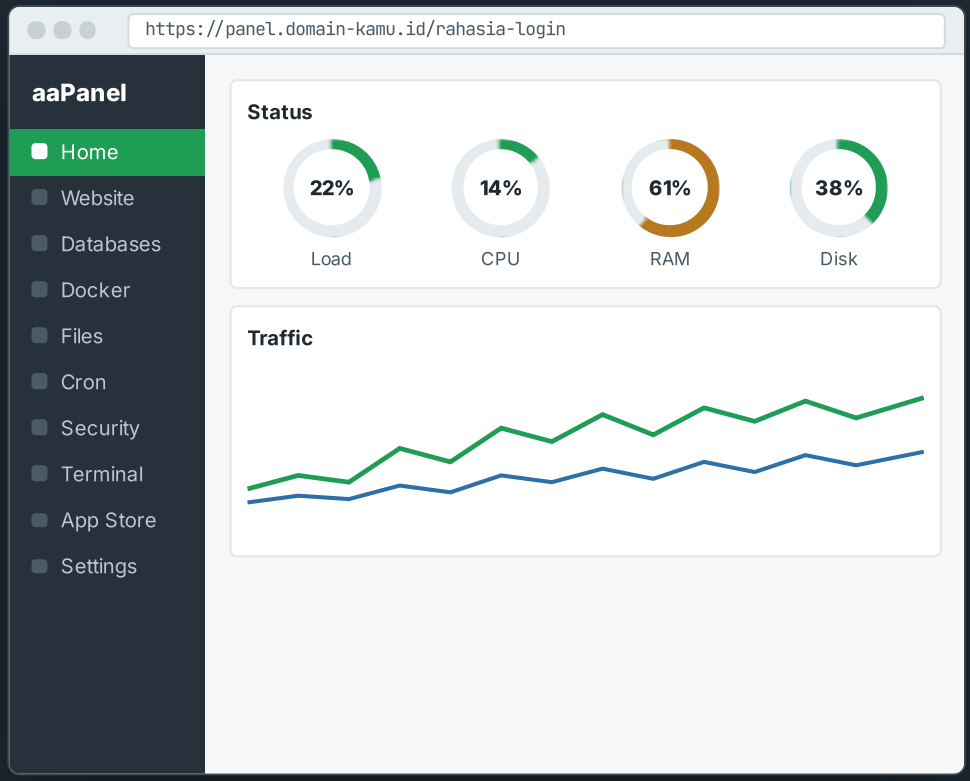
Handbook & Cheatsheet untuk Mahasiswa

aaPanel

dari instalasi sampai deploy aplikasi

Panduan visual untuk mengelola server Linux lewat aaPanel: instalasi, website PHP dan Node.js, database, SSL, cron, firewall, deploy dari GitHub, dan cara membaca error ketika ada yang tidak jalan.



Disusun untuk perkuliahan, administrasi server praktis, dan tugas mata kuliah Testing dan Implementasi Sistem.
Semua tampilan layar di handbook ini adalah ilustrasi. Letak menu dapat sedikit berbeda antar versi aaPanel.

Daftar isi

17 bab, disusun mengikuti urutan kerja nyata: pasang server, kenali panel, buat situs, lalu deploy dan rawat.

1 **Mengenal aaPanel**
Apa itu, kapan dipakai, perbandingan

2 **Instalasi & login pertama**
Syarat server, skrip install, pilih stack

3 **Tur dashboard**
Membaca halaman Home dan menu samping

4 **Struktur folder & izin file**
Peta /www, lokasi log dan konfigurasi

5 **Website**
Add Site, pengaturan situs, SSL

6 **Database**
Membuat DB, phpMyAdmin, impor, backup

7 **File Manager**
Upload, ekstrak, edit, chmod/chown

8 **Cron**
Tugas terjadwal dan format waktu

9 **Security**
Firewall, port, SSH

10 **App Store & Docker**
Memasang software tambahan

11 **Node.js & reverse proxy**
Menjalankan Astro, Next.js, Express

12 **Deploy dari GitHub**
Webhook otomatis dan git pull manual

13 **Backup & monitoring**
Strategi 3-2-1, memantau resource

14 **Keamanan produksi**
Checklist pengerasan panel

15 **Perintah terminal bt**
Cheatsheet CLI aaPanel

16 **Troubleshooting**
Membaca kode error dan solusinya

17 **Latihan & glosarium**
Checklist praktikum dan istilah

Cara membaca handbook ini

Setiap bab berisi penjelasan singkat di kiri dan tampilan layar atau diagram di kanan. Angka merah pada tampilan layar menunjuk ke penjelasan bernomor di sebelahnya.

Info
Konsep atau latar belakang yang perlu dipahami.

Tips
Cara kerja yang lebih cepat atau lebih aman.

Perhatian
Langkah yang sering salah atau berisiko.

Jangan
Kebiasaan yang membuat server rentan.

```
# blok gelap = perintah terminal
sudo bt 14
```

Mengenal aaPanel

aaPanel adalah control panel server berbasis web yang gratis dan open source. Panel ini memberi antarmuka grafis untuk mengelola web server, database, runtime bahasa pemrograman, file, dan keamanan server Linux tanpa harus selalu mengetik perintah manual.

Apa yang sebenarnya dilakukan aaPanel?

aaPanel tidak menggantikan Linux, Nginx, atau MySQL. Panel ini adalah **lapisan pengelola** di atas software tersebut. Ketika kamu menekan tombol "Add Site", aaPanel menulis file konfigurasi Nginx, membuat folder situs, dan me-reload web server untukmu.

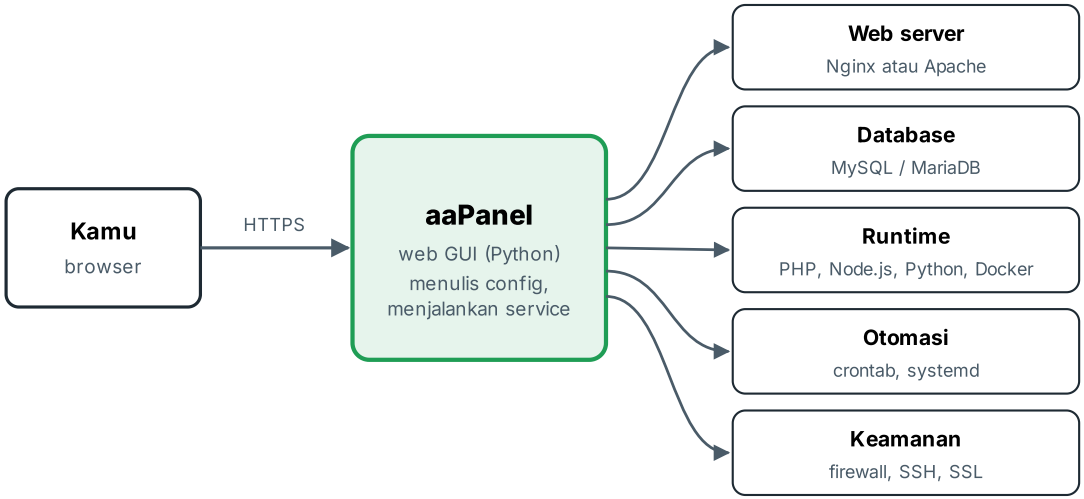
Artinya, semua yang dibuat lewat panel tetap berupa file dan service Linux biasa. Kalau panel bermasalah, situs yang sudah berjalan umumnya tetap hidup.

Kapan cocok memakai aaPanel?

- VPS pribadi untuk tugas kuliah, portofolio, atau staging.
- Server kecil yang menjalankan beberapa situs PHP dan Node.js sekaligus.
- Belajar administrasi server sambil tetap bisa melihat apa yang terjadi di balik layar.

Kapan kurang cocok?

- Infrastruktur besar yang dikelola dengan Kubernetes atau Infrastructure as Code.
- Server dengan RAM di bawah 512 MB.



Aspek	Terminal murni	cPanel	aaPanel
Lisensi	Gratis	Berbayar bulanan	**Gratis, open source**
Pemakaian RAM panel	Hampir nol	Cenderung berat	**Ringan, kira-kira 150–200 MB**
Aplikasi modern	Konfigurasi manual	Fokus PHP, shared hosting	**PHP, Node.js, Python, Docker**
Kemudahan	Butuh skill Linux	Ramah non-teknis	**Ramah developer, modular**
Cocok untuk	Admin berpengalaman	Penyedia hosting	**VPS pribadi dan tim kecil**

Instalasi aaPanel

aaPanel dipasang di server Linux yang masih bersih, biasanya VPS. Proses instalasinya hanya satu perintah, tetapi ada beberapa syarat yang perlu dipastikan dulu supaya tidak gagal di tengah jalan.

Sebelum mulai

Kebutuhan	Minimum	Disarankan
Sistem operasi	Ubuntu 22.04/24.04, Debian 11/12, AlmaLinux/Rocky 8–9	
RAM	512 MB	1–2 GB atau lebih
Disk	10 GB	20 GB atau lebih (SSD)
Akses	User `root` atau user dengan `sudo`	

Gunakan server yang masih bersih
Jangan pasang aaPanel di server yang sudah memiliki Nginx, Apache, PHP, atau MySQL hasil instal manual. Konfigurasinya bisa bentrok.

Langkah instalasi

1. Login ke server lewat SSH: `ssh root@IP_SERVER`
2. Perbarui paket: `apt update && apt upgrade -y`
3. Salin perintah instalasi terbaru dari halaman resmi **aapanel.com**, lalu jalankan.
4. Ketik `y` saat ditanya konfirmasi, tunggu 2–10 menit.
5. Catat alamat panel, username, dan password yang muncul di akhir.
6. Buka port panel di firewall penyedia VPS (security group), lalu akses alamat tersebut di browser.

