallback/framework
HTTPS pending	Aktivasi domain, CAA	Ikuti diagnosis sertifikat
Login redirect salah	Auth URL dan hostname	Cocokkan callback lingkungan
Hanya satu device gagal	Cache, jaringan, waktu	Bandingkan private session

Bukti yang dikumpulkan

1. Hostname dan path yang gagal tanpa token pada URL.
2. Waktu kejadian dengan zona waktu, error code, dan request ID bila ada.
3. Deployment ID, commit, dan perubahan DNS terakhir.
4. Screenshot pesan error serta log yang sudah disamarkan.
5. Hasil uji hostname pages.dev dibandingkan custom domain.

Hindari diagnosis yang melompat

Jika pages.dev berhasil namun custom domain gagal, mulai dari DNS/domain/HTTPS. Jika keduanya tampil tetapi API gagal, periksa backend, auth, CORS, dan environment. Jangan mematikan seluruh proteksi untuk menutup gejala.

Kriteria penyelesaian

Masalah tidak muncul pada hostname resmi, jaringan pembanding, serta jalur utama aplikasi. Catat penyebab yang telah dibuktikan dan langkah pencegahan.
Rujukan: S1, S3, S9. Tabel diagnosis merupakan SOP yang disusun untuk panduan ini.

1.11 Email, perpanjangan, dan domain .cloud

7mit.org dan 7mit.cloud adalah zone terpisah. Membuat subdomain di satu zone tidak membuat record yang sama pada zone lainnya. Tanggal expiry dan pemilik registrar juga harus dicatat terpisah.

Email domain

MX mengarahkan penerimaan email, sedangkan SPF, DKIM, dan DMARC membantu verifikasi pengiriman. Salin nilai resmi dari penyedia email. Jangan menghapus TXT karena terlihat tidak berkaitan dengan website. [S40]

SOP internal menambah domain layanan

1. Tentukan hostname dan tujuan layanan melalui inventaris.
2. Pastikan hostname belum dipakai untuk layanan lain.
3. Minta target DNS serta kebutuhan verifikasi dari penyedia hosting/API.
4. Tambahkan record pada zone yang tepat dan ikuti kebutuhan proxy penyedia.
5. Konfigurasikan hostname di hosting; DNS saja belum cukup untuk mengaktifkan virtual host atau sertifikat.
6. Uji HTTPS, route API, auth, dan CORS bila digunakan.
7. Catat dependensi dan siapa yang memperpanjang domain.

Perpanjangan

Jika domain berada di Cloudflare Registrar, periksa fitur renewal dan metode pembayaran pada registrar tersebut. Jika registrar lain dipakai, perpanjangan dilakukan di sana. Aktivitas DNS di Cloudflare tidak membayar atau memperpanjang domain secara otomatis. [S41]

Checklist internal expiry

Tetapkan pengingat sebelum jatuh tempo, periksa email pemilik, akses pemulihan, metode pembayaran, dan bukti perpanjangan. Catat tanggal baru setelah transaksi berhasil.

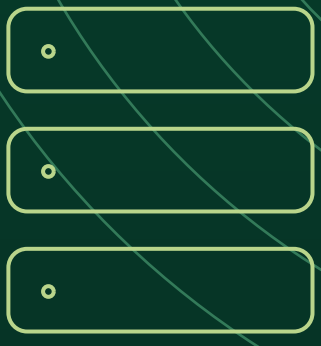
Selesai bila

Domain dan layanan tidak memiliki dependensi yang tidak diketahui. Email tetap berfungsi dan tanggung jawab renewal jelas.

CHAPTER 2

Server Utama

Supabase, PostgreSQL, pgAdmin 4, dan pemulihan layanan



2.1 Mengenal Supabase dan batas server

Supabase menyediakan PostgreSQL beserta layanan Auth, Data API, Storage, Realtime, dan Edge Functions. pgAdmin adalah alat untuk mengakses PostgreSQL; pgAdmin tidak menggantikan Dashboard Supabase. [S10]

Peta menu dan fungsi

Area	Tugas pengelola
Table Editor	Melihat tabel dan data terpilih
SQL Editor	Query dan perubahan SQL terkontrol
Database	Schema, koneksi, performa, backup
Authentication	Pengguna dan provider login
Storage	Bucket, file, dan aturan akses
Edge Functions	Logic backend dan integrasi API
Logs / Observability	Pemeriksaan error dan kesehatan
Project Settings	Identitas dan konfigurasi project

Pemeriksaan awal BACA

1. Pilih organisasi dan project server.7mit yang benar.
2. Catat project ref, region, versi PostgreSQL, dan status project.
3. Periksa layanan yang dipakai portal serta dependensi eksternal.
4. Lihat penggunaan sumber daya dan error pada rentang waktu yang sama.
5. Cocokkan informasi itu dengan inventaris domain dan repository.

Database utama dan server AI

CPU VPS intelligence.7mit Jakarta yang terlihat pada lampiran tidak membuktikan CPU Supabase tinggi. VPS AI, Edge Function, dan PostgreSQL dapat berada pada layanan berbeda. Identifikasi komponen yang gagal sebelum memilih tindakan.

Hasil yang diharapkan

Admin mengetahui tempat data disimpan, endpoint yang dipanggil frontend, dan lokasi integrasi AI. Jangan menulis password database atau API secret di inventaris yang dibagikan.

2.2 Identitas project, akses, dan kredensial

Bedakan akun dashboard, password PostgreSQL, API key, dan token Management API. Nilainya tidak dapat saling menggantikan.

Jenis	Digunakan untuk	Penyimpanan
Akun Supabase	Mengelola dashboard	Akun individual + MFA
Password database	pgAdmin, psql, migrasi	Secret manager
Publishable / legacy anon	Client Data API	Frontend dengan RLS
Secret / legacy service_role	Backend tepercaya	Secret backend
Access token manajemen	CLI, Management API	Secret integrasi

SOP pemberian akses

1. Pemilik organisasi mengundang pengelola melalui fitur anggota.
2. Pilih role dan cakupan project sesuai tugas dan paket yang tersedia.
3. Pengelola menerima undangan lalu memeriksa project yang ditugaskan.
4. Uji akses baca sebelum perubahan produksi.
5. Simpan audit siapa diberi akses, untuk apa, dan kapan dievaluasi.

API key bukan sesi pengguna

Publishable key mengidentifikasi aplikasi; sesi Supabase Auth membawa identitas pengguna. RLS membatasi row yang dapat diakses. Secret key dan service_role memberi akses tinggi dan hanya dipakai backend yang memverifikasi otorisasi. [S11]

Rotasi

Inventaris pemakai secret sebelum mengganti nilainya. Perbarui secret backend, deploy, uji, lalu cabut secret lama sesuai kemampuan platform. Untuk kebocoran aktif, pencabutan segera dapat didahulukan meskipun menyebabkan gangguan sementara.

Perubahan terbaru

Changelog 6 Oktober 2026 mencatat scoped personal access tokens. Saat tersedia pada alur yang dipakai, pilih project dan permission minimal serta expiry. Token manajemen berbeda dari API key aplikasi. [S28]

2.3 Memilih koneksi PostgreSQL

Buka tombol Connect pada project Supabase. Salin parameter dari mode koneksi yang dipilih, bukan dari contoh buku. Host, username, dan port harus berasal dari mode yang sama. [S12]

Mode	Port umum	Pemakaian
Direct	5432	Sesi panjang, admin, dump
Shared session pooler	5432	Admin pada jaringan IPv4
Shared transaction pooler	6543	Koneksi singkat/serverless

Direct dan jaringan

Direct memakai db.[PROJECT-REF].supabase.co dengan username postgres. Umumnya membutuhkan IPv6, atau IPv4 jika add-on yang relevan diaktifkan. Jika jaringan hanya IPv4, pilih session pooler yang ditampilkan Connect.

Session pooler

CONTOH bentuk: postgres.[PROJECT-REF] sebagai username dan hostname pooler dari dashboard, port 5432, database postgres. Jangan memakai username postgres direct bersama host pooler secara sembarang.

