

Rancangan Lengkap Aplikasi AutoClipper

Stack: Next.js + Electron (Cross-platform Windows/macOS/Linux)

Fokus: AI auto-cut (pilih bagian menarik), auto-render (subtitle + crop), workflow review, scheduler, dan auto-post via konektor resmi platform sosial media.

Tanggal	26 February 2026
Target OS	Windows (utama), macOS, Linux
Mode Operasi	Local-first (proses di device) + cloud opsional (login, posting, analytics)
Audiens	Creator, marketer, agency, tim konten

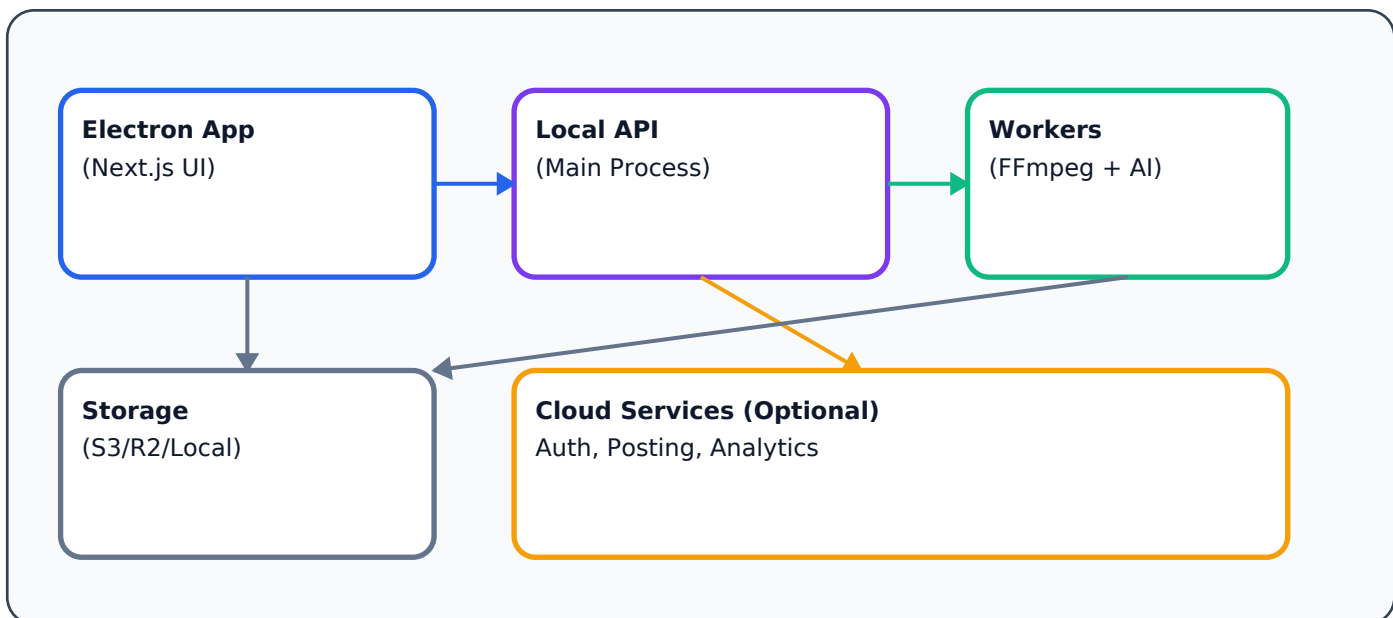
Dokumen ini berisi blueprint teknis end-to-end: arsitektur, modul, data model, API, job pipeline, keamanan login, sistem setting dinamis (ClipProfile), rendering FFmpeg, integrasi AI, serta rencana packaging untuk Windows.

1. Ringkasan Produk

AutoClipper adalah aplikasi desktop cross-platform yang mengubah video panjang menjadi beberapa klip pendek yang siap diposting (Shorts/Reels/TikTok) secara otomatis. Kualitas klip ditentukan oleh pipeline gabungan: transkripsi + kandidat segmen + scoring AI (hook/clarity/payoff) + sinyal audio/visual + auto-trim ke batas kalimat/silence.

- User dapat mengatur jenis konten (talking head/podcast/screen/gameplay/mixed) dan durasi target (min/ideal/max) secara dinamis melalui ClipProfile.
- Aplikasi menyediakan login untuk keamanan, workspace, dan kontrol akses (opsional multi-user).
- Workflow: Generate clips -> Review/Approve -> Render -> Schedule -> Post (jika API resmi tersedia).
- Mode Local-first: pemrosesan berat dilakukan di device (FFmpeg, optional ASR lokal). Cloud opsional untuk auth, sync, dan posting.