Lupa mencatat info login?
Jalankan `sudo bt 14` kapan saja untuk menampilkannya lagi.

```
root@vps: ~

# Contoh perintah instalasi (selalu salin versi terbaru dari aapanel.com)
root@vps:~#
 URL=https://www.aapanel.com/script/install_7.0_en.sh && \
  if [ -f /usr/bin/curl ];then curl -ksSO "$URL" ;else \
  wget --no-check-certificate -O install_7.0_en.sh "$URL";fi; \
  bash install_7.0_en.sh aapanel

Do you want to install aaPanel to the /www directory now?(y/n): y
...
==================================================================
Congratulations! Installed successfully!

==================================================================
 aaPanel Internet Address: https://203.0.113.10:xxxxx/a1b2c3d4
 aaPanel Internal Address: https://10.0.0.5:xxxxx/a1b2c3d4
 username: k8x2q9mw
 password: f3a9c1e7
==================================================================
 Warning:
 If you cannot access the panel, release the following port
 (xxxxx|80|443|20|21) in the security group
==================================================================
```

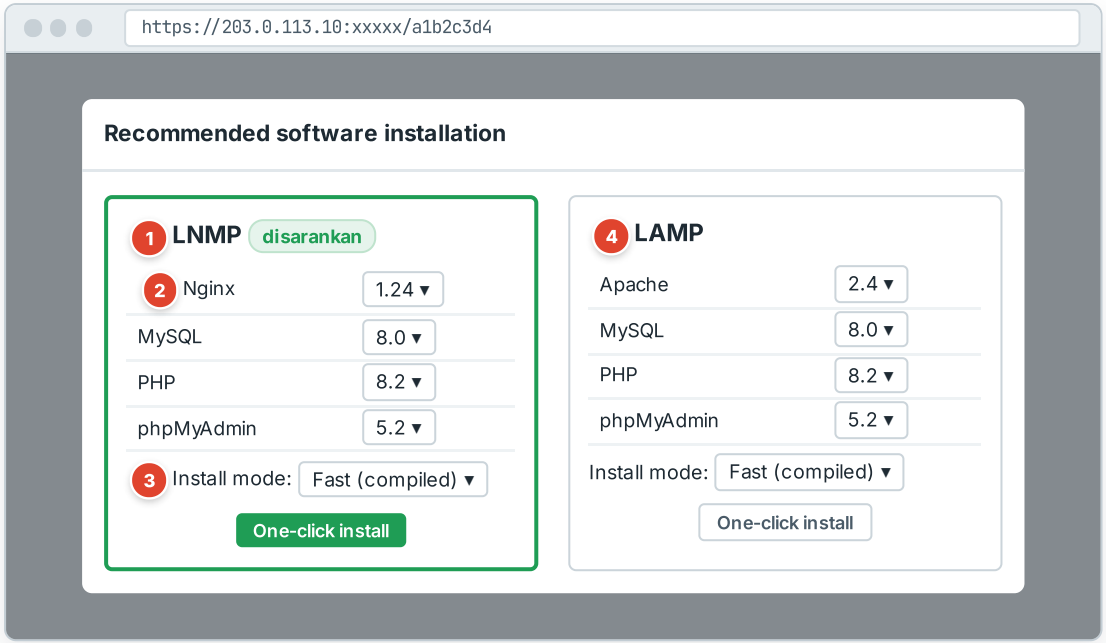
Ilustrasi output akhir instalasi. Port dan path rahasia dibuat acak pada setiap instalasi.

Arti tiap bagian alamat
`https://IP:port/entrance`
port = port panel, **entrance** = path rahasia. Tanpa path ini, halaman login tidak akan terbuka.

Browser memberi peringatan SSL?
Normal pada akses pertama karena panel memakai sertifikat self-signed. Pilih Advanced lalu lanjutkan, kemudian pasang SSL yang benar di Bab 14.

Login pertama & memilih stack

Setelah login pertama kali, aaPanel menawarkan paket software yang akan dipasang. Pilihan ini menentukan web server dan versi PHP serta database yang dipakai server.



Ilustrasi jendela rekomendasi yang muncul saat login pertama.

1. **Pilih LNMP** untuk kebanyakan kebutuhan kuliah. Satu server cukup memakai satu web server.
2. **Versi software.** Pilih versi yang dibutuhkan aplikasimu. Laravel 11 butuh PHP 8.2 ke atas. Versi PHP lain bisa ditambah nanti lewat App Store.
3. **Mode instal.** *Fast* memakai paket siap pakai sehingga lebih cepat. *Compiled* lebih lama tetapi dioptimalkan untuk server.
4. **LAMP** hanya bila aplikasi benar-benar butuh Apache.

> **RAM 1 GB atau kurang?**
> Pertimbangkan MariaDB atau MySQL 5.7 karena MySQL 8 cukup boros memori. Tambahkan juga swap 1–2 GB.

```
# membuat swap 2 GB (opsional, untuk VPS kecil)
sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile && sudo swapon /swapfile
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

Nginx atau Apache?

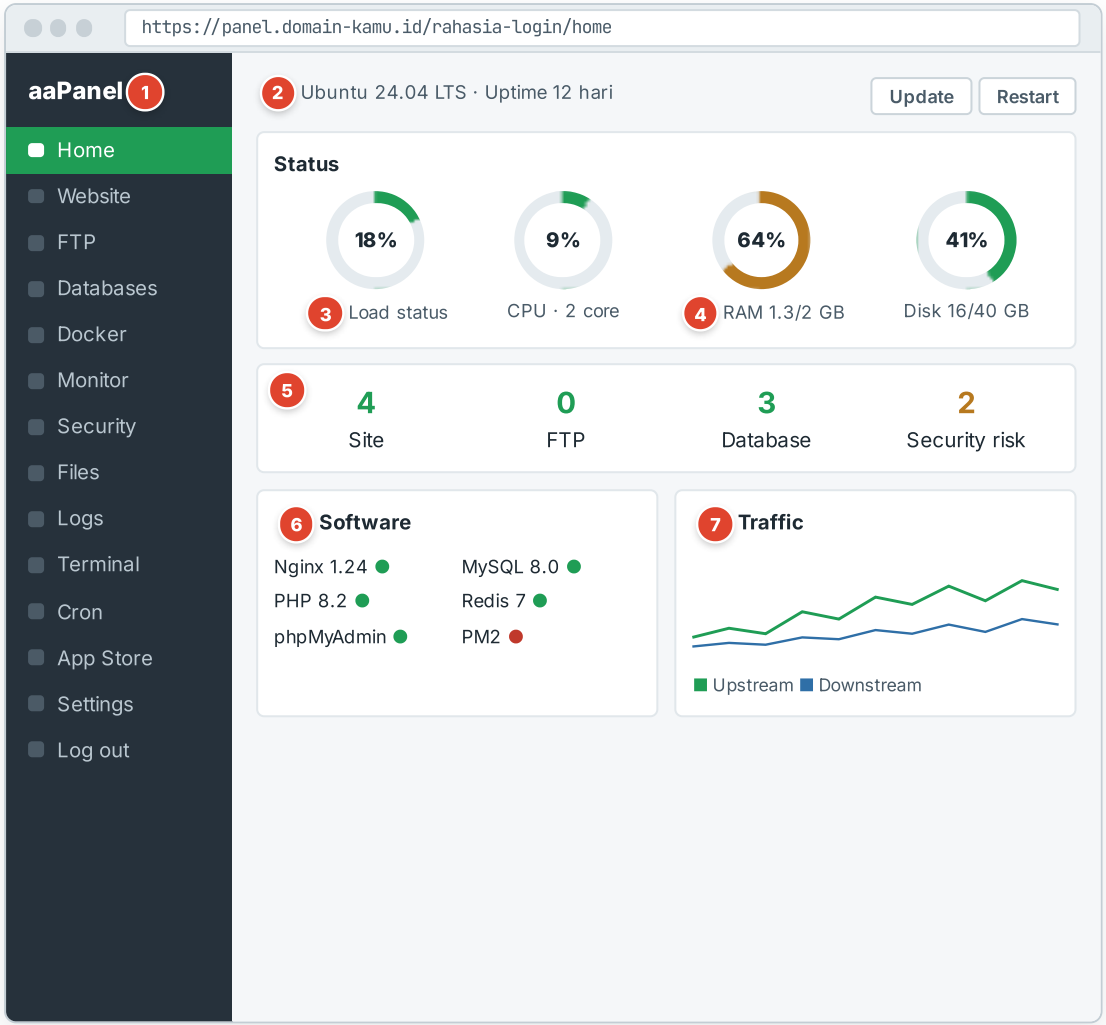
	Nginx (LNMP)	Apache (LAMP)
Kelebihan	Hemat RAM, cepat untuk file statis, umum dipakai sebagai reverse proxy Node.js	Mendukung `.htaccess`, cocok untuk aplikasi PHP lama
Pilih jika	VPS kecil, aplikasi Node.js atau Laravel modern	Aplikasi bawaan yang bergantung pada .htaccess

Langkah wajib setelah login pertama

- [ ] Ganti username dan password panel
- [ ] Aktifkan Google Authenticator (2FA)
- [ ] Bind domain panel dan pasang SSL
- [ ] Cek port terbuka di menu Security

Tur dashboard

Halaman Home adalah tempat pertama untuk mengecek kesehatan server. Biasakan membukanya setiap kali login untuk melihat apakah CPU, RAM, dan disk masih aman.



Ilustrasi halaman Home. Tata letak dapat sedikit berbeda antar versi.

1. **Menu samping.** Semua fitur ada di sini. Bab-bab selanjutnya mengikuti menu ini.
2. **Info sistem.** Versi OS, uptime, tombol update panel dan restart server.
3. **Load status.** Beban antrean proses. Hijau aman, kuning mulai sibuk, merah berarti server kewalahan.
4. **RAM dan disk.** Dua angka yang paling sering menyebabkan masalah. Disk penuh membuat MySQL dan panel berhenti.
5. **Ringkasan.** Jumlah situs, database, dan peringatan keamanan. Klik "Security risk" untuk melihat saran perbaikan.
6. **Software terpasang.** Titik hijau berarti service berjalan, merah berarti berhenti. Klik untuk membuka pengaturan atau restart.
7. **Traffic.** Lalu lintas jaringan masuk dan keluar. Lonjakan mendadak bisa menandakan serangan atau bot.

Patokan angka sehat

Indikator	aman	waspada	bahaya
CPU	< 60%	60–85%	> 85% terus-menerus
RAM	< 70%	70–90%	> 90%, swap penuh
Disk	< 70%	70–85%	> 90%

Struktur folder & izin file

Semua komponen aaPanel tinggal di direktori /www. Mengenal peta foldernya membuatmu tahu ke mana harus mencari ketika perlu melihat log, mengedit konfigurasi, atau memulihkan backup.