Transaction pooler

Mode ini mengembalikan koneksi ke pool setelah transaksi. Fitur state sesi dan prepared statements mempunyai keterbatasan. Untuk pgAdmin dan operasi admin yang membutuhkan sesi tetap, gunakan direct atau session pooler.

BACA sebelum menyambung

- Pastikan project tidak pause dan password database masih berlaku.
- Periksa network restrictions dan IP keluar jaringan bila fitur itu dipakai.
- Pastikan firewall mengizinkan port yang dipilih.
- Siapkan sertifikat CA dan mode SSL sesuai endpoint.

Selesai bila

Koneksi berhasil ke project yang benar melalui mode yang sesuai. Domain web server.7mit.org bukan otomatis hostname PostgreSQL. Cloudflare proxy web juga bukan pengganti koneksi database.

2.4 Menghubungkan pgAdmin 4

Pasang pgAdmin 4 dari situs resmi pada komputer pengelola. Jika memakai pgAdmin melalui browser, file sertifikat dan koneksi berada pada server pgAdmin, sehingga akses serta penyimpanannya harus dikelola di sana. [S13, S14]

SOP Register Server

1. Di object browser, klik kanan Servers > Register > Server.
2. General: beri nama jelas, misalnya server.7mit - PRODUKSI. Nama ini hanya label lokal.
3. Connection: isi Host name/address, Port, Maintenance database, dan Username dari Supabase Connect.
4. Masukkan password database, bukan password login dashboard atau API key.
5. Pada Parameters atau pengaturan SSL versi yang dipakai, atur sslmode dan Root certificate sesuai petunjuk Supabase.
6. Unduh sertifikat CA dari Database Settings jika disediakan. Pilih file CA tersebut pada Root certificate.
7. Pilih verify-full jika hostname dan rantai CA dapat diverifikasi. require mengenkripsi koneksi, tetapi tidak memberikan verifikasi identitas server setara verify-full.
8. Simpan, buka database postgres, lalu jalankan query identifikasi di bawah.

```
SELECT current_database(), current_user, version();
SELECT ssl, version, cipher
FROM pg_stat_ssl
WHERE pid = pg_backend_pid();
```

Hasil yang diharapkan

Database dan role sesuai project tujuan; pg_stat_ssl menunjukkan koneksi SSL aktif. Pengujian ini memeriksa koneksi, bukan keseluruhan permission aplikasi.

Jika gagal

Periksa typo host, pasangan username-host, IPv4/IPv6, password, port, CA, serta restriction IP. Jangan menjadikan sslmode=disable sebagai konfigurasi permanen. Simpan password hanya pada perangkat yang dikelola dengan baik.

2.5 Menjelajahi tabel dan Query Tool

Pada pgAdmin, buka Servers > server tujuan > Databases > postgres > Schemas. Schema public lazim berisi tabel aplikasi; schema auth dan storage dikelola layanan Supabase dan tidak boleh diedit sembarang.

Membaca tabel

1. Buka schema > Tables dan pilih tabel.
2. Gunakan Properties atau Columns untuk memahami tipe, primary key, foreign key, dan default.
3. Gunakan View/Edit Data dengan jumlah terbatas. Hindari mengambil seluruh tabel produksi berukuran besar.
4. Perhatikan bahwa grid edit dapat menyimpan perubahan; jangan menganggap semua panel bersifat baca saja.

Query Tool BACA

Pilih database lalu Tools > Query Tool. Query Tool menyediakan editor SQL, hasil Data Output, pesan, serta opsi transaksi. Tinjau koneksi dan Auto commit sebelum menjalankan perintah. [S15]

```
SELECT table_schema, table_name
FROM information_schema.tables
WHERE table_schema = 'public'
  AND table_type = 'BASE TABLE'
ORDER BY table_name;
```

Untuk tabel yang sudah dipahami, pilih kolom minimum dan batasi hasil. Contoh: SELECT id, created_at FROM public.nama_tabel LIMIT 20; hanya valid bila tabel dan kolom itu benar-benar ada.

Praktik internal

• Tandai koneksi produksi dengan nama dan warna yang jelas bila tersedia.
• Simpan query pemeriksaan sebagai file .sql tanpa secret.
• Tutup transaksi yang dibuka secara manual setelah selesai.
• Jangan membagikan hasil berisi data siswa atau token pada screenshot publik.

Hasil yang diharapkan

Admin dapat menemukan schema dan tabel, membaca struktur, serta menjalankan SELECT terbatas tanpa mengubah data.

2.6 Membaca Dashboard pgAdmin



Foto pgAdmin dari lampiran pengguna. Foto ini memberi konteks tampilan, bukan audit kesehatan server.

Cara membaca grafik

Grafik	Arti dan batas interpretasi
Sessions	Jumlah koneksi; active sedang menjalankan query, idle menunggu
Transactions/s	Aktivitas transaksi per waktu; commit dan rollback tidak selalu menunjukkan sukses/gagal aplikasi
Tuples in	Aktivitas insert, update, dan delete
Tuples out	Row yang diproses oleh scan; bukan jumlah row yang diterima browser
Block I/O	Read block dan hit cache PostgreSQL; bukan seluruh trafik disk OS

Pengamatan yang berguna

Bandingkan baseline dengan saat layanan lambat pada rentang waktu yang sama. Spike singkat dapat normal. Cari kondisi menetap: antrean koneksi, idle in transaction, blocking, durasi query panjang, atau error meningkat.

Hits tinggi sering berarti cache PostgreSQL membantu, tetapi Reads tidak selalu berarti pembacaan fisik disk karena OS juga memiliki cache. Grafik saja tidak membuktikan perlunya upgrade.

Rujukan tampilan Dashboard: S47. Definisi metrik PostgreSQL: S16. Pembacaan foto terbatas karena sudut dan resolusinya.

2.7 Metrik yang sering disalahartikan

Gunakan statistik database bersama latency aplikasi, error rate, dan metrik sumber daya dari penyedia. Nilai pg_stat_database bersifat kumulatif sejak reset statistik; grafik dapat menampilkan laju perubahan berdasarkan interval sampling. [S16, S48]

Sessions

Active berarti backend sedang mengerjakan query, termasuk menunggu sesuatu. Idle berarti tidak sedang menjalankan query. Idle in transaction berarti transaksi masih terbuka saat koneksi menunggu; ini dapat menahan lock atau menghambat pemeliharaan.
Banyak idle tidak otomatis berarti kebocoran koneksi karena pool sengaja mempertahankan koneksi. Bandingkan dengan kapasitas, pemakai koneksi, dan tren.

Transactions

Commit adalah transaksi yang selesai dengan commit. Rollback dapat muncul karena error, pembatalan sengaja, pengujian, atau logic aplikasi. Cari pesan error dan konteks, bukan menyamakan setiap rollback dengan insiden.

Tuples out

Returned terkait row yang dibaca sequential scan dan entri index scan. Fetched terkait row tabel yang diambil melalui index scan. Angka ini bukan penghitung response API kepada pengguna.

Block I/O

blks_hit adalah block yang ditemukan di buffer cache PostgreSQL. blks_read adalah block yang dibaca oleh PostgreSQL dan masih dapat dilayani cache OS. Rasio cache tinggi tidak membuktikan setiap query sudah optimal.

SOP interpretasi internal

1. Tetapkan baseline pada jam normal dan jam sibuk.
2. Catat durasi kondisi, bukan hanya puncak satu titik.
3. Cocokkan dengan query aktif, blocking, dan deploy terakhir.
4. Uji perbaikan pada ukuran beban yang sebanding.
5. Catat hasil sebelum dan sesudah dengan rentang waktu yang jelas.