Arsitektur Tingkat Tinggi



2. Scope, Personas, dan Use Case

Personas:

- Solo creator: ingin cepat mengubah podcast/talking head jadi 10-20 klip per episode.
- Agency: butuh workflow review, template brand, dan penjadwalan multi-platform.
- Tim marketing: butuh konsistensi subtitle/cover/caption dan audit trail keputusan AI.

Use case utama:

- Upload video 30-120 menit -> hasil 5-30 klip 20-60 detik yang menarik.
- Preset ClipProfile untuk masing-masing platform + gaya subtitle.
- Approval sebelum posting, atau auto-approve untuk workflow cepat.
- History proyek, re-render, dan tracking status posting.

Non-functional requirements:

- Cross-platform build: Windows (installer), macOS (dmg), Linux (ApplImage/deb).
- Stabil untuk file besar (1-5GB), pemrosesan background via queue.
- Keamanan: token disimpan aman (OS keychain), enkripsi at-rest untuk data sensitif.
- Observability: log job, error report, dan audit output AI.

3. Tech Stack (Next.js + Electron)

Lapisan	Teknologi	Catatan
Desktop Shell	Electron	Main process untuk IPC, file system, worker orchestration, secure shell.
UI	Next.js (React) + Tailwind UI	Modern, komponen konsisten, dark mode, routing.
Local API	IPC (Electron) + HTTP local	IPC untuk prioritas sensitif; HTTP untuk worker internal jika perlu.
Database lokal	SQLite (better-sqlite3) / SQLite	Cepat & Portable. Untuk multi-user cloud: Postgres.
Queue/Jobs	BullMQ (Redis) atau queue in-process	Untuk produksi: Redis + BullMQ. Untuk dev: SQLite in-process; produksi disarankan BullMQ.
Video processing	FFmpeg (bundled binary)	Cut, crop/reframe, subtitle burn-in, loudness normalize.
ASR	Cloud Whisper/Deepgram/AssemblyAI	AssemblyAI atau Whisper API di Settings.
LLM	OpenAI/Anthropic/Gemini (Text-to-Text)	Generate title/caption/hashtags, refinement.
Storage	Local disk + optional S3/R2	Local-first; cloud untuk kolaborasi/backup.
Auto-post	API resmi platform + OAuth	Posting dijalankan via backend cloud (opsional) untuk keamanan.
Update	electron-updater	Auto update (Windows/macOS) dengan signing.

Catatan penting: untuk fitur auto-post yang robust dan aman, disarankan ada komponen backend cloud (meski aplikasi tetap desktop). Token OAuth dan refresh token lebih aman dikelola server-side.

4. Arsitektur Detail (Local-first + Cloud Opsional)

Aplikasi dibagi menjadi 3 bagian: Renderer UI (Next.js), Main Process (Electron) sebagai pengendali sistem/keamanan, dan Workers untuk job berat (FFmpeg + AI calls). Cloud opsional dipakai untuk login terpusat, sinkronisasi, dan auto-post.

Komponen:

- Renderer (Next.js): halaman proyek, player preview, editor klip, settings, scheduler.
- Main Process: auth session, secure storage, file access sandboxing, job queue, IPC API.
- Worker: proses terpisah untuk FFmpeg, transkripsi, dan LLM scoring (agar UI tetap responsif).
- Cloud (opsional): Auth, billing, social connectors (OAuth), webhook, analytics.

Alur data ringkas:

- User membuat Project -> file disimpan lokal -> metadata dibaca (ffprobe).
- Job transcribe menghasilkan transcript bertimestamp -> candidate generator membuat segmen kandidat sesuai durasi profile.
- Ranker memilih klip terbaik + auto-trim -> render menghasilkan mp4 + thumbnail + subtitle.
- Klip masuk scheduler -> jika konektor tersedia, dibuat PostJob dan dieksekusi sesuai jadwal.

5. Pipeline AI Auto-Cut (Memilih Bagian Paling Menarik)

Kualitas klip ditentukan oleh pipeline 2 tahap: (A) membangkitkan banyak kandidat segmen yang valid durasinya, lalu (B) menilai dan meranking kandidat menggunakan gabungan sinyal teks, audio, dan visual.

5.1 Candidate Generation (durasi dinamis)

Input: transcript + silence boundaries + (opsional) speaker turns. Output: ratusan segmen kandidat yang sudah memenuhi min/ideal/max dari ClipProfile.