```
/www/
├── server/
│   ├── panel/            inti aaPanel (Python)
│   │   ├── data/          database internal panel
│   │   ├── vhost/         config situs Nginx/Apache
│   │   │   └── nginx/domain.conf
│   │   ├── logs/          error.log panel
│   │   └── pyenv/         Python bawaan panel
│   ├── nginx/ atau apache/ binary & nginx.conf
│   ├── php/
│   │   ├── 74/  81/  82/   tiap versi PHP
│   │   │   └── etc/php.ini
│   ├── data/             file mentah MySQL
│   └── nodejs/           versi Node.js
├── wwwroot/              folder situs
│   └── domain-kamu.id/
├── wwwlogs/              log akses & error situs
│   ├── domain-kamu.id.log
│   └── domain-kamu.id.error.log
└── backup/               hasil backup lokal
    ├── site/
    └── database/
```

Prinsip penting

Konfigurasi situs sebaiknya diubah lewat panel. Kalau diedit manual di `vhost/`, cek dulu dengan `nginx -t` sebelum reload.

Jangan pakai chmod 777

Izin 777 membuat siapa pun di server bisa mengubah file, dan sering dimanfaatkan untuk menanam malware.

Lokasi yang paling sering dicari

Saya ingin...	Lokasinya
Melihat error situs	`/www/wwwlogs/DOMAIN.error.log`
Melihat error panel	`/www/server/panel/logs/error.log`
Mengubah config Nginx situs	`/www/server/panel/vhost/nginx/DOMAIN.conf`
Mengubah php.ini	`/www/server/php/82/etc/php.ini`
Mengunggah kode situs	`/www/wwwroot/DOMAIN/`
Mencari file backup	`/www/backup/`

Izin file (permission)

Situs dijalankan oleh user `www`. Jika kepemilikan file salah, aplikasi tidak bisa menulis log, cache, atau file unggahan.

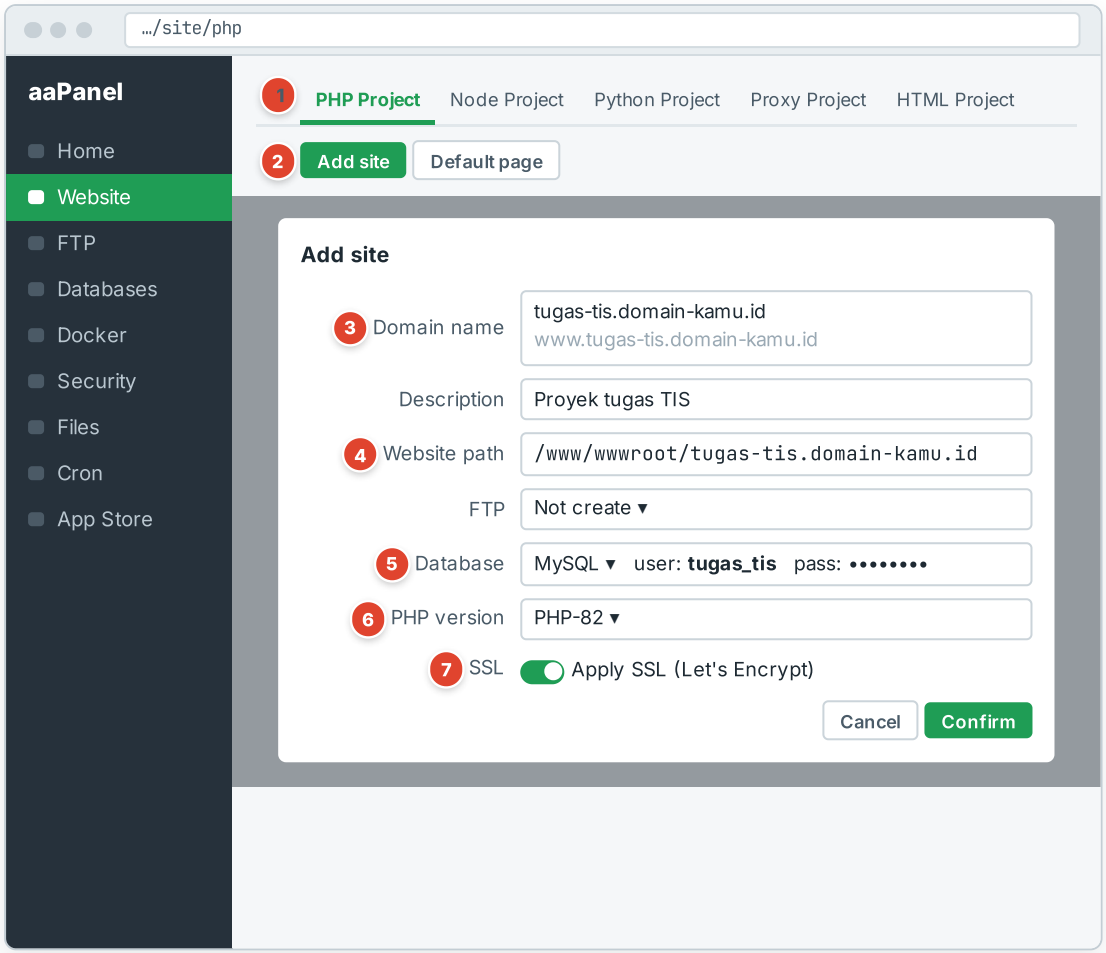
Owner	Group	Others
r w x	r - x	r - x
4+2+1 = **7**	4+0+1 = **5**	4+0+1 = **5**

Objek	Izin	Pemilik
Folder	`755`	`www:www`
File biasa	`644`	`www:www`
`.env`, config rahasia	`600` atau `640`	`www:www`

```
# memperbaiki kepemilikan dan izin satu situs
sudo chown -R www:www /www/wwwroot/domain-kamu.id
sudo find /www/wwwroot/domain-kamu.id -type d -exec chmod 755 {} \;
sudo find /www/wwwroot/domain-kamu.id -type f -exec chmod 644 {} \;
```

Website: menambah situs

Menu Website dipakai untuk membuat situs baru dan mengatur semua hal yang berhubungan dengan domain. Satu server bisa menampung banyak situs, masing-masing dengan folder dan konfigurasi sendiri.



Ilustrasi dialog Add site pada tab PHP Project.

1. **Jenis proyek.** PHP untuk WordPress, Laravel, CodeIgniter. Node untuk Astro, Next.js, Express. HTML untuk situs statis.
2. **Add site** membuka dialog pembuatan situs.
3. **Domain.** Satu domain per baris. Domain harus sudah diarahkan (record A) ke IP server.
4. **Path.** Folder kode situs, otomatis di `/www/wwwroot/`.
5. **Database.** Bisa dibuat sekalian. Catat username dan password-nya.
6. **Versi PHP** per situs. Situs lama dan baru bisa memakai versi berbeda.
7. **SSL langsung** hanya berhasil jika DNS sudah mengarah ke server.

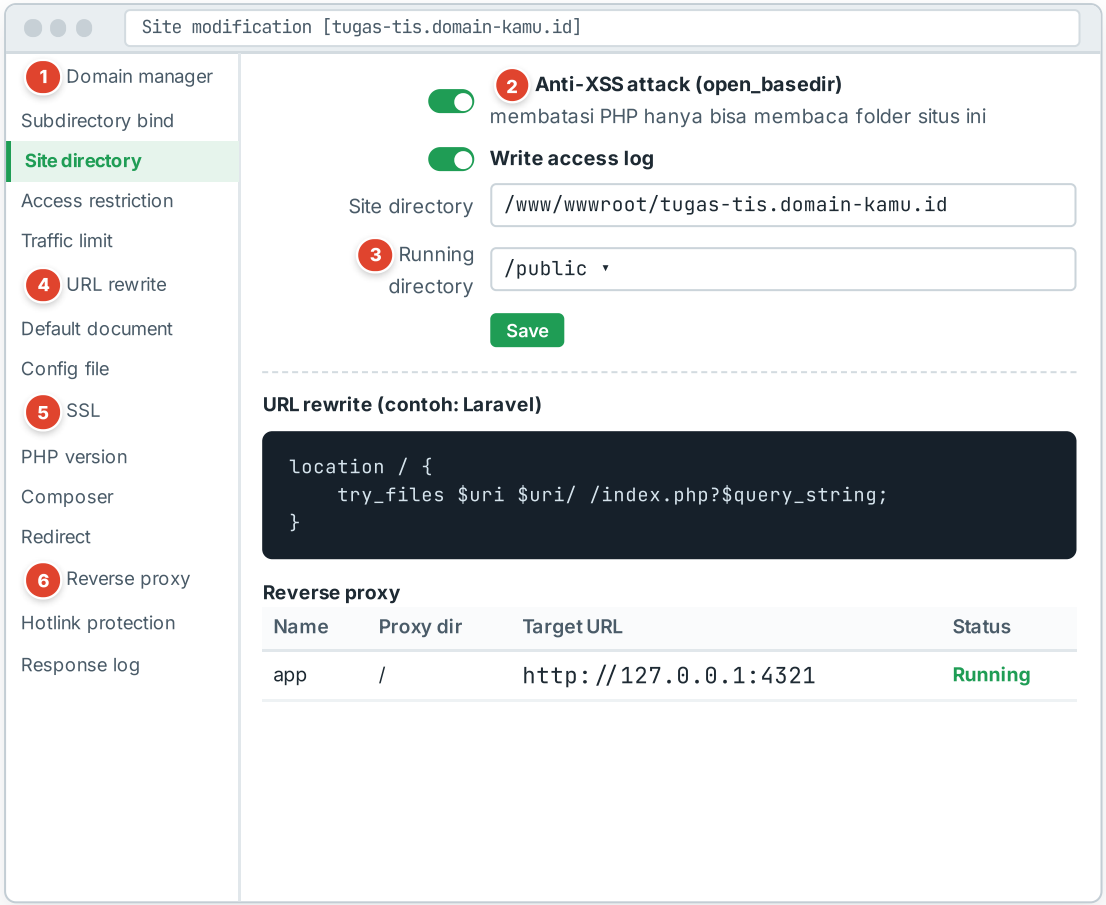
Sebelum Add site: arahkan DNS

Type	Name	Content
A	`tugas-tis`	`203.0.113.10`
A	`www.tugas-tis`	`203.0.113.10`