2.8 SQL kesehatan: koneksi dan query aktif

BACA. Query berikut tidak mengubah data. Hak akses dapat membatasi informasi query milik role lain. Jalankan pada database tujuan dan hindari menyalin teks query yang mengandung data privat. [S16]

Menghitung koneksi per state

```
SELECT COALESCE(state, '(internal)') AS state,
       count(*) AS connections
FROM pg_stat_activity
WHERE datname = current_database()
GROUP BY state
ORDER BY connections DESC;
```

Query aktif yang berjalan lebih dari 30 detik

```
SELECT pid, usename, application_name,
       now() - query_start AS elapsed,
       wait_event_type, wait_event,
       left(query, 300) AS query_sample
FROM pg_stat_activity
WHERE datname = current_database()
  AND state = 'active'
  AND pid <> pg_backend_pid()
  AND query_start < now() - interval '30 seconds'
ORDER BY query_start;
```

Interpretasi operasional

Ambang 30 detik adalah filter contoh untuk investigasi, bukan batas resmi semua layanan. Query laporan dapat sah berjalan lama. Query login yang biasanya cepat tetapi berulang kali melampaui baseline memerlukan pemeriksaan.

wait_event membantu melihat apa yang ditunggu; NULL tidak otomatis berarti query sehat. Gunakan application_name dan waktu deploy untuk mempersempit asal koneksi.

Hasil yang diharapkan

Daftar sesi atau query mencurigakan untuk diperiksa lebih lanjut. Jangan langsung menghentikan semua PID. Pastikan pemilik aplikasi, fungsi query, serta transaksi yang terpengaruh dahulu.

2.9 SQL kesehatan: blocking dan transaksi

BACA. PostgreSQL menyediakan pg_blocking_pids untuk mengidentifikasi backend yang menghalangi sesi. Teks query dapat berisi informasi sensitif. [S29]

Sesi yang terblokir

```
SELECT pid, usename,
       now() - query_start AS elapsed,
       pg_blocking_pids(pid) AS blocking_pids,
       left(query, 250) AS query_sample
FROM pg_stat_activity
WHERE datname = current_database()
  AND cardinality(pg_blocking_pids(pid)) > 0
ORDER BY query_start;
```

Transaksi terbuka yang menganggur

```
SELECT pid, usename, application_name,
       now() - xact_start AS transaction_age,
       left(query, 250) AS last_query
FROM pg_stat_activity
WHERE datname = current_database()
  AND state LIKE 'idle in transaction%'
ORDER BY xact_start;
```

Tindak lanjut internal

1. Identifikasi PID blocker, aplikasi, dan pemilik transaksi.
2. Periksa apakah ada migrasi atau proses admin yang berjalan.
3. Hubungi pemilik proses dan tentukan apakah transaksi bisa diselesaikan normal.
4. Bila penghentian memang diperlukan, pahami bahwa pembatalan query dan penghentian koneksi mempunyai dampak berbeda.
5. Catat tindakan dan validasi fungsi aplikasi setelahnya.

Batas kewenangan

pg_cancel_backend dan pg_terminate_backend adalah tindakan produksi, bukan pemeriksaan rutin. Tidak semua admin Supabase dapat mengendalikan semua sesi. Buku ini tidak memberikan perintah penghentian massal.

Selesai bila

Antrean blocking hilang, transaksi ditutup sesuai kebutuhan, dan akar penyebab diperbaiki, misalnya aplikasi yang lupa commit atau rollback.

2.10 Performa query dan kapasitas

Mulai dari query lambat yang terbukti. Jangan menambah index pada semua kolom atau menaikkan compute hanya berdasarkan satu lonjakan grafik.

SOP investigasi internal

1. Catat endpoint, waktu, latency, error, dan volume request.
2. Periksa query aktif serta blocking; cocokkan dengan metrik CPU, memori, dan I/O penyedia.
3. Periksa apakah query mengambil kolom atau row berlebihan.
4. Uji filter, pagination, dan index pada lingkungan uji.
5. Bandingkan execution plan dan latency sebelum serta sesudah perubahan.
6. Terapkan perubahan melalui migrasi dan amati produksi.

EXPLAIN

```
-- UJI: sesuaikan tabel dan kolom yang benar.
EXPLAIN
SELECT id
FROM public.nama_tabel
WHERE owner_id = 'UUID-CONTOH'::uuid
LIMIT 20;
```

Ganti UUID-CONTOH dengan UUID valid; contoh ini tidak dapat langsung dieksekusi. EXPLAIN menampilkan rencana, sedangkan EXPLAIN ANALYZE benar-benar menjalankan query. Hindari ANALYZE pada query write produksi tanpa pengendalian transaksi. [S30]

Index dan beban

Index dapat membantu filter, join, dan urutan tertentu, tetapi memakai disk dan menambah biaya write. Sequential scan dapat tepat untuk tabel kecil atau query yang membaca banyak row. Pilih index berdasarkan workload.

Koneksi dan scale

Gunakan pooling yang cocok dengan lifecycle aplikasi. Tutup koneksi sesuai driver, batasi concurrency, dan cegah retry tanpa batas. Upgrade sumber daya dipertimbangkan setelah sumber beban dan kebutuhan kapasitas jelas.

Hasil yang diharapkan

Perbaikan terukur pada latency atau beban tanpa mengubah hasil query maupun akses data.

2.11 Data API, grants, dan RLS

Aplikasi tidak cukup hanya menyembunyikan tombol admin. Otorisasi harus diterapkan pada backend dan database. Grant menentukan akses objek; Row Level Security menentukan row yang boleh diakses. [S11, S17]

Contoh UJI: catatan milik pengguna

Contoh ini menggunakan Supabase Auth. Jangan menerapkannya pada tabel produksi atau SSO custom tanpa menyesuaikan model identitas.

```
CREATE TABLE public.demo_notes (
  id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
  owner_id uuid NOT NULL REFERENCES auth.users(id),
  body text NOT NULL
);
ALTER TABLE public.demo_notes ENABLE ROW LEVEL SECURITY;
GRANT SELECT, INSERT ON public.demo_notes TO authenticated;
CREATE POLICY notes_read ON public.demo_notes
  FOR SELECT TO authenticated
  USING ((SELECT auth.uid()) = owner_id);
CREATE POLICY notes_add ON public.demo_notes
  FOR INSERT TO authenticated
  WITH CHECK ((SELECT auth.uid()) = owner_id);
```

Uji yang wajib disiapkan

• Akun A dapat membuat dan membaca row miliknya.
• Akun B tidak dapat membaca row A atau membuat row atas nama A.
• Pengunjung tanpa sesi tidak mendapat akses.
• Secret backend hanya dipakai setelah otorisasi server yang sesuai.

Pengujian melalui pgAdmin sebagai role postgres tidak mewakili akses browser karena role berhak tinggi dapat melewati RLS. Uji melalui Data API menggunakan sesi pengguna uji.

Model akun.7mit custom

Username, tabel portal_login_accounts, atau klaim role pada frontend tidak otomatis menghasilkan auth.uid(). Buat pemetaan identitas dan verifikasi server yang jelas. Jangan menambahkan policy permisif USING (true) untuk menutupi masalah login.

2.12 Auth dan perubahan domain login

Pada Supabase Auth, Site URL serta Redirect URLs menentukan lokasi callback yang diizinkan. URL produksi sebaiknya tepat; pola wildcard lebih cocok untuk preview yang dikendalikan. [S18]

SOP ketika hostname berubah

1. Catat URL frontend, Site URL, callback, provider OAuth, dan lingkungan.
2. Tambahkan URL tujuan yang diperlukan di konfigurasi Auth.
3. Cocokkan redirectTo pada aplikasi dengan URL yang diizinkan.
4. Periksa callback provider OAuth di dashboard provider terkait.
5. Uji sign in, sign out, reset password, dan email verification memakai akun uji.
6. Cabut URL lama setelah perpindahan tervalidasi dan pengguna tidak lagi memerlukannya.

Authentication dan authorization

Login membuktikan identitas. Role dan policy menentukan tindakan yang diizinkan. Gunakan data peran yang dikelola server; user_metadata yang dapat diedit pengguna tidak boleh menjadi dasar izin admin. [S17, S31]

Sesi dan pencabutan

Perubahan role dapat memerlukan refresh token sebelum klaim baru terlihat. Penghapusan user tidak boleh diasumsikan langsung membatalkan semua access token yang masih berlaku. Sesuaikan pencabutan sesi dengan kebutuhan operasi sensitif. [S46]

Model akun.7mit

Jika portal memakai akun.7mit melalui backend custom, dokumentasikan validasi sesi, expiry, pemetaan role, dan aturan logout. Jangan menganggap konfigurasi Supabase Auth saja memperbaiki semua alur custom tersebut.