- Boundary sources: jeda (silence), akhir kalimat (punctuation), pergantian speaker.
- Strategi: boundary pairing + sliding window (step 1-2 detik) lalu snap ke boundary terdekat.
- Filter cepat: minimal kata, batas silence, hindari potong di tengah kata/kalimat.
- Dedup: buang kandidat yang overlap > 85% dengan kandidat lain yang skornya lebih rendah.

5.2 Scoring & Ranking (gabungan AI + heuristik)

- Score Text (LLM rubric): hook (0-25), clarity (0-25), payoff (0-25), standalone (0-25).
- Score Audio: energi, tempo bicara (WPM), puncak energi (peak moments).
- Score Visual (opsional): scene-change density, motion, face presence.
- Total: $wT \cdot \text{Text} + wA \cdot \text{Audio} + wV \cdot \text{Visual}$ - penalties (filler, context, dead start, dangling end).

5.3 Auto-trim (memperbaiki cut point)

- Geser start agar hook muncul dalam 0-2 detik pertama (sesuai hook_policy).
- Buang filler awal (mis. 0.5-3 detik) tanpa memotong kata.
- End di kalimat payoff atau akhir poin, kecuali allow_cliffhanger=true.
- Snap ke boundary kalimat/silence untuk menghindari cut terasa kasar.

6. Sistem Settings & Preset (ClipProfile)

Semua perilaku pemotongan, rendering, dan copywriting ditentukan oleh ClipProfile. User bisa memilih preset lalu mengubah parameter secara dinamis.

6.1 Mode Simple vs Advanced

- Simple: pilih content type, platform, durasi (min/ideal/max), jumlah klip, gaya subtitle, mode approval.
- Advanced: bobot scoring (text/audio/visual), kebijakan hook, strict_context, allow_cliffhanger, threshold silence, template ASS, aturan hashtag, dll.

6.2 Contoh ClipProfile (JSON)

```
{
  "name": "Shorts 25-45 Mixed",
  "content_type": "mixed",
  "platform_targets": ["youtube_shorts"],
  "target_duration_sec": {"min": 25, "ideal": 35, "max": 45},
  "num_clips": {"min": 8, "max": 15},
  "weights": {"text": 0.55, "audio": 0.30, "visual": 0.15},
  "hook_policy": "high",
  "strict_context": true,
  "allow_cliffhanger": false,
  "render": {"aspect_ratio": "9:16", "subtitle_style": "clean", "crop_mode": "auto"},
  "providers": {"asr": "deepgram", "llm": "openai"}
}
```

7. Login, Keamanan, dan Kontrol Akses

Karena aplikasi desktop memiliki akses file lokal dan token sosial media, desain keamanan harus jelas: session management, secure storage, dan batasan hak akses.

7.1 Model Auth yang disarankan

- Email + Password dengan server auth (cloud) dan token JWT/opaque session token.
- Support OAuth (Google) opsional untuk onboarding cepat.
- Mode offline opsional: akun lokal (tanpa sync) - tetap simpan token terenkripsi di device.

7.2 Secure storage (wajib)

- Simpan token session dan refresh token di OS Keychain (Windows Credential Manager, macOS Keychain, libsecret Linux).
- Encrypt konfigurasi sensitif di SQLite (mis. provider keys) menggunakan key dari Keychain.
- Pisahkan data proyek besar (video) dari DB: simpan di disk; DB hanya menyimpan path + metadata.

7.3 RBAC (opsional multi-user/workspace)

- Roles: Owner, Admin, Editor, Viewer.
- Owner/Admin mengelola integrasi sosial media dan billing.
- Editor bisa generate/render/schedule; Viewer hanya lihat & download.

8. Data Model & API Contracts

8.1 Database lokal (SQLite) - tabel inti

Tabel	Tujuan	Field kunci
projects	Satu sumber video	id, title, source_path, duration_ms, status
clip_profiles	Preset setting dinamis	id, name, config_json
transcripts	Transcript bertimestamp	project_id, provider, segments_json
clip_candidates	Kandidat segmen	project_id, start_ms, end_ms, stats_json
clips	Klip final + metadata posting	start_ms, end_ms, scores, caption, status
assets	Output mp4/thumbnail/subtitle	clip_id, kind, storage_path
jobs	Queue eksekusi	type, payload_json, status, attempts

8.2 API internal (Renderer <-> Main process)

Gunakan IPC sebagai API utama untuk operasi sensitif (file access, job start, token read). Kontrak IPC dibuat seperti RPC: channel + payload schema.