```
# cek apakah DNS sudah mengarah ke server
host tugas-tis.domain-kamu.id 1.1.1.1
```

Website: pengaturan situs

Klik nama situs atau tombol Conf untuk membuka jendela pengaturan. Hampir semua kebutuhan situs diatur dari sini tanpa perlu menyentuh file konfigurasi secara langsung.



Ilustrasi jendela pengaturan situs (tab Site directory dengan contoh isi tab lain).

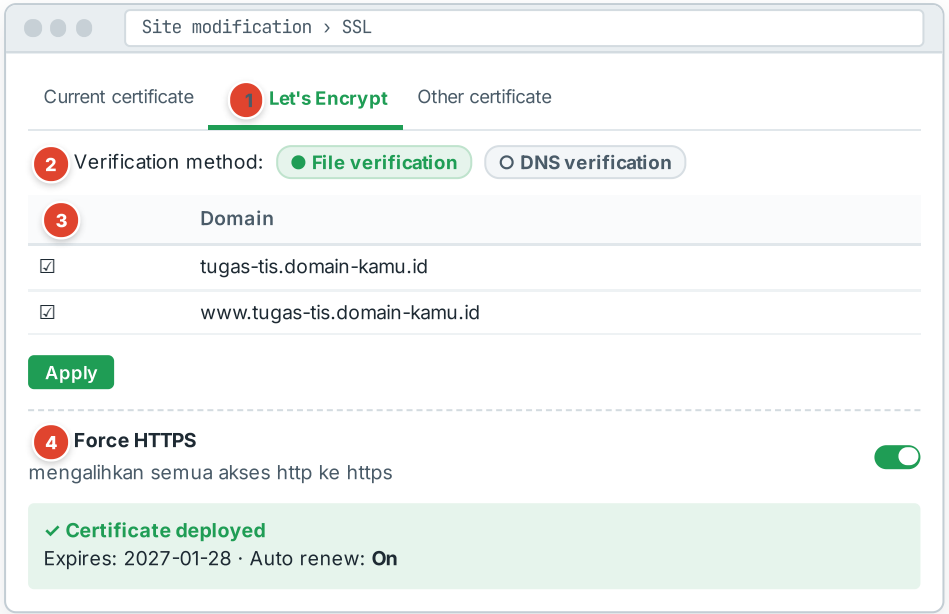
1. **Domain manager.** Menambah alias domain, misalnya versi www, ke situs yang sama.
2. **Anti-XSS (open_basedir).** Mengunci PHP agar tidak bisa membaca folder situs lain. Biarkan aktif.
3. **Running directory.** Untuk Laravel wajib diarahkan ke `/public`, agar file `.env` tidak bisa diakses dari browser.
4. **URL rewrite.** Aturan routing. Tersedia template siap pakai untuk Laravel, WordPress, CodeIgniter, dan lainnya.
5. **SSL.** Memasang sertifikat Let's Encrypt gratis. Dibahas di halaman berikutnya.
6. **Reverse proxy.** Meneruskan domain ke aplikasi yang berjalan di port lokal. Dibahas di Bab 11.

Tab lain yang berguna

Tab	Fungsi
Redirect	Mengalihkan domain lama ke domain baru (301)
Access restriction	Memasang password pada folder, misalnya halaman admin
Config file	Melihat dan mengedit config Nginx situs
Response log	Melihat log akses dan error situs dari panel
Composer	Menjalankan `composer install` untuk proyek PHP

SSL & HTTPS

Sertifikat SSL membuat situs bisa diakses lewat HTTPS. aaPanel memasang sertifikat Let's Encrypt gratis yang berlaku 90 hari dan diperpanjang otomatis sebelum habis.



Ilustrasi tab SSL.

1. **Let's Encrypt** untuk sertifikat gratis. *Other certificate* untuk sertifikat berbayar.
2. **File verification** paling mudah, tetapi port 80 harus terbuka dan DNS harus mengarah langsung ke server.
3. Centang semua domain yang ingin dilindungi, lalu **Apply**.
4. Aktifkan **Force HTTPS** setelah sertifikat berhasil terbit.

Memakai Cloudflare?

Saat pertama kali membuat sertifikat dengan file verification, matikan proxy Cloudflare supaya server Let's Encrypt bisa menghubungi IP server secara langsung.

DNS only (awan abu-abu)
Pengunjung langsung ke IP server. **Pakai saat generate SSL pertama.**

Proxied (awan oranye)
Lalu lintas lewat Cloudflare. Boleh diaktifkan kembali setelah SSL terbit.

1. DNS only Set awan abu-abu › **2. Apply SSL** di aaPanel › **3. Proxied** Awan oranye lagi › **4. Full (strict)** Mode SSL di Cloudflare

Gunakan mode Full (strict)
Mode *Flexible* di Cloudflare membuat koneksi ke server tetap HTTP dan sering menyebabkan redirect loop jika Force HTTPS aktif.

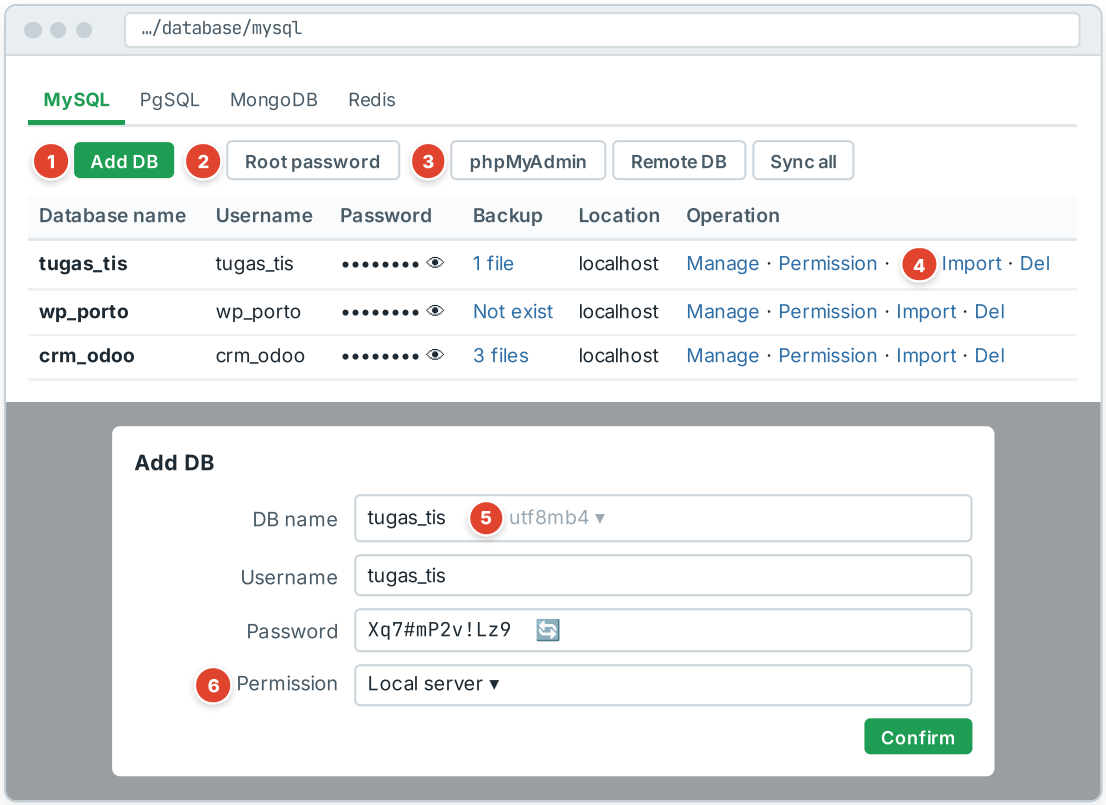
Alternatif: DNS verification
Jika port 80 tidak bisa dibuka atau ingin sertifikat wildcard `*.domain.id`, pilih DNS verification dan hubungkan API Cloudflare.

Penyebab umum SSL gagal

- DNS belum mengarah ke server atau belum selesai propagasi.
- Port 80 tertutup di firewall atau security group.
- Proxy Cloudflare masih aktif saat verifikasi file.
- Terlalu sering mencoba sehingga terkena batas percobaan Let's Encrypt. Tunggu sekitar satu jam.

Database

Menu Databases membuat database MySQL atau MariaDB beserta user dan password-nya dalam sekali klik. Dari sini kamu juga bisa membuka phpMyAdmin, mengimpor file SQL, dan membuat backup.



Ilustrasi menu Databases dan dialog Add DB.

1. **Add DB.** Satu aplikasi sebaiknya punya database dan user sendiri.
2. **Root password.** Mengganti atau mereset password root MySQL jika lupa.
3. **phpMyAdmin.** Melihat tabel, menjalankan query, ekspor dan impor lewat browser.
4. **Import.** Unggah `.sql` atau `.sql.gz` untuk memulihkan data.
5. **Charset utf8mb4** mendukung semua karakter, termasuk emoji.
6. **Permission.** *Local server* berarti hanya bisa diakses dari server itu sendiri. Ini paling aman.

Menghubungkan aplikasi ke database

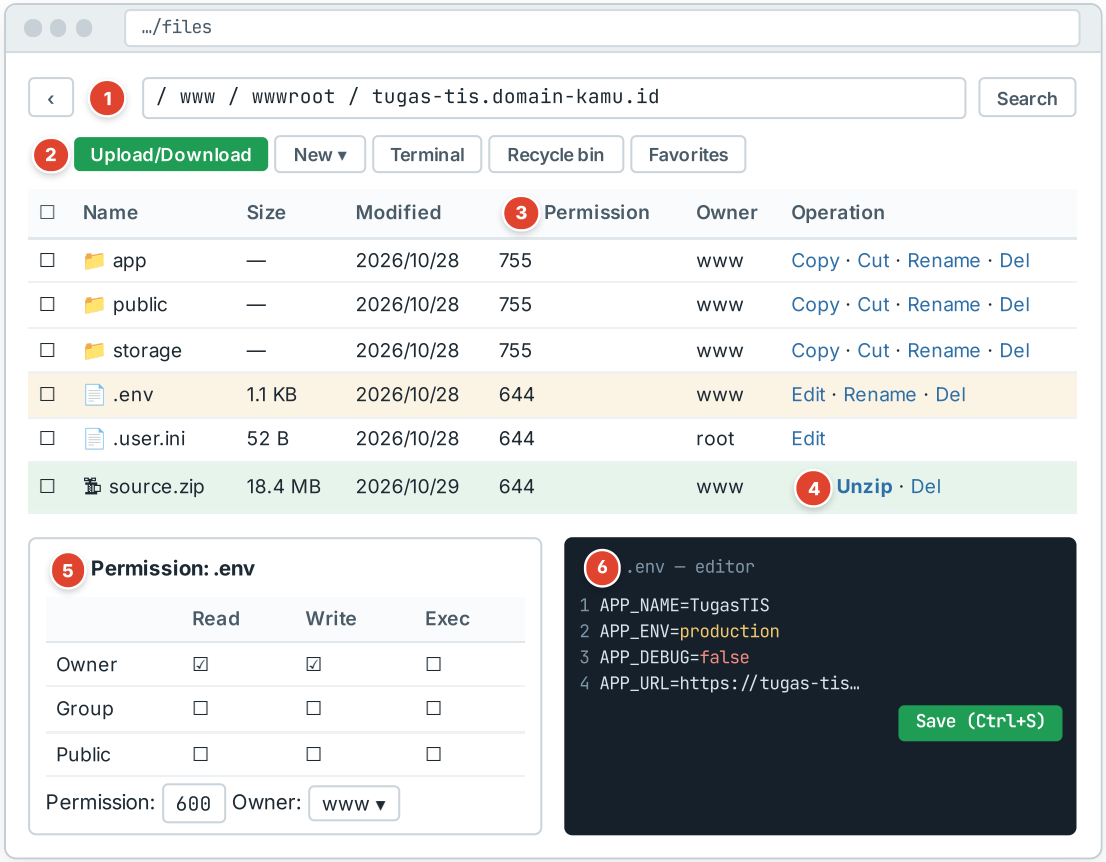
```
# contoh .env Laravel
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=tugas_tis
DB_USERNAME=tugas_tis
DB_PASSWORD=Xq7#mP2v!Lz9
```

Butuh akses dari laptop?
Jangan buka port 3306 ke publik. Gunakan SSH tunnel: `ssh -L 3307:127.0.0.1:3306 root@IP`, lalu sambungkan aplikasi desktop ke `127.0.0.1:3307`.

File SQL besar?
Unggah lewat File Manager ke `/www/backup/database`, lalu impor dari terminal: `mysql -u tugas_tis -p tugas_tis < data.sql`

File Manager

File Manager menggantikan FTP dan SFTP langsung dari browser. Kamu bisa mengunggah, mengekstrak arsip, mengedit kode, dan mengatur izin file tanpa membuka terminal.



Ilustrasi File Manager pada folder sebuah situs Laravel.

1. **Alamat folder.** Klik bagian path untuk berpindah folder dengan cepat.
2. **Upload.** Untuk banyak file, kompres dulu jadi `.zip` di laptop. Satu file besar jauh lebih cepat daripada ribuan file kecil.
3. **Permission dan Owner.** Owner `www` dengan izin 755/644 adalah kondisi normal.
4. **Unzip** mengekstrak arsip langsung di server.
5. **Izin per file.** Klik angka izin untuk mengubahnya. `.env` cukup 600.
6. **Editor kode** dengan syntax highlighting. Simpan dengan Ctrl+S.

Tidak bisa menghapus folder situs?

File `.user.ini` dikunci (immutable) oleh fitur Anti-XSS. Matikan fitur itu di pengaturan situs, atau buka kuncinya: `chattr -i .user.ini`

Recycle bin

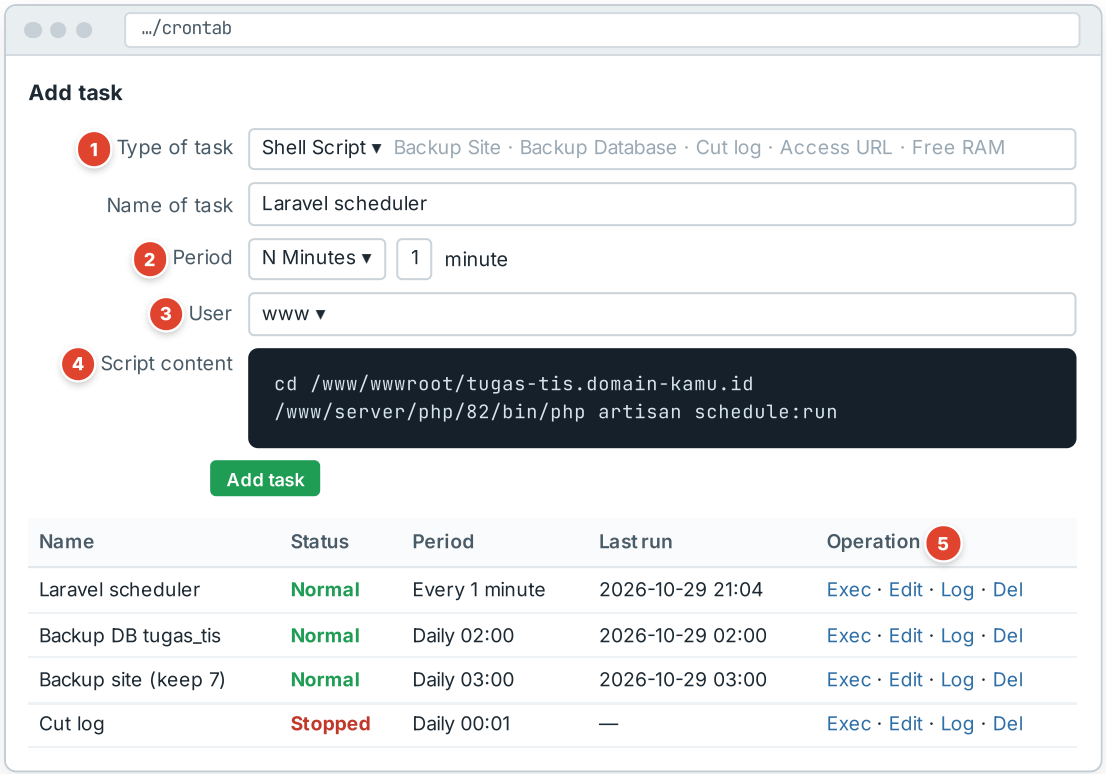
File yang dihapus masuk ke recycle bin dulu. Kosongkan berkala agar disk tidak penuh.

Jangan unggah folder node_modules atau vendor

Unggah kode sumber saja, lalu jalankan `npm install` atau `composer install` di server.

Cron: tugas terjadwal

Cron menjalankan pekerjaan secara otomatis pada waktu tertentu, misalnya backup setiap malam atau menjalankan scheduler aplikasi setiap menit. aaPanel membuatnya cukup lewat formulir, tanpa harus menulis crontab.



Ilustrasi formulir Add task dan daftar tugas.

1. **Jenis tugas.** Shell Script untuk perintah bebas, Backup Site/Database untuk backup otomatis, Access URL untuk memanggil webhook.
2. **Periode.** Pilih per menit, per jam, harian, mingguan, atau bulanan.
3. **User.** Jalankan sebagai `www` untuk tugas aplikasi agar file yang dibuat tidak dimiliki root.
4. **Path lengkap.** Cron tidak mengenal PATH seperti terminal, jadi tulis path PHP atau Node secara lengkap.
5. **Log.** Klik untuk melihat output terakhir. Selalu cek setelah membuat tugas baru.

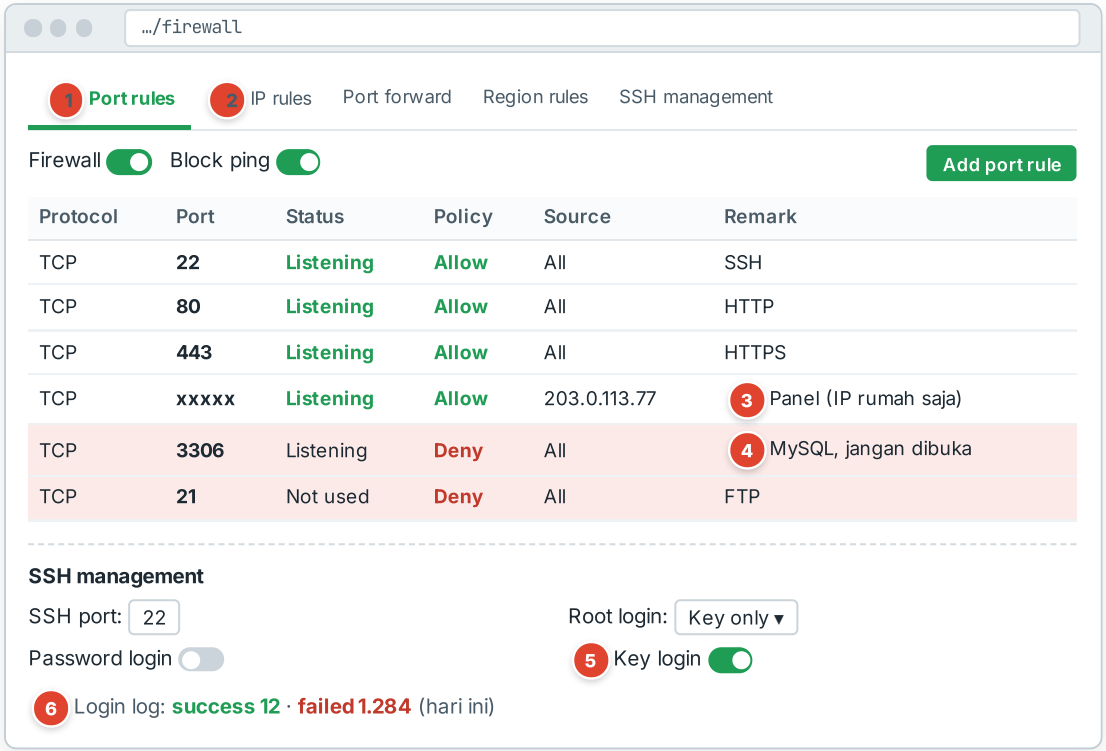
Membaca format cron