Hasil yang diharapkan

Login kembali ke hostname yang benar. Pengguna biasa tidak memperoleh akses admin dengan mengganti nilai lokal, URL, atau metadata profil.

2.13 Storage, Realtime, dan Edge Functions

Storage

Bucket public memungkinkan pengambilan file melalui URL publik, tetapi operasi write tetap memerlukan kontrol akses. Bucket private memerlukan akses terotorisasi atau signed URL. Simpan dokumen internal pada bucket private sesuai kebutuhan. [S19]

Uji upload baru, download, overwrite, dan delete dengan pengguna yang sesuai. Upsert dapat membutuhkan INSERT, SELECT, serta UPDATE. Mengubah status bucket dapat mengubah keterbukaan file; tinjau dulu isi dan penggunaannya.

Realtime

Catat tabel atau channel yang memakai realtime, jenis subscription, dan siapa yang boleh menerima event. Realtime tidak menggantikan verifikasi izin. Jika event tidak muncul, periksa konfigurasi publikasi/fitur, policy, koneksi websocket, serta listener aplikasi berdasarkan jenis Realtime yang dipakai. [S32]

Edge Functions

Gunakan function untuk logic server dan integrasi yang memakai secret. Simpan secret di pengaturan atau perintah secrets resmi; jangan menulis API key pada source. [S20]

SOP function internal

1. Identifikasi input, output, pengguna yang diizinkan, dan layanan eksternal.
2. Verifikasi token/sesi serta role sesuai model autentikasi.
3. Validasi ukuran input, tipe data, timeout, dan batas request.
4. Konfigurasikan CORS untuk origin yang diperlukan; CORS bukan sistem otorisasi.
5. Simpan log tanpa token, password, atau isi privat yang tidak diperlukan.
6. Uji sukses, input salah, tanpa sesi, upstream timeout, dan rate limit.

Integrasi AI 7 MIT

Endpoint API AI dan secret berada pada konfigurasi backend yang telah diverifikasi. Browser tidak boleh memperoleh secret upstream. Gangguan upstream perlu menghasilkan pesan yang jelas serta retry terbatas.

2.14 Migrasi dan perubahan data

Schema database harus mempunyai riwayat yang dapat direview. Jangan mengandalkan perubahan Table Editor yang tidak pernah dicatat pada repository. Gunakan workflow migrasi sesuai versi Supabase CLI dan project. [S21]

Alur UJI sebelum PRODUKSI

1. Sinkronkan branch kerja dan periksa versi CLI melalui supabase --version.
2. Baca supabase --help serta bantuan subcommand yang diperlukan.
3. Identifikasi perubahan schema, grants, RLS, trigger, dan dependensi aplikasi.
4. Buat atau hasilkan migration melalui CLI yang sesuai alur project.
5. Review SQL dan uji dari database latihan dengan kondisi awal yang representatif.
6. Pastikan frontend lama dan baru dapat melewati periode transisi.
7. Siapkan backup, rollback/forward fix, jadwal, serta pengamatan setelah penerapan.

Mengubah row secara terkontrol

Contoh berikut hanya pola latihan. Nama tabel, id, dan kolom harus disesuaikan.

```
BEGIN;
SELECT id, status FROM public.nama_tabel
WHERE id = 'UUID-VALID'::uuid FOR UPDATE;
UPDATE public.nama_tabel SET status = 'baru'
WHERE id = 'UUID-VALID'::uuid
RETURNING id, status;
ROLLBACK; -- latihan: batalkan perubahan
```

Hal yang perlu dicegah

UPDATE atau DELETE tanpa filter dapat mengubah banyak row. TRUNCATE, DROP, reset database, dan perubahan schema terkelola bukan langkah troubleshooting rutin.

Hasil yang diharapkan

Perubahan tercatat, diuji, dan sesuai row atau objek yang direncanakan. Untuk produksi, keputusan COMMIT dilakukan setelah hasil diperiksa; jangan menyalin contoh transaksi tanpa memahami efeknya.

2.15 Backup dan pemulihan

Backup harus diuji lewat restore. Retensi, daily backup, dan point-in-time recovery bergantung pada paket serta fitur project. Periksa ketersediaan aktual di dashboard; jangan menganggap seluruh project punya kemampuan sama. [S22]

Yang dipisahkan

Aset	Metode pencadangan yang dicatat
Data dan schema database	Backup platform atau logical dump
File Storage	Salinan objek/file terpisah
Source code	Git repository dan arsip yang diperlukan
Secret/config	Penyimpanan aman dan inventaris
DNS dan deployment	Catatan konfigurasi serta versi

Backup database tidak mencakup isi objek Storage. Metadata bucket atau object dalam database berbeda dari file fisiknya. [S10, S33]

SOP backup internal

1. Tentukan ruang lingkup, target pemulihan waktu (RTO), dan toleransi kehilangan data (RPO).
2. Jadwalkan pencadangan sesuai aktivitas layanan dan paket.
3. Jika memakai pg_dump, gunakan versi tool yang kompatibel dan koneksi admin yang sesuai. [S23]
4. Enkripsi hasil, batasi akses, simpan salinan terpisah, dan catat checksum serta waktu.
5. Restore ke lingkungan terisolasi, lalu uji jumlah data penting, login, RLS, file, dan function.

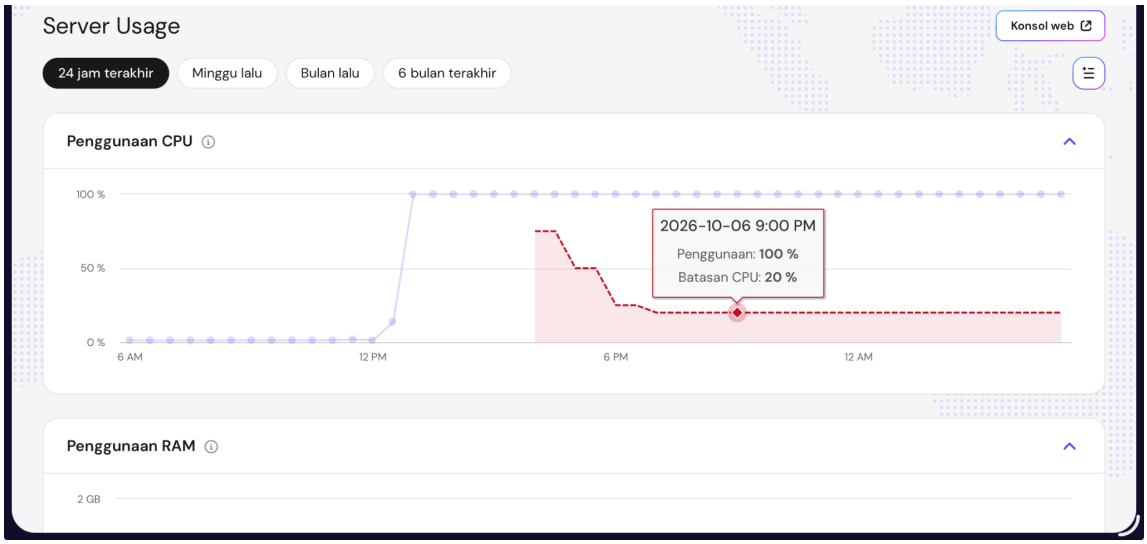
Pemulihan produksi

Tentukan titik restore dan dampak kehilangan perubahan setelah titik itu. Hentikan write bila rencana memerlukannya. Pulihkan database/file sesuai cakupan, lalu verifikasi endpoint, sesi, secret, dan integrasi.

Selesai bila

Latihan restore mempunyai bukti berhasil dan waktu pemulihan terukur. Satu file dump yang belum pernah dibuka bukan bukti bahwa seluruh layanan bisa pulih.

2.16 Insiden AI Jakarta: membaca bukti



Lampiran pengguna: penggunaan CPU 100% dan batas CPU 20% pada titik waktu yang ditampilkan.