Channel	Input	Output
auth.login	{email,password}	{session, user}
project.create	{title, sourcePath}	{projectId}
pipeline.run	{projectId, profileId, mode}	{jobId}
clips.list	{projectId}	{clips[]}
clip.update	{clipId, startMs,endMs,metadata}	{clip}
clip.render	{clipId}	{renderJobId}
post.schedule	{clipId, platform, scheduleAt}	{postJobId}

9. Rendering Engine (FFmpeg) & Subtitle

Rendering dilakukan di worker terpisah. Semua output dirender dari ClipPlan yang sudah final (start/end, crop mode, subtitle style). FFmpeg dibundling di installer agar konsisten di Windows/macOS/Linux.

9.1 Langkah render standar

- Cut video sesuai start/end.
- Reframe/crop sesuai aspect ratio (9:16 default). Mode auto: face-tracking atau center crop atau screen-fit.
- Generate subtitle (SRT/ASS) dari transcript segment; burn-in via filter subtitles.
- Normalisasi audio (optional) agar konsisten.
- Generate thumbnail (frame di 20% durasi klip) dan metadata.

9.2 Template subtitle (ASS) - prinsip

- Safe area 10-12% bawah agar tidak ketutup UI platform.
- Stroke/outline cukup untuk kontras di background ramai.
- Opsi karaoke highlight per-kata (V2) jika diperlukan.

10. Scheduler & Auto-Posting

Auto-post bergantung pada ketersediaan API resmi platform. Desain sistem harus menyediakan fallback: export dan reminder jika platform tidak mendukung posting otomatis.

10.1 Model eksekusi yang disarankan

- Desktop: user membuat PostJob (platform, caption, schedule).
- Cloud worker (opsional tapi direkomendasikan): menjalankan posting menggunakan token OAuth dan menangani webhook/status.
- Jika tanpa cloud: batasi ke platform yang memungkinkan token disimpan lokal dan upload dari client secara resmi.

10.2 Status lifecycle PostJob

- queued -> scheduled -> uploading -> posted
- failed (dengan error detail) -> retry (backoff) -> failed_permanent

11. UI Modern (Next.js) - Halaman & Komponen

11.1 Halaman utama

- Login: email/password + OAuth opsional, remember session.
- Dashboard: daftar project, status job, quick actions.
- Project Detail: timeline, tombol Generate Clips, progress pipeline, daftar clip.
- Clip Editor: preview player, trim handles, subtitle preview, title/caption editor, render & schedule.
- Settings: provider AI, ClipProfile preset, template subtitle, posting connectors, storage, performance.

11.2 Komponen UI kunci

- Player + scrubber + marker start/end (drag).
- Clip list dengan score breakdown (text/audio/visual) dan reason singkat.
- Preset manager (buat/duplikat/ubah profile).
- Job monitor (queue) dengan log per job.
- Theme: light/dark, typography konsisten (Tailwind + shadcn/ui).

12. Packaging & Build Cross-platform

Gunakan electron-builder untuk menghasilkan installer Windows (NSIS), macOS (dmg), dan Linux (AppImage/deb). FFmpeg dan (opsional) binary tambahan (whisper.cpp) dibundling sebagai extraResources.

12.1 Distribusi Windows (rekomendasi)

- NSIS installer + auto-update (electron-updater).
- Code signing untuk mengurangi warning SmartScreen.
- Folder kerja user: %AppData%/AutoClipper (DB, config), dan folder Projects untuk video.

12.2 Strategy dependency

- Pastikan FFmpeg path dipetakan per-OS dan dipanggil dari worker.
- Gunakan spawn dengan argumen yang aman (hindari shell injection).
- Sediakan setting performance: jumlah concurrent render, limit CPU, dan lokasi cache.

13. Roadmap Implementasi

MVP (fase 1)

- Login + workspace lokal, project upload, transcribe (cloud), generate candidates, LLM scoring, render 9:16 + subtitle clean, download/export.
- Settings: ClipProfile simple + provider keys + storage path.
- UI: dashboard, project, clip list, preview editor sederhana.

Fase 2

- Approval workflow tim, template subtitle lebih banyak, crop mode auto lebih pintar (face/screen).
- Scheduler + integrasi posting minimal 1 platform via API resmi (jika tersedia).
- Observability: job logs, crash reporting opt-in.

Fase 3

- Analytics + feedback loop untuk scoring, A/B test caption/hook.
- Multi-language subtitle + translate, brand kit per workspace.

Lampiran: Checklist keamanan & kualitas

- Token disimpan di keychain; config sensitif terenkripsi.
- Selalu snap cut ke boundary kalimat/silence.
- Simpan audit: score breakdown + prompt version.
- Fallback posting: export jika konektor tidak tersedia.