```
┌ menit (0-59)
│ ┌ jam (0-23)
│ │ ┌ tanggal (1-31)
│ │ │ ┌ bulan (1-12)
│ │ │ │ ┌ hari (0-6, 0=Minggu)
* * * * *  perintah
```

Ekspresi	Artinya
`* * * * *`	Setiap menit
`*/15 * * * *`	Setiap 15 menit
`0 2 * * *`	Setiap hari pukul 02.00
`0 3 * * 0`	Setiap Minggu pukul 03.00
`0 0 1 * *`	Setiap tanggal 1 tengah malam

Security: firewall & SSH

Firewall menentukan port mana yang boleh diakses dari internet. Prinsipnya sederhana: buka hanya port yang benar-benar dipakai, tutup sisanya.



Ilustrasi menu Security.

1. **Port rules.** Daftar port yang dibuka atau ditutup.
2. **IP rules.** Memblokir IP penyerang, atau mengizinkan IP tertentu saja.
3. **Port panel** sebaiknya hanya diizinkan dari IP kamu.
4. **Port database** tidak perlu dibuka ke internet.
5. **Login SSH dengan key** jauh lebih aman daripada password.
6. **Login log.** Ribuan percobaan gagal itu normal, tanda bot sedang mencoba. Itulah alasan login password sebaiknya dimatikan.

> **Ada dua lapis firewall**
> Firewall aaPanel berjalan di dalam server. Penyedia VPS (AWS, GCP, Biznet, dll.) biasanya punya **security group** sendiri. Port harus dibuka di keduanya.

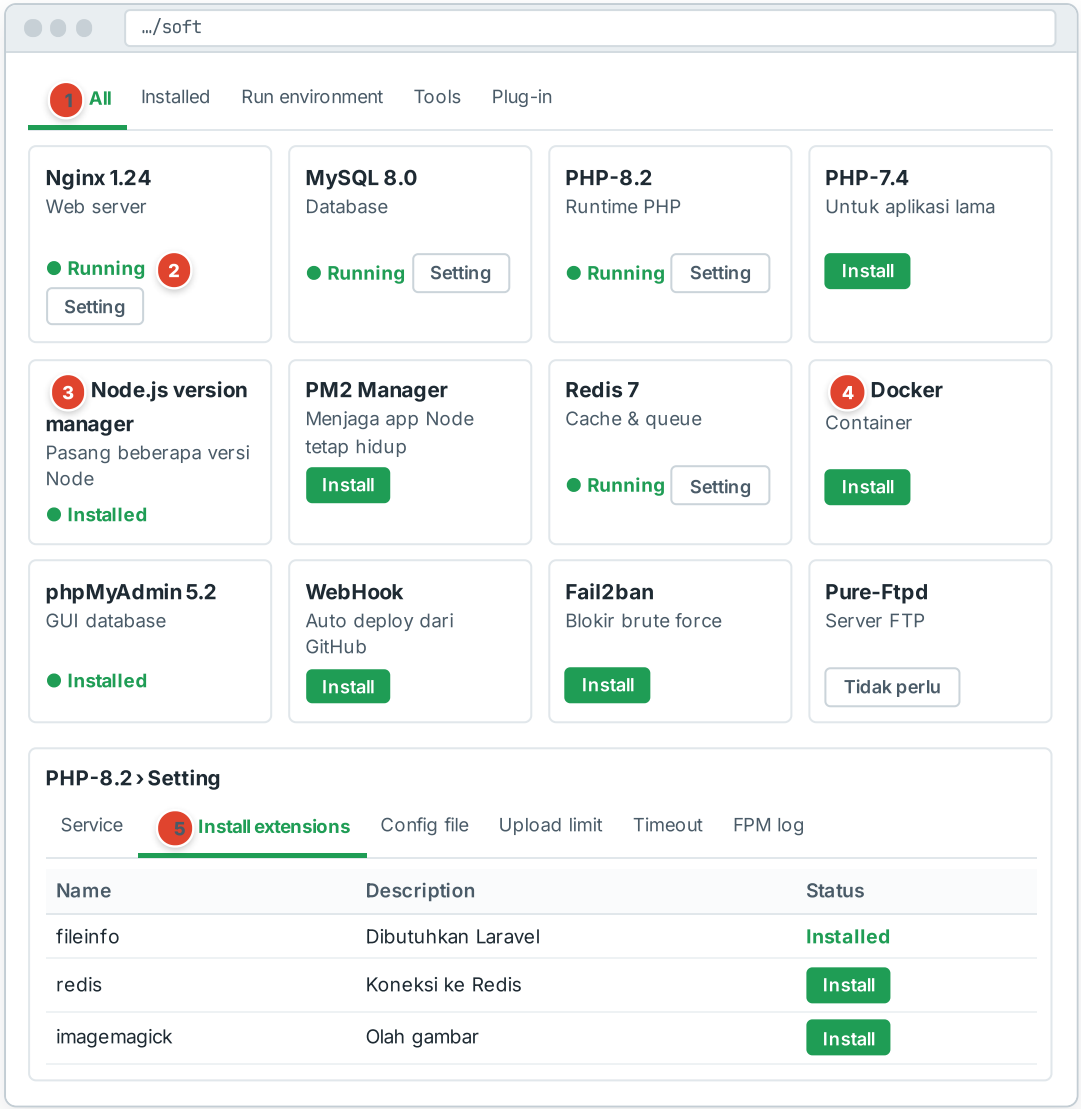
Membuat login SSH dengan key