Screenshot sebelumnya menunjukkan penggunaan 100% dengan batas 75%; screenshot berikutnya menunjukkan batas 20%. Itu bukti nilai pada grafik, bukan bukti penyebab throttling atau kondisi saat buku dibaca.

SOP investigasi internal

1. Konfirmasikan hostname VPS, lokasi Jakarta, dan proses AI yang berjalan.
2. Bandingkan CPU, RAM, disk, request masuk, dan antrean pada periode yang sama.
3. Periksa proses dominan, perubahan deploy, task berkala, serta retry berulang.
4. Hubungkan kegagalan function/API dengan request ID dan waktu upstream.
5. Batasi concurrency atau aktivitas nonkritis sesuai hasil diagnosis.
6. Koordinasikan batas CPU dengan penyedia bila mekanismenya belum jelas.

Komunikasi status.7mit

Saat investigating, tulis dampak yang teramati, komponen, waktu, dan tindakan investigasi. Hindari menyatakan akar penyebab sebelum terbukti. Setelah perbaikan, monitoring perlu membuktikan request kembali stabil sebelum insiden ditutup.

Contoh pesan: Kami sedang menyelidiki gangguan pada server intelligence.7mit Jakarta. Sebagian permintaan AI dapat lambat atau gagal. Tim IT sedang memeriksa beban server dan koneksi layanan.

2.17 Troubleshooting server utama

Gejala	Pemeriksaan awal	Langkah lanjutan
pgAdmin timeout	Host, port, IPv4/IPv6, IP rule	Gunakan mode Connect yang sesuai
Password rejected	Role dan password DB	Verifikasi atau rotasi terencana
SSL error	CA, hostname, sslmode	Perbaiki trust chain/config
API permission denied	Grant, schema, role	Tinjau grant sebelum policy
API data kosong	Filter, sesi, RLS	Uji akun dan ownership
Update nol row	WHERE dan SELECT policy	Uji policy read/write
Upload/overwrite gagal	Bucket, policy, ukuran	Periksa operasi yang dibutuhkan
Function 401/403	Model token dan izin	Verifikasi auth pada backend
Function 5xx/timeout	Log, DB, upstream	Cocokkan request dan waktu
Layanan lambat	Blocking, query, sumber daya	Diagnosis sebelum scale

SOP pengumpulan bukti

1. Catat lingkungan, endpoint, waktu WIB, serta perubahan terakhir.
2. Reproduksi dengan akun uji dan input minimum.
3. Simpan status HTTP dan pesan error yang disamarkan.
4. Bandingkan gejala pada browser, backend, dan database.
5. Ubah satu penyebab yang terbukti, lalu ulangi pengujian yang sama.

Batas pemeriksaan

Jangan mematikan RLS, membuka bucket privat, atau menyebarkan service_role untuk memperbaiki akses. Periksa pemetaan identitas dan permission yang sebenarnya.

Kriteria penyelesaian

Endpoint berhasil untuk pengguna yang berhak dan tetap ditolak untuk pengguna yang tidak berhak. Koneksi serta performa stabil selama periode pengamatan yang sesuai pola beban. Rujukan teknis: S11-S20. Urutan diagnosis disusun sebagai SOP internal.

2.18 Backup dan restore melalui pgAdmin

UJI dahulu. Backup pgAdmin memakai utilitas PostgreSQL; versi binary harus kompatibel dengan server. Gunakan direct atau session pooler dan role dengan izin membaca objek yang dibutuhkan. [S23, S44]

SOP membuat logical backup

1. Pastikan database dan lingkungan yang dipilih benar.
2. Klik kanan database > Backup; pilih format Custom bila ingin arsip yang dapat dipilih melalui pg_restore.
3. Pilih lokasi file yang aman. Pada pgAdmin web, lokasi ini ada pada server pgAdmin.
4. Tinjau bagian data/schema, pemilihan objek, serta opsi role/ownership berdasarkan tujuan restore.
5. Jalankan backup, baca Process Watcher atau log, lalu pastikan proses sukses dan file terbentuk.
6. Catat waktu, versi tool, cakupan, ukuran, dan checksum. Enkripsi atau pindahkan hasil ke penyimpanan backup berwenang.

SOP restore latihan

1. Siapkan database/project terisolasi yang sesuai, bukan produksi.
2. Klik kanan target > Restore, pilih arsip yang didukung dan cocokkan opsi restore.
3. Review ownership, grants, schema, extension, dan objek terkelola sebelum eksekusi.
4. Jalankan restore, periksa log sampai selesai, lalu validasi tabel, RLS, dan data penting. [S45]

Batas Supabase

Auth, Storage, role, extension, dan komponen terkelola membuat pemindahan project lebih kompleks dari restore satu dump. Gunakan prosedur Supabase untuk migrasi penuh; arsip pgAdmin tidak otomatis memulihkan file Storage, secret, atau konfigurasi dashboard. [S33]

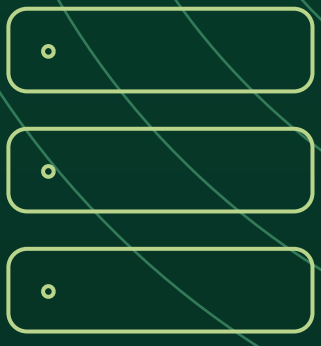
Selesai bila

Restore latihan terbukti berhasil dan cakupannya jelas. Jangan memakai opsi clean/drop pada produksi tanpa rencana yang telah direview.

CHAPTER 3

GitHub

Source code 7 MIT di github.com/7-MIT



3.1 Organisasi 7-MIT dan repository

Organisasi GitHub adalah ruang bersama untuk repository, anggota, dan integrasi. Repository menyimpan file serta riwayat commit; repository bukan backup otomatis seluruh data Supabase.

Repository yang ditemukan

Repository	Default branch pada pemeriksaan
7-MIT/7mit-guru	claude/portal-interactive-performance-g8tl22
7-MIT/7mit-eksekutif	claude/portal-interactive-performance-g8tl22
7-MIT/7mit-siswa	claude/portal-interactive-performance-g8tl22
7-MIT/home	claude/focused-ptolemy-3wj18o
7-MIT/7mit-denahkelas	claude/nice-cray-navmmg

Diperoleh melalui konektor GitHub pada 7 Oktober 2026. Kelimanya berstatus public saat pemeriksaan. Daftar ini adalah hasil pencarian yang tersedia, bukan jaminan seluruh repository organisasi. [G1]

Hal penting untuk deployment

Jangan mengasumsikan main ada atau menjadi produksi. Default branch adalah pilihan utama GitHub; Pages mempunyai pengaturan production branch sendiri. Periksa keduanya, lalu catat pemetaan ke hostname.

SOP BACA repository

1. Buka https://github.com/7-MIT dan pilih repository.
2. Periksa branch yang aktif, README, struktur folder, dan commit terakhir.
3. Buka Settings jika memiliki izin untuk melihat default branch dan integrasi.
4. Cocokkan repository dengan project hosting melalui inventaris.

Batas verifikasi

Isi source dan konfigurasi build setiap repository belum diaudit dalam buku ini. Jangan menyimpulkan bahwa semua repo memakai framework, backend, atau deployment yang sama.

3.2 Anggota, teams, dan akses repository

Gunakan akun pribadi anggota. Beri akses melalui team bila sesuai kebutuhan agar pergantian pengurus lebih mudah dikelola.

Peran repository organisasi

Peran	Fokus pemakaian
Read	Membaca dan meninjau
Triage	Mengelola issue dan PR sesuai izin
Write	Mengirim perubahan kode
Maintain	Mengelola repo tanpa semua aksi sensitif
Admin	Pengaturan serta izin sensitif repo

Kemampuan detail dan custom roles bergantung paket GitHub. Organization owner berbeda dari repository admin. Gunakan matriks izin resmi saat memilih role. [S24]

SOP pemberian akses internal

1. Verifikasi username GitHub dan identitas penerima.
2. Undang ke organisasi atau sebagai collaborator sesuai kebutuhan.
3. Pilih team dan repository yang ditugaskan.
4. Berikan role minimum; hindari owner untuk pekerjaan kode rutin.
5. Pastikan undangan diterima dan MFA memenuhi kebijakan organisasi.
6. Uji akses baca, lalu tindakan kerja pada branch latihan bila perlu.