```
# di laptop
ssh-keygen -t ed25519 -C "laptop-kamu"
ssh-copy-id root@203.0.113.10
# uji login dulu, baru matikan password login
ssh root@203.0.113.10
```

> **Jangan matikan password login sebelum key terbukti bisa dipakai**
> Kalau tidak, kamu bisa terkunci di luar server sendiri.

App Store & Docker

App Store adalah tempat memasang software tambahan seperti versi PHP lain, Redis, Node.js, PM2, atau Docker. Pasang seperlunya saja, karena setiap software memakan RAM.



Ilustrasi App Store. Nama dan versi software dapat berbeda.

1. **Filter.** Tab Installed menampilkan apa saja yang sudah terpasang.
2. **Setting** pada software yang sudah terpasang membuka konfigurasi, log, dan tombol restart.
3. **Node.js dan PM2** dibutuhkan untuk aplikasi Node (Bab 11).
4. **Docker** untuk menjalankan aplikasi dalam container, misalnya Odoo atau PostgreSQL.
5. **Ekstensi PHP.** Jika aplikasi error "Class not found" atau "extension missing", pasang ekstensinya di sini.

Contoh: menjalankan Odoo dengan Docker

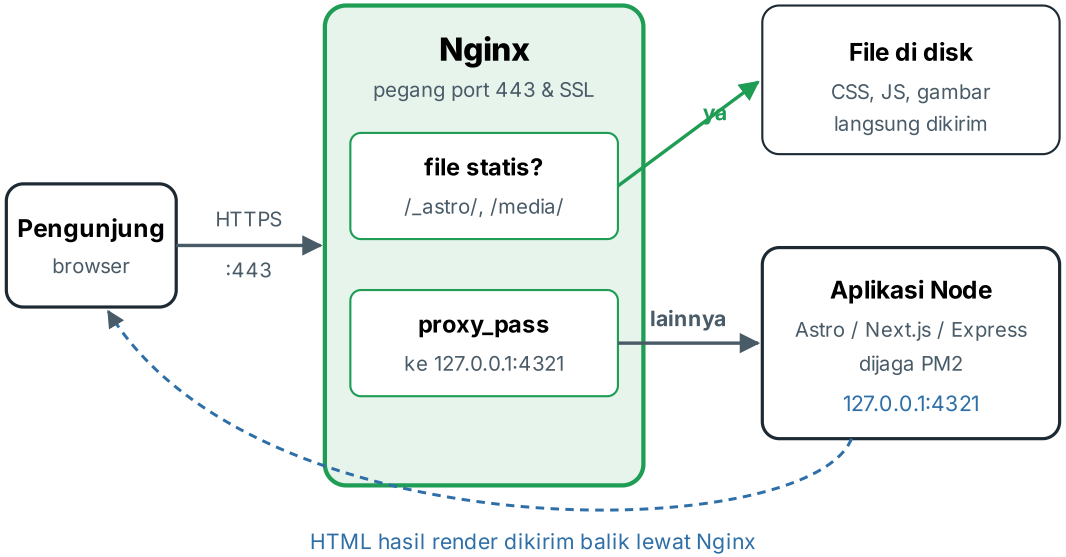
```
docker run -d --name db -e POSTGRES_USER=odoo \
  -e POSTGRES_PASSWORD=odoo -e POSTGRES_DB=postgres postgres:15
docker run -d --name odoo -p 127.0.0.1:8069:8069 \
  --link db:db odoo:17
```

Port diikat ke `127.0.0.1` agar tidak terbuka langsung ke internet. Akses publiknya lewat reverse proxy di situs aaPanel, ke `http://127.0.0.1:8069`.

> **Batas upload PHP**
> Ubah di tab *Upload limit* pada setting PHP. Untuk Nginx, naikkan juga `client_max_body_size` pada config situs.

Node.js & reverse proxy

Aplikasi seperti Astro, Next.js, Express, atau Flask tidak dijalankan langsung oleh Nginx seperti file PHP. Mereka berjalan sebagai proses server sendiri di port lokal, lalu Nginx meneruskan pengunjung ke port tersebut. Cara ini disebut reverse proxy.



Satu port publik, banyak aplikasi
App di port 3000, 3001, 4321 cukup diakses lewat 443 dengan domain berbeda.

File statis lebih cepat
Gambar, CSS, dan JS dilayani Nginx tanpa membebani Node.

SSL terpusat
Sertifikat dipasang di Nginx, bukan di dalam kode aplikasi.

Isi config yang dibuat panel

Saat kamu menambah reverse proxy, aaPanel menulis blok seperti ini ke config situs:

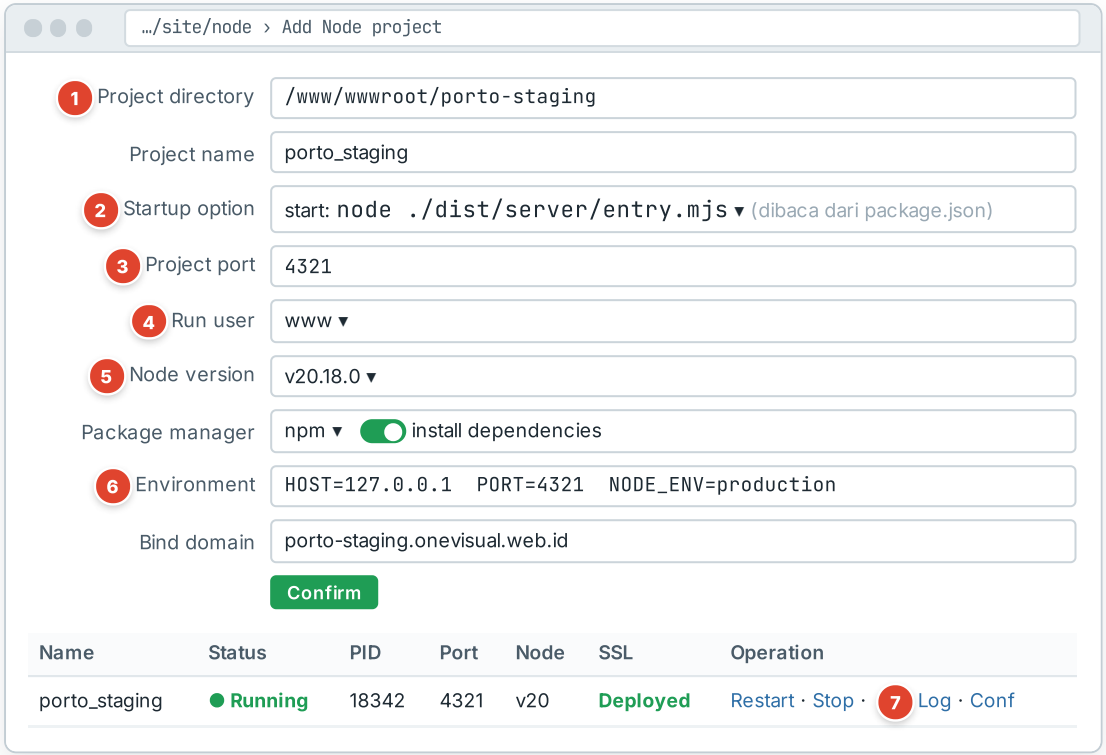
```
location / {
    proxy_pass http://127.0.0.1:4321;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}
```

Baris	Gunanya
`proxy_pass`	Alamat aplikasi tujuan
`Host`, `X-Forwarded-*`	Memberi tahu aplikasi domain asli, IP pengunjung, dan bahwa aksesnya HTTPS
`Upgrade`, `Connection`	Diperlukan untuk WebSocket, misalnya fitur realtime atau hot reload

Ikat aplikasi ke 127.0.0.1
Jalankan app di `127.0.0.1` atau `localhost`, bukan `0.0.0.0`, dan jangan buka port 4321 di firewall. Dengan begitu, satu-satunya pintu masuk adalah Nginx.

Menjalankan proyek Node

Menu Website › Node Project membuat aaPanel menjalankan, me-restart, dan menjaga aplikasi Node tetap hidup, sekaligus memasang reverse proxy ke domainnya.



Ilustrasi formulir Node Project. Nama kolom dapat berbeda antar versi.

Perintah start per framework

Framework	Build	Start
Astro (SSR, adapter node)	`npm run build`	`node ./dist/server/entry.mjs`
Next.js	`npm run build`	`npm run start -- -p 3000`
Express	tidak perlu	`node server.js`
Astro (statis)	`npm run build`	Tidak perlu Node, arahkan situs HTML ke `dist/`

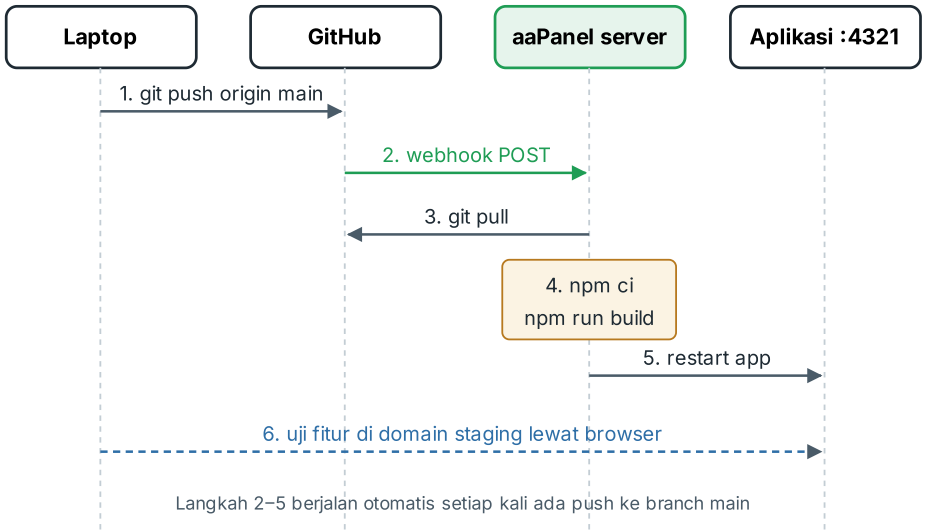
1. **Folder proyek** berisi `package.json`.
2. **Startup option** diambil dari bagian `scripts` di package.json, atau tulis perintah sendiri.
3. **Port** harus sama dengan port yang dipakai aplikasi.
4. **Run user** `www`, bukan root.
5. **Versi Node** sesuaikan dengan kebutuhan framework. Astro dan Next.js terbaru butuh Node 18.17 ke atas, lebih aman Node 20.
6. **Environment.** HOST dan PORT dibaca oleh adapter Astro.
7. **Log** adalah tempat pertama yang dicek jika status Stopped atau muncul 502.

> **Alternatif tanpa menu Node Project**
>
> Jalankan dengan PM2 lalu tambahkan reverse proxy secara manual di pengaturan situs.