Integrasi pihak ketiga

Tinjau GitHub Apps, token, deploy keys, dan webhook. Untuk Pages, pilih repository yang diperlukan saja. Hak manusia dan installation GitHub App adalah dua pengaturan berbeda.

Offboarding

Cabut membership/collaborator, token, SSH atau deploy key terkait, dan akses secret. Pastikan ada pengelola pengganti sebelum menghapus satu-satunya akses pemulihan.

Hasil yang diharapkan

Orang yang ditugaskan dapat bekerja pada repo yang tepat, sedangkan repo dan pengaturan lain tetap di luar cakupannya.

3.3 Membaca source dan mengunduh

Mulai dari README dan struktur folder. Jangan menjalankan script instalasi dari source yang belum dipahami. Kode frontend, function backend, dan migration dapat berada di repository atau folder berbeda.

Struktur yang perlu dicari

File/folder	Yang diperiksa
README.md	Cara setup, build, dan deployment
package.json	Scripts serta dependency jika Node digunakan
Lockfile	Versi dependency yang disepakati
src / public	Kode aplikasi dan aset bila ada
supabase	Function, config, dan migration bila ada
.github/workflows	Otomasi CI jika dipakai
.env.example	Nama variabel tanpa secret

Download ZIP

Pada Code > Download ZIP, periksa branch yang dipilih. ZIP berisi snapshot source dan tidak menyediakan riwayat Git untuk workflow commit/push. Download ZIP bukan build output siap hosting. [S25]

Clone untuk pengembangan

Clone membuat salinan repository beserta metadata Git yang dibutuhkan untuk branch dan commit. Pilih URL HTTPS atau SSH melalui menu Code. Private repo memerlukan metode autentikasi yang berhak. [S26]

```
git clone https://github.com/7-MIT/7mit-siswa.git
cd 7mit-siswa
git remote -v
git branch -a
git status
```

Selesai bila

Remote menunjuk organisasi dan repo yang benar, branch diketahui, dan working tree dipahami. Jangan menambahkan token ke URL remote yang dapat tersimpan pada log atau konfigurasi.

3.4 Branch, commit, dan sinkronisasi

Branch memisahkan pekerjaan. Commit menyimpan perubahan yang bisa ditelusuri. Push mengirim commit ke GitHub; pada repo yang terhubung hosting, push juga dapat memicu build.

SOP branch kerja internal

1. Pastikan working tree bersih atau simpan perubahan yang masih diperlukan.
2. Jalankan git fetch origin untuk memperbarui informasi remote.
3. Tentukan branch dasar yang disepakati tim, bukan otomatis main.
4. Buat branch kerja, ubah file, jalankan pemeriksaan, dan review diff.
5. Commit hanya file yang relevan lalu push branch kerja.

```
git fetch origin
# Ganti BRANCH-DASAR dengan nama yang diverifikasi.
git switch -c fix/login-portal origin/BRANCH-DASAR
git status
git diff
git add path/file-yang-diubah
git commit -m "Perbaiki redirect login portal"
git push -u origin fix/login-portal
```

Review sebelum commit

Periksa apakah diff mengandung secret, data pribadi, output build besar, atau file yang tidak terkait. Hindari git add . tanpa memeriksa seluruh perubahan.

Konflik

Jika basis berubah, sinkronkan sesuai workflow tim. Merge mempertahankan hubungan histori; rebase menyusun ulang commit dan mengubah SHA. Rebase pada branch yang dipakai orang lain memerlukan koordinasi. [S27]

Hasil yang diharapkan

Branch kerja berisi perubahan terfokus dari basis yang benar. Jangan force push branch produksi atau menghapus branch hanya karena local workspace bermasalah.

3.5 Pull request dan review

Pull request (PR) menyajikan perubahan branch untuk ditinjau sebelum merge. Pastikan base branch benar; branch tujuan yang salah dapat menerbitkan perubahan ke lingkungan yang keliru. [S34]

SOP membuat PR

1. Push branch kerja dan buka Compare & pull request.
2. Pilih base branch dan compare branch yang tepat.
3. Tulis judul yang menjelaskan perilaku yang berubah.
4. Jelaskan masalah, hasil perubahan, cara pengujian, serta dampak migrasi/env jika ada.
5. Tambahkan reviewer yang memahami komponen terkait.
6. Periksa Files changed, status checks, dan preview.
7. Selesaikan diskusi serta konflik sebelum merge sesuai kebijakan tim.

Contoh isi PR internal

Masalah: setelah login, portal kembali ke hostname lama. Perubahan: callback diarahkan ke hostname produksi yang telah diverifikasi. Validasi: login dan reset password berhasil dengan akun uji pada preview. Deployment: update allowlist Auth dan environment sebelum rilis.

Checklist reviewer

• Apakah perubahan menyelesaikan pemicu masalah yang dijelaskan?
• Apakah izin pengguna dan RLS masih sesuai?
• Apakah ada secret atau data privat di diff?
• Apakah test dan build relevan lulus?
• Apakah frontend dan database tetap kompatibel saat rilis?

Setelah merge

Catat commit hasil merge dan deployment yang dibuat. Hapus branch kerja hanya jika tidak lagi diperlukan. PR merged belum membuktikan hosting menerbitkan commit yang sama; cek deployment secara terpisah.

3.6 Rulesets dan branch protection

Gunakan aturan agar branch penting menerima perubahan melalui review dan pemeriksaan. Ketersediaan fitur pada private repository bergantung paket. Admin perlu memeriksa aturan yang benar-benar aktif. [S35]

Rancangan kebijakan internal

Kontrol	Tujuan
Require pull request	Perubahan produksi dapat direview
Required approvals	Ada pemeriksa yang ditugaskan
Required status checks	Build/test yang dipilih harus lulus
Resolve conversations	Masalah review dituntaskan
Batasi force push/delete	Melindungi riwayat dan branch
Bypass terbatas	Jalur darurat dapat ditelusuri

SOP pengaturan

1. Tentukan branch yang dipakai produksi berdasarkan inventaris.
2. Buka Settings dan pilih Rules/Rulesets atau Branches sesuai fitur repo.
3. Tentukan target branch dengan pola yang tepat.
4. Pilih kontrol dan status checks yang memang tersedia serta pernah berjalan.
5. Periksa enforcement, siapa yang boleh bypass, dan apakah admin juga terikat.
6. Uji dengan PR latihan, lalu pastikan merge langsung yang seharusnya dilarang benar-benar ditolak.

Hindari aturan yang mengunci tim

Required check yang salah nama atau tidak pernah berjalan dapat membuat PR tertahan. Simpan jalur pemulihan oleh owner yang sah. Perubahan default branch tidak otomatis memperbarui semua aturan dan integrasi.

Hasil yang diharapkan

Perubahan tidak dapat melewati review dan checks secara tidak sengaja. Kebijakan tertulis harus cocok dengan enforcement aktual, bukan hanya screenshot pengaturan.

3.7 GitHub Actions dan secret

GitHub Actions menjalankan workflow dari file YAML pada .github/workflows. Mulai dari build dan test sebelum mengotomatisasi deployment produksi. Baca permission workflow serta trigger yang digunakan. [S36]

Contoh UJI: CI untuk project Node

Contoh hanya berlaku bila package-lock.json dan script build ada. Versi Node harus disesuaikan project; versi action yang dipakai perlu diverifikasi dan dapat dipin ke commit SHA tepercaya.

```
name: Validate
on: [pull_request]
permissions:
  contents: read
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v7
      - uses: actions/setup-node@v7
        with:
          node-version: '22'
          cache: npm
      - run: npm ci
      - run: npm run build
```

Versi action contoh mengikuti sumber resmi yang diperiksa pada tanggal edisi. [S42, S43]

Secrets dan variables

Simpan secret melalui Settings > Secrets and variables > Actions pada cakupan repo, organization, atau environment yang sesuai. Referensikan secrets dengan mekanisme Actions, bukan nilai literal YAML. [S37]

Kontrol internal

Batasi token workflow. Jangan mencetak secret atau mengunggah .env sebagai artifact. Perhatikan bahwa PR dari fork mempunyai pembatasan secret; jangan mengatasinya dengan workflow berhak tinggi yang menjalankan kode tidak tepercaya.