```
cd /www/wwwroot/porto-staging
npm ci && npm run build
HOST=127.0.0.1 PORT=4321 pm2 start dist/server/entry.mjs \
  --name porto-staging
pm2 save && pm2 startup   # hidup lagi setelah reboot
pm2 logs porto-staging    # lihat log
```

Deploy dari GitHub

Alur kerja yang rapi: kode diubah di laptop, didorong ke GitHub, lalu server menarik versi terbaru dan menjalankan ulang aplikasi. Bab ini membahas cara otomatis lewat webhook dan cara manual lewat git pull.



Cara 2: git pull manual

Cocok saat ingin kontrol penuh, misalnya sebelum demo.

```
cd /www/wwwroot/porto-staging
git pull origin main
npm ci && npm run build
pm2 restart porto-staging    # atau Restart di menu Node Project
```

Cara 1: webhook otomatis

1. Hubungkan server ke repo dengan **deploy key** (lihat di bawah), lalu `git clone` ke `/www/wwwroot/nama-proyek`.
2. App Store › pasang **WebHook** › Add, lalu isi skrip deploy.
3. Salin URL webhook yang diberikan aaPanel.
4. GitHub › repo › **Settings › Webhooks › Add webhook**, tempel URL, pilih event *Just the push event*.
5. Lakukan push kecil, lalu cek log WebHook di aaPanel.

```
#!/bin/bash
set -e                                  # berhenti jika ada langkah gagal
cd /www/wwwroot/porto-staging
git pull origin main
npm ci                                  # butuh devDependencies untuk build
npm run build
pm2 restart porto-staging
echo "Deploy selesai: $(date)"
```

Koreksi dari versi lama handbook

`npm install --production` sebelum `npm run build` biasanya gagal, karena alat build ada di devDependencies. Gunakan `npm ci`. Skrip WebHook juga sudah berjalan sebagai root, jadi tidak perlu `sudo`.

Deploy key: akses repo private tanpa password

```
ssh-keygen -t ed25519 -f ~/.ssh/deploy_porto -N ""
cat ~/.ssh/deploy_porto.pub   # tempel ke GitHub › Settings › Deploy keys
GIT_SSH_COMMAND="ssh -i ~/.ssh/deploy_porto" \
  git clone git@github.com:user/porto.git /www/wwwroot/porto-staging
```

Simpan .env di server, bukan di GitHub

Tambahkan `.env` ke `.gitignore`. Buat file-nya sekali lewat File Manager.

Backup & monitoring

Server bisa rusak karena salah perintah, update gagal, atau serangan. Backup yang teratur dan pemantauan resource adalah dua kebiasaan yang membedakan server coba-coba dengan server yang siap dipakai.

Aturan 3-2-1

3	2	1
salinan data	**media berbeda**	**di luar server**
data asli + 2 backup	disk server + cloud storage	Google Drive, S3, atau laptop

Jadwal backup yang disarankan

Apa	Kapan	Simpan	Lewat
Database	Setiap hari 02.00	7 salinan terakhir	Cron › Backup Database
File situs	Setiap hari 03.00	7 salinan terakhir	Cron › Backup Site
Kode sumber	Setiap commit	Seluruh riwayat	GitHub
Snapshot VPS	Sebelum update besar	1–2 snapshot	Dashboard penyedia VPS

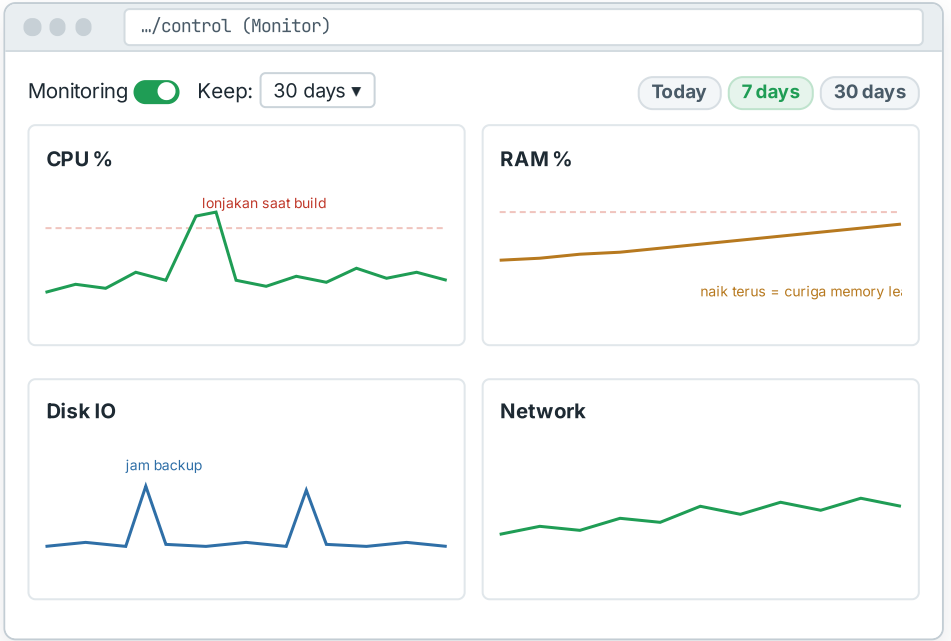
Simpan ke cloud

Di App Store tersedia plugin penyimpanan seperti Google Drive dan S3. Setelah terpasang, pilihannya muncul di kolom *Backup to* saat membuat tugas backup di Cron.

Backup yang belum pernah diuji belum tentu backup

Sesekali coba pulihkan database ke database baru, untuk memastikan file backup benar-benar bisa dipakai.

Memantau resource



Ilustrasi menu Monitor dengan pola grafik yang perlu dikenali.

Cara membaca pola

- **Lonjakan singkat** saat build atau backup itu normal.
- **Garis RAM yang naik terus** tanpa turun menandakan aplikasi bocor memori. Restart sementara, lalu cari penyebabnya.
- **CPU tinggi terus** tanpa ada pengunjung bisa berarti proses nyangkut atau server disusupi. Cek dengan `htop`.

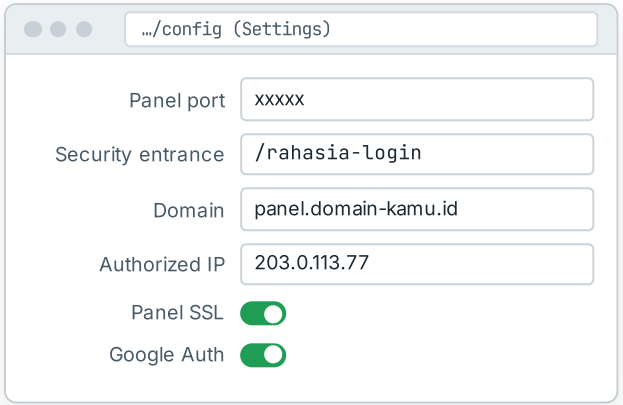
```
htop                  # proses paling boros
df -h                 # sisa disk
du -sh /www/* | sort -h  # folder terbesar
```

Keamanan produksi

Panel yang terbuka ke internet akan terus dicoba dibobol oleh bot. Checklist berikut membuat server jauh lebih sulit diserang. Kerjakan semuanya sebelum server dipakai untuk data sungguhan.

Akses panel

- [ ] **Username unik**, bukan `admin` atau `root`. Contoh: `it_core`, `onevisual_ops`.
- [ ] **Password panjang**, minimal 16 karakter, simpan di password manager.
- [ ] **2FA aktif**: Settings › Google Authenticator.
- [ ] **Security entrance** dengan path rahasia, bukan path default.
- [ ] **Port panel** diganti dan hanya diizinkan dari IP tertentu jika memungkinkan.
- [ ] **Bind domain panel**, misalnya `panel.domain.id`, lalu pasang SSL.



Ilustrasi pengaturan keamanan panel.

Server

- [ ] Login SSH dengan key, login password dimatikan (Bab 9).
- [ ] Hanya port 22, 80, 443, dan port panel yang terbuka.
- [ ] Port database dan Redis tidak dibuka ke publik.
- [ ] Pasang **Fail2ban** dari App Store untuk memblokir brute force.
- [ ] Update sistem berkala: `apt update && apt upgrade`.
- [ ] Update aaPanel lewat tombol Update di Home.
- [ ] Hapus software yang tidak dipakai, misalnya FTP.

Aplikasi

- [ ] `APP_DEBUG=false` atau `NODE_ENV=production` di server produksi.
- [ ] File `.env` tidak bisa diakses dari browser.
- [ ] Setiap aplikasi punya user database sendiri.
- [ ] Force HTTPS aktif di semua situs.
- [ ] Anti-XSS (open_basedir) aktif untuk situs PHP.

Cloudflare dan SSL

- Generate SSL pertama dengan DNS only (awan abu-abu).
- Setelah terbit, proxy (awan oranye) boleh diaktifkan lagi.
- Set SSL/TLS mode Cloudflare ke **Full (strict)**.

Tanda server bermasalah

Gejala	Kemungkinan
CPU 100% tanpa pengunjung	Proses penambang kripto dari malware
File PHP asing di folder situs	Web shell hasil upload celah aplikasi
User Linux baru yang tidak dikenal	Akses SSH dibobol
Traffic keluar sangat besar	Server dipakai untuk spam atau serangan

Jika server sudah disusupi

Jangan hanya menghapus file mencurigakan. Ambil data penting, buat server baru, pulihkan dari backup bersih, lalu ganti semua password dan key.

Perintah terminal bt

aaPanel menyediakan perintah bt untuk mengelola panel dari terminal. Sangat berguna saat panel tidak bisa dibuka dari browser, misalnya lupa password atau terkunci karena salah login.

```
root@vps: ~

root@vps:~#
 sudo bt
===============aaPanel cli tools=================
(1) Restart panel        (8) Change panel port
(2) Stop panel           (9) Clear panel cache
(3) Start panel          (10) Clear login limit
(4) Reload panel         (11) IP + UA check on/off
(5) Change password      (12) Remove domain bind
(6) Change username      (13) Remove IP limit
(7) Change MySQL root    (14) View default info
(15) Clear system junk   (16) Repair panel
(22) Panel error log     (24) Turn off 2FA
(28) Change entrance     (0) Cancel
================================================
Please enter command number: 14
```

Ilustrasi menu bt. **Nomor menu dapat berbeda antar versi.** Selalu baca daftar di layar sebelum memilih.

Semua perintah bt butuh akses root

Jalankan sebagai root atau awali dengan `sudo`.

Lebih aman lewat menu

Ketik `sudo bt` tanpa angka, lalu pilih nomornya. Dengan begitu kamu bisa membaca dulu apa yang akan dijalankan.

Situasi	Perintah	Keterangan
Lupa alamat, username, atau port panel	`sudo bt 14`	Menampilkan info login default
Lupa password panel	`sudo bt 5`	Masukkan password baru
Ganti username panel	`sudo bt 6`	Masukkan username baru
Terkunci karena salah login berkali-kali	`sudo bt 10`	Menghapus batasan login
IP kamu tidak diizinkan masuk panel	`sudo bt 13`	Menghapus pembatasan IP
Domain panel bermasalah, tidak bisa login	`sudo bt 12`	Kembali akses lewat IP
HP hilang, tidak bisa membuka 2FA	`sudo bt 24`	Mematikan Google Authenticator
Panel lambat atau tampil aneh setelah ubah config	`sudo bt reload`	Memuat ulang service panel
Panel tidak merespons	`sudo bt restart`	Restart penuh service panel
Muncul error di panel	`sudo bt 22`	Menampilkan log error panel
File inti panel rusak	`sudo bt 16`	Memperbaiki file inti panel
Disk hampir penuh	`sudo bt 15`	Membersihkan sampah sistem dan panel

Perintah Linux pendamping

```
/etc/init.d/nginx restart          # restart Nginx
/www/server/nginx/sbin/nginx -t    # cek config Nginx sebelum reload
/etc/init.d/mysqld restart         # restart MySQL
/etc/init.d/php-fpm-82 restart     # restart PHP 8.2
ss -tulpn | grep 4321              # cek port aplikasi aktif
```

Troubleshooting: membaca kode error

Hampir setiap masalah situs bisa dipersempit dari kode error yang muncul di browser. Mulai dari kode error, ikuti cabangnya, lalu cek log yang ditunjuk.

Situs tidak tampil? Lihat kode error di browser

403
Forbidden

- **Izin file salah**
  → chown www:www, chmod 755/644
- **Tidak ada file index**
  → cek Default document
- **Running dir salah**
  → Laravel harus ke /public
- **IP diblokir**
  → cek IP rules di Security

404
Not found

- **Path situs salah**
  → cek Site directory
- **Route tidak dikenali**
  → pasang URL rewrite
- **Hasil build belum ada**
  → jalankan npm run build
- **Domain belum terdaftar**
  → tambah di Domain manager

500
Server error

- **Error di kode**
  → baca DOMAIN.error.log
- **Laravel**
  → storage/logs/laravel.log
- **Folder storage tidak bisa ditulis**
  → chown www
- **Ekstensi PHP kurang**
  → pasang di setting PHP

502 / 504
Bad gateway

- **App Node mati**
  → cek log proyek, ss -tulpn
- **Port proxy beda**
  → samakan port app dan proxy
- **PHP-FPM mati**
  → restart php-fpm
- **Proses terlalu lama (504)**
  → naikkan timeout

Timeout
Tidak terbuka

- **DNS belum ke IP server**
  → host domain 1.1.1.1
- **Port 80/443 tertutup**
  → firewall dan security group
- **Nginx mati**
  → cek Home, jalankan nginx -t
- **Server kehabisan RAM**
  → cek Monitor, restart

Di mana mencari petunjuk

Log	Lokasi
Error situs (Nginx/PHP)	`/www/wwwlogs/DOMAIN.error.log`
Akses situs	`/www/wwwlogs/DOMAIN.log`
Aplikasi Node	Log di Node Project, atau `pm2 logs`
Panel aaPanel	`/www/server/panel/logs/error.log`
MySQL	Databases › Setting › Error log

Membaca log dari terminal

```
# 50 baris error terakhir sebuah situs
tail -n 50 /www/wwwlogs/DOMAIN.error.log
# pantau log secara langsung saat membuka situs
tail -f /www/wwwlogs/DOMAIN.error.log
```

Biasakan alur ini

Buka situs → lihat kode error → buka log yang sesuai → baca baris terakhir. Pesan error di log hampir selalu menyebut file dan baris penyebabnya.

Ubah satu hal, uji, baru lanjut

Mengubah banyak pengaturan sekaligus membuatmu tidak tahu perubahan mana yang memperbaiki atau malah merusak.

Troubleshooting: kasus umum

Kumpulan masalah yang paling sering dialami saat memakai aaPanel, lengkap dengan cara mengecek dan memperbaikinya.

Masalah	Penyebab	Solusi
"An error occurred while the panel was running!"	Exception pada skrip Python panel, misalnya `KeyError: 'project_script'`	Cek `tail -n 50 /www/server/panel/logs/error.log`, hapus cache `find /www/server/panel -name "*.pyc" -delete`, lalu `bt reload`. Jika masih, `bt 16`.
Panel tidak bisa dibuka	Port panel tertutup, path entrance salah, atau service panel mati	`bt 14` untuk melihat alamat yang benar, buka port di security group, `bt restart`.
502 Bad Gateway	Reverse proxy mengarah ke port yang aplikasinya mati	`ss -tulpn | grep 4321`, cek log proyek Node, restart aplikasi.
SSL gagal terbit	DNS belum mengarah, port 80 tertutup, atau proxy Cloudflare aktif	`host domain 1.1.1.1`, buka port 80, set DNS only, coba lagi.
Redirect loop (terlalu banyak pengalihan)	Force HTTPS aktif sementara Cloudflare memakai mode Flexible	Ubah mode Cloudflare ke Full (strict).

Masalah	Penyebab	Solusi
413 Request Entity Too Large	Batas ukuran upload terlalu kecil	Naikkan *Upload limit* PHP dan `client_max_body_size` Nginx.
Tidak bisa menghapus folder situs	`.user.ini` dikunci oleh Anti-XSS	Matikan Anti-XSS, atau `chattr -i .user.ini`.
Database connection refused	MySQL mati karena RAM atau disk penuh	Cek `df -h` dan `free -h`, bersihkan disk atau tambah swap, restart MySQL.
Disk penuh	Log, backup lama, recycle bin	`du -sh /www/* | sort -h`, hapus backup lama, aktifkan Cut log di Cron, `bt 15`.
Webhook tidak menjalankan deploy	URL salah, repo private tanpa deploy key, atau skrip error	Cek *Recent Deliveries* di GitHub dan log WebHook di aaPanel. Jalankan skrip manual untuk melihat error-nya.
Perubahan kode tidak muncul	Belum build ulang, app belum restart, atau cache browser/Cloudflare	Build, restart app, hard refresh (Ctrl+Shift+R), purge cache Cloudflare.

Latihan & glosarium

Gunakan checklist ini untuk praktikum Testing dan Implementasi Sistem. Setiap langkah membangun di atas langkah sebelumnya, dari server kosong sampai aplikasi yang teruji di domain staging.

Alur uji mahasiswa

1. Ubah kode di laptop, commit › **2. git push** ke branch main › **3. Auto deploy** ke staging › **4. Uji fitur** di browser › **5. Pantau** log, CPU, RAM

Checklist praktikum

- [ ] VPS siap, aaPanel terpasang
- [ ] Username, password, 2FA panel diganti
- [ ] Stack LNMP terpasang
- [ ] Domain staging diarahkan ke IP server
- [ ] Situs dibuat dan SSL aktif
- [ ] Database dan user aplikasi dibuat
- [ ] Firewall hanya membuka port yang perlu
- [ ] Repo di-clone dengan deploy key
- [ ] Aplikasi berjalan dan reverse proxy aktif
- [ ] Webhook deploy otomatis berhasil
- [ ] Backup DB dan situs terjadwal
- [ ] Uji fungsional fitur utama di staging
- [ ] Uji beban ringan, catat CPU/RAM
- [ ] Screenshot setiap langkah untuk laporan

Bahan laporan

Screenshot dashboard sebelum dan sesudah uji beban, log deploy webhook, dan hasil uji fitur di domain staging adalah bukti implementasi yang kuat.

Glosarium

Istilah	Arti singkat
VPS	Server virtual sewaan dengan akses root
LNMP / LAMP	Linux + Nginx/Apache + MySQL + PHP
DocumentRoot	Folder yang isinya bisa diakses dari browser
Vhost	Config satu situs di web server
Reverse proxy	Web server meneruskan permintaan ke aplikasi di port lain
DNS record A	Penunjuk nama domain ke alamat IP
SSL / TLS	Enkripsi yang membuat situs bisa diakses lewat HTTPS
Cron	Penjadwal tugas otomatis di Linux
Webhook	URL yang dipanggil otomatis saat suatu kejadian terjadi, misalnya push
Deploy key	SSH key khusus untuk server membaca satu repo
PM2 / systemd	Penjaga proses agar aplikasi hidup lagi jika mati atau server reboot
Staging	Salinan lingkungan produksi untuk pengujian
Swap	Ruang disk yang dipakai sebagai cadangan RAM
2FA	Login dua langkah dengan kode dari aplikasi authenticator