Hasil yang diharapkan

Checks berulang pada PR yang sama, menghasilkan build terverifikasi tanpa membuka kredensial. Tambahkan test relevan sesuai framework sebelum check dijadikan syarat merge.

3.8 Menghubungkan repo ke deployment

GitHub menyimpan perubahan; Pages dan Supabase menerapkan komponen masing-masing. Tentukan satu jalur deployment utama agar admin mengetahui apa yang akan terjadi setelah merge.

Pemetaan yang perlu dicatat

Source	Penerapan	Validasi
Frontend	Pages Git integration / jalur resmi lain	Commit aktif, rute, login
Edge Function	Deploy function sesuai workflow	Version, log, request uji
Migration SQL	Workflow database terkontrol	Schema, grants, RLS
Konfigurasi/env	Dashboard atau secret manager	Nilai lingkungan dan hasil uji

SOP rilis lintas komponen internal

1. Identifikasi PR frontend, function, dan migration yang saling bergantung.
2. Pilih urutan rilis yang menjaga kompatibilitas. Perubahan schema additive sering membantu transisi.
3. Terapkan pada lingkungan uji dan jalankan alur pengguna lengkap.
4. Pastikan backup serta rencana pemulihan tiap komponen tersedia.
5. Deploy produksi melalui jalur yang disepakati dan catat commit/version.
6. Uji homepage, login, akses pengguna, data, serta fitur yang diubah.
7. Amati error dan latency selama periode yang mencakup beban representatif.

Menghindari deploy ganda

Jika Pages Git integration sudah menerbitkan frontend, jangan menambah Actions deployment ke target yang sama tanpa tujuan jelas. Dua pipeline dapat menimpa versi satu sama lain.

Hasil yang diharapkan

Dari hostname publik, admin dapat menelusuri deployment, commit source, dan perubahan backend terkait. Rollback frontend tetap harus mempertimbangkan schema database yang aktif.

3.9 Source hygiene dan secret bocor

Repository private tetap tidak boleh dijadikan tempat secret. Riwayat commit, fork, clone, artifact, dan log bisa menyimpan nilai lama.

Pola .gitignore CONTOH

```
.env
.env.*
!.env.example
node_modules/
dist/
*.pem
*.key
```

Sesuaikan project. Jangan mengabaikan file yang sengaja diperlukan oleh build hanya karena contoh ini. .env.example berisi nama variabel dan nilai dummy, bukan kredensial nyata.

SOP kebocoran secret

1. Identifikasi jenis secret, cakupan izin, serta layanan pemakainya.
2. Cabut atau rotasi secret pada penyedia terlebih dahulu; menghapus file saja tidak cukup.
3. Perbarui secret pada backend/CI dan validasi layanan.
4. Tinjau log penggunaan yang tersedia dan tindakan yang mencurigakan.
5. Jika perlu membersihkan histori, ikuti prosedur GitHub serta koordinasi dengan seluruh pemilik clone/fork. [S38]
6. Dokumentasikan insiden tanpa mengulang nilai secret.

Pemeriksaan sebelum PR

Cari pola key, password, connection string, dump database, dan data siswa pada diff. Jangan mempublikasikan log diagnostik mentah. Secret scanning jika tersedia membantu, tetapi review manusia dan penyimpanan yang benar tetap diperlukan.

Hasil yang diharapkan

Kredensial yang bocor tidak lagi bisa dipakai, layanan memakai nilai baru, dan source tidak mengandung secret aktif. Membersihkan histori bukan pengganti rotasi.

3.10 Troubleshooting GitHub

Gejala	Periksa	Tindakan
Repository tidak terlihat	Account, org, invitation	Verifikasi membership dan izin
Clone private repo gagal	HTTPS token/SSH dan scope	Gunakan autentikasi yang benar
Push rejected	Role, branch rule, remote update	Ikuti PR/sinkronisasi
Merge conflict	File dan basis kedua branch	Review hasil gabungan
Actions gagal	Step error pertama, runtime	Reproduksi pada lingkungan uji
Repo tidak muncul di Pages	GitHub App installation	Pilih repo yang benar
Commit tidak terdeploy	Branch, trigger, checks	Periksa log pipeline
ZIP tidak siap hosting	Build dan output folder	Bangun source lebih dulu

SOP konflik internal

1. Simpan pekerjaan lokal dan pahami file yang berkonflik.
2. Baca kedua versi serta tujuan perubahan, bukan hanya memilih seluruh ours/theirs.
3. Gabungkan logika yang benar, hapus conflict markers, dan review diff.
4. Jalankan test/build yang relevan sebelum commit penyelesaian.
5. Push melalui workflow branch yang disepakati.

Restore source

Gunakan commit atau release yang diketahui sebagai referensi. Mengembalikan source berbeda dari mengembalikan data. Jangan force push atau reset branch produksi untuk mengatasi kesalahan lokal.

Kriteria penyelesaian

PR dapat direview, checks lulus, target deployment benar, dan hasil publik sesuai commit yang dicatat. Setelah insiden, perbarui README atau SOP agar pengelola berikutnya dapat mengulangi langkah dengan tepat.

A. Checklist harian, mingguan, dan rilis

Checklist berikut adalah usulan ritme internal 7 MIT. Sesuaikan frekuensinya dengan tingkat perubahan dan jam pemakaian layanan.

Harian

• Periksa akses hostname utama, status HTTPS, serta status layanan.
• Lihat error Auth, Data API, Storage, Functions, dan upstream AI.
• Periksa latency, query panjang, blocking, dan antrean koneksi bila ada gejala.
• Periksa deployment gagal atau perubahan yang tidak direncanakan.
• Catat insiden dan tindak lanjut dengan waktu WIB.

Mingguan

• Tinjau anggota, token, GitHub Apps, serta scope integrasi.
• Periksa tanggal expiry domain/sertifikat dan status perpanjangan.
• Periksa backup yang benar-benar tersedia dan lakukan restore latihan berkala.
• Tinjau kapasitas, penggunaan Storage, dan biaya aktual pada dashboard.
• Tinjau PR tertahan, dependency, serta Advisor yang relevan.
• Perbarui inventaris hostname, repo, branch, project, dan pemilik layanan.

Sebelum rilis

• Ada commit/version, review, test, serta preview yang lolos.
• Migration, function, env, dan frontend punya urutan yang jelas.
• Backup serta rencana pemulihan telah diperiksa.
• Data pengguna dan secret tidak masuk source/log publik.

Setelah rilis

Uji alur pengguna utama dan satu uji penolakan izin. Catat commit aktif pada hosting dan versi backend. Amati error serta performa, lalu tutup pekerjaan setelah hasilnya terbukti.

Bukti yang disimpan

Tanggal, pemeriksa, layanan, temuan, tindakan, hasil, dan tautan PR/incident. Catatan tindakan boleh dibagikan sesuai kewenangan; secret disimpan terpisah.

B. Template inventaris dan perubahan

Inventaris per layanan

Kolom	Isi yang perlu dicatat
Nama layanan	Nama portal atau API
Hostname	Domain resmi dan hostname cadangan
Domain zone	7mit.org / 7mit.cloud / lainnya
Hosting project	Nama dan ID project
Source repository	URL repository
Branch produksi	Nama tepat, termasuk slash
Backend	Project ref dan region
Pemilik operasional	Orang/team yang berwenang
Dependensi	Auth, function, API eksternal
Backup	Cakupan, jadwal, lokasi aman
Terakhir diverifikasi	Tanggal, waktu, pemeriksa

Catatan perubahan

Tanggal/waktu WIB: ...
Layanan dan lingkungan: ...
Tujuan perubahan: ...
Nilai lama dan baru tanpa secret: ...
PR/commit/deployment: ...
Backup dan rencana pemulihan: ...
Pengujian sebelum dan sesudah: ...
Hasil serta tindak lanjut: ...

Aturan pengisian

Jangan memasukkan password, token, connection string berkredensial, recovery code, atau data siswa. Gunakan nama secret dan lokasi penyimpanan yang berwenang. Catat perbedaan default branch dan branch produksi agar handover tidak salah.
Template merupakan rancangan internal, bukan formulir audit resmi dari penyedia.

C. Glosarium untuk pengelola

Istilah	Arti praktis
Apex	Domain utama, misalnya 7mit.org
Zone	Kumpulan konfigurasi DNS satu domain
Registrar	Penyedia registrasi/perpanjangan domain
Nameserver	Server yang menjawab authoritative DNS
TTL	Masa cache record DNS
Origin	Server tujuan di belakang proxy
Deployment	Versi aplikasi yang diterapkan
Preview	Lingkungan peninjauan sebelum produksi
Schema	Namespace objek PostgreSQL
Transaction	Unit perubahan yang di-commit/rollback
Pooler	Pengelola penggunaan koneksi database
RLS	Aturan akses per row database
Grant	Izin role terhadap objek database
Auth	Layanan identitas dan sesi login
Secret	Kredensial dengan akses sensitif
Migration	Riwayat perubahan struktur database
Repository	Source beserta riwayat perubahan
Branch	Jalur kerja perubahan pada Git
Commit SHA	Identitas versi perubahan Git
Pull request	Permintaan review sebelum merge
CI	Pengujian/build otomatis perubahan
RPO	Toleransi kehilangan data dalam waktu
RTO	Target waktu pemulihan layanan

Prinsip pengingat

Nama domain, hosting, source code, serta database adalah komponen berbeda. Selalu cocokkan target, lingkungan, dan hak akses sebelum menjalankan perubahan.

D. Sumber resmi / 1

Dokumentasi diperiksa pada 7 Oktober 2026. Link menunjuk sumber resmi. Cocokkan dokumentasi PostgreSQL dan pgAdmin dengan versi yang digunakan. SOP internal serta contoh bukan konfigurasi produksi yang terverifikasi.

S1 / Cloudflare Pages: Custom domains
https://developers.cloudflare.com/pages/configuration/custom-domains/
S2 / Cloudflare Pages: GitHub integration
https://developers.cloudflare.com/pages/configuration/git-integration/github-integration/
S3 / Cloudflare Pages: Build configuration
https://developers.cloudflare.com/pages/configuration/build-configuration/
S4 / Cloudflare DNS: Proxy status
https://developers.cloudflare.com/dns/proxy-status/
S5 / Cloudflare SSL: Full (strict)
https://developers.cloudflare.com/ssl/origin-configuration/ssl-modes/full-strict/
S6 / Cloudflare DNS: DNSSEC
https://developers.cloudflare.com/dns/dnssec/
S7 / Cloudflare: Create API token
https://developers.cloudflare.com/fundamentals/api/get-started/create-token/
S8 / Cloudflare Pages: Rollbacks
https://developers.cloudflare.com/pages/configuration/rollbacks/
S9 / Cloudflare Pages: Serving Pages
https://developers.cloudflare.com/pages/configuration/serving-pages/

D. Sumber resmi / 2

S10 / Supabase: Database overview
https://supabase.com/docs/guides/database/overview
S11 / Supabase: API keys
https://supabase.com/docs/guides/getting-started/api-keys
S12 / Supabase: Connecting to Postgres
https://supabase.com/docs/guides/database/connecting-to-postgres
S13 / Supabase: Connecting with pgAdmin
https://supabase.com/docs/guides/database/pgadmin
S14 / pgAdmin 4: Server Dialog
https://www.pgadmin.org/docs/pgadmin4/latest/server_dialog.html
S15 / pgAdmin 4: Query Tool
https://www.pgadmin.org/docs/pgadmin4/latest/query_tool.html
S16 / PostgreSQL: Cumulative statistics
https://www.postgresql.org/docs/current/monitoring-stats.html
S17 / Supabase: Row Level Security
https://supabase.com/docs/guides/database/postgres/row-level-security
S18 / Supabase Auth: Redirect URLs
https://supabase.com/docs/guides/auth/redirect-urls

D. Sumber resmi / 3

S19 / Supabase Storage: Access control
https://supabase.com/docs/guides/storage/security/access-control
S20 / Supabase Functions: Environment variables
https://supabase.com/docs/guides/functions/secrets
S21 / Supabase: Database migrations
https://supabase.com/docs/guides/deployment/database-migrations
S22 / Supabase: Production checklist
https://supabase.com/docs/guides/deployment/going-into-prod
S23 / PostgreSQL: SQL Dump
https://www.postgresql.org/docs/current/backup-dump.html
S24 / GitHub: Repository roles for an organization
https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles/repository-roles-for-an-organization
S25 / GitHub: Downloading source code archives
https://docs.github.com/en/repositories/working-with-files/using-files/downloading-source-code-archives
S26 / GitHub: Cloning a repository
https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository
S27 / GitHub: About Git rebase
https://docs.github.com/en/get-started/using-git/about-git-rebase

D. Sumber resmi / 4

S28 / Supabase: Changelog, diperiksa 7 Oktober 2026
https://supabase.com/changelog
S29 / PostgreSQL: System information functions
https://www.postgresql.org/docs/current/functions-info.html
S30 / PostgreSQL: Using EXPLAIN
https://www.postgresql.org/docs/current/using-explain.html
S31 / Supabase: Custom claims and RBAC
https://supabase.com/docs/guides/api/custom-claims-and-role-based-access-control-rbac
S32 / Supabase Realtime: Postgres Changes
https://supabase.com/docs/guides/realtime/postgres-changes
S33 / Supabase: Backup and Restore using CLI
https://supabase.com/docs/guides/platform/migrating-within-supabase/backup-restore
S34 / GitHub: Creating a pull request
https://docs.github.com/en/pull-requests/how-tos/create-pull-requests/creating-a-pull-request
S35 / GitHub: Managing a branch protection rule
https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/managing-a-branch-protection-rule
S36 / GitHub Actions: Quickstart
https://docs.github.com/en/actions/get-started/quickstart

D. Sumber resmi / 5

S37 / GitHub Actions: Using secrets
https://docs.github.com/en/actions/how-tos/write-workflows/choose-what-workflows-do/use-secrets
S38 / GitHub: Removing sensitive data
https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/removing-sensitive-data-from-a-repository
S39 / Cloudflare Pages: Direct Upload
https://developers.cloudflare.com/pages/get-started/direct-upload/
S40 / Cloudflare DNS: Manage DNS records
https://developers.cloudflare.com/dns/manage-dns-records/how-to/create-dns-records/
S41 / Cloudflare Registrar: Domain renewal
https://developers.cloudflare.com/registrar/account-options/renew-domains/
S42 / GitHub Actions: Checkout official source
https://github.com/actions/checkout
S43 / GitHub Actions: Setup Node official source
https://github.com/actions/setup-node
S44 / pgAdmin 4: Backup Dialog
https://www.pgadmin.org/docs/pgadmin4/latest/backup_dialog.html
S45 / pgAdmin 4: Restore Dialog
https://www.pgadmin.org/docs/pgadmin4/latest/restore_dialog.html

D. Sumber resmi / 6

S46 / Supabase Auth: User sessions
https://supabase.com/docs/guides/auth/sessions
S47 / pgAdmin 4: Tabbed Browser dan Dashboard
https://www.pgadmin.org/docs/pgadmin4/latest/tabbed_browser.html
S48 / pgAdmin 4: Preferences dan sampling grafik
https://www.pgadmin.org/docs/pgadmin4/latest/preferences.html
G1 / GitHub organisasi 7-MIT; metadata konektor 7 Oktober 2026
https://github.com/7-MIT

Lampiran pengguna: foto pgAdmin dan screenshot CPU dipakai sebagai konteks. Nilai dalam screenshot tidak membuktikan status layanan terbaru. Tidak ada kredensial produksi dimuat dalam buku.